

# ITC: Introdução à Teoria da Computação

Marcos Castilho

DInf/UFPR

14 de maio de 2021

# Autômatos Finitos não-Determinísticos

Definição: Um Autômato Finito não-Determinístico (AFN) é uma quintupla  $M = (Q, \Sigma, \delta, q_0, F)$  onde:

- ▶  $Q$ : conjunto finito de estados;
- ▶  $\Sigma$ : conjunto finito de símbolos, é o alfabeto;
- ▶  $q_0 \in Q$ : é o estado inicial;
- ▶  $F \subseteq Q$  é o conjunto de estados finais;
- ▶  $\delta : Q \times \Sigma \rightarrow 2^Q$ : é a função de transição.

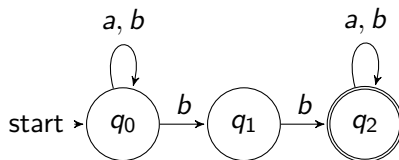
## Exemplo

Um AFN que aceita o conjunto de palavras sobre  $\{a, b\}$  que contêm a subpalavra  $bb$ .

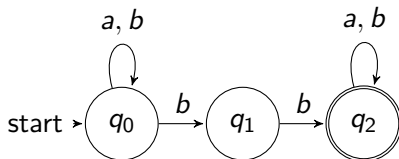
$$L(M) = (a + b)^* bb(a + b)^*.$$

$$M : Q = \{q_0, q_1, q_2\}; \Sigma = \{a, b\}; F = \{q_2\}.$$

| $\delta$ | a           | b              |
|----------|-------------|----------------|
| $q_0$    | $\{q_0\}$   | $\{q_0, q_1\}$ |
| $q_1$    | $\emptyset$ | $\{q_2\}$      |
| $q_2$    | $\{q_2\}$   | $\{q_2\}$      |



## Computações possíveis



$[q_0, ababb]$

$\vdash [q_0, babb]$

$\vdash [q_0, abb]$

$\vdash [q_0, bb]$

$\vdash [q_0, b]$

$\vdash [q_0, \lambda]$

$q_0 \notin F$

$[q_0, ababb]$

$\vdash [q_0, babb]$

$\vdash [q_1, abb]$

$q_1 \notin F$

$[q_0, ababb]$

$\vdash [q_0, babb]$

$\vdash [q_0, abb]$

$\vdash [q_0, bb]$

$\vdash [q_1, b]$

$\vdash [q_2, \lambda]$

$q_2 \in F$

# Definição

A linguagem de um AFN,  $L(M)$ , é o conjunto de palavras aceitas por  $M$ , isto é:

$$L(M) = \{w \mid \text{existe uma computação } [q_0, w] \vdash^* [q_i, \lambda] \text{ onde } q_i \in F\}$$

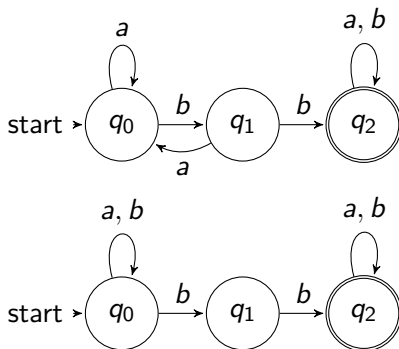
# Definição

O diagrama de estados de um AFN  $M = (Q, \Sigma, \delta, q_0, F)$  é o grafo dirigido rotulado  $G$  definido assim:

1. Os nodos de  $G$  são os elementos de  $Q$ ;
2. Os rótulos dos arcos de  $G$  são os elementos de  $\Sigma$ ;
3.  $q_0$  é o nodo inicial representado por um círculo;
4.  $F$  é o conjunto dos nodos representados por um círculo duplo;
5. Existe um arco de  $q_i$  a  $q_j$  rotulado  $a$  se  $q_j \in \delta(q_i, a)$ .

