Automatos Finitos

EDUARDO FREIRE NAKAMURA

Instituto de Computação
Universidade Federal do Amazonas
nakamura@icomp.ufam.edu.br

[1]Este material utiliza conteúdo das aulas fornecidas pelo Prof. Vilar da Câmara Neto (disponível em http://http://prof.vilarneto.com).
[2]Permissão de uso fornecida pelos autores.
[3]As figuras utilizadas neste material são de domínio público, disponíveis na Internet sem informações de direitos autorais.

Autômatos Finitos Determinísticos

OBJETIVO

COMPREENDER E PROJETAR AFDS

Introdução

- Os autômatos são formalismos particularmente adequados para o reconhecimento de linguagens

- Dentre os diversos tipos de autômatos existentes, neste curso abordaremos os mais conhecidos
  - Autômatos Finitos Determinísticos, ou AFDs
  - Autômatos Finitos Não Determinísticos, ou AFNs
  - Autômatos Finitos Não Determinísticos Estendidos, ou AFNEs
  - Autômatos Finitos Não Determinísticos com Transições λ, ou AFNλs
  - Autômatos de Pilha Determinísticos, ou APDs;
  - Autômatos de Pilha Não Determinísticos, ou APNs;
  - Máquinas de Turing, ou MTs

Autômatos finitos determinísticos (AFDs)

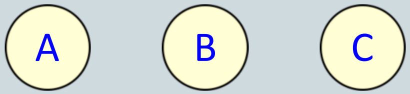
- "Máquinas" virtuais com poder de processamento extremamente limitado
- Um AFD é composto por
  - Um conjunto finito e não vazio de estados
  - Sempre se encontra em um (e somente um) estado a cada instante
  - Um dos estados é o estado inicial e estabelece o estado da máquina no início do seu funcionamento
- O AFD é alimentado com uma palavra de entrada, composta por uma sequência arbitrária de símbolos do alfabeto

Autômatos finitos determinísticos (AFDs)

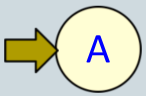
- A passagem de um estado para outro é determinada por regras de transição
- A cada passo, o primeiro símbolo da palavra é consumido e a máquina adota o novo estado determinado pela regra de transição
- A máquina pára quando todos os símbolos da palavra de entrada são consumidos
- Alguns estados podem ser finais
  - Após consumir toda a palavra de entrada, se a máquina pára em um estado final, então diz-se que a máquina reconhece (ou aceita) a palavra de entrada

Autômatos finitos determinísticos (AFDs)

- Um estado é representado por uma circunferência contendo um rótulo (seu nome)

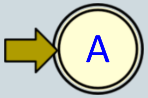


- O estado inicial é identificado por uma seta



- Cada estado final é identificado por borda dupla

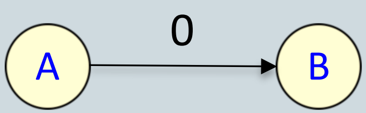
- Um estado pode ser ao mesmo tempo inicial e final



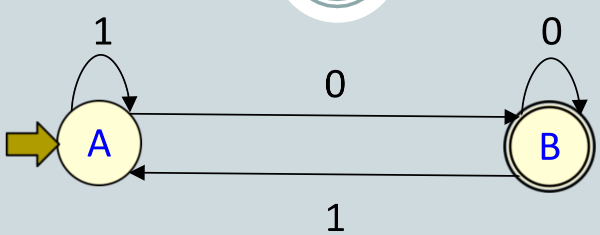
Autômatos finitos determinísticos (AFDs)

- Uma transição é representada por uma seta

- Diz-se que esta transição consome o símbolo 0 e leva a máquina do estado A para o estado B



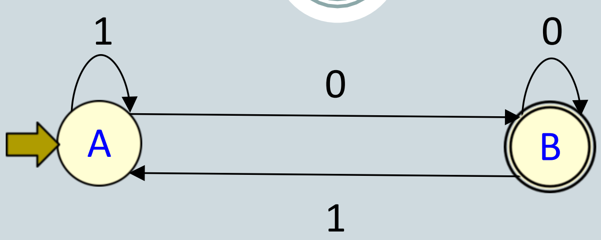
Exemplo



- Os estados são A e B
- O estado Inicial é A
- O estado final é B

- Há quatro transições
  1. De A, consome 0 e vai para B
  2. De A, consome 1 e vai para A
  3. De B, consome 0 e vai para B
  4. De B, consome 1 e vai para A

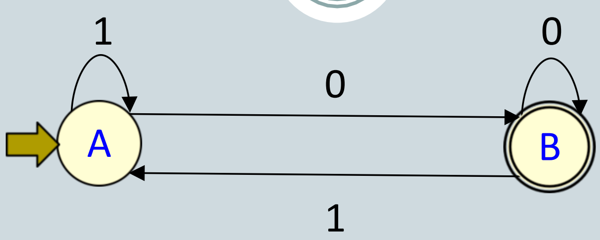
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

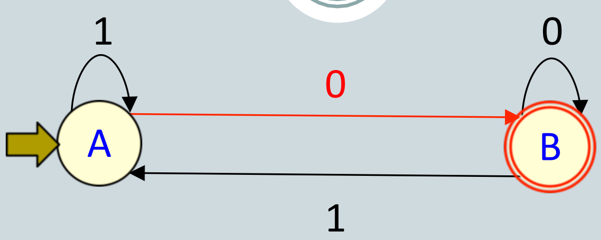
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	<u>0</u>01100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

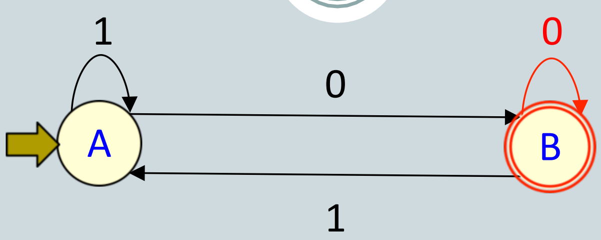
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

