Teoria da Computação

Autômatos Finitos Não Determinísticos (AFN)

Felipe Cunha

Material extraído do livro e slides do Prof. Newton Vieira (http://dcc.ufmg.br/~nvieira) e do prof. Andrei

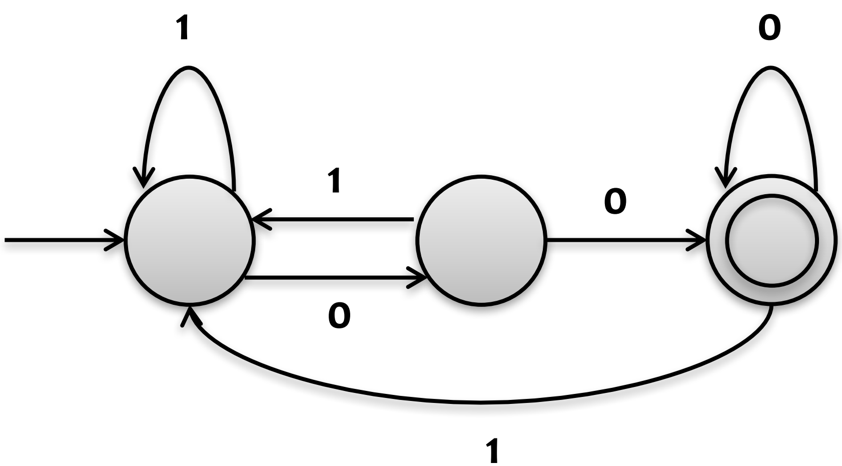
Sumário

- Introdução
- Autômatos Finitos Não Determinísticos (AFN)
- Equivalência entre AFD's e AFN's
- AFN estendido

INTRODUÇÃO

Introdução

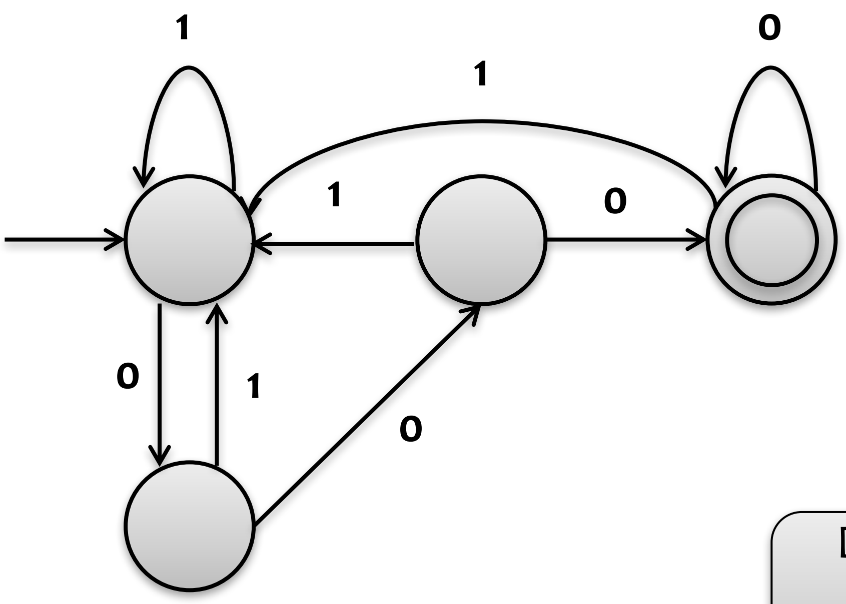
- Considere o AFD para a linguagem $L = \{0,1\}^*\{00\}$



E para a linguagem
$L = \{0,1\}^*\{000\}$

Introdução

- Considere o AFD para a linguagem $L = \{0,1\}^*\{000\}$



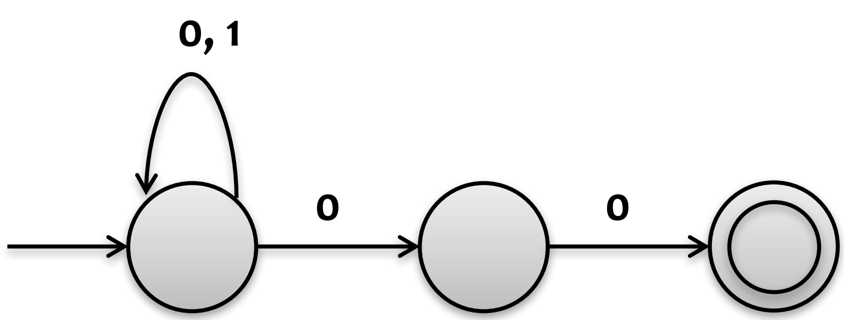
Deveria ser mais fácil, não?