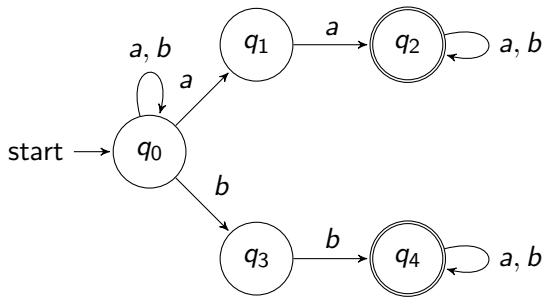
## Exemplo 1

AFN's que aceitam  $(a + b)^*bb(a + b)^*$



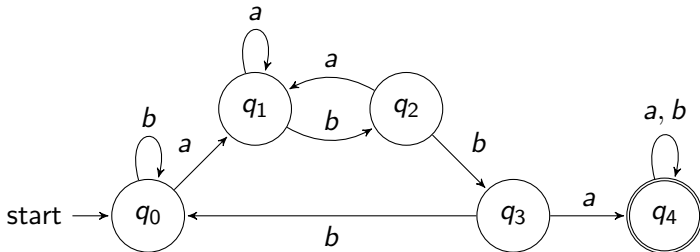
## Exemplo 2

AFN que aceita  $(a + b)^* bb(a + b)^* + (a + b)^* aa(a + b)^*$



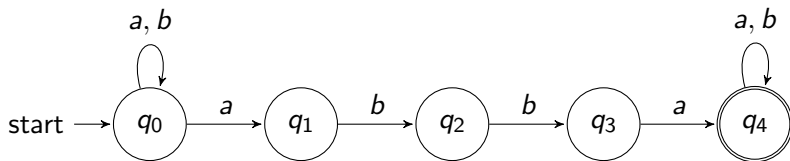
## Exemplo 3

Palavras que contém *abba* são aceitas pelo AFD seguinte:



## Exemplo 3, continuação

E também são aceitas pelo AFN abaixo:



## Exercício

Construir um AFN para reconhecer as palavras sobre  $\{a, b\}$  que contêm a subpalavra  $aa$  ou não contêm a subpalavra  $bb$ .

# Transições $\lambda$

Definição:

Um Autômato Finito não-Determinístico com transições  $\lambda$  é uma quintupla  $M = (Q, \Sigma, q_0, \delta, F)$  onde  $Q, \Sigma, q_0$  e  $F$  são como nos caso dos AFN's. A função de transição é:

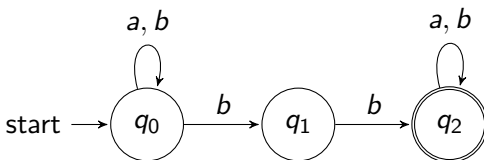
$$\delta : Q \times (\Sigma \cup \{\lambda\}) \longrightarrow 2^Q.$$

Uma entrada é aceita por um AFN- $\lambda$  se existe uma computação que processa toda a palavra de entrada e para em um estado final.

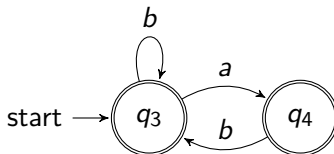
## Exemplo

Sejam  $M_1$  e  $M_2$  os automatos que reconhecem:

$(a + b)^*bb(a + b)^*$ :

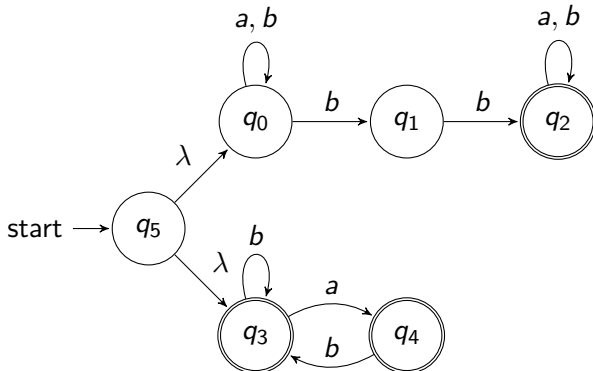


$(b + ab)^*(a + \lambda)$ :