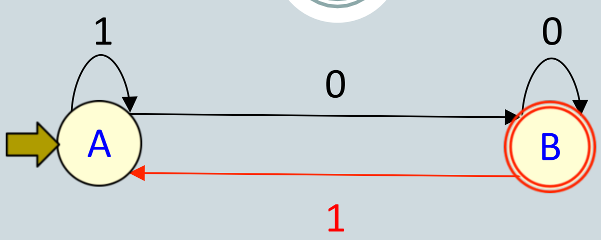
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

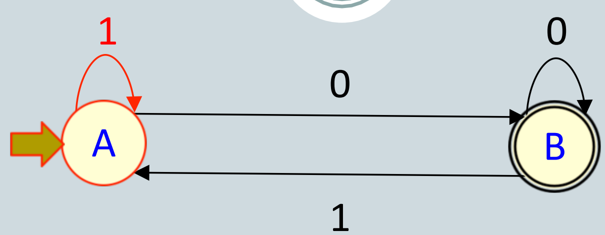
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

Exemplo

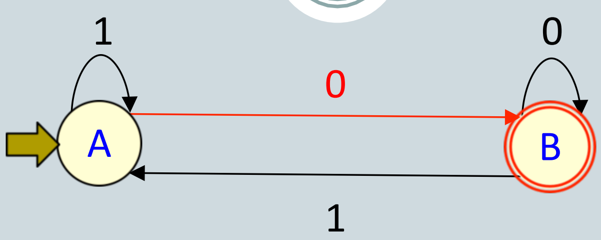


Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

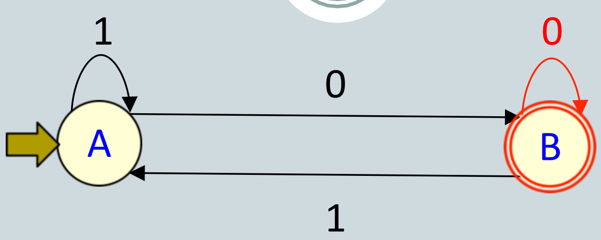
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

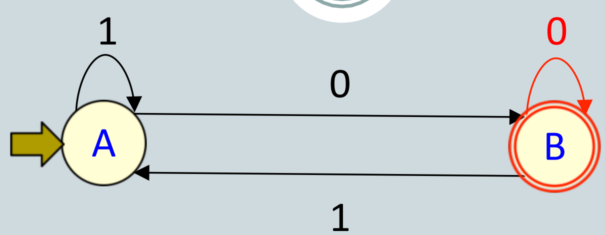
Exemplo



Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	A	100
1	A	00
0	B	0
0	B	λ

Exemplo

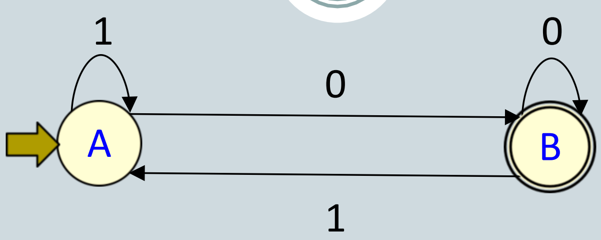


Exemplo para consumo da palavra 001100

Após consumir	chega-se ao estado	e resta consumir
λ	A	001100
0	B	01100
0	B	1100
1	[illegible]	[illegible]
1	A	00
0	B	0
0	B	λ

AFD pára no estado B, final. Portanto, a palavra é aceita!

Exemplo

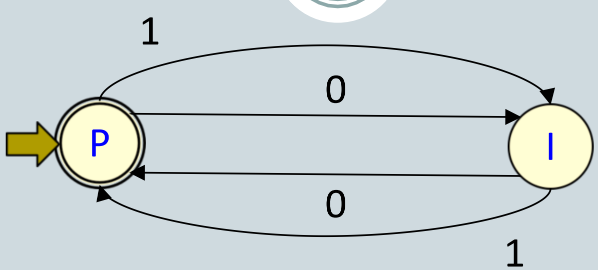


- Observações sobre a máquina
  - Somente chega no estado B após consumir um símbolo 0
  - Retorna incondicionalmente ao estado A quando consome um símbolo 1

- Qual linguagem é reconhecida por este AFD?

$$\{0,1\}^*\{0\}$$

Exemplo

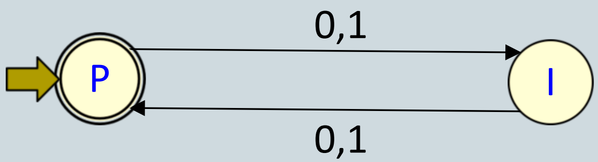


- Este AFD intercala entre os estados P e I, independente do símbolo consumido

- A única maneira de chegar a um estado final (no caso, P) é consumir um número par de símbolos

$$L = (\{0,1\}^2)^*$$

- Uma forma mais compacta, consolidando transições



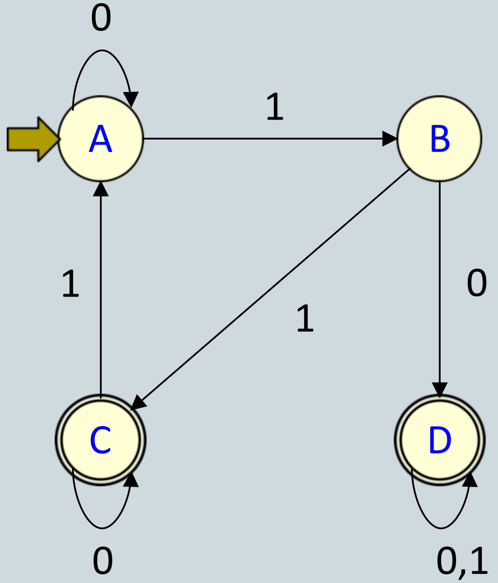
Autômatos finitos determinísticos (AFDs)

- Em um AFD, é obrigatório que
  - Todos os estados tenham regras de transição partindo deles para cada um dos símbolos do alfabeto
  - Não haja mais de uma transição que parta do mesmo estado e que consuma o mesmo símbolo

- Essas restrições garantem que
  - Enquanto houver símbolos a serem consumidos, é sempre possível realizar um passo de execução
  - Um passo de execução sempre leva a máquina a um único estado