Introdução

- Em ambos os exemplos a maior parte da lógica foi para evitar que a palavra terminasse com 00 ou 000
  - Mas parece tão fácil reconhecer que uma palavra termina com 00 ou 000

- Uma definição mais natural seria

O que têm de diferente aqui?

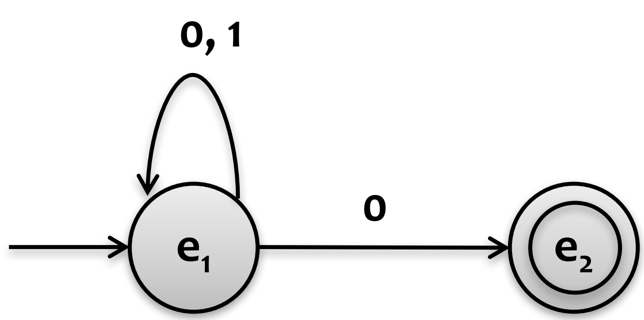


$$L = \{0,1\}^*\{00\}$$

Autômato Finito Não Determinístico (AFN)

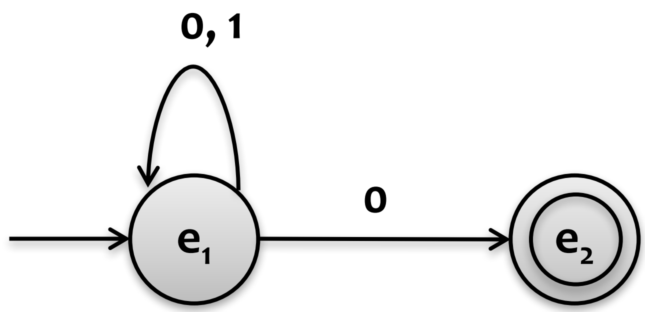
AFN

- Em um autômato finito determinístico (AFD), cada par (estado, símbolo) é uma transição para um **único estado**
- Se está restrição for eliminada, se para algum par (estado, símbolo) houver transições para dois ou mais estados, então têm-se um **autômato finito não determinístico** (**AFN**)



AFN

- O não determinismo se verifica pela **indecisão** associada ao estado e_1, que possui duas transições sob o símbolo 0



- Para a palavra 1010 têm-se as seguintes computações possíveis, dependendo do caminho tomado

$$
\begin{array}{lllllllll}
[e_1, 1010] & \vdash & [e_1, 010] & \vdash & [e_1, 10] & \vdash & [e_1, 0] & \vdash & [e_1, \lambda] \\
 & & & \vdash & [e_2, 10] & & & \vdash & [e_2, \lambda]
\end{array}
$$

Quando um AFN reconhece uma palavra?

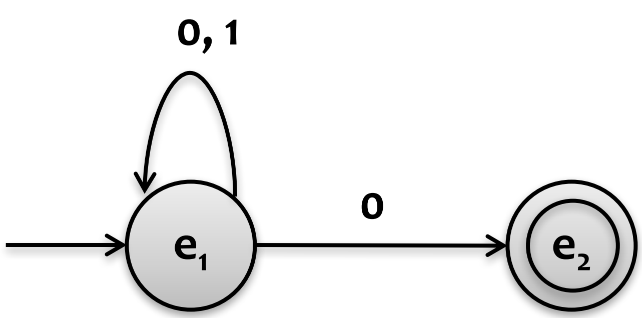
AFN

- Com relação ao reconhecimento para AFN
  - Uma palavra é reconhecida se, e somente se, **existe** uma computação que a consome e termina em estado final
  - Em todo ponto de indecisão, a máquina **adivinha** qual escolha (se houver alguma) leva a uma computação que resulta em sucesso no reconhecimento

