Departamento de Ciência da Computação - DCC
Prof. Ricardo Martins
Site: https://ricardofm.com
Email: ricardo.martins@udesc.br
Ramal: 3481-7823
Sala: Bloco F – 2° piso (sala 8)

LINGUAGENS FORMAIS E AUTÔMATOS

LFA0001:
Ciência da Computação
3ª fase

EXPRESSÕES REGULARES

- Expressão Regular
  - Toda LR pode ser descrita por uma
    - Expressão Regular
  - Formalismo
    - simples
    - denotacional (gerador)
  - Definida a partir de
  - Em particular:
    - conjuntos (linguagens) básicos
    - operação de concatenação
    - operação de união

EXPRESSÕES REGULARES

- Definição. Expressão Regular
  - ou simplesmente ER, sobre um alfabeto Σ
    - $\varnothing$ é ER, e denota a linguagem vazia
    - ε é ER, e denota a linguagem $\{ \varepsilon \}$
    - x é ER, onde $x \in \Sigma$ e denota a linguagem $\{ x \}$
  - se r e s são ER, e denotam as linguagens R e S, então:
    - $(r + s)$ é ER, e denota $R \cup S$
    - (rs) é ER, e denota RS
    - (r^*) é ER, e denota R^*

- Obs.: É muito comum representar $+$ também por $|$.

EXPRESSÕES REGULARES

- Omissão de parênteses em ER
  - concatenação sucessiva tem precedência sobre
    - concatenação
    - união
  - concatenação tem precedência sobre
    - união
- Linguagem gerada
  - por uma ER r é representada por L(r) ou GERA(r)

EXPRESSÕES REGULARES

- Exemplos (para $\Sigma = \{ a, b \}$)

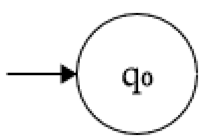
ER	Linguagem representada
aa	Somente a palavra aa
ba^*	Todas as palavras que iniciam por b, seguido por zero ou mais a´s
$(a + b)^*$	Todas as palavras sobre { a, b }
$(a + b)^* aa (a + b)^*$	Todas as palavras contendo aa como sub-palavra
$a^*ba^*ba^*$	Todas as palavras contendo exatamente dois b´s
$(a + b)^* (aa + bb)$	Todas as palavras que terminam com aa ou bb
$(a + \varepsilon) (b + ba)^*$	Todas as palavras que não possuem dois a´s consecutivos

EXPRESSÕES REGULARES

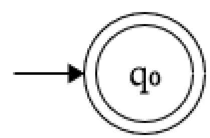
- As ER denotam exatamente as LR
  - <u>Teorema</u>
    - Se r é uma ER,
    - então GERA(r) é uma LR
  - <u>Prova</u>
    - L é LR sse é possível construir um
    - AF (AFD, AFN ou AFε) que reconheça L
    - Portanto, é necessário mostrar que:
      - dado uma ER r qq,
      - é possível construir um AF M tq
        - ACEITA(M) = GERA(r)
    - demonstração de que ACEITA(M) = GERA(r)
      - indução no número de operadores

EXPRESSÕES REGULARES

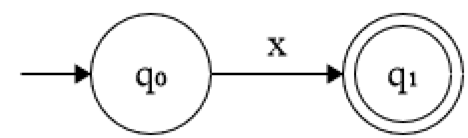
- Base (ER com zero operadores)
  - ✓ se r tem <u>zero operadores</u>, então é da forma:
    - $r = \varnothing$
    - $r = \varepsilon$
    - $r = x$ (com $x \in \Sigma$)



$M_1 = < \varnothing, \{q_0\}, \delta_1, q_0, \varnothing >$



$M_2 = < \varnothing, \{q_0\}, \delta_2, q_0, \{q_0\} >$



$M_3 = < \{x\}, \{q_0, q_1\}, \delta_3, q_0, \{q_1\} >$

δ3	x
q0	q1
q1	-

EXPRESSÕES REGULARES

- Hipótese (ER com até $n > 0$ operadores)
  - suponha que é possível construir um AF que aceita a linguagem GERA(r)
- Indução (ER com $n + 1$ operadores)
  - se r possui $n + 1$ operadores, então a ER pode ser representada por: (r_1 e r_2 possuem conjuntamente, no máximo, n operadores)
    - $r = r_1 + r_2$
    - $r = r_1 r_2$
    - $r = r_1^*$
  - por hipótese de indução, existem:

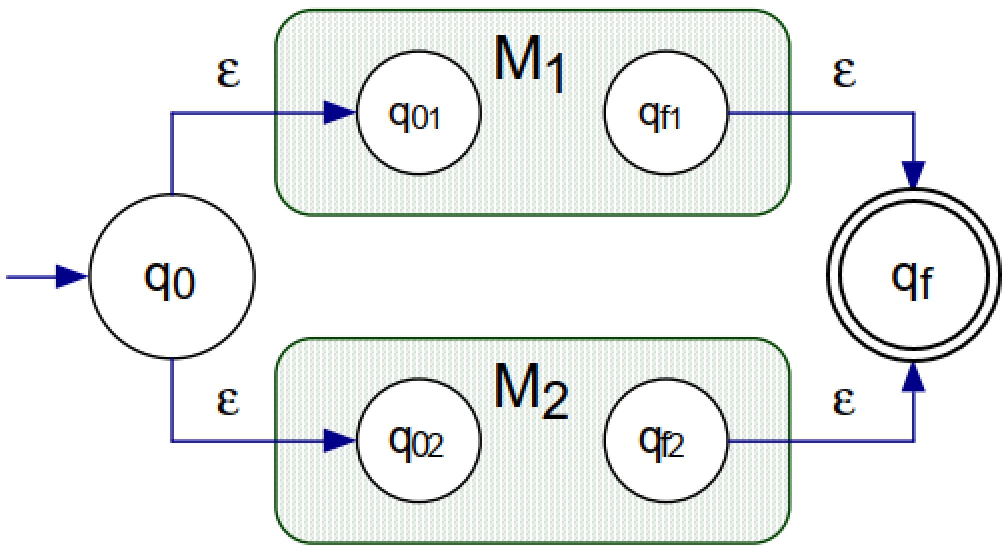
$$M_1 = < \Sigma_1, Q_1, \delta_1, q_{01}, \{ q_{f1} \} > \text{ tq } L(M_1) = \text{GERA}(r_1)$$

$$M_2 = < \Sigma_2, Q_2, \delta_2, q_{02}, \{ q_{f2} \} > \text{ tq } L(M_2) = \text{GERA}(r_2)$$

EXPRESSÕES REGULARES

- $r = r_1 + r_2$

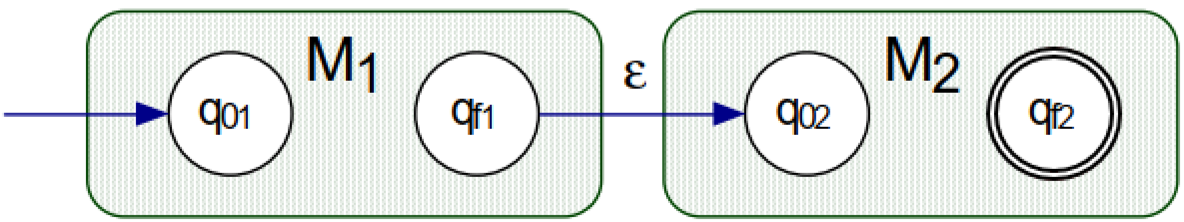
  $M = < \Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2 \cup \{ q_0, q_f \}, \delta, q_0, \{ q_f \} >$