Exemplo 1

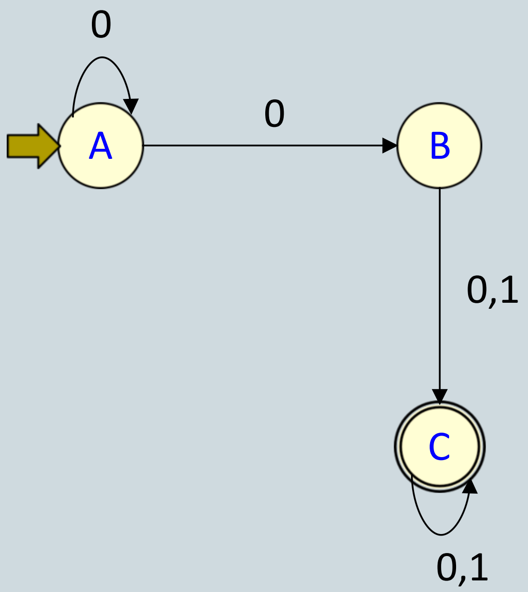
- Considerando Σ = {0,1}
- A máquina de estados ao lado é um AFD?
- Por quê?



Sim, pois há transições saindo de cada estado para cada símbolo, sem repetições

Exemplo 2

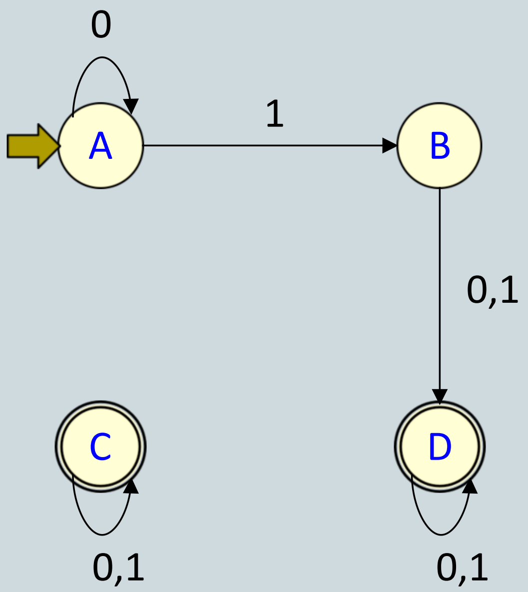
- Considerando $\Sigma = \{0,1\}$
- A máquina de estados ao lado é um AFD?
- Por quê?



Não, há duas transições saindo de A, consumindo o símbolo 0

Exemplo 3

- Considerando Σ = {0,1}
- A máquina de estados ao lado é um AFD?
- Por quê?



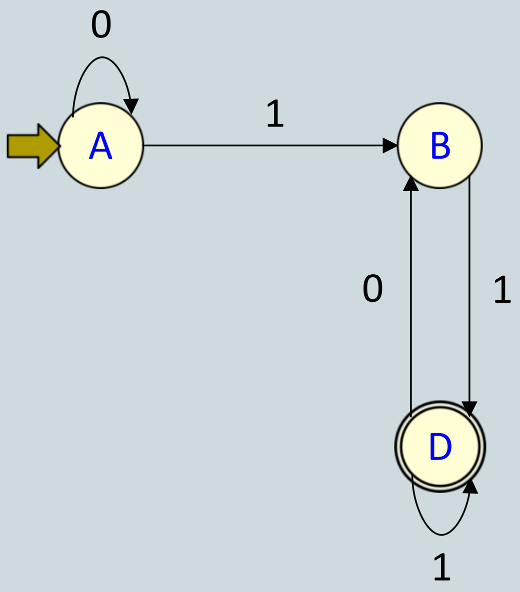
Sim, não há obrigação de que todos os estados estejam conectados entre si

Exemplo 4

- Considerando Σ = {0,1}

- A máquina de estados ao lado é um AFD?

- Por quê?



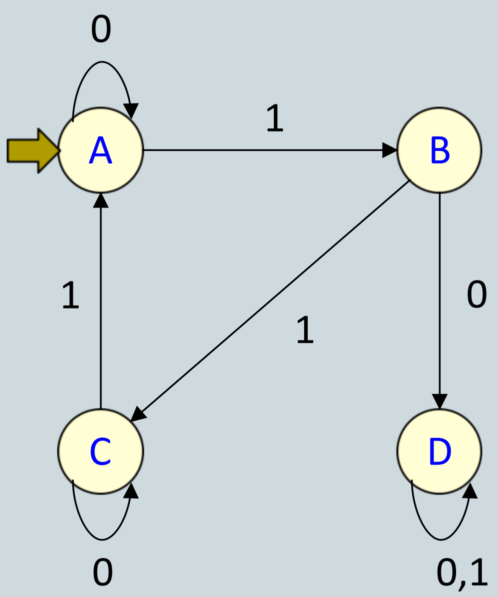
Não, falta a transição de saída de B consumindo o símbolo 0

Exemplo 5

- Considerando Σ = {0,1}

- A máquina de estados ao lado é um AFD?

- Por quê?



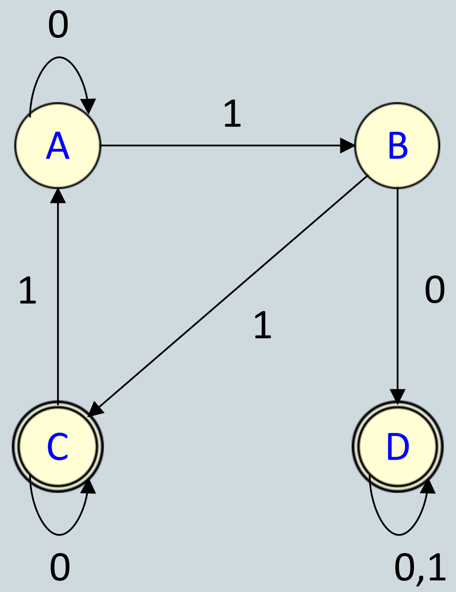
Sim, pois não há obrigação de ter estados finais

Exemplo 6

- Considerando Σ = {0,1}

- A máquina de estados ao lado é um AFD?

- Por quê?



