Aspectos Teóricos da Computação

CIÊNCIAS DA COMPUTAÇÃO

Aspectos Teóricos da Computação

Prof. César C. Xavier

ROTEIRO

Máquinas de Turing – Parte II

- Exercícios
- Expressões Regulares
- Autômato Finito Não-Determinístico (NFA)
- Conversão NFA para DFA
- Máquina de Turing Não Determinísticas
- Máquina de Turing com Várias Fitas
- Máquina de Turing com Acesso Aleatório

Aspectos Teóricos da Computação

Expressões Regulares

- Para todo autômato finito existe uma expressão regular equivalente!
- Se A_1 e A_2 são linguagens regulares então $A_3 = A_1 \cup A_2$ também é.
- **Demonstração**: Existem dois autômatos finitos e determinísticos M_1 e M_2 que reconhecem A_1 e A_2.

  $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, reconhece A_1.
  $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$, reconhece A_2.

  Construir AFD $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 \cup A_2$.

  M deverá aceitar w como entrada se M_1 ou M_2 aceitar w.

Aspectos Teóricos da Computação

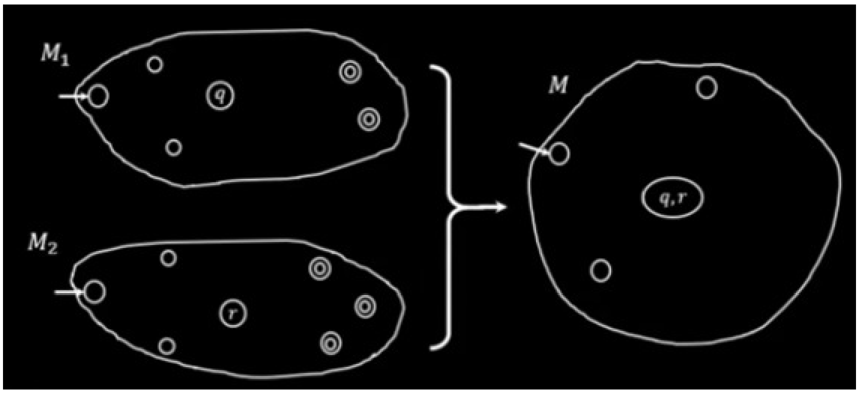
Expressões Regulares

- Construir AFD $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 \cup A_2$.
  - M deverá aceitar w como entrada se M_1 ou M_2 aceitar w.
  - É possível testar primeiro em M_1 e depois em M_2?
    - Não!
  - A solução será apresentar a entrada nas duas máquinas em paralelo!!!
- Neste caso, se provado, diz-se que as linguagens A_1 e A_2 são fechadas em relação à operação $\cup$.
- Ex.: o conjuntos dos números naturais forma um conjunto fechado em relação às operações:
  - Soma, Multiplicação, Subtração e Divisão?
  - V V F F

Aspectos Teóricos da Computação

Expressões Regulares

- Construir M = **(Q, Σ, δ, q_0, F)**, reconhece $A_1 \cup A_2$.
  - M deverá estar monitorando o par de estados de M_1 e M_2



Aspectos Teóricos da Computação

Expressões Regulares

- Construir M = **(Q, Σ, δ, q_0, F)**, reconhece $A_1 \cup A_2$.

Componentes de M:

i) Estados – par de estados:

$Q = Q_1 \times Q_2 =$

$= \{(q_i, q_j) \mid q_i \in Q_1 \text{ e } q_j \in Q_2\}$

ii) Estado inicial:

$q_0 = (q_{01}, q_{02})$

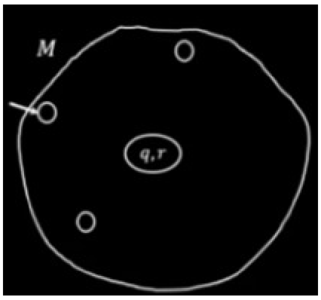
iii) FunçãoTransição: quando se está no par (q,r) ao receber o símbolo a:

$\delta((q, r), a) = (\delta_1(q, a), \delta_2(r, a))$

iv) Estados Aceitáveis F:

- $F = F_1 \times F_2$.

ERRADO! Pois apenas um dos estados deverá ser aceito!



Aspectos Teóricos da Computação

UNIVERSIDADE PAULISTA

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 \cup A_2$.

  Componentes de M:

  i) Estados – par de estados:

  $Q = Q_1 \times Q_2 =$

  $= \{(q_i, q_j) \mid q_i \in Q_1 \text{ e } q_i \in Q_2\}$

  ii) Estado inicial:

  $q_0 = (q_1, q_2)$

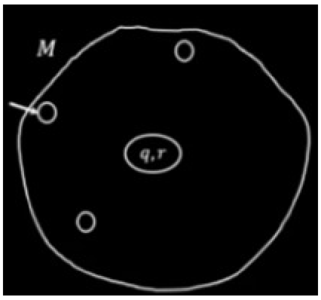
  iii) FunçãoTransição: quando se está no par (q,r) ao receber a:

  $\delta((q, r), a) = (\delta_1(q, a), \delta_2(q, a))$

  iv) Estados Aceitáveis:

  $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

  Ou o primeiro estado está em um estado de aceitação qualquer estado para o segundo elemento ou qualquer estado para o primeiro elemento e um estado de aceitação para o segundo elemento.