AFN

- Um AFN é uma quíntupla $(E, \Sigma, \delta, I, F)$, em que
  - E é um conjunto finito não vazio de estados
  - Σ é um alfabeto
  - $\delta : E \times \Sigma \rightarrow \mathrm{P}(E)$ é a função de transição (função total)
  - $I \subseteq E$ é um conjunto não vazio de estados iniciais
  - $F \subseteq E$ é um conjunto de estados finais

- Exemplo de AFN



$$(\{e_1, e_2\}, \{0,1\}, \delta, \{e_1\}, \{e_2\})$$

δ	0	1
e_1	$\{e_1, e_2\}$	$\{e_1\}$
e_2	$\varnothing$	$\varnothing$

Linguagem Reconhecida por AFN

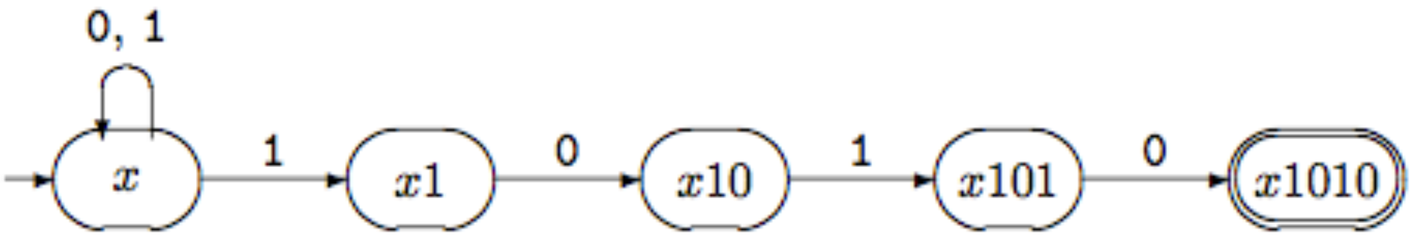
- A linguagem reconhecida por um AFN $M = (E, \Sigma, \delta, I, F)$ é dada por

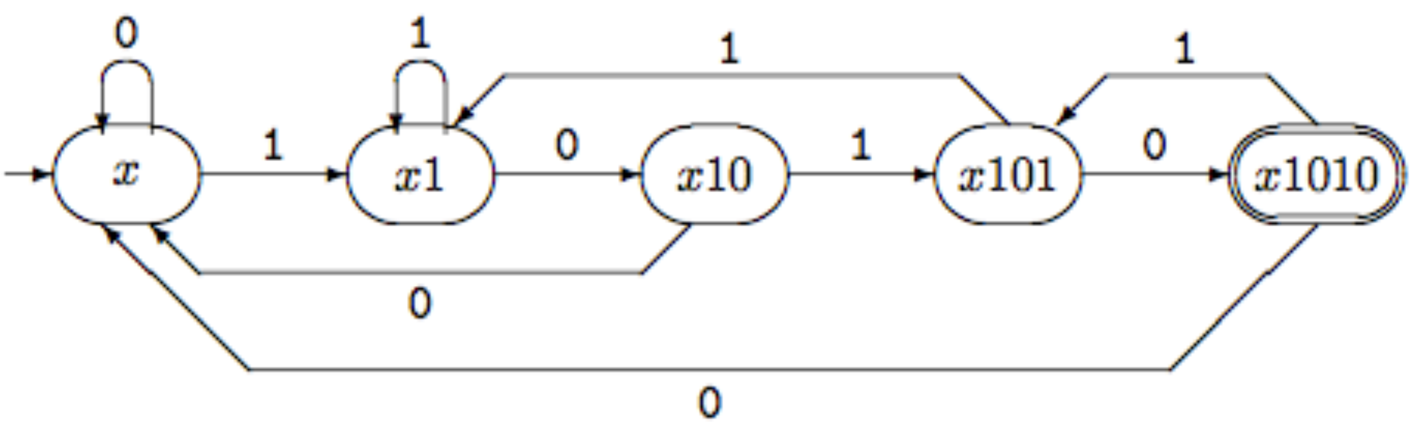
$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(I, w) \cap F \neq \phi\}$$

AFN's possuem maior poder computacional que AFD's?