Não, pois não há estado inicial

Configuração instantânea

- A configuração instantânea de uma máquina é o conjunto de informações que representa uma “fotografia” da execução dessa máquina em um determinado instante

- No caso de um AFD, a configuração instantânea engloba
  - O estado atual
  - A sequência de símbolos que falta consumir

Configuração instantânea

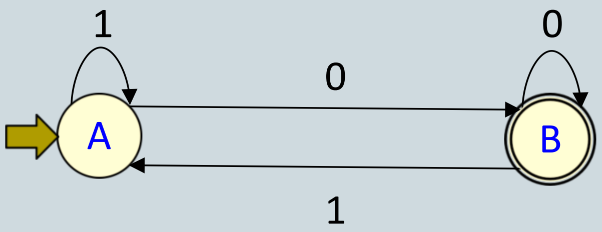
- Voltando a exemplo inicial
  - Palavra a consumir 001100

- Configuração instantânea inicial é

  (A, 001100)

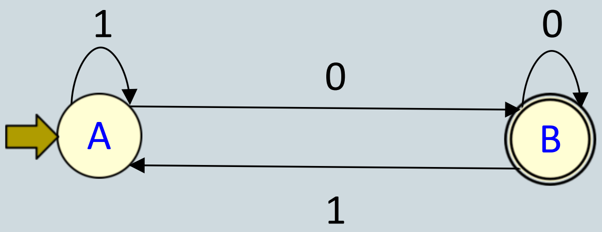
- Após consumir o primeiro símbolo da palavra (0)

  (B, 01100)



Configuração instantânea

- Voltando a exemplo inicial
  - Palavra a consumir 001100
- Diz-se que
  - A configuração (A, 001100) leva à configuração (B, 01100)
- Este passo de execução é representado pelo símbolo ⊢, que significa “leva a”
  - (A, 001100) ⊢ (B,01100)



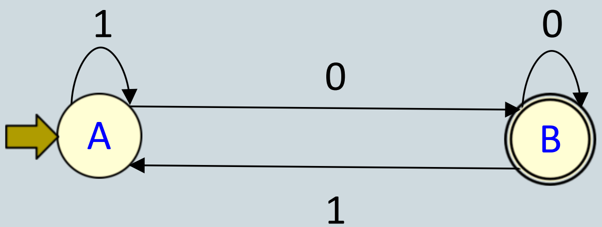
Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Configuração instantânea

- Voltando a exemplo inicial
  - Palavra a consumir 001100

- A sequência de execução é

$$
\begin{aligned}
(A, 001100) &\vdash (B, 01100)\\
&\vdash (B, 1100)\\
&\vdash (A, 100)\\
&\vdash (A, 00)\\
&\vdash (B, 0)\\
&\vdash (B, \lambda)
\end{aligned}
$$



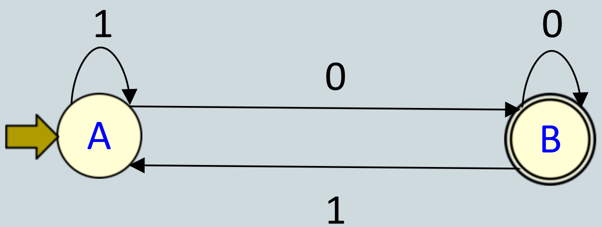
Configuração instantânea

- Voltando a exemplo inicial
  - Palavra a consumir 001100

- Ou simplesmente

$$(A, 001100) \vdash^{6} (B, \lambda)$$

que significa que (A, 001100) leva a (B, λ) em 6 passos



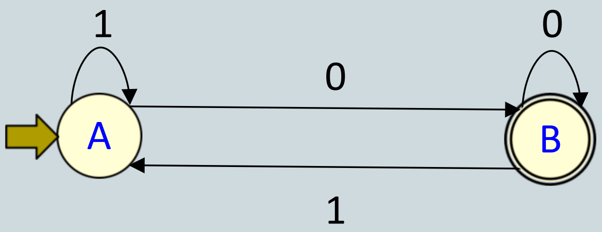
Configuração instantânea

- Voltando a exemplo inicial
  - Palavra a consumir 001100

- Em geral o número de passos não importa, podendo escrever

$$(A, 001100) \overset{*}{\vdash} (B, \lambda)$$

que significa que (A, 001100) leva a (B, λ)

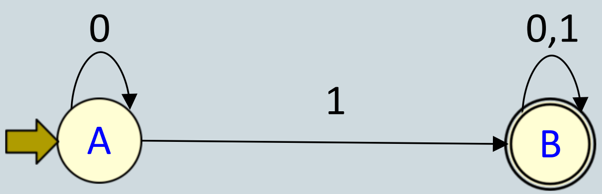


Exemplo 1

- Qual linguagem é reconhecida pelo AFD?

$$
\begin{aligned}
(A, p1q) &\overset{*}{\vdash} (A, 1q)\\
&\vdash (B, q)\\
&\overset{*}{\vdash} (B, \lambda)
\end{aligned}
$$

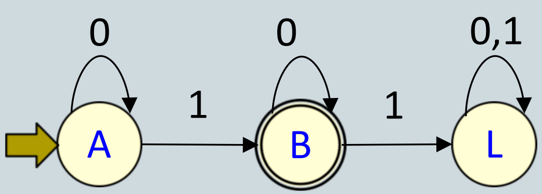
para $p \in \{0\}^*$ e $q \in \{0,1\}^*$



Exemplo 2

- Qual linguagem é reconhecida pelo AFD?
  - O estado L serve como "lixeira"
  - Se o AFD entra neste estado, que não é final, fica "preso"

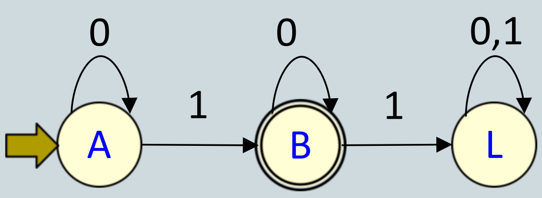
$$
\begin{aligned}
(A, p1q) &\overset{*}{\vdash} (A, 1q) && \text{para } p \in \{0\}^* \\
&\vdash (B, q) && \\
&\overset{*}{\vdash} (B, \lambda) && \text{para } q \in \{0\}^*
\end{aligned}
$$



