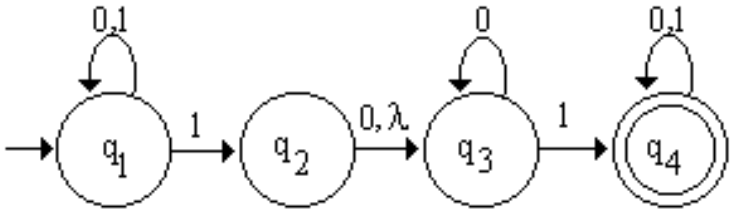
Expressões regulares

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC
Professor: Andrey Brito Período: 2023.2

Lembrando do nosso AFND do início...

- Como transformá-lo em um AFD?

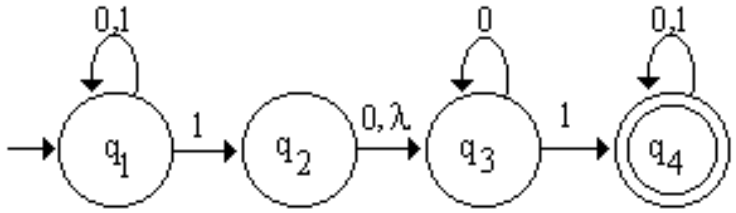
N_1:

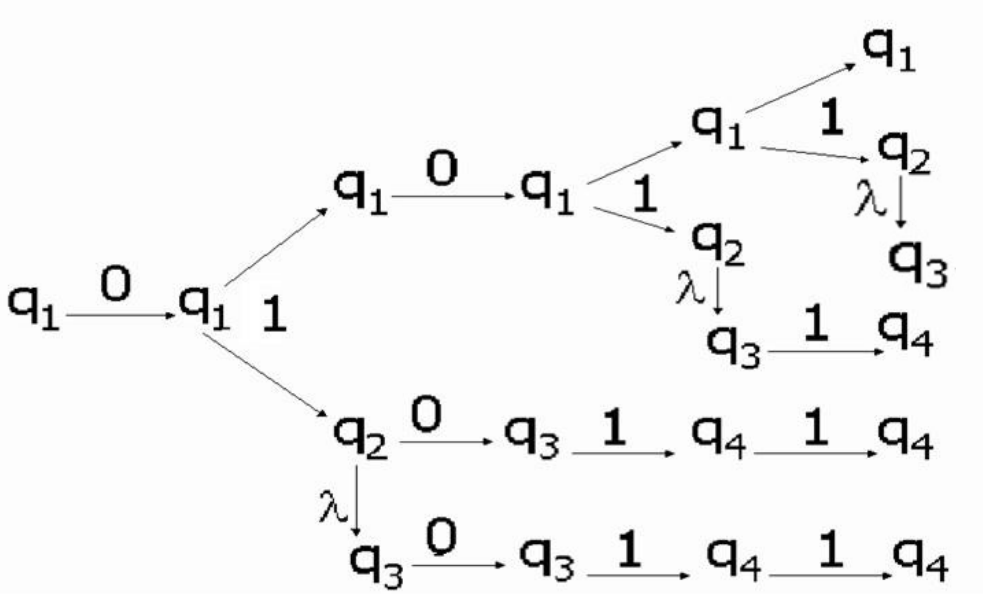


Lembrando do nosso AFND do início...

- Como ele reagiu à entrada 01011?

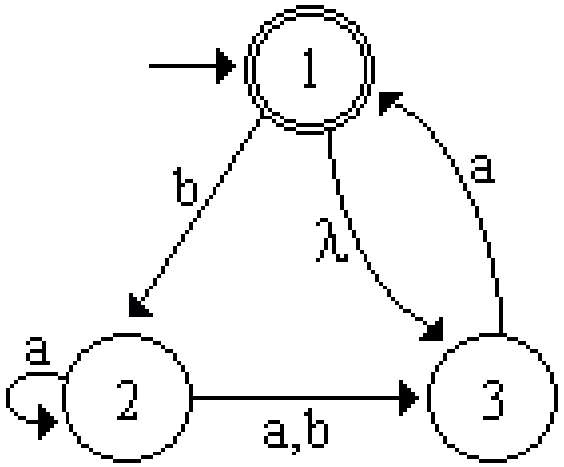
N_1:





- Como isso nos ajuda a entender a conversão? E como comparar os procedimentos de união para AFDs e para AFNDs ajudaria?

