Aspectos Formais da Computação

Prof. Sergio D. Zorzo
Departamento de Computação – UFSCar
1º semestre / 2017
Aula 06

Expressões regulares

Expressões regulares

- Autômatos denotam linguagens
- Autômatos possuem duas notações
  - Diagrama de estados
  - Tabela de transições
- Vimos apenas uma notação para linguagens
  - Notações de conjuntos / formadores de conjuntos
  - Ex: {w | regra sobre w}

- Existe outra notação
  - Expressões regulares

Expressões regulares

Definição de álgebra:

Um conjunto A e

uma coleção de operações sobre A

Operações que podem ser k-árias:

- 0-árias: ex: constantes, 2, x, y
- 1-árias: ex: -10
- 2-árias: ex: 2+2, 3*y

Expressões regulares

Na teoria da computação

Álgebra envolve

- Conjunto A = alfabeto
- Operações = operações regulares

Operações regulares

- Operações sobre membros de um alfabeto

- Linguagens regulares são fechadas sob as operações regulares

Conjuntos fechados sob uma operação

Exemplo:

N = {1,2,3,…} (conjunto de números naturais)

N é fechado sob multiplicação

Ou seja: para quaisquer x e y em N

x * y também está em N

N não é fechado sob divisão

Contra-exemplo: 1 e 2 estão em N, mas ½ não está!

Definição:

Uma coleção de objetos é fechada sob alguma operação se, aplicando-se essa operação a membros da coleção, recebe-se um objeto ainda na coleção

Operações regulares

Operações que:

- Aplicadas sobre elementos de linguagens regulares
- Resultam em linguagens regulares

Em outras palavras:

- Sejam L1 e L2 duas linguagens regulares
- L1 op_{reg} L2 é regular

Operações regulares

São 3 as operações regulares:

- União: $A \cup B = \{x \mid x \in A \text{ ou } x \in B\}$
- Concatenação: $A.B = \{xy \mid x \in A \text{ e } y \in B\}$
- Estrela (ou fechamento, ou fechamento de Kleene):
  $A^* = \{x_1x_2...x_k \mid k \geq 0 \text{ e cada } x_i \in A\}$

União e concatenação são operações binárias

Estrela é uma operação unária

Operações regulares

União

L = {001, 10, 111} e M = {ε, 001}

$L \cup M$ = {ε, 10, 001, 111}

Concatenação

L = {001, 10, 111} e M = {ε, 001}

L.M (ou LM) = {001,10,111,001001,10001,111001}

Estrela

L = {0, 11}

L* = {ε, 0, 00, 000, 000, 11, 011, 1111, 00011011, ... (não há uma ordem lógica aqui)}

Operações regulares

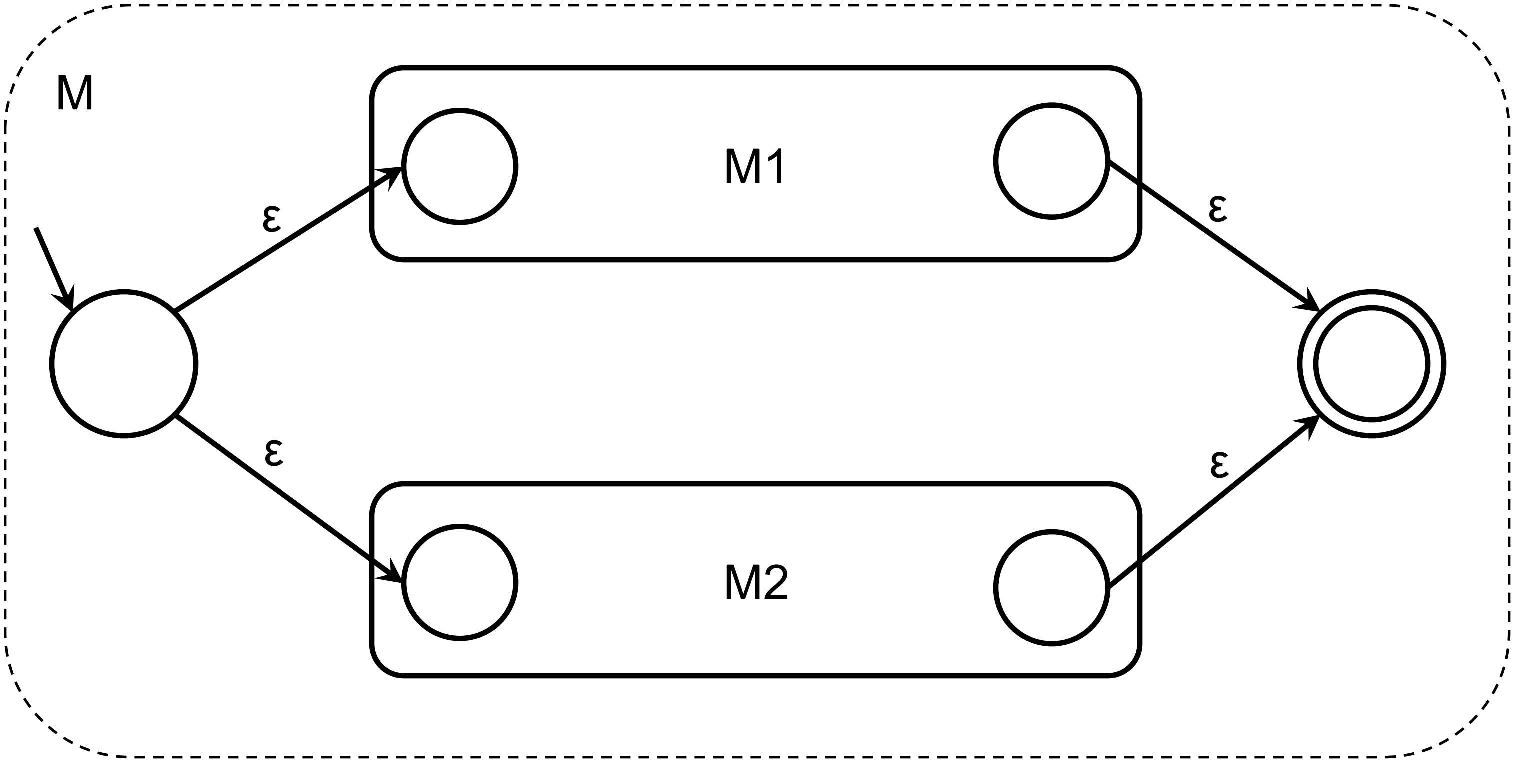
Teorema: A classe de linguagens regulares é fechada sob a operação de união

- Em outras palavras: se A1 e A2 são linguagens regulares, A1 $\cup$ A2 é regular

Prova por construção:

- A1 é regular, existe um autômato M1
- A2 é regular, existe um autômato M2
- Construímos um autômato M que simula M1 e M2, aceitando se uma das simulações aceita