Exemplo 2

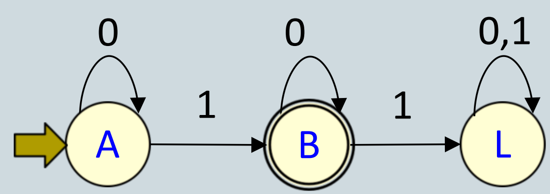
- Qual linguagem é reconhecida pelo AFD?
  - O estado L serve como “lixeira”
  - Se o AFD entra neste estado, que não é final, fica “preso”

$$
\begin{aligned}
(A, p1q1) &\overset{*}{\vdash} (A, 1q1) && \text{para } p \in \{0\}^* \\
&\vdash (B, q1) && \\
&\overset{*}{\vdash} (B, 1) && \text{para } q \in \{0\}^* \\
&\vdash (L, \lambda) &&
\end{aligned}
$$



Exemplo 2

- Qual linguagem é reconhecida pelo AFD?
  - O estado L serve como “lixeira”
  - Se o AFD entra neste estado, que não é final, fica “preso”



$$
\begin{array}{llll}
(A, p1q1r) & \overset{*}{\vdash} & (A, 1q1r) & \text{para } p \in \{0\}^* \\
 & \vdash & (B, q1r) & \\
 & \overset{*}{\vdash} & (B, 1r) & \text{para } q \in \{0\}^* \\
 & \vdash & (L, r) & \\
 & \overset{*}{\vdash} & (L, \lambda) & \text{para } r \in \{0,1\}^*
\end{array}
$$

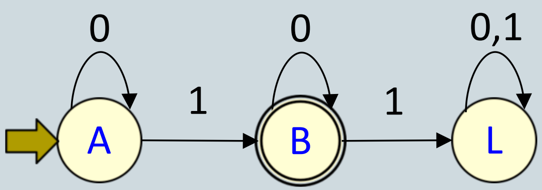
Formalização

- Um AFD qualquer pode ser representado por uma quíntupla $M = (E,\Sigma,\delta,i,F)$, onde
  - E é um conjunto finito de um ou mais elementos denominados estados;
  - Σ é o alfabeto;
  - $\delta: E \times \Sigma \rightarrow E$ é a função de transição, uma função total;
  - i, um estado de E, é o estado inicial;
  - F, um subconjunto de E, é o conjunto de estados finais.

Formalização - exemplo

- Este AFD é representado pela quíntupla $M = (E,\Sigma,\delta,i,F)$, onde
  - $E = \{A,B,L\}$
  - $\Sigma = \{0,1\}$
  - $i = A$
  - $F = \{B\}$
  - δ é definido por

$\delta(A,0) = A$	$\delta(B,0) = B$	$\delta(L,0) = L$
$\delta(A,1) = B$	$\delta(B,1) = L$	$\delta(L,1) = L$



Exercícios

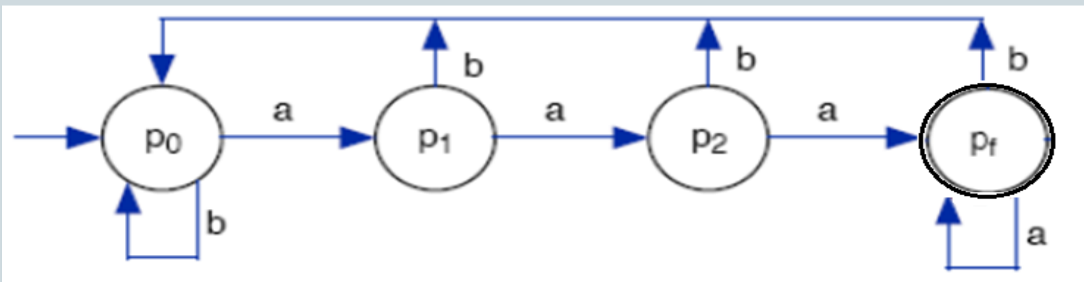
Escreva um AFD para cada linguagem abaixo

1. Palavras sobre {0,1} com um número par de 1s.
2. Sequências de dígitos binários com um número par de 1s e que terminam em 0.
3. Sequências de dígitos binários com um número par de 1s ou que terminam em 0.
4. Números binários que contêm três 0s em sequência.
5. Números binários que não contêm três 0s em sequência.

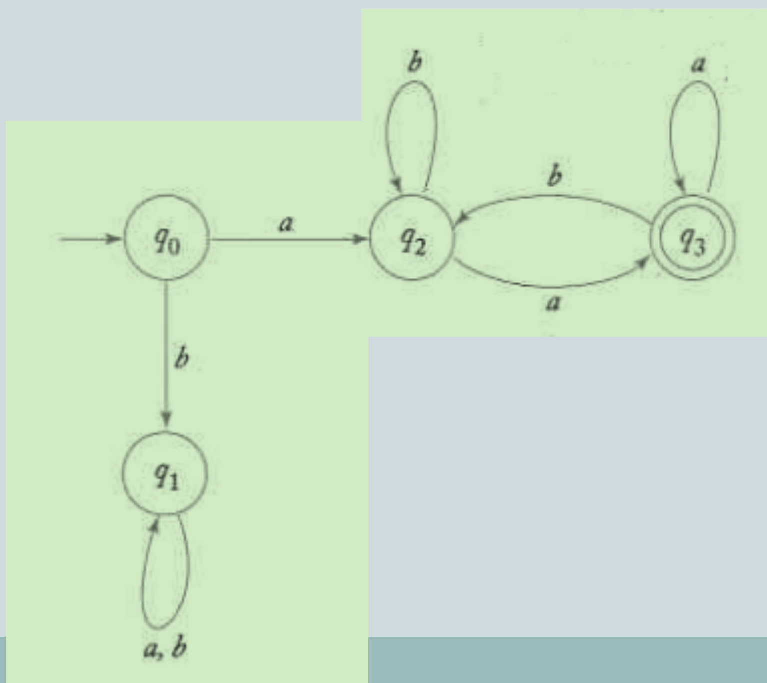
Exercícios

Identifique a linguagem reconhecida por cada AFD abaixo:

a)



b)