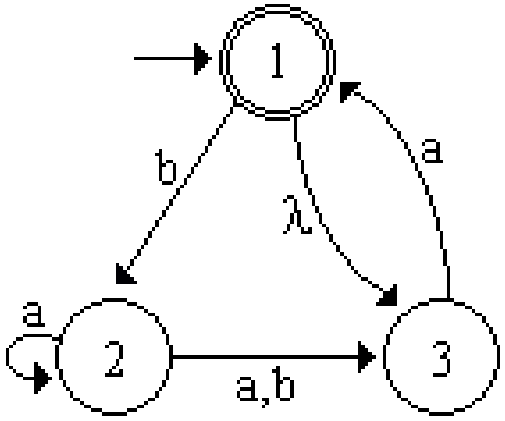
Outro exemplo (primeiro, iterativamente)



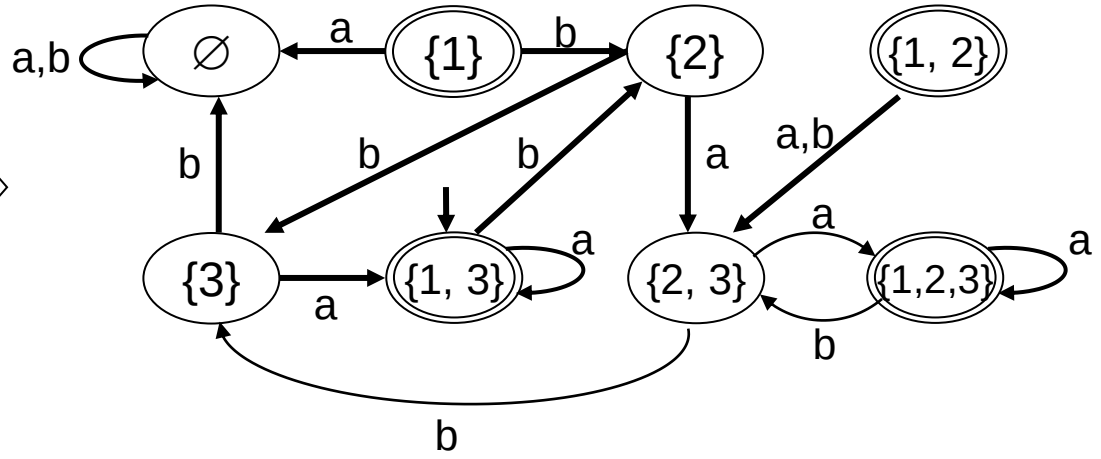
Detalhes do exaustivo

Um AFND é uma 5-tupla $<Q, \sum, \delta, q_0, F>$, onde
Q é um conjunto finito de estados
$\sum$ é um alfabeto finito
$\delta: Q \times \sum_\varepsilon \rightarrow P(Q)$ é a função de transição
$q_0 \in Q$ é o estado inicial
$F \subseteq Q$ é o conjunto de estados de aceitação

- M = <Q', $\sum$, δ', q', F'>, onde
  - Q' = < ∅, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3} >
  - q_0' = E({q_0}) = E({1}) = {1, 3}
  - F' = { {1}, {1, 2}, {1, 3}, {1, 2, 3} }