Operações regulares



$$L(M) = L(M1) \cup L(M2)$$

Operações regulares

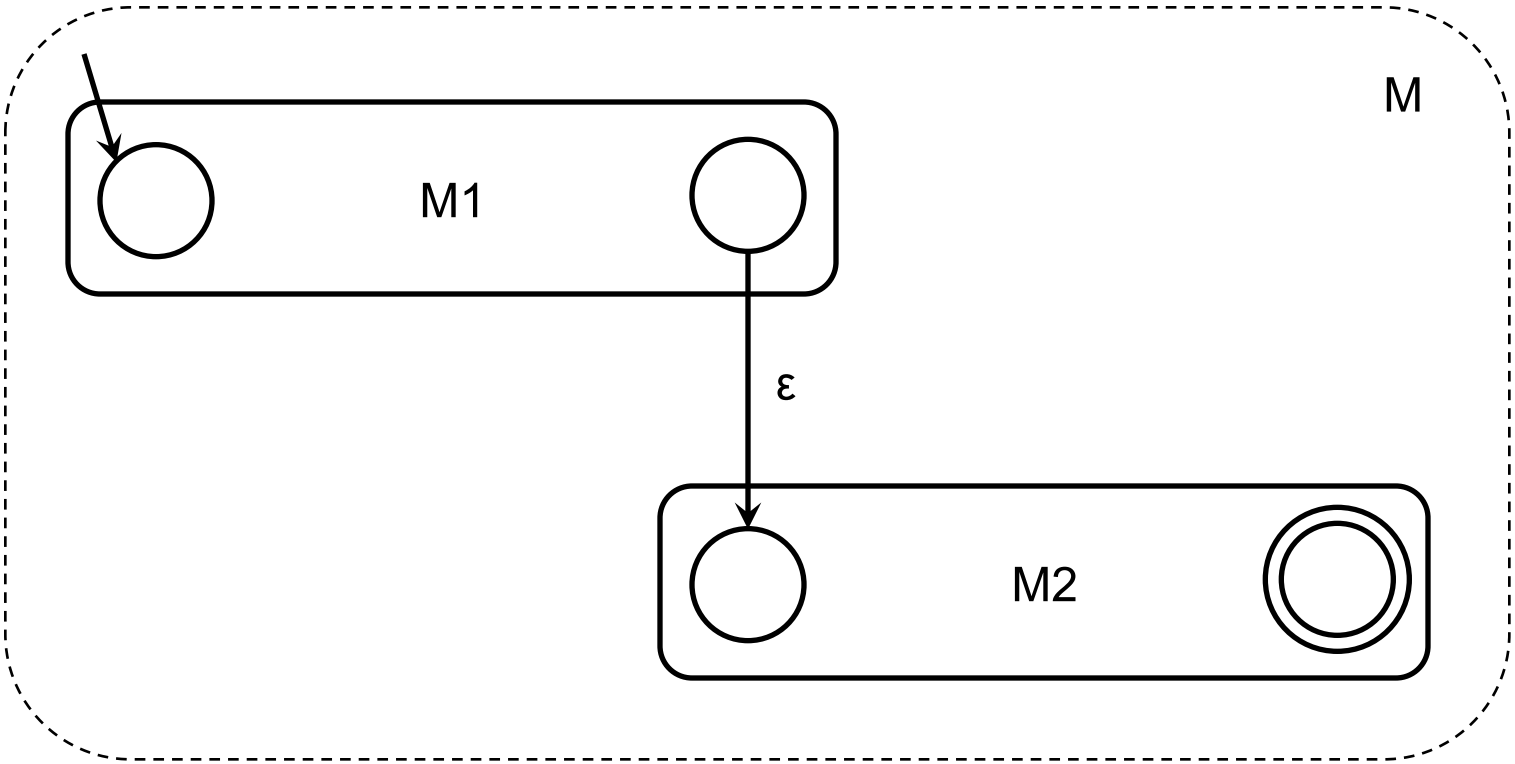
Teorema: A classe de linguagens regulares é fechada sob a operação de concatenação

- Em outras palavras, se A1 e A2 são linguagens regulares, A1.A2 é regular

Prova por Construção

- A1 é regular, existe um autômato M1
- A2 é regular, existe um autômato M2
- Construímos um autômato M que simula M1 e em seguida M2, passando de M1 para M2 quando M1 aceita, e aceitando quando M2 aceita

Operações regulares



$$L(M) = L(M1) \,.\, L(M2)$$

Operações regulares

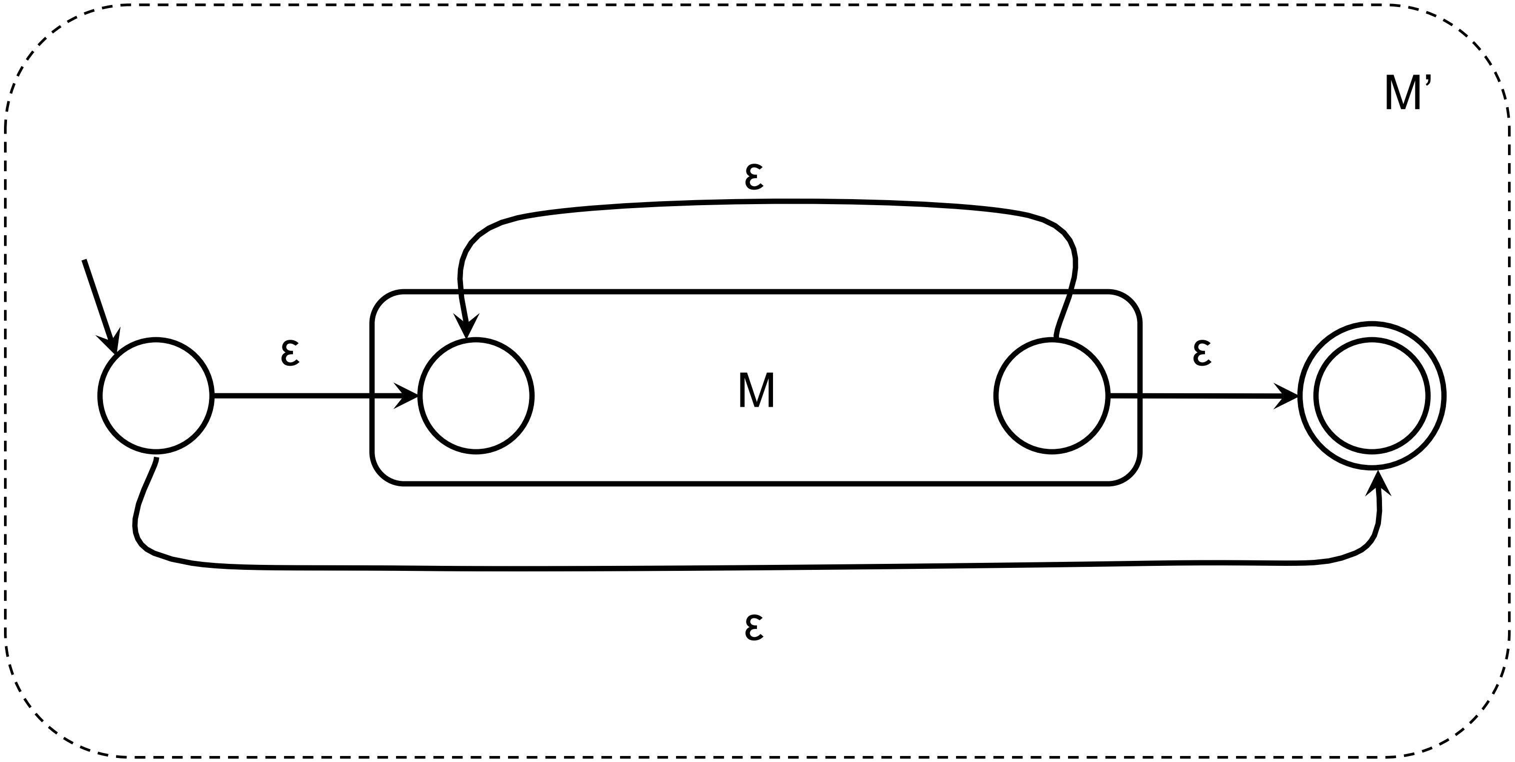
Teorema: A classe de linguagens regulares é fechada sob a operação de estrela