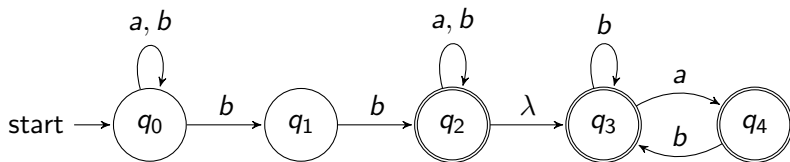
## Exemplo, continuação

Então o seguinte autômato aceita  $L(M_1) \cup L(M_2)$ :



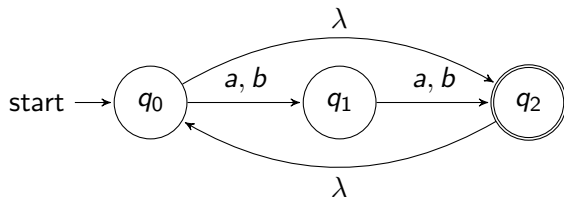
## Exemplo, continuação

Este AFN- $\lambda$  aceita  $L(M_1)L(M_2)$ :



## Exemplo

Este AFN- $\lambda$  aceita todas as palavras sobre  $\{a, b\}$  que têm tamanho par:



## Lema

Seja  $M = (Q, \Sigma, \delta, q_0, F)$  um AFN- $\lambda$ . Existe um AFN- $\lambda$  equivalente  $M' = (Q \cup \{q'_0, q_f\}, \Sigma, \delta', q'_0, \{q_f\})$  onde:

- ▶ O grau de entrada do nodo inicial é zero;
- ▶ O único estado final de  $M'$  é  $q_f$ ;
- ▶ O grau de saída de  $q_f$  é zero.

# Prova

Construímos  $\delta'$  como sendo exatamente como  $\delta$ , e adicionamos as seguintes transições sobre às do autômato  $M$ :

- ▶  $\delta'(q'_0, \lambda) = \{q_0\}$
- ▶  $\delta'(q'_i, \lambda) = \{q_f\}, \forall q_i \in F.$

Assim é fácil simular as transições de  $M$  em  $M'$ .

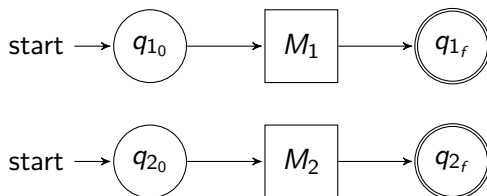
# Teorema

Sejam  $M_1$  e  $M_2$  dois AFN- $\lambda$ . Existe um AFN- $\lambda$  que aceita:

- ▶  $L(M_1) \cup L(M_2)$ ,
- ▶  $L(M_1)L(M_2)$ ,
- ▶  $L(M_1)^*$ .

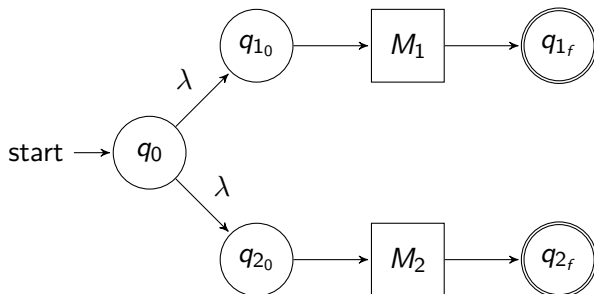
# Prova

Assumimos, sem perda de generalidade, que  $M_1$  e  $M_2$  satisfazem as condições do Lema anterior. Assim  $M_1$  e  $M_2$  são:



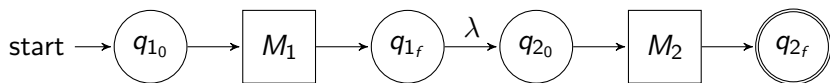
## Prova, continuação

A linguagem  $L(M_1) \cup L(M_2)$  é aceita pela seguinte máquina:



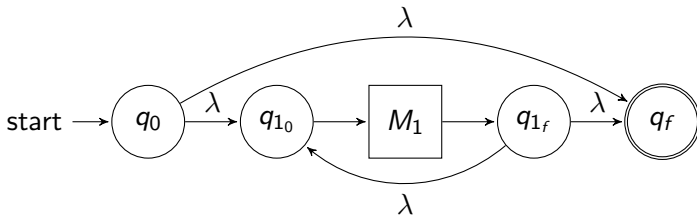
## Prova, continuação

A linguagem  $L(M_1)L(M_2)$  é aceita pela seguinte máquina:



## Prova, continuação

A linguagem  $L(M_1)^*$  é aceita pela seguinte máquina:



## Removendo o não Determinismo

Definição: O fecho lambda de um estado  $q_i$ , denotado  $fecho-\lambda(q_i)$ , é assim definido:

- ▶ BASE:  $q_i \in fecho-\lambda(q_i)$ ;
- ▶ PASSO RECURSIVO: Seja  $q_j \in fecho-\lambda(q_i)$ . Se  $q_k \in \delta(q_j, \lambda)$  então  $q_k \in fecho-\lambda(q_i)$ ;
- ▶ FECHO:  $q_j \in fecho-\lambda(q_i)$  apenas se pode ser obtido de  $q_i$  por um número finito de aplicações do passo recursivo.

## Definição

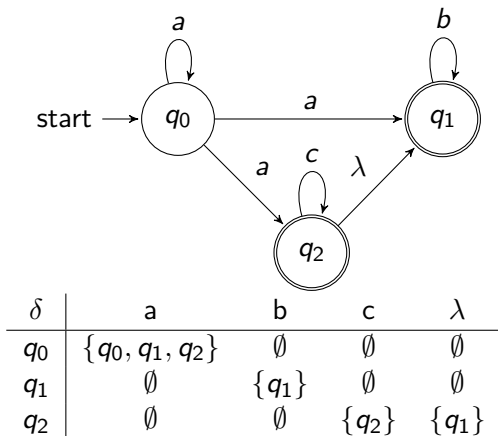
A função de transição de entrada  $t$  de um AFN- $\lambda$   $M$  é uma função de  $Q \times \Sigma$  em  $2^Q$  definida por:

$$t(q_i, a) = \bigcup_{q_j \in \text{fecho-}\lambda(q_i)} \text{fecho-}\lambda(\delta(q_j, a))$$

onde  $\delta$  é a função de transição de  $M$ .

## Exemplo

Seja  $M$  o AFN- $\lambda$  seguinte, que aceita a linguagem  $a^+c^*b^*$ :



## Exemplo, continuação

Apresentamos aqui a função de transição  $t$  para a função  $\delta$  acima:

| $\delta$ | a                   | b           | c           | $\lambda$   |
|----------|---------------------|-------------|-------------|-------------|
| $q_0$    | $\{q_0, q_1, q_2\}$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |
| $q_1$    | $\emptyset$         | $\{q_1\}$   | $\emptyset$ | $\emptyset$ |
| $q_2$    | $\emptyset$         | $\emptyset$ | $\{q_2\}$   | $\{q_1\}$   |

| $t$   | a                   | b           | c              |
|-------|---------------------|-------------|----------------|
| $q_0$ | $\{q_0, q_1, q_2\}$ | $\emptyset$ | $\emptyset$    |
| $q_1$ | $\emptyset$         | $\{q_1\}$   | $\emptyset$    |
| $q_2$ | $\emptyset$         | $\{q_1\}$   | $\{q_1, q_2\}$ |

## Algoritmo para gerar AFD a partir de AFN- $\lambda$

Entrada: Um AFN- $\lambda$   $M = (Q, \Sigma, \delta, q_0, F)$  e a função de transição de entrada de  $M$ ,  $t$ . A saída é  $DM$ , o AFD equivalente à  $M$ .