AFN x AFD

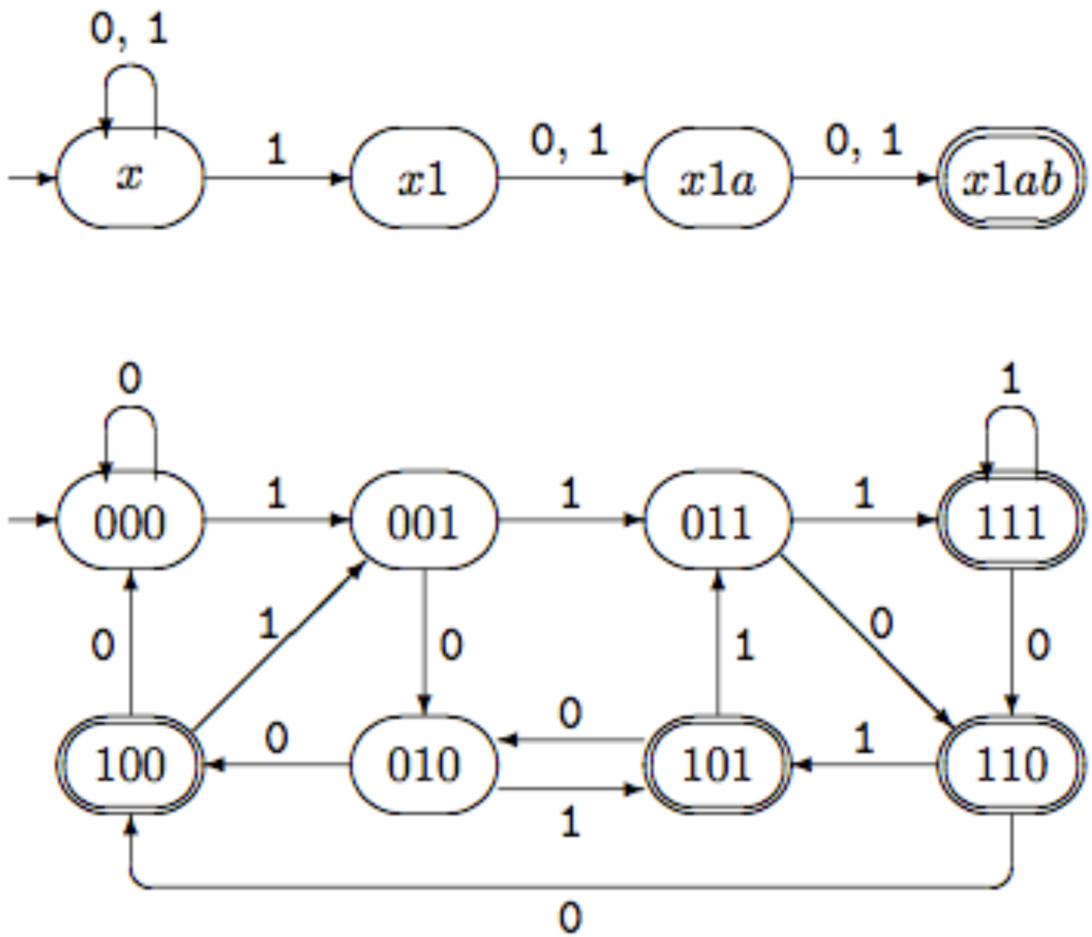
- AFN e AFD para a linguagem $L = \{0,1\}^* \{1010\}$