- Em outras palavras, se A é uma linguagem regular, A* é regular

Prova por Construção

- A é regular, existe um autômato M
- Construímos um autômato M’ que simula M, com a possibilidade de ir direto para o estado de aceitação, e a possibilidade de voltar de um estado de aceitação para o inicial

Operações regulares



$$L(M') = L(M)^*$$

Definição: Expressões Regulares

Constantes:

- ε e ∅ são expressões regulares
  - $L(\varepsilon) = \{\varepsilon\}$
  - $L(\emptyset) = \emptyset$
- Se a é um símbolo qualquer, a é uma expressão regular
  - $L(a) = \{a\}$

União

- Se E e F são expressões regulares, E + F é uma expressão regular
  - $L(E+F) = L(E) \cup L(F)$

Definição: Expressões Regulares

Concatenação

- Se E e F são expressões regulares, EF é uma expressão regular
  - $L(EF) = L(E).L(F)$

Estrela

- Se E é uma expressão regular, E* é uma expressão regular
  - $L(E^*) = (L(E))^*$

Parêntesis

- Se E é uma expressão regular, (E) é uma expressão regular
  - $L((E)) = L(E)$

Definição: Expressões Regulares

Precedência

Estrela → concatenação → união

Ex: 01*+1

Parêntesis

Mudam a precedência

Ex: (01)*+1 ou 0(1*+1)

Exemplos de expressões regulares

Alfabeto = {0,1}

- 0^*10^* = {w | w contém um único 1}
- $01 + 10$ = {01,10}
- $(\varepsilon + 0)1^*$ = {w | w é uma sequência de zero ou mais 1s, começando opcionalmente com 0}
- $(0+1)^*$ = Conjunto das partes do alfabeto ou conjunto de todas as cadeias possíveis sobre o alfabeto, incluindo a cadeia vazia ($|w| \geq 0$)
- $(0+1)(0+1)^*$ = Idem ao exemplo acima, mas sem a cadeia vazia ($|w| \geq 1$)

Exercícios

Escreva expressões regulares correspondentes às seguintes linguagens:

{w | w começa com um 1 e termina com um 0}

Resp: $1(0+1)^*0$

{w | w contém pelo menos três 1s}

Resp: $0^*10^*10^*1(0+1)^*$

{w | o comprimento de w é no máximo 5}

Resp: $(0+1+\varepsilon)(0+1+\varepsilon)(0+1+\varepsilon)(0+1+\varepsilon)(0+1+\varepsilon)$

{w | toda posição ímpar de w é um 1}

Resp: $(1(0+1))^* + 1((0+1)1)^*$ ou $(1(0+1))^*(1+\varepsilon)$

Leis algébricas para expressões regulares

Leis algébricas para expressões regulares

É possível (e muitas vezes necessário) simplificar expressões regulares

Ex: $1^*0 + 1^*0(\varepsilon+0+1)^*(\varepsilon+0+1) = 1^*0(0+1)^*$

Existem algumas leis algébricas que facilitam esse processo

Leis algébricas para expressões regulares

Associatividade e comutatividade

$L+M=M+L$

Ex: $0+1 = 1+0$

$(L+M)+N=L+(M+N)$

Ex: $(a^* + bc) + a = a^* + (bc + a)$

$(LM)N=L(MN)$

Ex: $(00(1+0))111=00((1+0)111)$

Leis algébricas para expressões regulares

Identidades (elemento neutro) e aniquiladores

$\emptyset + L = L + \emptyset = L$ ($\emptyset$ é identidade para união)

$\varepsilon L = L\varepsilon = L$ (ε é identidade para concatenação)

$\emptyset L = L\emptyset = \emptyset$ ($\emptyset$ é aniquilador para concatenação)

Exs:

$\varepsilon a(b+c)+aa\varepsilon = a(b+c)+aa$

$\emptyset(\varepsilon+1)^*(1+0(01^*10(0+1))) + 01 = \emptyset + 01 = 01$

Leis algébricas para expressões regulares

Leis distributivas

- $L(M+N)=LM+LN$
- $(M+N)L=ML+NL$
  - Exs:
    - $0(0+1) = 00 + 01$
    - $(0+1)(0+1) = (0+1)0 + (0+1)1$
    - $(0+1)(0+1) = 0(0+1) + 1(0+1)$

Leis algébricas para expressões regulares

Lei da idempotência

$L+L=L$

Exs:

$(0+1+\varepsilon) + (0+1+\varepsilon)=(0+1+\varepsilon)$

$(0+1+\varepsilon)+01^*0+(\varepsilon+0+1)+(\varepsilon+1+0)=(0+1+\varepsilon)+01^*0$

Leis algébricas para expressões regulares

Leis envolvendo fechamentos

$(L^*)^*=L^*$

Ex: $((01)^*)^*=(01)^*$

$\emptyset^*=\varepsilon$

$\varepsilon^*=\varepsilon$

Operadores de fechamento adicionais

$L^+=LL^*=L^*L$

$L^*=L^++\varepsilon$

$L?=\varepsilon+L$

Exemplos de simplificação

0+010

0**ε**+010

0**(ε+10)**

0**(10)?**

ab*b(a+ε)

ab*b**a?**

a**b^+**a?

a+(b+c+ε)a(b+c)+ca+ba

a+**(b+c)?**a(b+c)+ca+ba

a+(b+c)?a(b+c)+**(c+b)**a

εa+(b+c)?a(b+c)+(c+b)a

εa+**(c+b)a+(b+c)?a(b+c)**

(ε+c+b)a+(b+c)?a(b+c)

(b+c)?a+(b+c)?a(b+c)

(b+c)?a**ε**+(b+c)?a(b+c)