Autômatos Finitos Não Determinísticos

OBJETIVO

COMPREENDER E PROJETAR AFNS

Eduardo Freire Nakamura (nakamura@icomp.ufam.edu.br)

Introdução

- AFD
  - Transições são definidas para apenas um único estado
  - Uma transição para cada símbolo do alfabeto
- Um Autômato Finito Não Determinístico (AFN) não impõe essas restrições
  - Várias transições podem ser definidas para o mesmo estado de origem e para o mesmo símbolo do alfabeto
  - Não é obrigatório definir transições para cada símbolo
- Vantagens
  - Maior facilidade na construção dos autômatos
  - Autômatos mais “limpos”

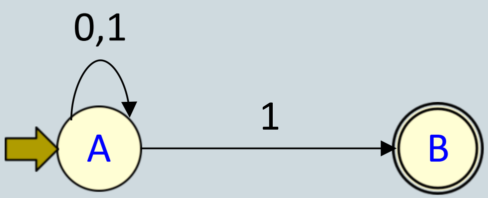
Introdução

- O que muda em relação aos AFDs?
  - Em um dado momento, a máquina pode estar em vários estados ao mesmo tempo
  - Não se pode mais falar em "estado atual", mas conjunto de estados atuais

- E qual é o critério de "aceitação de uma palavra" por um AFN?
  - Ao consumir toda a palavra de entrada, se pelo menos um dos estados atingidos for final, então a palavra de entrada é aceita

Exemplo 1

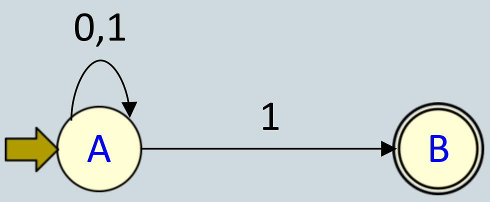
- Números binários terminados em 1
  - A máquina sempre permanece no estado A
  - Se o símbolo 1 for consumido, então a máquina transita ao mesmo tempo para os estados A e B
  - Não é necessário criar transições saindo do estado B



Exemplo 1

- Para a palavra 00101

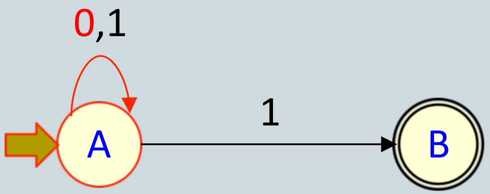
Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Exemplo 1

- Para a palavra 00101

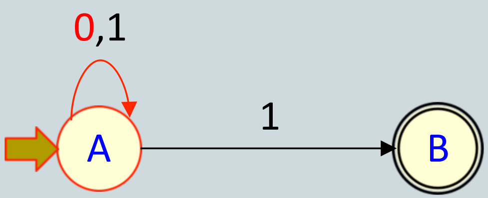
Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Exemplo 1

- Para a palavra 00101

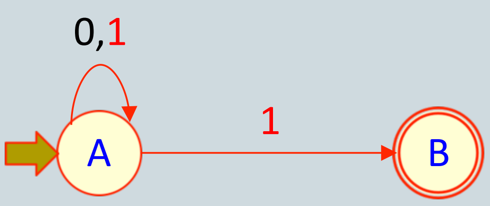
Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Exemplo 1

- Para a palavra 00101

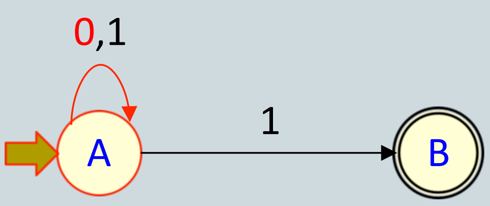
Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Exemplo 1

- Para a palavra 00101

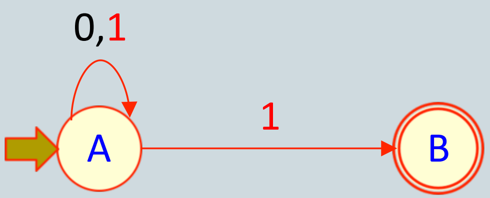
Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Exemplo 1

- Para a palavra 00101

Após consumir	chega-se aos estados	e resta consumir
λ	{A}	00101
0	{A}	0101
0	{A}	101
1	{A,B}	01
0	{A}	1
1	{A,B}	λ



Configuração instantânea

- “Fotografia” da execução da máquina em um dado instante

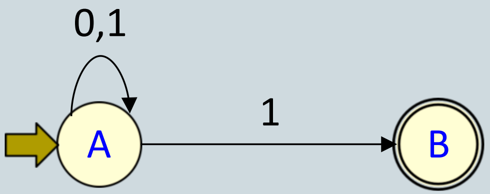
- No caso de um AFD, a configuração instantânea engloba
  - O estado atual
  - A sequência de símbolos que falta consumir

- No caso de um AFN, a configuração instantânea engloba
  - O conjunto de estados atuais
  - A sequência de símbolos que falta consumir

Exemplo 2

- A configuração instantânea é representada por uma tupla (*E*, *p*), onde
  - *E* - conjunto de estados atuais
  - *p* - a palavra restante

$$
\begin{aligned}
(\{A\}, 00101) &\vdash (\{A\}, 0101) \\
&\vdash (\{A\}, 101) \\
&\vdash (\{A,B\}, 01) \\
&\vdash (\{A\}, 1) \\
&\vdash (\{A,B\}, \lambda)
\end{aligned}
$$

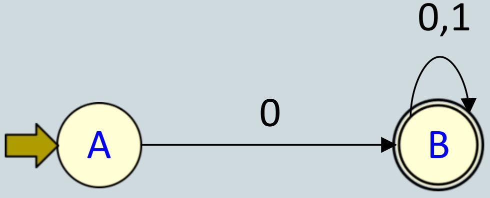


Exemplo 2

- E quando não há transição definida para um símbolo?

- Consumo da palavra 1010

$(\{A\}, 1010) \vdash (\varnothing, 010)$
$\vdash (\varnothing, 10)$
$\vdash (\varnothing, 0)$
$\vdash (\varnothing, \lambda)$

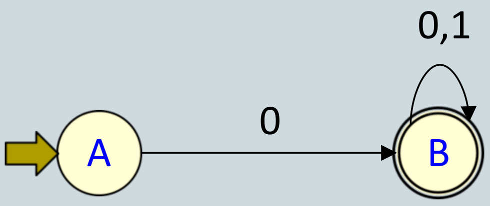


Exemplo 2

- E quando não há transição definida para um símbolo?