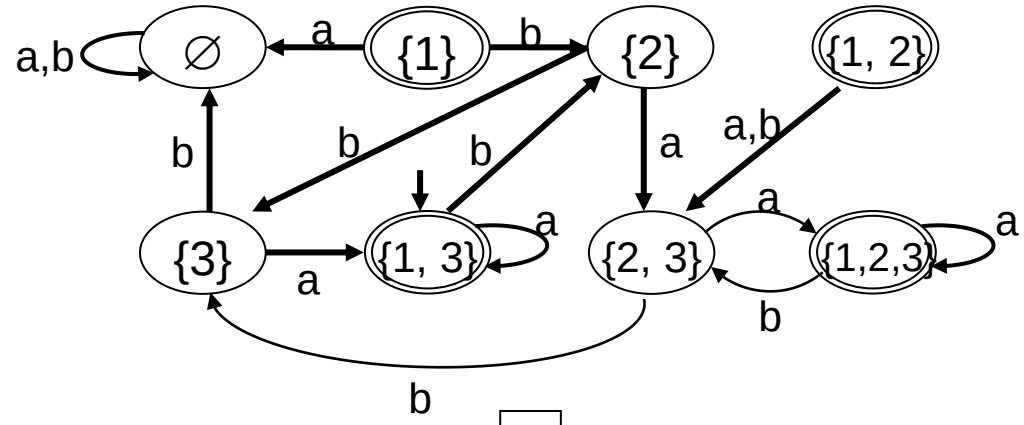
AFND N



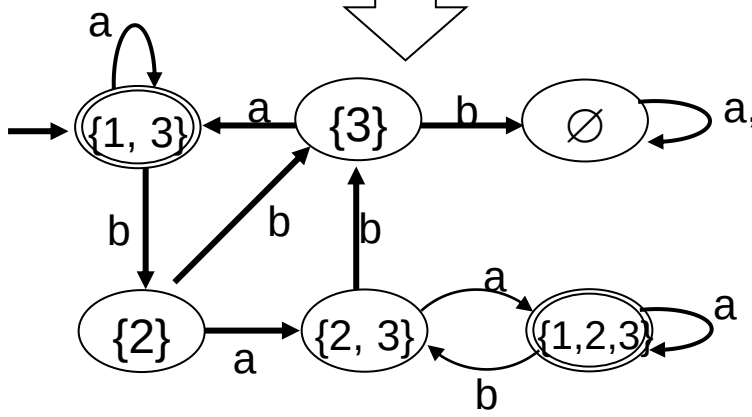
AFD M

Otimizando o exaustivo

- Eliminando os estados inacessíveis

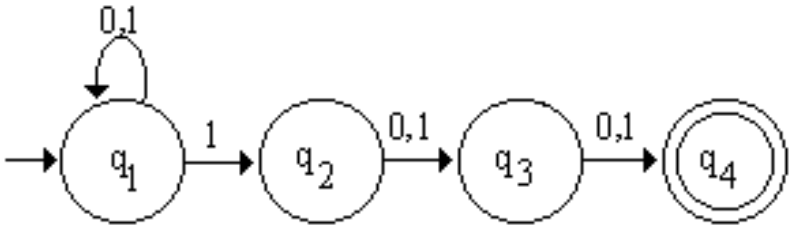


AFD M

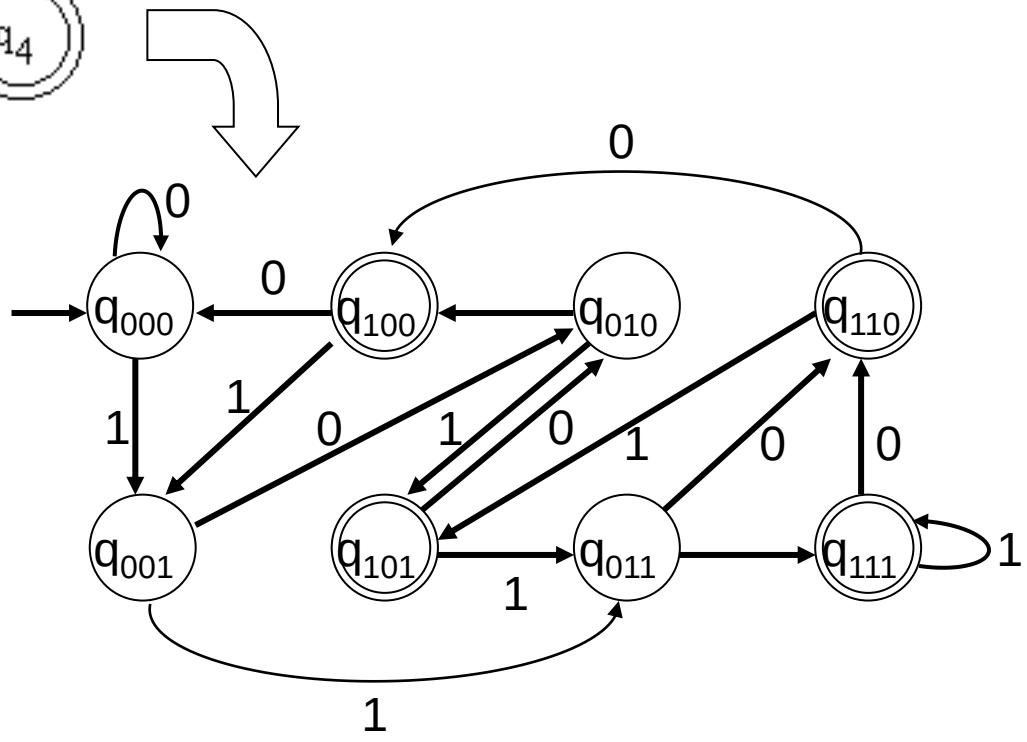


AFD M’

Equivalência – Não faça!