AFN x AFD

- AFN e AFD para a linguagem $L = \{0,1\}^* \{1\}\{0,1\}\{0,1\}$



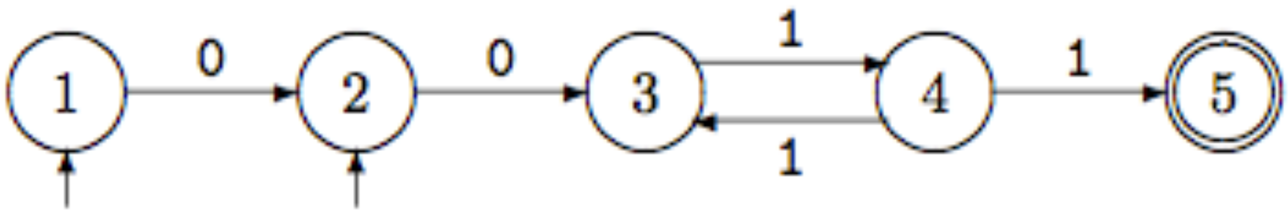
Equivalência Entre AFN's e AFD's

Equivalência Entre AFN's e AFD's

- **Teorema:** para qualquer AFN existe um AFD equivalente
  - A ideia é que um estado será um conjunto, significando todos os estados do AFN atingidos por todas as computações possíveis para a mesma palavra
- Seja um AFN $M = (E, \Sigma, \delta, I, F)$, o AFD equivalente é dado por $M' = (\mathrm{P}(E), \Sigma, \delta', I, F')$, onde
  - $\delta'(\phi, a) = \phi, \forall a \in \Sigma^*$
  - $\delta'(X, a) = \bigcup_{e \in X} \delta(e, a), \forall a \in \Sigma, X \subseteq E$
  - $F' = \{X \subseteq E \mid X \cap F \neq \varnothing\}$