Aspectos Teóricos da Computação

UNIVERSIDADE PAULISTA

Expressões Regulares

- Se M_1 e M_2 são autômatos finitos com k_1 e k_2 estados respectivamente, quantos estados M terá?

  (a) $k_1 + k_2$

  (b) $(k_1)^2 + (k_2)^2$

  (c) $k_1 \times k_2$

Aspectos Teóricos da Computação

Expressões Regulares

- Se M_1 e M_2 são autômatos finitos com k_1 e k_2 estados respectivamente, quantos estados M terá?

  (a) $k_1 + k_2$

  (b) $(k_1)^2 + (k_2)^2$

  (c) $k_1 \times k_2$

Aspectos Teóricos da Computação

Expressões Regulares

- Se A_1 e A_2 são linguagens regulares então $A_3 = A_1 A_2$ (concatenação) também é.

- **Demonstração**: Existem dois autômatos M_1 e M_2 que reconhecem A_1 e A_2.

  $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, reconhece A_1.
  $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$, reconhece A_2.

  Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

  M deverá aceitar w como entrada se $w = xy$ se M_1 aceita x e M_2 aceita y.

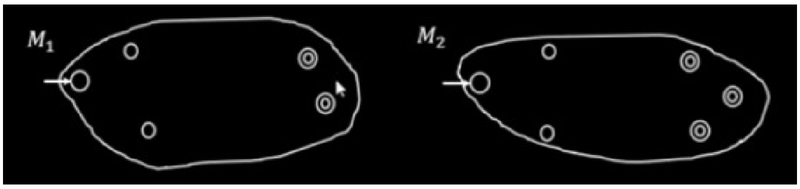
UNIVERSIDADE PAULISTA

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1\,A_2$.

M deverá aceitar *w* como entrada se $w = xy$ se M_1 aceita x e M_2 aceita *y*.



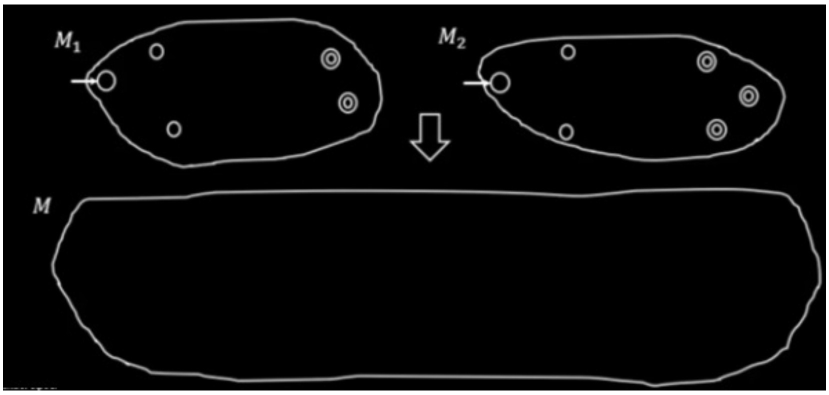
UNIVERSIDADE PAULISTA

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1\,A_2$.

M deverá aceitar *w* como entrada se $w = xy$ se M_1 aceita *x* e M_2 aceita *y*.



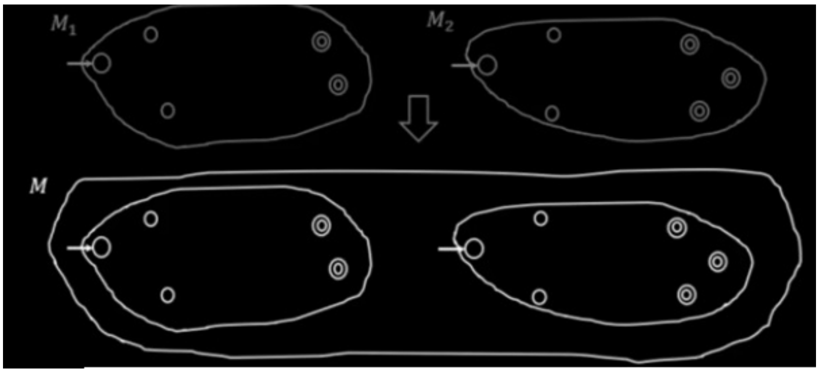
Passo #1:
- copiar as duas máquinas

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

M deverá aceitar *w* como entrada se $w = xy$ se M_1 aceita x e M_2 aceita *y*.