(b+c)?a**(ε+b+c)**

(b+c)?a**(b+c)?**

Autômatos finitos e expressões regulares

Autômatos finitos e expressões regulares

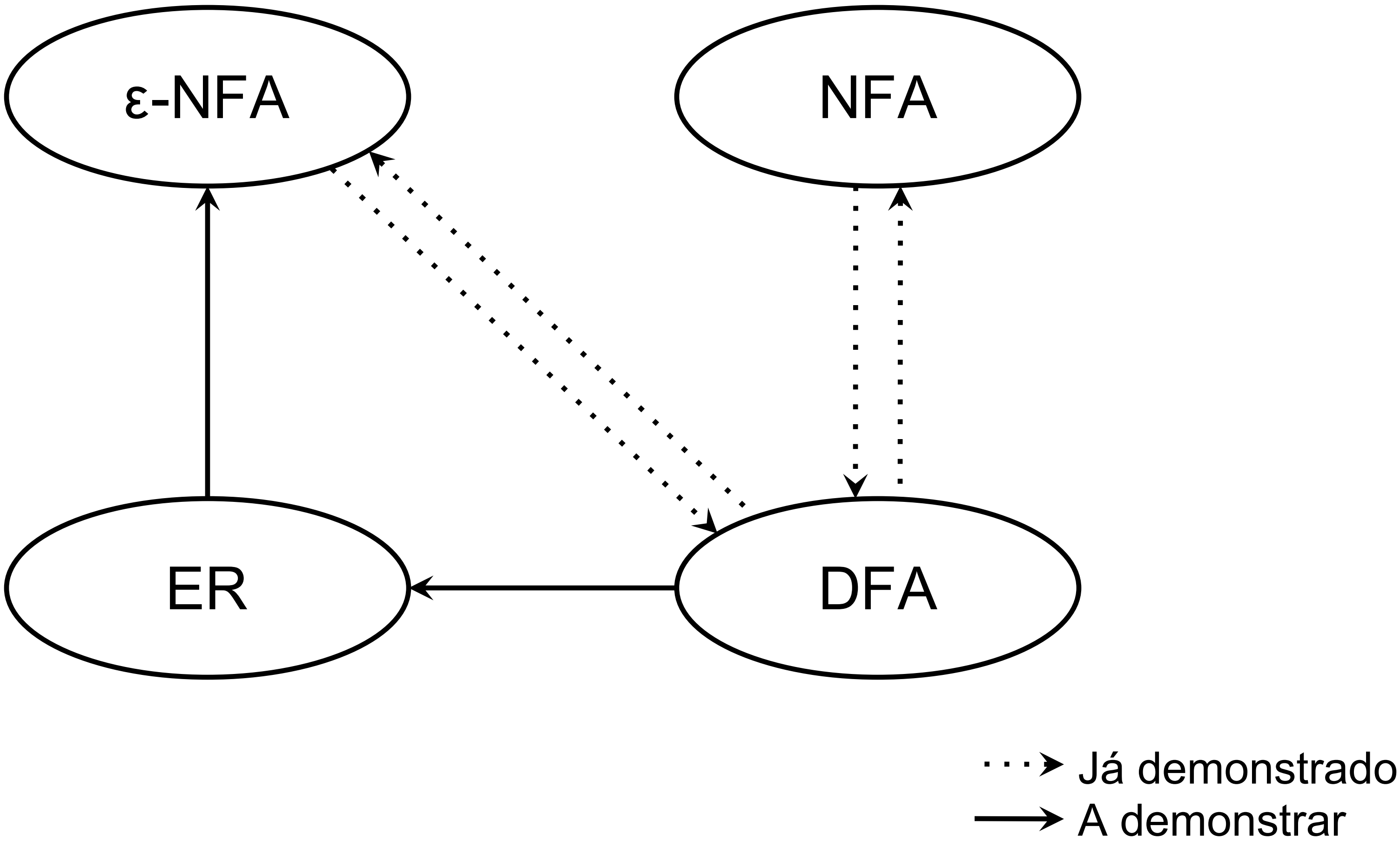
São diferentes na notação

- 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

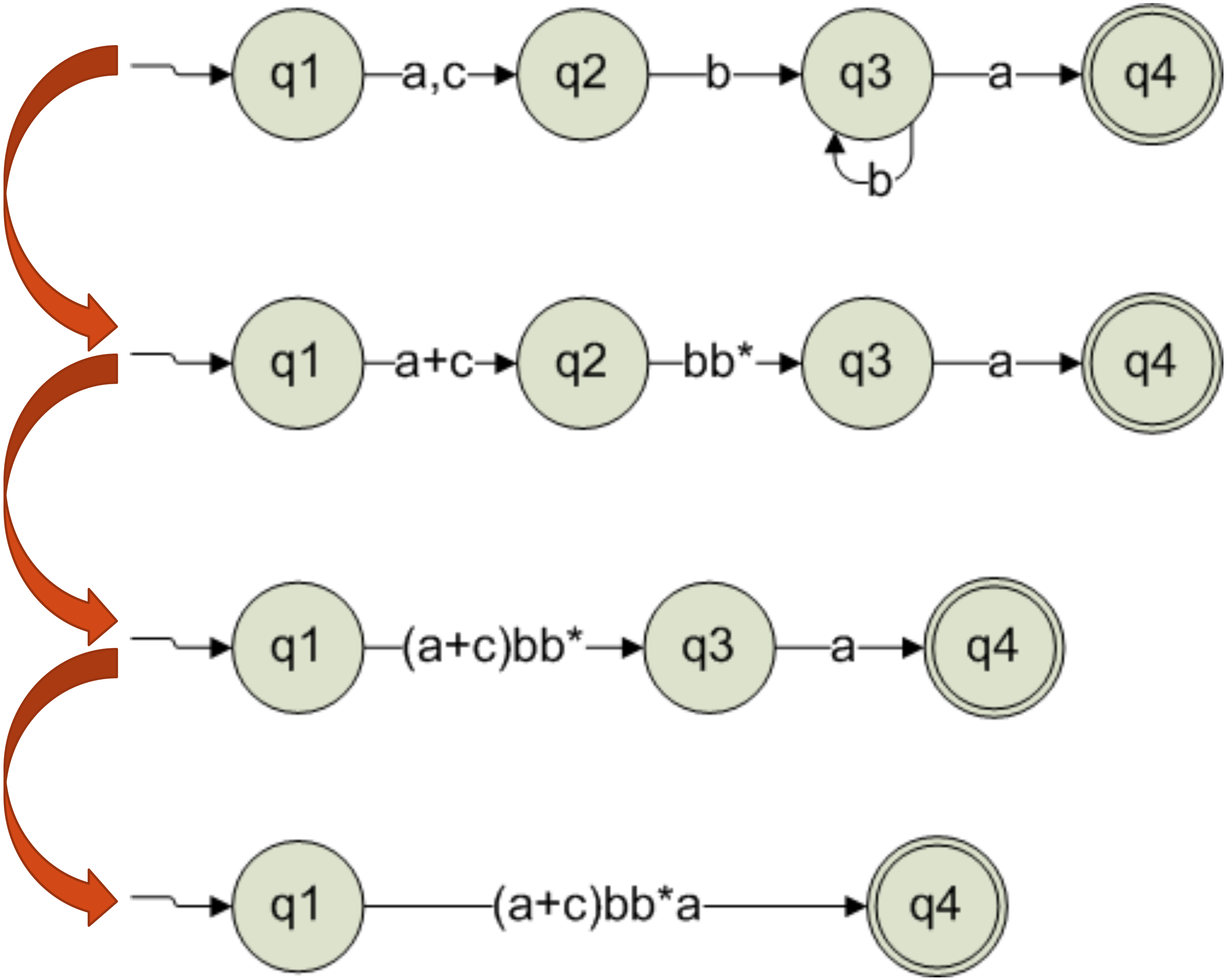
Autômatos finitos e expressões regulares