AFND N_2



AFD equivalente

Não é uma transformação, somente um AFD com a mesma linguagem.

AFND e linguagens regulares

- Uma linguagem é regular se e somente se é reconhecida por um AFND

AFND e linguagens regulares

- Uma linguagem é regular se e somente se é reconhecida por um AFND
  - Se a linguagem é regular existe um AFD que reconhece, todo AFD é um AFND

AFND e linguagens regulares

- Uma linguagem é regular se e somente se é reconhecida por um AFND
  - Se a linguagem é regular existe um AFD que reconhece, todo AFD é um AFND
  - Se a linguagem é reconhecida por um AFND, existe um AFD que simula o AFND, logo existe um AFD que a reconhece

Expressões regulares

Expressões regulares

- Assim como vimos para operações regulares...

- Aritmética
  - Objeto de estudo: números
  - Expressão aritmética: **(3 + 4)** $\times$ **5**
  - Expressão aritmética gera: número

- Teoria da computação
  - Objeto de estudo: linguagens
  - Expressão regular: **(0** $\cup$ **1)** $\cdot$ **0*** ou **(0** + **1)** $\cdot$ **0***
  - Expressão regular gera: linguagem

Uso

- Detecção de padrões
  - AWK, GREP
  - Editores de texto (p.ex., Microsoft Word)
  - Pearl, Java, Python
  - Validação de entrada
- Especificações de protocolo
- Compiladores (descrevendo expressões, identificadores, constantes)

Exemplos de Expressões Regulares

- E-mails válidos em formulários:
  [A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}

- Constantes numéricas em uma linguagem de programação
  (**+** ∪ **-** ∪ **λ**) (D^+ ∪ $D^+.D^*$ ∪ $D^*.D^+$) → 72, 3.14159, +7, -.01

- IPs em um busca em um arquivo de log
  \d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} → quatro números de 3 dígitos
  (25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?) → números de 0 a 255

  Exemplo de procura por IPs em grande volumes de dados:
  $ bin/hadoop jar hadoop-examples.jar grep input output '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'

  Ou, se para um único arquivo: ... | grep -E -o "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}"

Exemplo de entrada de log

50.97.138.111 - - [21/Sep/2014:19:08:48 -0300] "GET /index.php?option=com_content&view=article&id=22:workshop-testes-tolerancia-falhas&catid=10:workshop&Itemid=25 HTTP/1.0" 200 12404

50.97.138.111 - - [21/Sep/2014:19:08:49 -0300] "GET /administrator/templates/khepri/favicon.ico HTTP/1.0" 200 1150

207.46.13.86 - - [21/Sep/2014:19:09:30 -0300] "GET /index.php?option=com_content&view=category&layout=blog&id=10&Itemid=25 HTTP/1.1" 200 3389

157.55.39.93 - - [21/Sep/2014:19:16:19 -0300] "GET /index.php?option=com_content&view=article&id=22:workshop-testes-tolerancia-falhas&catid=10:workshop&Itemid=25 HTTP/1.1" 200 4527

37.59.69.33 - - [21/Sep/2014:19:21:15 -0300] "GET /wp-login.php HTTP/1.1" 404 210

37.59.69.33 - - [21/Sep/2014:19:21:16 -0300] "GET /administrator/index.php HTTP/1.1" 200

Exemplo de saída de log

```
366251  150.165.85.172
30484   46.45.32.79
25617   195.128.126.91
20067   216.144.247.50
18648   129.82.100.202
18643   142.4.17.151
18252   80.93.26.151
17977   37.1.223.254
15599   95.47.137.56
14258   213.251.189.203
12178   198.24.185.168
11127   142.4.211.151
10671   77.235.33.28
10168   91.121.165.158
```

Definição formal

- Seja $\sum$ um alfabeto
  1. Se a $\in \sum$, então a é uma expressão regular
  2. Se λ é a palavra nula, então λ é uma expressão regular
  3. Se $\varnothing$ é o conjunto vazio, então $\varnothing$ é uma expressão regular

Definição formal

- Seja $\sum$ um alfabeto
  1. Se a $\in \sum$, então **a** é uma expressão regular
  2. Se λ é a palavra nula, então λ 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
     $(R_1 \cup R_2)$ e $(R_1 \bullet R_2)$ são expressões regulares
  5. Se R_1 é uma expressão regular, então (R_1^*) é uma expressão regular

Expressões e linguagens: ERs descrevem conjuntos de palavras...