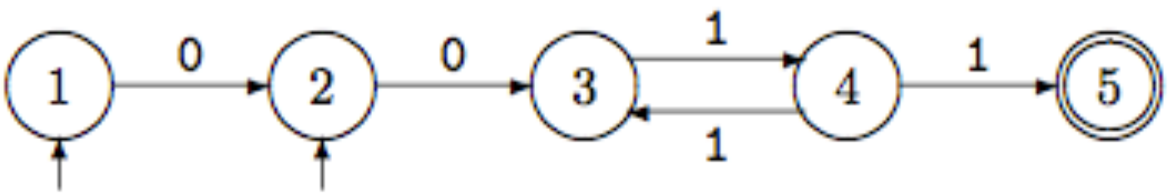
Equivalência Entre AFN's e AFD's

- Obtenha o diagrama de estados do AFD equivalente ao AFN dado a seguir

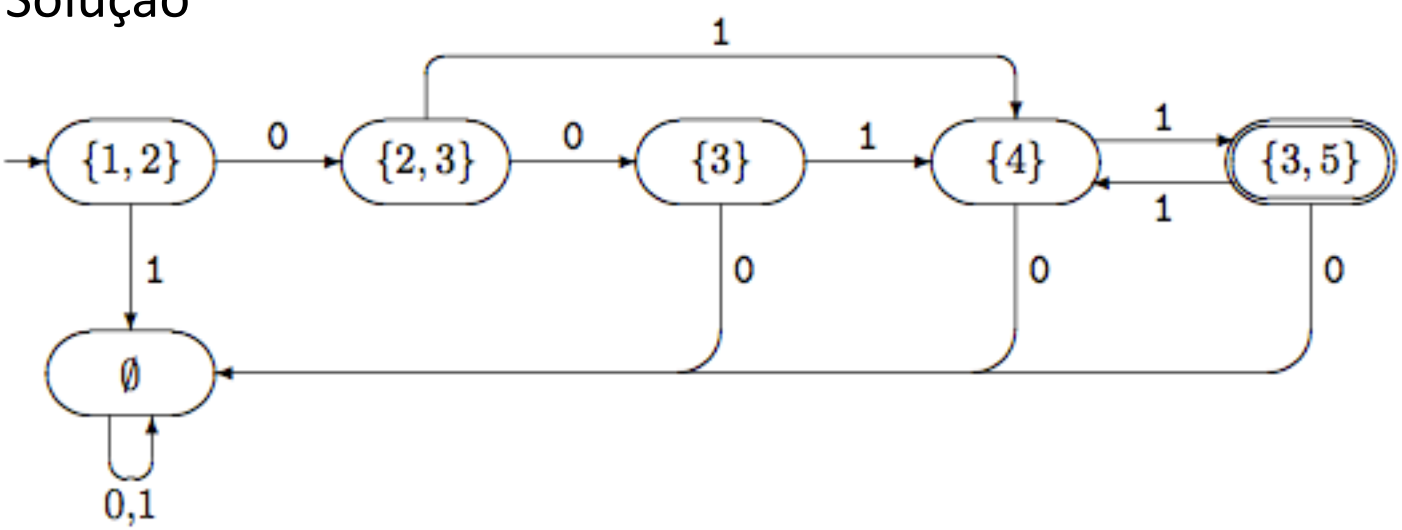


Equivalência Entre AFN's e AFD's

- Obtenha o diagrama de estados do AFD equivalente ao AFN dado a seguir



- Solução

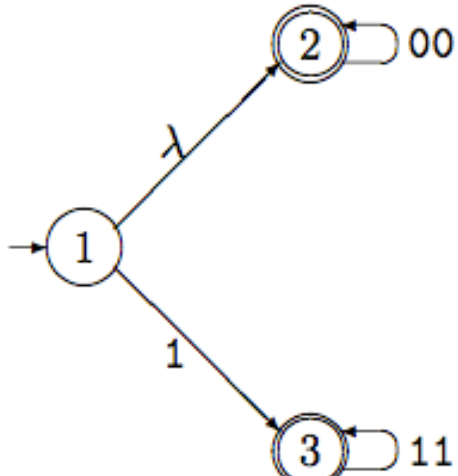


AFN ESTENDIDO

AFN Estendido

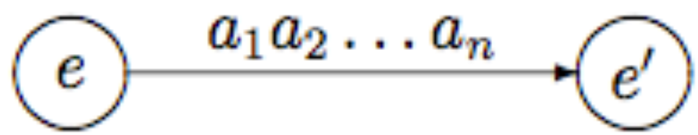
- Um AFN **estendido** é uma quíntupla $(E, \Sigma, \delta, I, F)$, em que
  - E, Σ, I, F são como em AFN's
  - δ é uma função parcial $E \times D \rightarrow P(E)$, em que D é algum subconjunto finito de Σ^*

- Exemplo: AFNE para $\{w \in \{0\}^* \mid |w| \text{ é par}\} \cup \{w \in \{1\}^* \mid |w| \text{ é ímpar}\}$

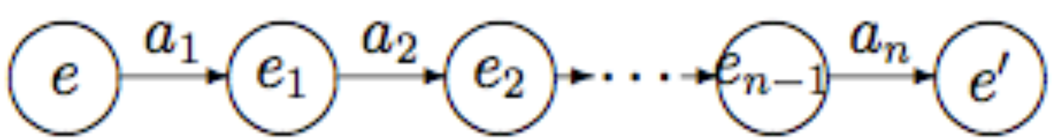


AFN Estendido

- Uma transição da forma



pode ser substituída por n transições



AFN λ

- Um autômato finito não determinístico com transições λ (AFN λ) é uma quíntupla $M = (E, \Sigma, \delta, I, F)$, onde
  - E, Σ, I, F são como em AFN's
  - δ é uma função total $E \times (\Sigma \cup \{\lambda\}) \rightarrow P(E)$

- A função **fecho** λ para M, $f\lambda$: $P(E) \rightarrow P(E)$ é definida recursivamente como
  a) $X \subseteq f\lambda(X)$
  b) se $e \in f\lambda(X)$, então $\delta(e, \lambda) \subseteq f\lambda(X)$

AFN λ

- A **função de transição estendida** para um AFN λ $M = (E, \Sigma, \delta, I, F)$, $\delta : \mathrm{P}(E) \times \Sigma^* \rightarrow P(E)$ é definida recursivamente como segue