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

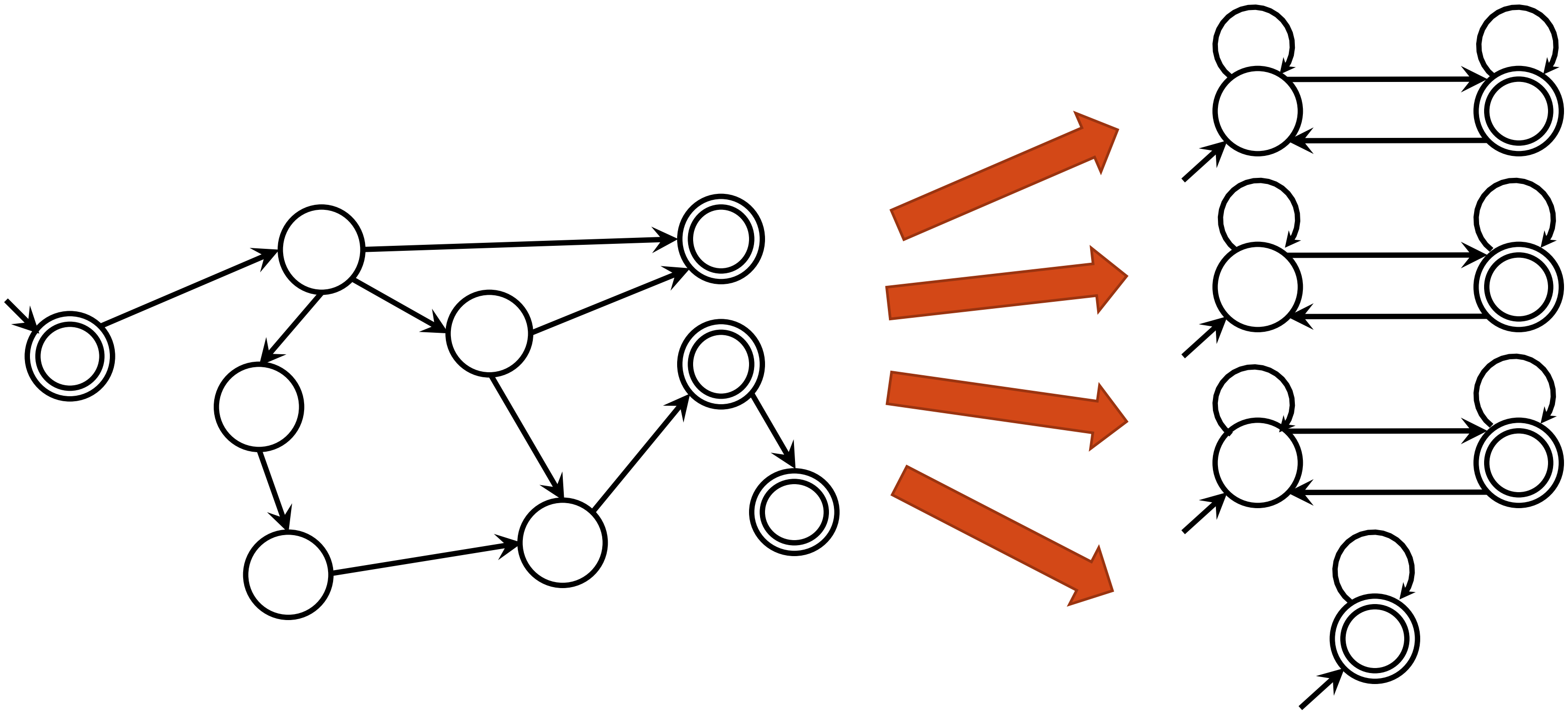
Eliminamos todos os estados, um por um

- 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

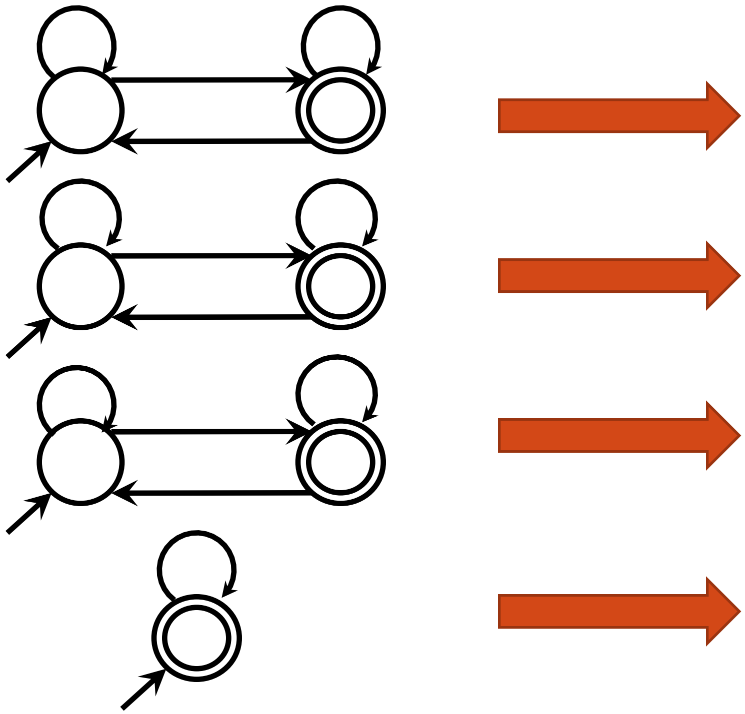
Conversão DFA→ER

- 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



01*(1+0)*11

0

0+1*

11

1

1(0+ε)10

Conversão DFA→ER

- Basta fazer a união de todas as expressões

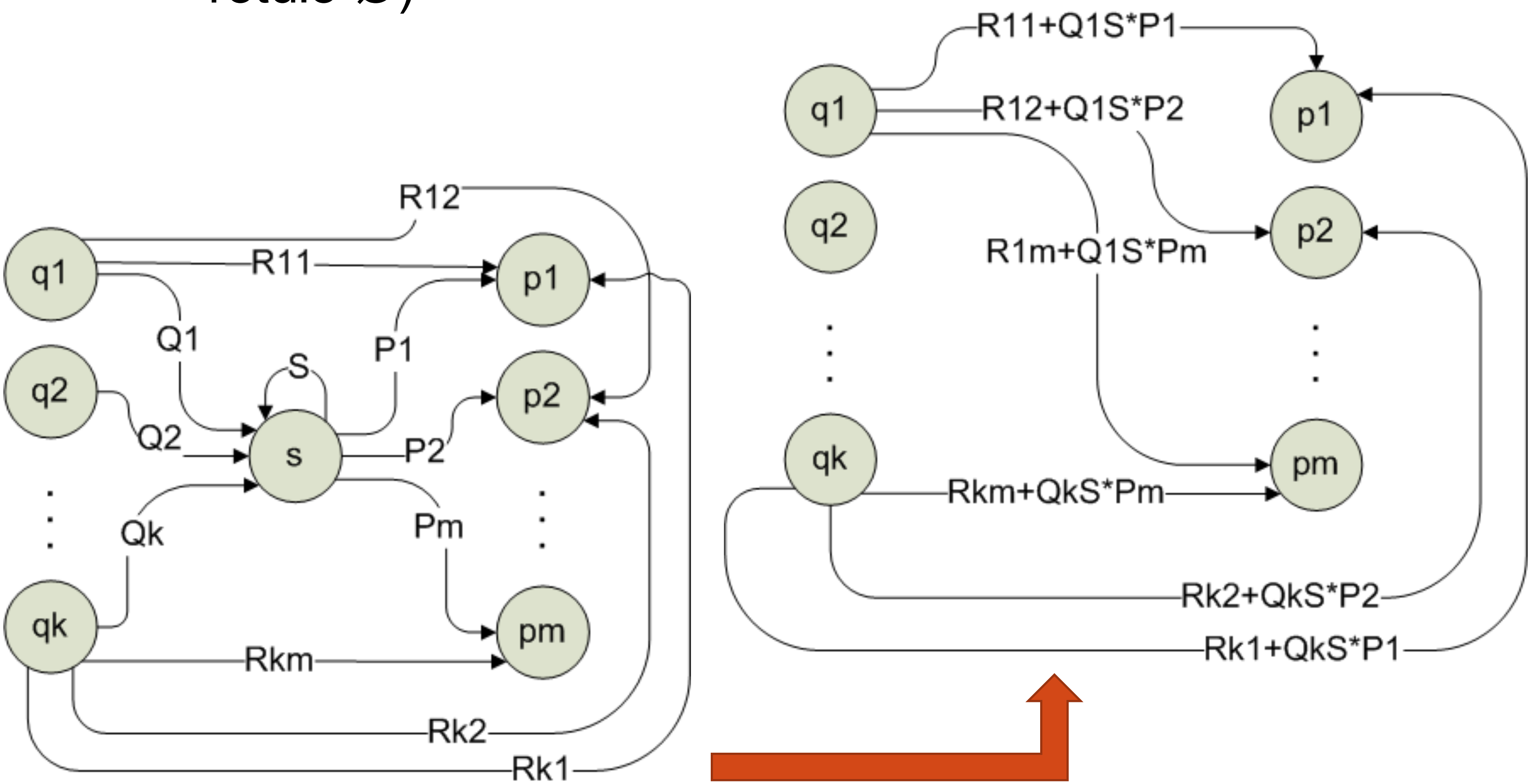
01*(1+0)*11
0
0+1*

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)