Expressão

- 0*10*

Expressões e linguagens

Expressão

- 0*10*

Linguagem

- w contém um 1

Expressões e linguagens: ERs descrevem conjuntos de palavras...

Expressão

- 0*10*

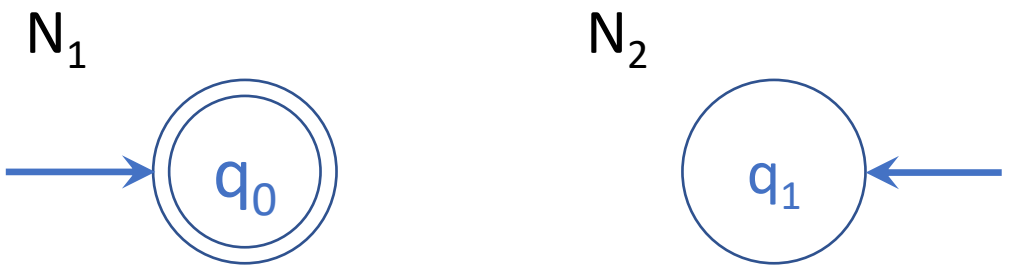
Aqui na realidade são várias expressões combinadas!

Outras definições

- λ versus $\varnothing$
  - λ: antes representava uma palavra, a palavra vazia, mas aqui, como ER, é a ER que tem como linguagem um conjunto unitário, cujo único elemento é a palavra vazia
  - $\varnothing$: aqui, como ER, representa um conjunto vazio, então não tem nenhuma palavra

Outras definições

- λ versus $\varnothing$
  - λ: como ER descreve uma linguagem unitária
  - $\varnothing$: nenhuma palavra, conjunto vazio



Outras definições

- Linguagem descrita por R: L(R)
  - Equivalência: $R_1 \equiv R_2 \leftrightarrow L(R_1) = L(R_2)$
  - $R^+ = RR^*$, ou seja, $R^* = R^+ \cup \{\lambda\}$

Expressões e linguagens

- Se $\sum = \{0, 1\}$
  - $R_1 = \sum \rightarrow L(R_1) = \{0, 1\}$
  - $R_2 = (0 \cup 1) \rightarrow L(R_2) = \{0, 1\}$
  - $R_1 \equiv R_2$
  - $R_3 = \sum^* \rightarrow L(R_3) = \{$todas as palavras sobre o alfabeto, incluindo $\lambda\}$
- $R_4 = 1^*.\varnothing \rightarrow$?
- $R_5 = \varnothing^* \rightarrow$?

Expressões e linguagens

- Se $\sum = \{0, 1\}$
  - $R_1 = \sum \rightarrow L(R_1) = \{0, 1\}$
  - $R_2 = (0 \cup 1) \rightarrow L(R_2) = \{0, 1\}$
  - $R_1 \equiv R_2$
  - $R_3 = \sum^* \rightarrow L(R_3) = \{$todas as palavras sobre o alfabeto, incluindo $\lambda\}$
- $R_4 = 1^*. \varnothing \rightarrow L(R_4) = \varnothing$
- $R_5 = \varnothing^* \rightarrow L(R_5) = \{\lambda\}$

Expressões e linguagens

Expressão

- 0^*10^*
- $\Sigma^*1\,\Sigma^*$
- $\Sigma^*001\,\Sigma^*$
- $(01^+)^*$
- $(\Sigma\,\Sigma)^*$
- $(01 \cup 10)$
- $0\Sigma^*0 \cup\ 1\Sigma^*1 \cup 0 \cup 1$

Expressões e linguagens ($\Sigma = \{0,1\}$)

Expressão

- 0^*10^*
- $\Sigma^*1\,\Sigma^*$
- $\Sigma^*001\,\Sigma^*$
- $(01^+)^*$
- $(\Sigma\,\Sigma)^*$
- $(01 \cup 10)$
- $0\Sigma^*0 \cup\ 1\Sigma^*1 \cup 0 \cup 1$