EXPRESSÕES REGULARES

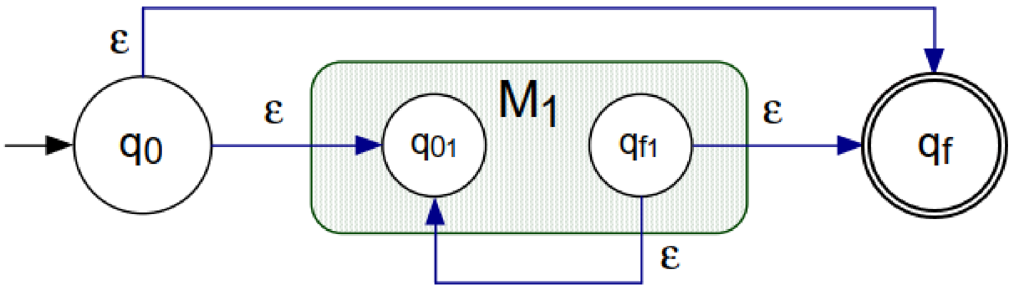
- $r = r_1 r_2$

  $M = < \Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, \delta, q_{01}, \{ q_{f2} \} >$



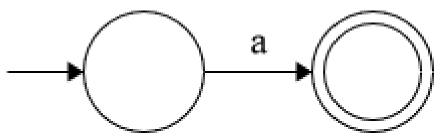
- $r = r_1^*$

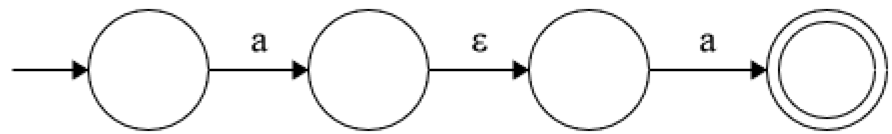
  $M = < \Sigma_1, Q_1 \cup \{q_0, q_f\}, \delta, q_0, \{ q_f \} >$

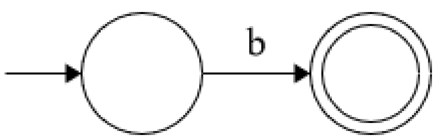


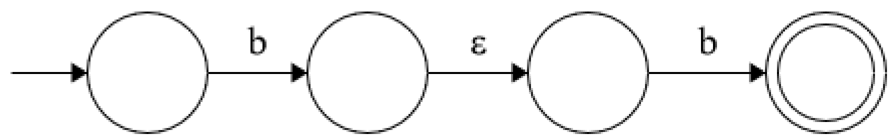
EXPRESSÕES REGULARES

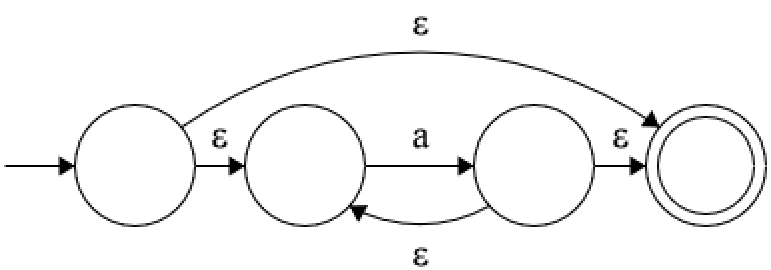
- Exemplo: a^{*} (aa + bb)

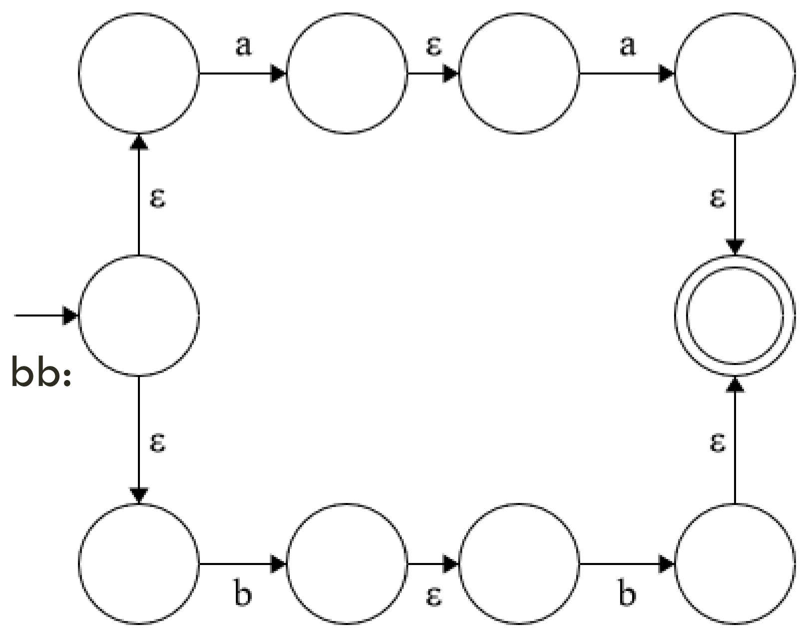
a:


aa:


b:


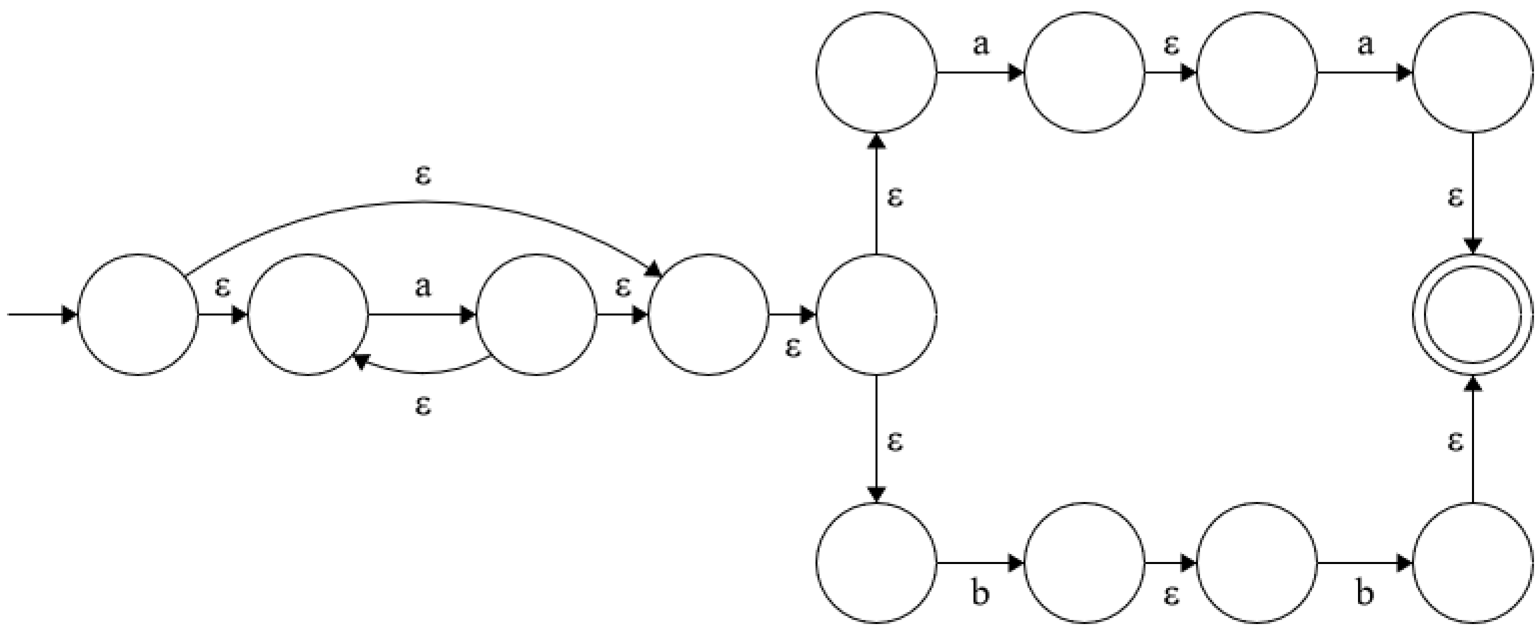
bb:


a*:


aa + bb:


EXPRESSÕES REGULARES

- Exemplo: a^* (aa + bb)



EXPRESSÕES REGULARES

- Exercícios:

Escreva, se possível (caso contrário, justifique sua resposta) as expressões regulares que representam as seguintes linguagens:

(a) $L = \{w \in \{a, b\}^* \mid w$ tem 1 ou 2 a\`s, começa e termina com um $b\}$

(b) $L = \{w \in \{0, 1\}^* \mid w$ tem um número ímpar de 0\`s$\}$

(c) $L = \{w \in \{0, 1\}^* \mid |w|_0$ é par$\}$ (obs.: $|w|_0$ representa o número de 0\`s em w)

(d) $L = \{w \in \{0, 1\}^* \mid |w|_0 = |w|_1\}$

(e) $L = \{w \in \{0, 1\}^* \mid |w|_1$ é divisível por 3$\}$

Diga se as expressões abaixo são verdadeiras ou falsas, justificando sua resposta.

(a) $abababaaba \in L\{r\}$, $r = (a^*b^*)^*a^*bb^*ba^*$

(b) $aabbaa \in L\{r\}$, $r = (aa)^*(bb)^*$

(c) $aaaaabbbbb \in L\{r\}$, $r = a^*b^*ba(a^*b^*)^*$

EXPRESSÕES REGULARES

- Exercícios:

  Destaque: $L = \{ w \in \{ 0,1 \}^* \mid w$ tem um número impar de 0´s $\}$

Restrições: número de zeros (impar)

Liberdades: localização e quantidade quanto aos 1´s

Primeira situação (restrição): 0 (0 0)*

Segunda situação (liberdade): 1* 0 1* (1* 0 1* 0 1*)* 1*

É possível simplificar? 1* 0 $\boxed{1^* (1^*}$ 0 1* 0 $\boxed{1^*)^* 1^*}$

EXPRESSÕES REGULARES

- Exercícios:

  Destaque: L = { w ∈ { 0,1 }* | w tem um número impar de 0´s }

Restrições: número de zeros (impar)

Liberdades: localização e quantidade quanto aos 1´s

Primeira situação (restrição): 0 (0 0)*

Segunda situação (liberdade): 1* 0 1* (1* 0 1* 0 1*)* 1*

É possível simplificar? 1* 0 1* (1* 0 1* 0 1*)* 1*

Simplificação correta? 1* 0 (1* 0 1* 0 1*)*

EXPRESSÕES REGULARES

- Exercícios:

  Destaque: $L = \{ w \in \{0,1\}^* \mid w$ tem um número impar de 0´s $\}$

Restrições: número de zeros (impar)

Liberdades: localização e quantidade quanto aos 1´s

Primeira situação (restrição): $0\,(0\,0)^*$

Segunda situação (liberdade): $1^*\,0\,1^*\,(\,1^*\,0\,1^*\,0\,1^*\,)^*\,1^*$

É possível simplificar? $1^*\,0\,\boxed{1^*\,(\,1^*}\,0\,1^*\,0\,\boxed{1^*\,)^*\,1^*}$

Simplificação correta? $1^*\,0\,(\,1^*\,0\,1^*\,0\,1^*\,)^*$

Não! Por que?

EXPRESSÕES REGULARES

- Exercícios:

  Destaque: $L = \{ w \in \{0,1\}^* \mid w$ tem um número impar de 0´s $\}$

Restrições: número de zeros (impar)

Liberdades: localização e quantidade quanto aos 1´s

Primeira situação (restrição): $0\ (0\ 0)^*$

Segunda situação (liberdade): $1^*\ 0\ 1^*\ (\ 1^*\ 0\ 1^*\ 0\ 1^*\ )^*\ 1^*$

É possível simplificar? $1^*\ 0\ \boxed{1^*\ (\ 1^*}\ 0\ 1^*\ 0\ \boxed{1^*\ )^*\ 1^*}$

Simplificação correta? $1^*\ 0\ (\ 1^*\ 0\ 1^*\ 0\ 1^*\ )^*$

Não! Por que?

Resposta: $1^*\ 0\ 1^*\ (\ 0\ 1^*\ 0\ 1^*\ )^*$