  a) $\hat{\delta}(\phi, w) = \phi, \forall w \in \Sigma^*$

  a) $\hat{\delta}(A, \lambda) = f\lambda(A), \forall A \subseteq E$

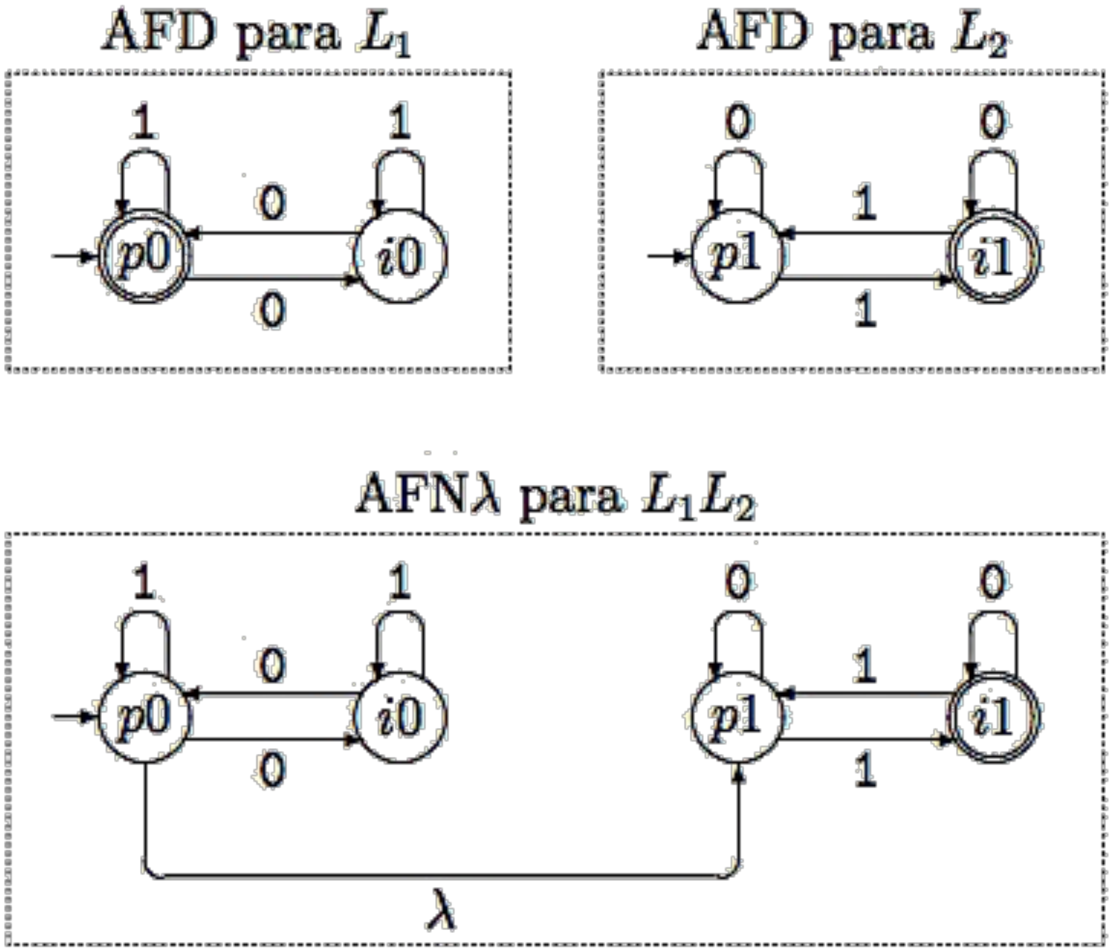
  a) $\hat{\delta}(A, ay) = \hat{\delta}(\bigcup_{e \in f\lambda(A)} \delta(e, a), y), \forall A \subseteq E, a \in \Sigma, y \in \Sigma^*$