Linguagem

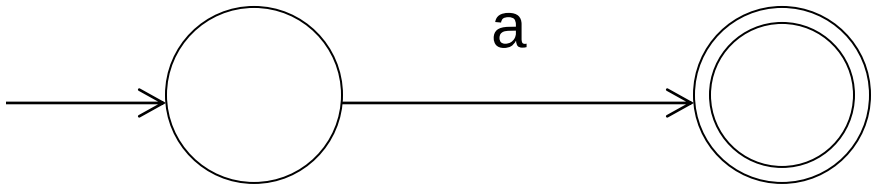
- w contém um 1
- w contém ao menos um 1
- w contém uma sequência 001
- Todo 0 é seguido de um 1
- w tem comprimento par
- {01, 10}
- w começa e termina com o mesmo símbolo

Expressão regular vs. autômato finito

- Teorema 3: uma linguagem é regular se e somente se ela pode ser descrita por uma expressão regular
  - Isso significa que ERs e AFs são equivalentes

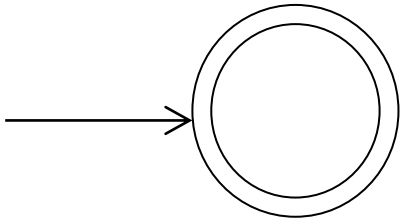
Expressão regular vs. autômato finito

- R = **a** , para algum a em Σ
  - Então, L(R) = {a}
  - O seguinte AFND também reconhece L(R)



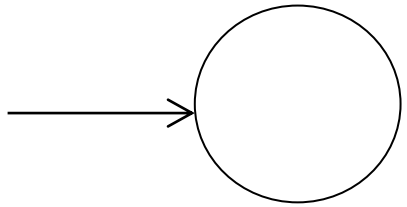
Expressão regular vs. autômato finito

- R = λ
  - Então, L(R) = {λ}
  - O seguinte AFND também reconhece L(R)



Expressão regular vs. autômato finito

- R = ∅
  - Então, L(R) = ∅
  - O seguinte AFND também reconhece L(R)



Novamente: expressões regulares complexas

- Seja $\sum$ um alfabeto
  1. Se a $\in \sum$, 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 $(R_1 \cup R_2)$ e $(R_1 \bullet R_2)$ são expressões regulares.
  5. Se R_1 é uma expressão regular, então (R_1^*) é uma expressão regular.

Expressão regular vs. autômato finito

- As regras restantes: ERs podem ser geradas a partir da...
  - União
  - Concatenação
  - Ou operação estrela sobre outras ERs

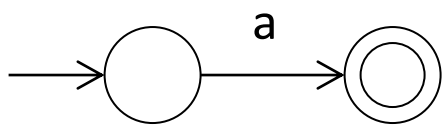
- A classe de linguagens regulares também é fechada por essas mesmas operações
  - Usar essas operações para gerar expressões maiores gera linguagens regulares
  - Sabemos gerar AFNDs que reconhecem linguagens geradas por essas operações regulares

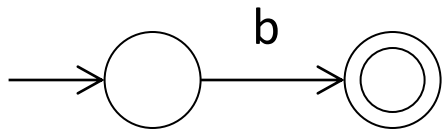
Expressão regular vs. autômato finito

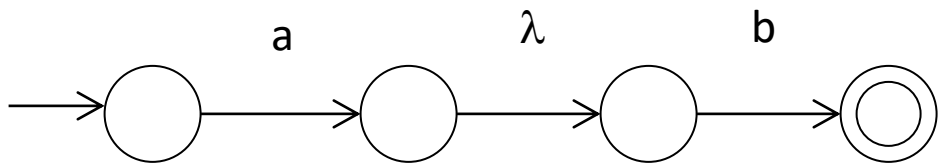
- Conclusões
  - Expressões podem ser convertidas em autômatos
  - Logo, expressões geram linguagens regulares