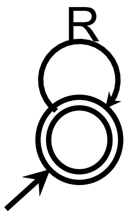


Conversão DFA→ER

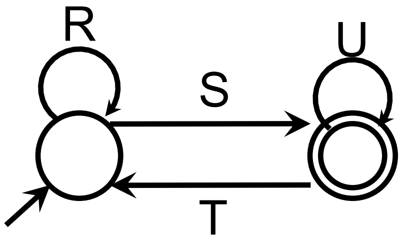
Repetir esse procedimento para todos os estados

No final, existem duas possibilidades:

- q0=q
  - Resta um único estado, com uma transição R
  - R é a expressão regular equivalente



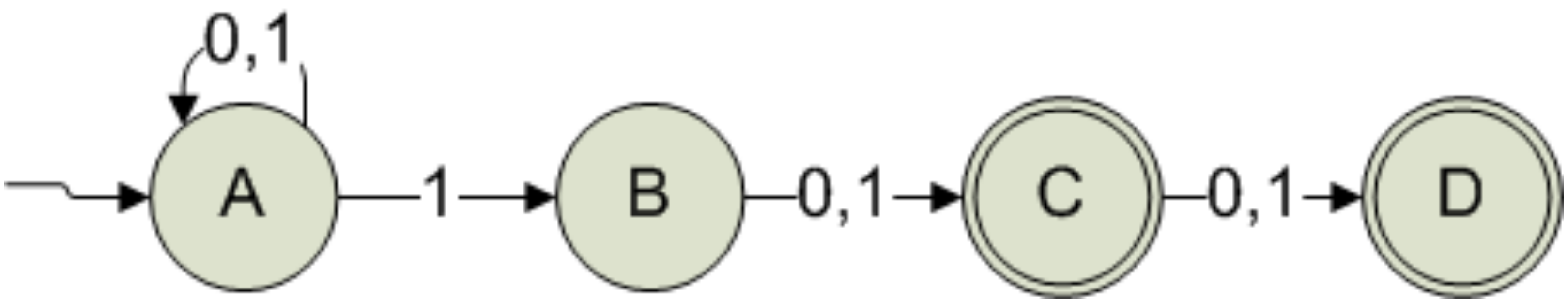
- q0≠q
  - Restam dois estados, no seguinte formato genérico:



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

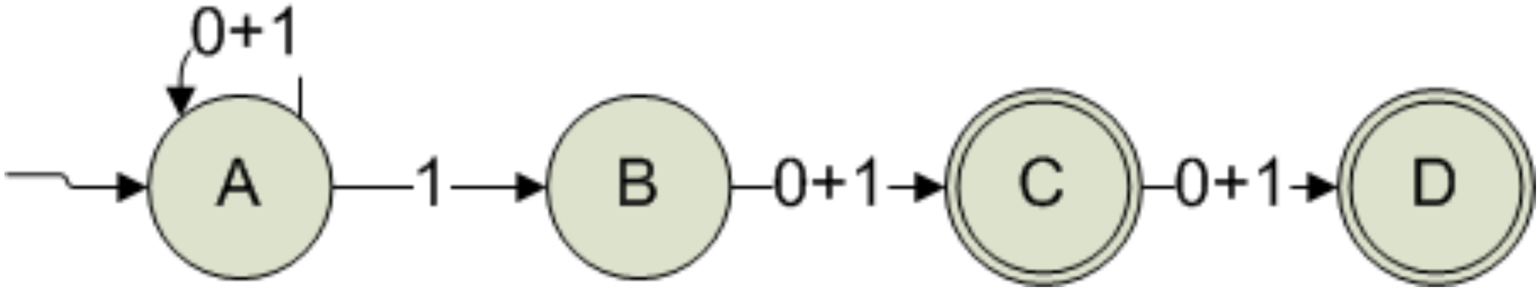
Conversão DFA→ER

- 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)