Inicialize  $Q' := \{\text{fecho-}\lambda(q_0)\}$

Repita

SE existe um nodo  $X \in Q'$  e um símbolo  $a \in \Sigma$   
tal que não há um arco saindo de  $X$  rotulado  $a$ , ENTÃO

Seja  $Y = \bigcup_{q_i \in X} t(q_i, a)$

SE  $Y \notin Q'$  ENTÃO  $Q' := Q' \cup \{Y\}$

Adicione um arco de  $X$  para  $Y$  rotulado  $a$

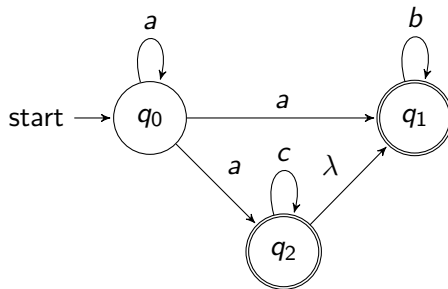
SENÃO feito:= true

ATÉ feito;

$F' := \{x \in Q' \mid x \text{ contém um elemento } q_i \in F\}$

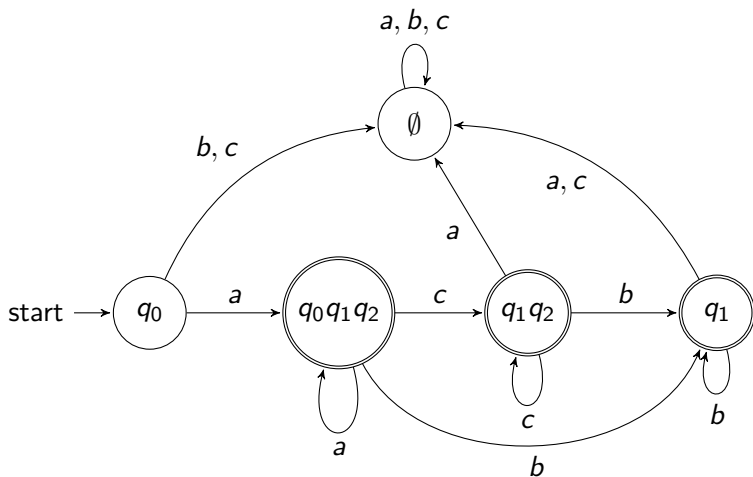
## Exemplo

Considere o AFN- $\lambda$   $M$  logo abaixo, e o fato de que o  $\text{fecho-}\lambda(q_0) = \{q_0\}$ .



## Exemplo

O AFD equivalente é este:



# Teorema

Sejam  $w \in \Sigma^*$  e  $Q_w = \{q_{w1}, q_{w2}, \dots, q_{wj}\}$  o conjunto de estados atingidos após o processamento da palavra  $w$  em  $M$ .  
Então processar  $w$  em  $DM$  resulta no estado  $Q_w$ .

# Prova

Por indução no tamanho da palavra  $w$ .

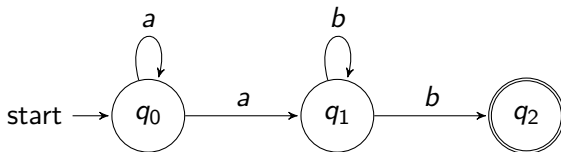
- ▶ BASE: Uma computação de  $\lambda$  em  $M$  termina em um nodo que pertence a  $\text{fecho-}\lambda(q_0)$ , que é o estado inicial de  $DM$ .
- ▶ HI: Assuma que o teorema é válido para palavras de tamanho  $n \geq 0$  fixo.
- ▶ PASSO: Seja  $w = ua$  tal que  $|w| = n + 1$ . Seja  $Q_u = \{q_{u1}, q_{u2}, \dots, q_{uk}\}$  o conjunto de terminais obtidos após processar  $u$  em  $M$ . Por HI, processar  $u$  em  $DM$  resulta em  $Q_u$ . Para processar  $ua$  em  $M$  basta processar um  $a$  em  $Q_u$ . Seja  $Q_w$  obtido por  $Q_w = \bigcup_{i=1}^k t(q_{ui}, a)$ .  $Q_w$  é o estado que resulta de processar  $a$  em  $DM$ .