Exemplo

- Converter (**ab** ∪ **a**)* em um AFND
  - **a**
  - **b**
  - **ab**



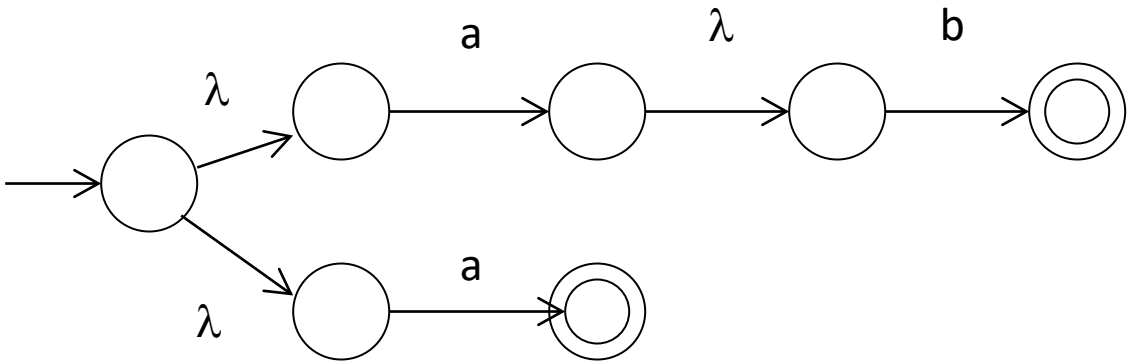




Exemplo

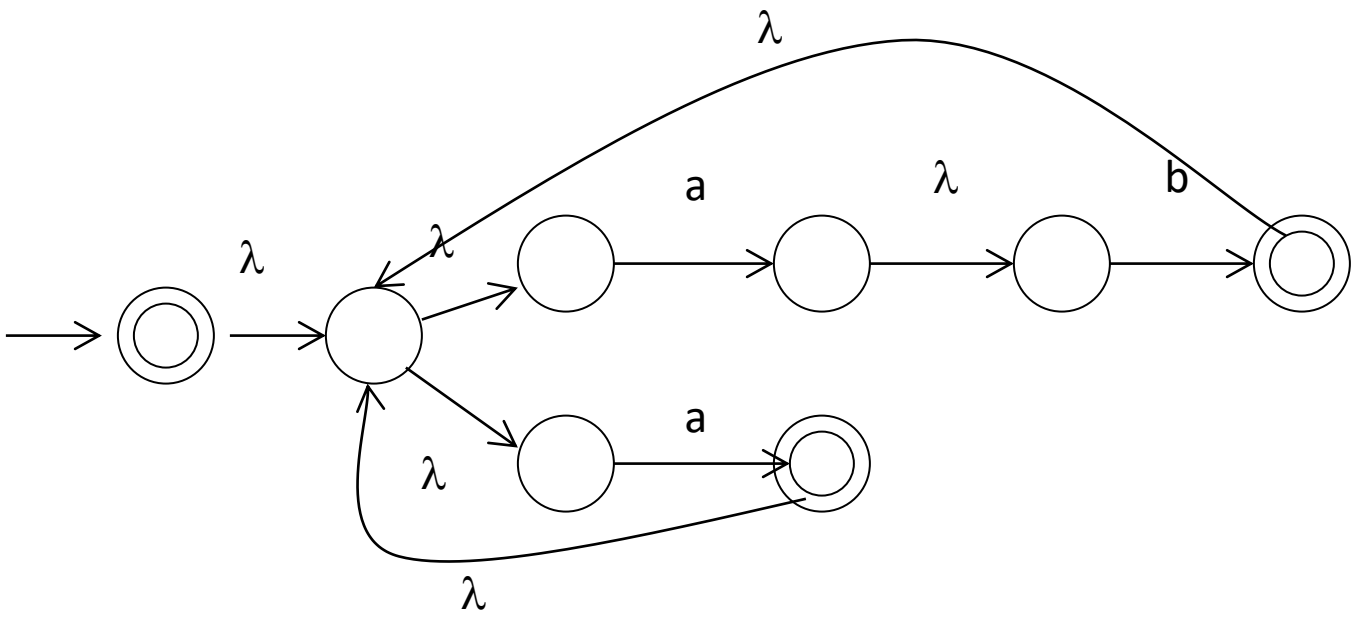
- Converter (**ab** ∪ **a**)* em um AFND

  - **ab** ∪ **a**



Exemplo

- Converter (**ab** ∪ **a**)* em um AFND



AFND → ER

Equivalência AFND→ER

- A transformação nem sempre é intuitiva, precisamos de um processo

- Mas em geral queremos converter o autômato...
  - ... eliminado os estados um a um até que sobre somente dois estados
  - Estes estados são um estado inicial e um final
  - O estado inicial tem uma única transição, saindo dele
  - O estado final tem uma única transição, chegando nele
  - Quando eliminarmos um estado, vamos compensar a remoção adicionando símbolos a outras transições diretas
  - As transições podem ter agora expressões regulares, que vão crescendo para representar partes cada vez maiores do autômatos