$Q1=1$

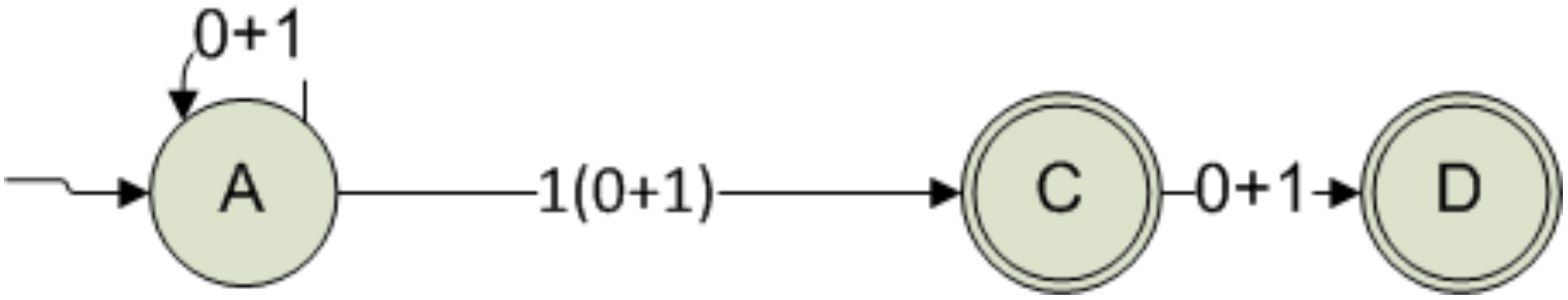
$P1=0+1$

$R11=\emptyset$

$S=\emptyset$

Arco de A para C = $R11+Q1S^*P1$ = $\emptyset+1\emptyset^*(0+1)$

Simplificando: $1(0+1)$



Conversão DFA→ER

Eliminando C

$Q1=1(0+1)$

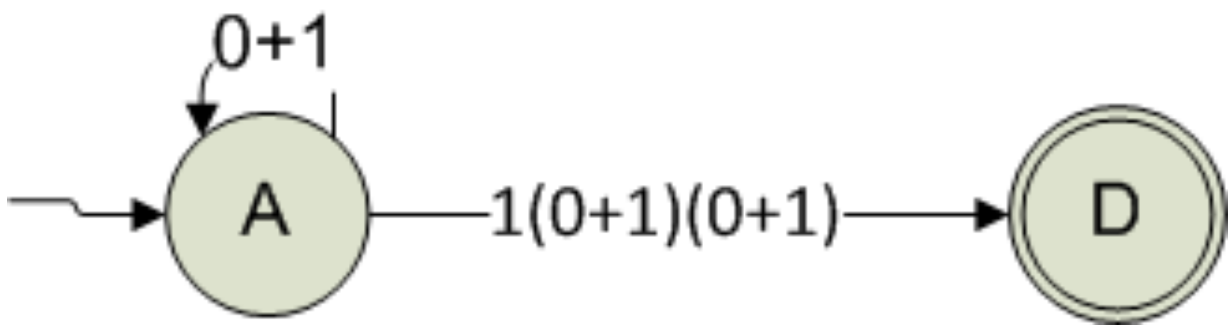
$P1=0+1$

$R11=\emptyset$

$S=\emptyset$

Arco de A para D = $R11+Q1S^*P1 = \emptyset+1(0+1)\emptyset^*(0+1)$

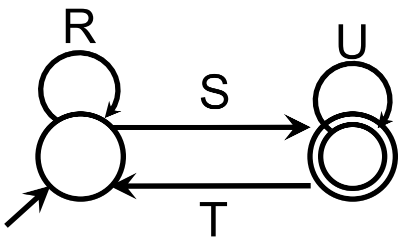
Simplificando: $1(0+1)(0+1)$



Conversão DFA→ER

Restou um autômato de 2 estados

- R=0+1
- S=1(0+1)(0+1)
- U=Ø
- T=Ø



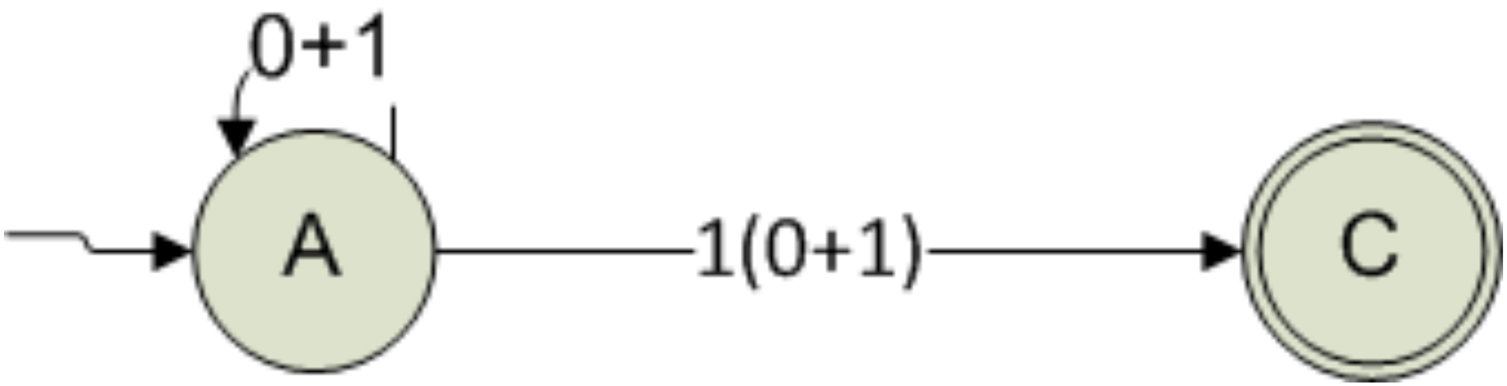
Fórmula: (R+SU*T)*SU*

- 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)?

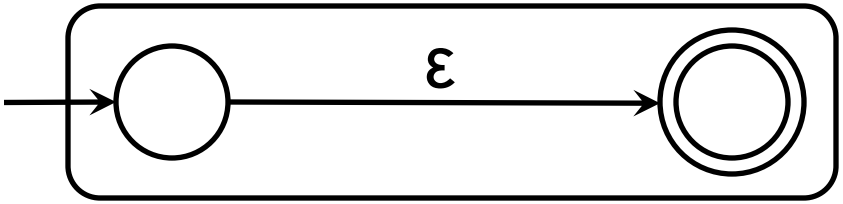
Conversão ER→ε-NFA

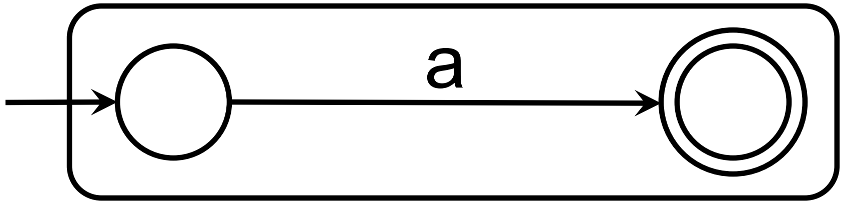
Teorema: Toda linguagem definida por uma expressão regular também é definida por um autômato finito

Prova por construção: ER→ε-NFA

Base + indução

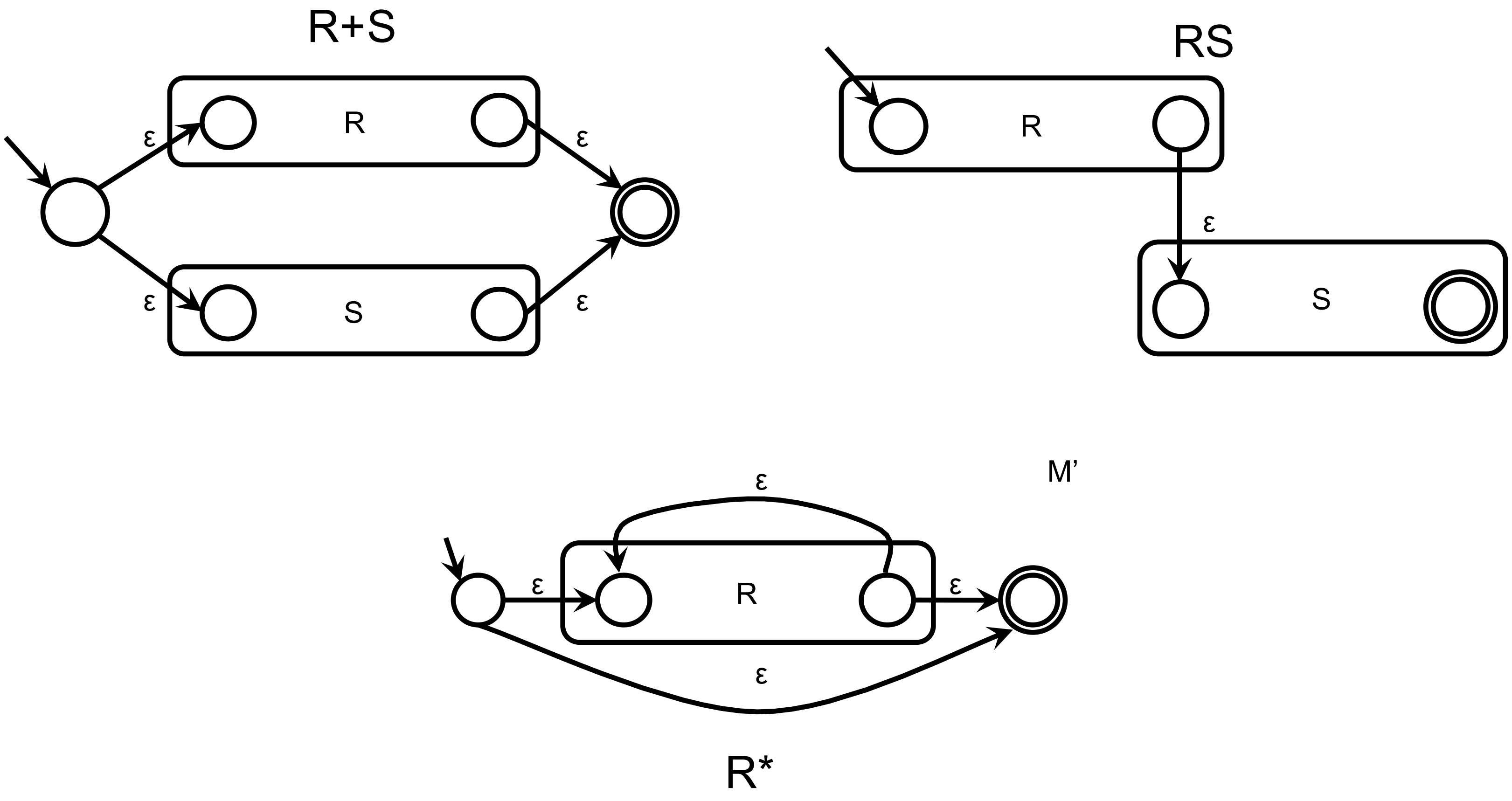
Base: ε, Ø e a (um símbolo qualquer)