- A linguagem reconhecida por um AFN λ é dada por

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(I, w) \cap F \neq \phi\}$$

AFN λ

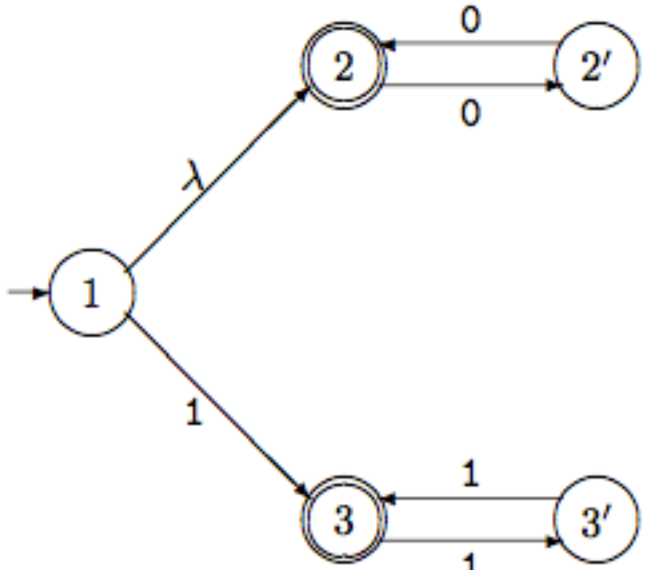
- Exemplo de AFN λ



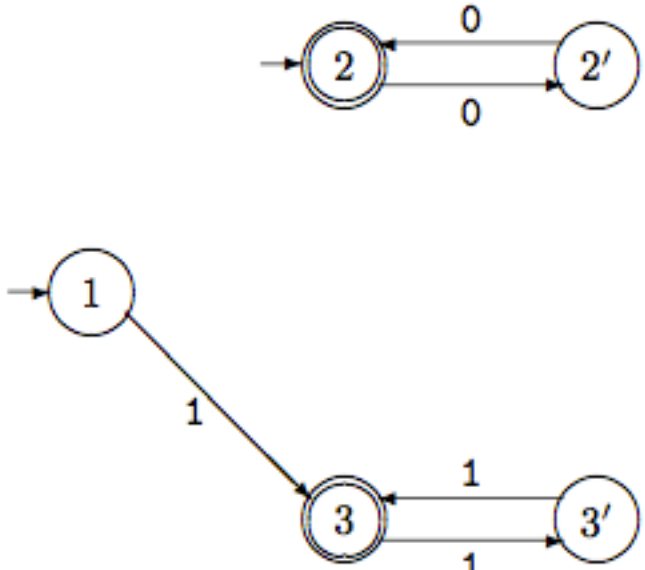
Equivalência entre AFN λ e AFN

- Seja um AFN λ $M = (E, \Sigma, \delta, I, F)$, o AFN equivalente é dado por $M' = (E, \Sigma, \delta', I', F)$, onde
  - $I' = f\lambda(I)$
  - $\delta'(e,a) = f\lambda(\delta(e,a)), \forall e \in E, a \in \Sigma$

- Exemplo



(a) AFNλ

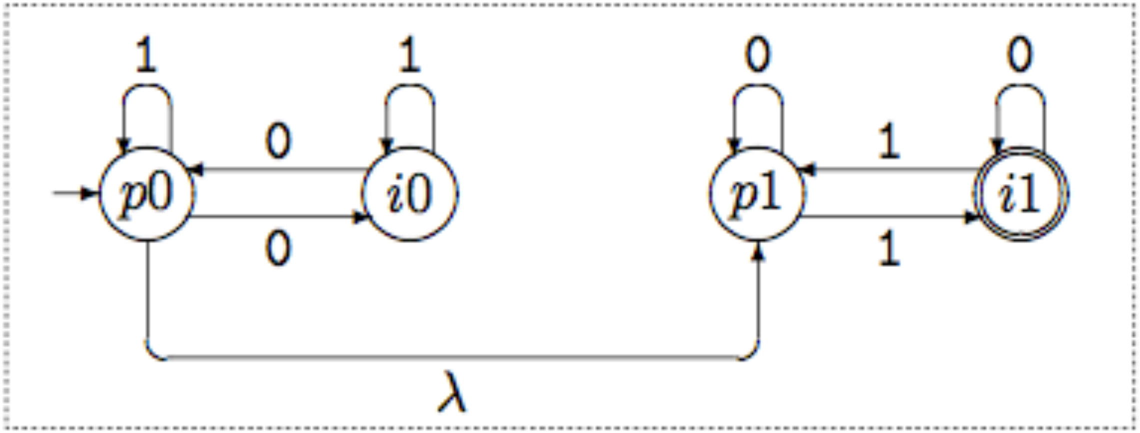


(b) AFN

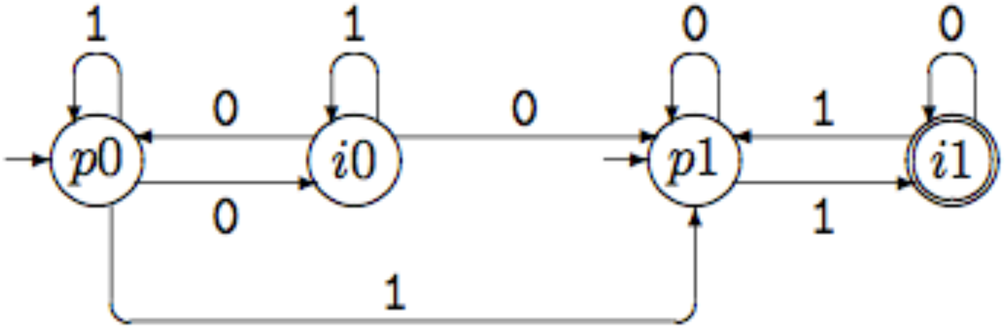
Equivalência entre AFN λ e AFN

- Outro exemplo

AFNλ para L_1L_2



AFN para L_1L_2



Relação Entre Autômatos Finitos

- O diagrama abaixo ilustra a relação entre autômatos finitos