- Consumo da palavra 1010

$$
\begin{aligned}
(\{A\}, 1010) &\vdash (\varnothing, 010)\\
&\vdash (\varnothing, 10)\\
&\vdash (\varnothing, 0)\\
&\vdash (\varnothing, \lambda)
\end{aligned}
$$

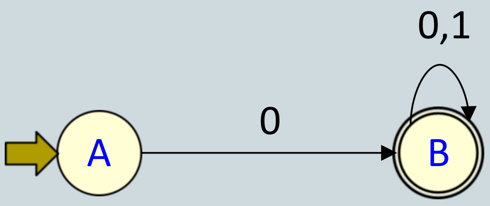


Exemplo 2

- E quando não há transição definida para um símbolo?

- Consumo da palavra 1010

$$(\{A\}, 1010) \vdash (\varnothing, 010)$$
$$\vdash (\varnothing, 10)$$
$$\vdash (\varnothing, 0)$$
$$\vdash (\varnothing, \lambda)$$

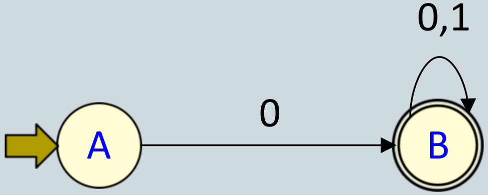


Exemplo 2

- E quando não há transição definida para um símbolo?

- Consumo da palavra 1010

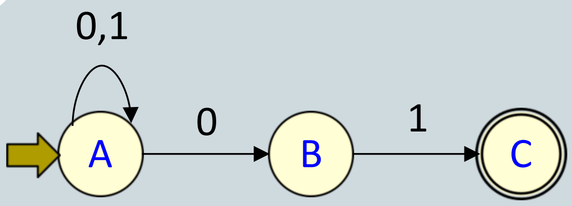
$$
\begin{aligned}
(\{A\}, 1010) &\vdash (\varnothing, 010) \\
&\vdash (\varnothing, 10) \\
&\vdash (\varnothing, 0) \\
&\vdash (\varnothing, \lambda)
\end{aligned}
$$



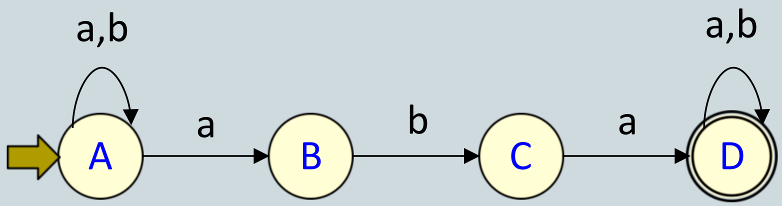
$(\varnothing, p) \overset{*}{\vdash} (\varnothing, \lambda)$ para qualquer palavra p

Exemplos

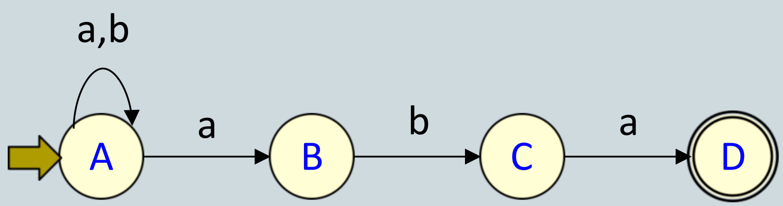
- Números binários que terminam em 01



- Palavras sobre {a,b} que contêm aba



- Palavras sobre {a,b} que terminam em aba



Formalização

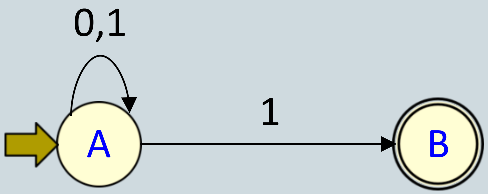
- Um AFN pode ser representado por uma quíntupla $M=(E,\Sigma,\delta,I,F)$
  - E é um conjunto finito de um ou mais estados
  - Σ é o alfabeto
  - $\delta : E \times \Sigma \rightarrow \mathcal{P}(E)$ é a função de transição, uma função total, onde $\mathcal{P}(E)$ é o conjunto de todos os subconjuntos de E
  - I, um subconjunto de E, é o conjunto de estados iniciais
  - F, um subconjunto de E, é o conjunto de estados finais

Formalização - exemplo

- Este AFN é representado pela quíntupla $M = (E,\Sigma,\delta,i,F)$, onde
  - $E = \{A,B\}$
  - $\Sigma = \{0,1\}$
  - $i = A$
  - $F = \{B\}$
  - δ é definido por

$\delta(A,0) = \{A\}$ $\delta(B,0) = \varnothing$

$\delta(A,1) = \{A,B\}$ $\delta(B,1) = \varnothing$



Autômato Finito Não Determinístico Estendido

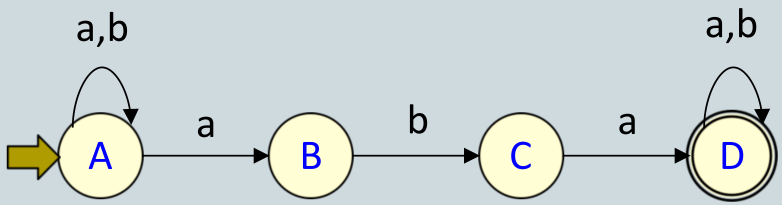
- Um AFN consome sempre exatamente um símbolo
  - δ(*estado*, *símbolo*) = {*estados-destino*}

- Um Autômato Finito Não Determinístico Estendido (AFNE), permite que cada transição consuma uma palavra não nula
  - δ(*estado*, *palavra*) = {*estados-destino*}
  - Sendo *palavra* pertencente a Σ^+