Conversão ER→ε-NFA

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



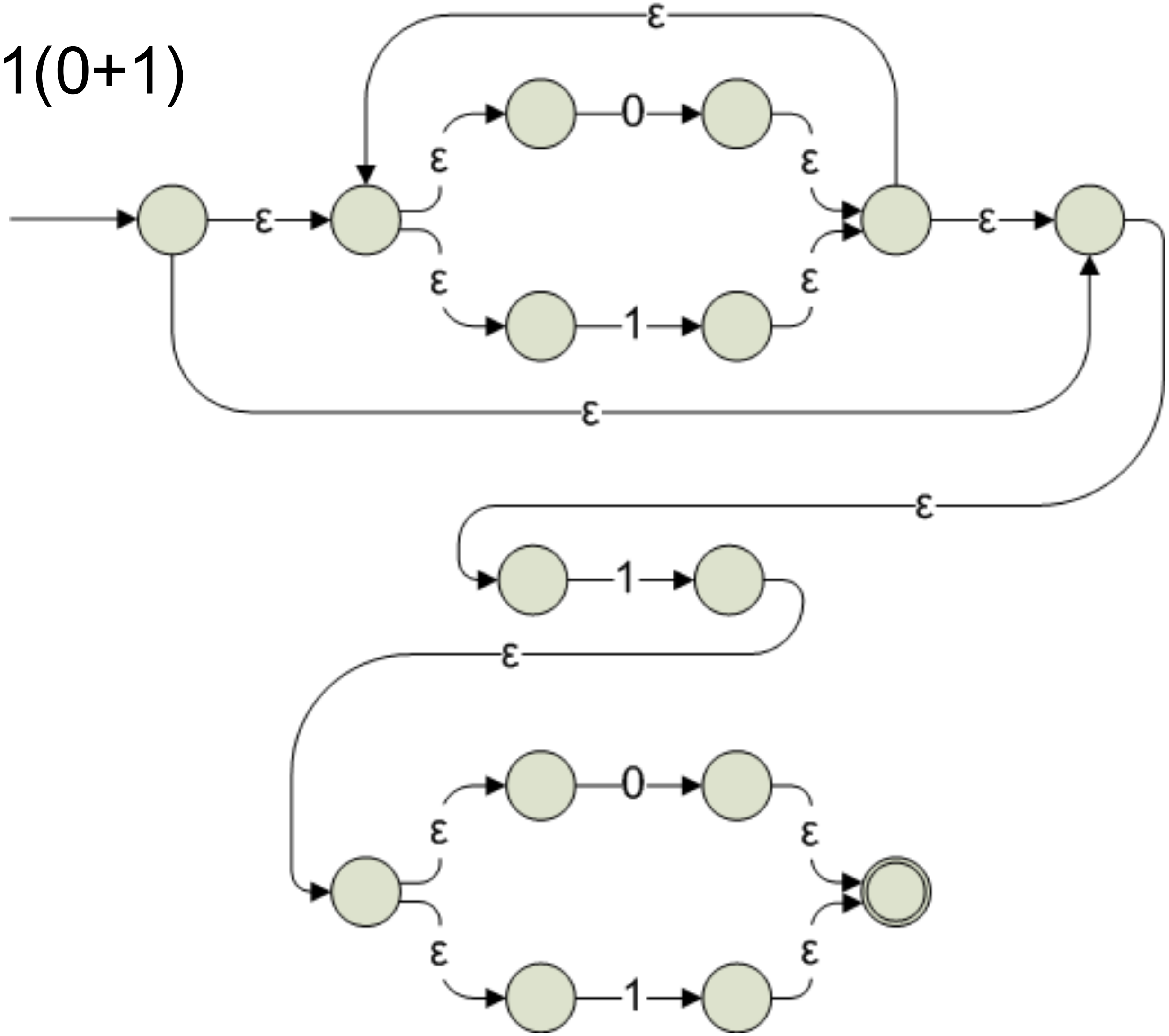
Conversão ER→ε-NFA

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)

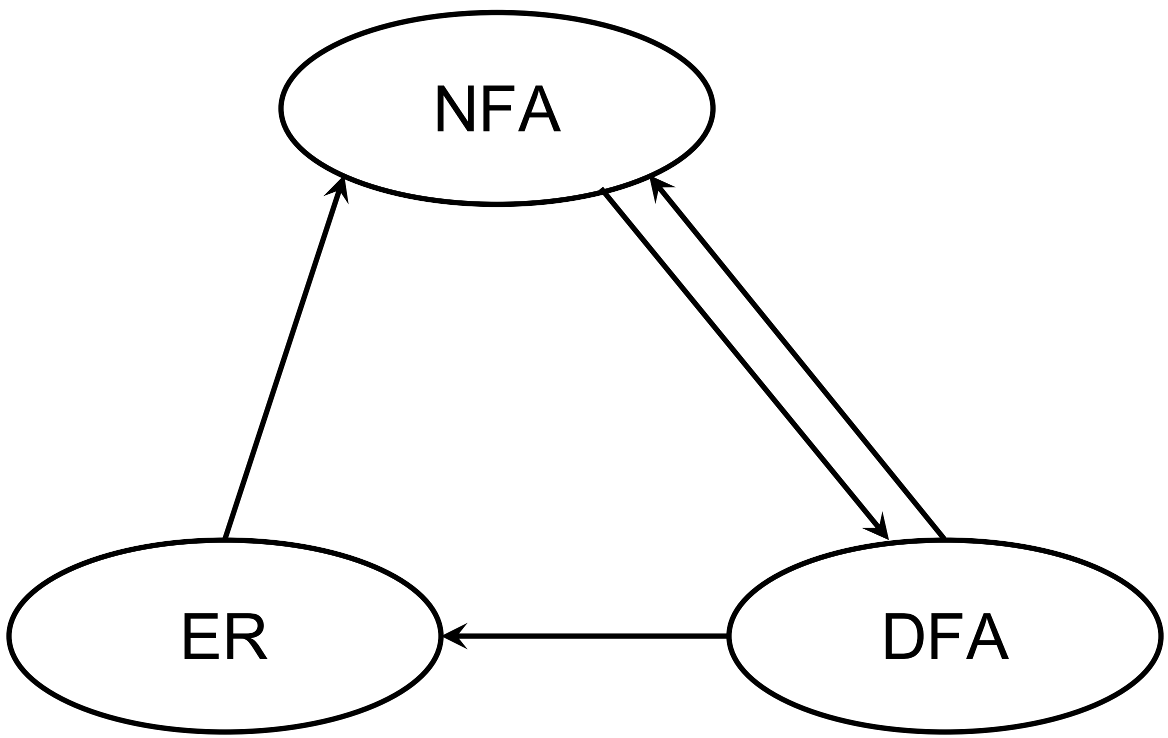


Questões sobre linguagens regulares

Questões sobre linguagens regulares

- Linguagens regulares existem sob muitas formas
  - DFA
  - NFA
  - ε-NFA
  - Expressões regulares
- Cada uma tem suas características
  - DFA = rápida execução
  - NFA (e ε-NFA) = mais fácil projeto, porém execução mais lenta
  - ER = boa legibilidade e projeto fácil
- É possível passar de uma para outra

Conversão entre representações



Fim