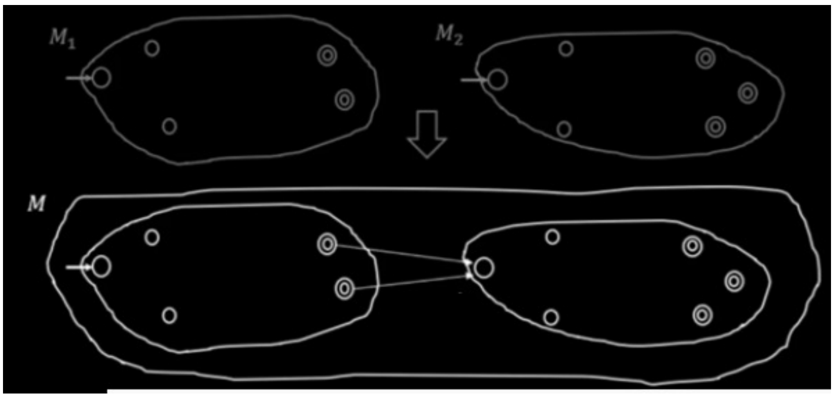
Passo #2:
- desclassificar o estado inicial de M_2
- estabelecer transição entre os estados finais de M_1 com o estado inicial M_2
- remover os estados finais de M_1.

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

M deverá aceitar *w* como entrada se $w = xy$ se M_1 aceita x e M_2 aceita *y*.



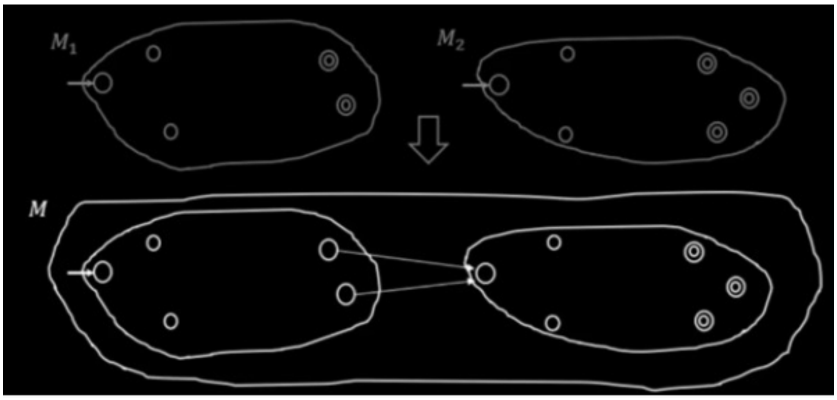
Passo #3:
- remover os estados finais de M_1.

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

M deverá aceitar w como entrada se $w = xy$ se M_1 aceita x e M_2 aceita y.



Não funciona…

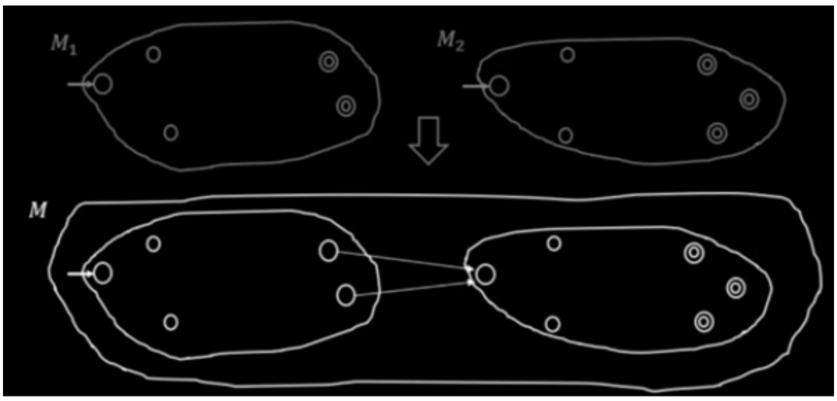
Não sabemos se dividimos a palavra na posição certa. Pode ser que exista um estado mais a frente de M_1 no qual aceite a nova palavra.

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

M deverá aceitar w como entrada se $w = xy$ se M_1 aceita x e M_2 aceita y.



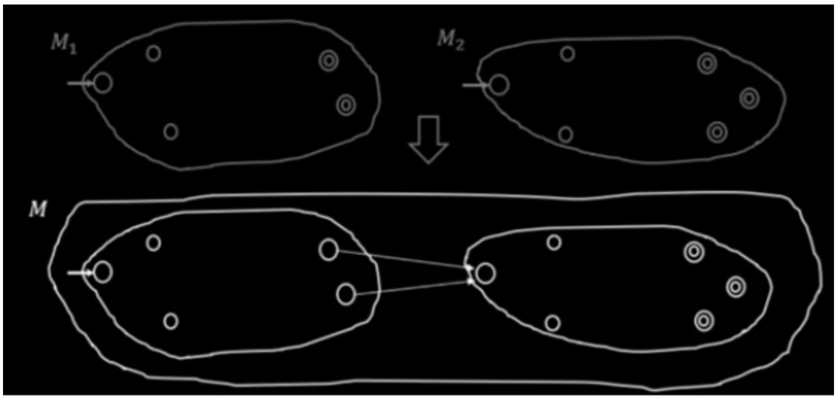
Onde dividir a palavra?

Aspectos Teóricos da Computação

Expressões Regulares

- Construir $M = (Q, \Sigma, \delta, q_0, F)$, reconhece $A_1 A_2$.

  M deverá aceitar w como entrada se $w = xy$ se M_1 aceita x e M_2 aceita y.



Onde dividir a palavra?