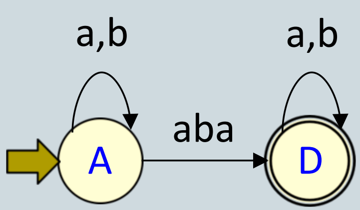
AFNE

- Palavras sobre o alfabeto {a,b} que contêm a sequência *aba*

- Usando um AFN



- Usando um AFNE



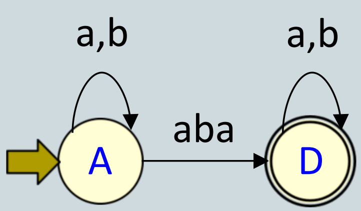
AFNE

- Palavras sobre o alfabeto {a,b} que contêm a sequência *aba*

- Usando um AFN

- Usando um AFNE

A transição do estado A para o estado D somente poderá ocorrer se todos os símbolos puderem ser consumidos imediatamente e na ordem especificada



Formalização

- Um AFNE é representado por uma quíntupla $M = (E,\Sigma,\delta,I,F)$
  - E é um conjunto finito de um ou mais estados
  - Σ é o alfabeto
  - $\delta : E \times \Sigma^+ \rightarrow \mathcal{P}(E)$ é a função de transição, uma função total, onde $\mathcal{P}(E)$ é o conjunto de todos os subconjuntos de E
  - I, um subconjunto de E, é o conjunto de estados estados iniciais
  - F, um subconjunto de E, é o conjunto de estados finais

Autômato Finito Não Determinístico com Tansições λ

- Um AFN consome sempre exatamente um símbolo
  - δ(*estado*, *símbolo*) = {*estados-destino*}

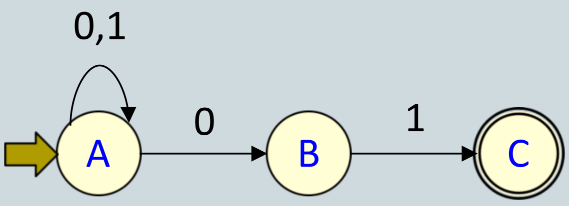
- Um Autômato Finito Não Determinístico com Transições λ (AFNλ), permite transições que não consomem símbolo
  - δ(*estado*, *palavra*) = {*estados-destino*}
  - Sendo *palavra* pertencente a $\Sigma \cup \{\lambda\}$

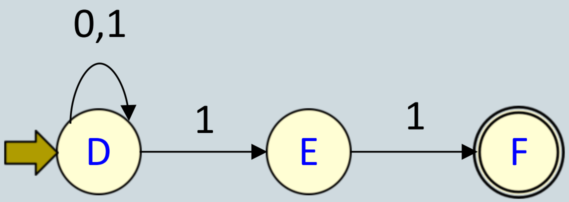
AFNλ

- As transições λ são muito úteis em várias situações
  - Uma das mais comuns é "fingir" que o autômato possui vários estados iniciais
  - União de linguagens

- Exemplo
  - Linguagem sobre {0,1} cujas palavras terminam em 01 ou 11

Palavras terminadas em 01



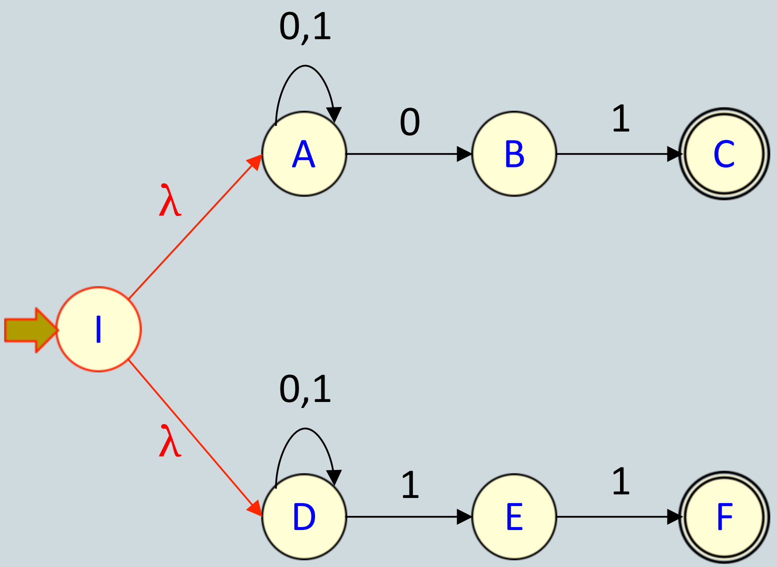


Palavras terminadas em 11

AFNλ

- As transições λ são muito úteis em várias situações
  - Uma das mais comuns é “fingir” que o autômato possui vários estados iniciais
  - União de linguagens

- Exemplo
  - Linguagem sobre {0,1} cujas palavras terminam em 01 ou 11



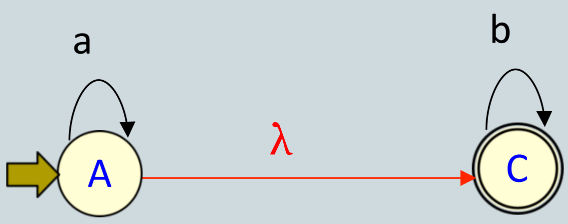
Palavras terminadas em 01 ou 11

AFNλ

- As transições λ são muito úteis em várias situações
  - Uma das mais comuns é “fingir” que o autômato possui vários estados iniciais
  - União de linguagens
  - Concatenação de linguagens

- Exemplo
  - Linguagem sobre {a,b} definida como:

  $L=\{a^n b^m \mid n>=0 \text{ e } m>=0\}$



$L=\{a^n b^m \mid n>=0 \text{ e } m>=0\}$

Exercícios

- Faça um AFN que reconheça

1. Palavras sobre o alfabeto {*a*,*b*} que possuam a sequência *baba*
2. Palavras sobre o alfabeto {*x*,*y*,*z*} que terminam em *x* ou *yz*
3. Palavras sobre o alfabeto {*x*,*y*} que começam com *xyx* ou que terminam com *yxy*
4. Palavras sobre o alfabeto {*m*,*n*} que contêm um número par de *m*'s ou que terminam em *nnn*
5. Palavras em {*a*, *b*}* de tamanho par