# Corolário

Os autômatos finitos  $D$  e  $DM$  são equivalentes.

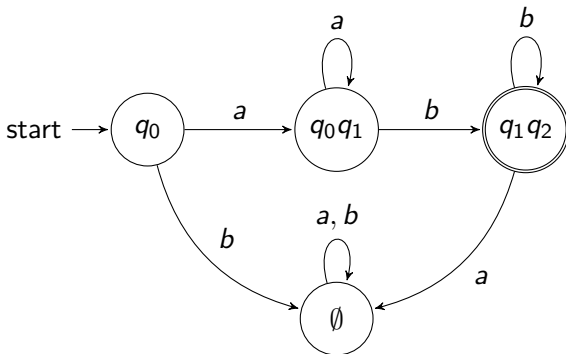
## Exemplo

Este é um AFN que reconhece a linguagem  $a^+b^+$ .



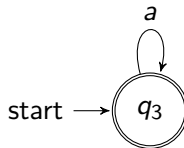
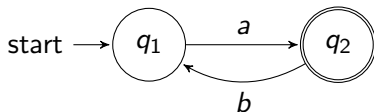
## Exemplo, continuação

Este é o AFD equivalente.



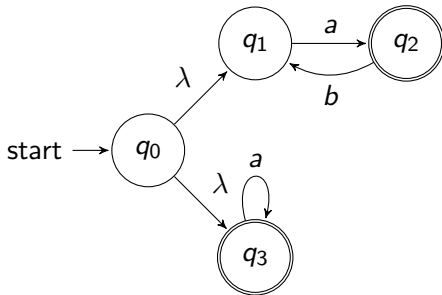
## Exemplo 2

Os AFN's  $M_1$  e  $M_2$  aceitam respectivamente as linguagens  $a(ba)^*$  e  $a^*$ :

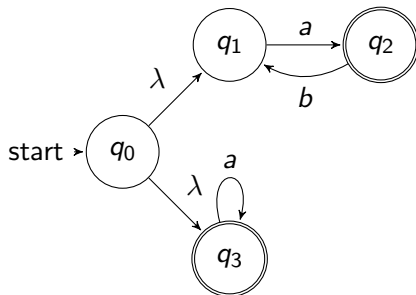


## Exemplo 2, continuação

O AFN abaixo aceita  $L(M_1) \cup L(M_2)$ :



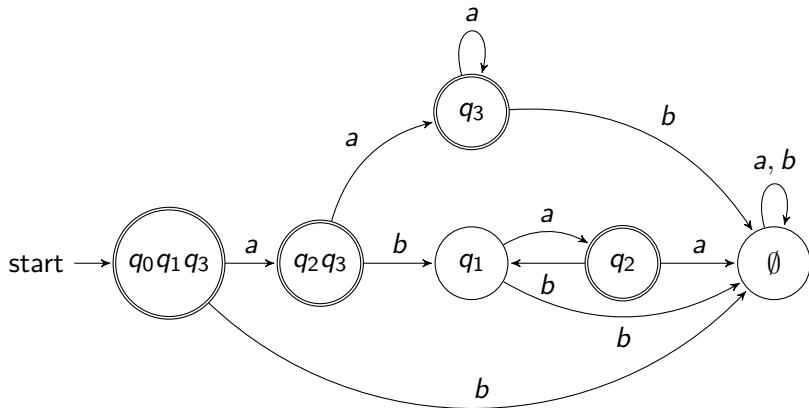
## Exemplo 2, continuação



| t     | a              | b           |
|-------|----------------|-------------|
| $q_0$ | $\{q_2, q_3\}$ | $\emptyset$ |
| $q_1$ | $\{q_2\}$      | $\emptyset$ |
| $q_2$ | $\emptyset$    | $\{q_1\}$   |
| $q_3$ | $\{q_3\}$      | $\emptyset$ |

## Exemplo 2, continuação

O AFD equivalente é:



# Resumo

- ▶ AFD's são um tipo especial de AFN's
- ▶ AFN's são um tipo especial de AFN- $\lambda$ 's
- ▶ AFN- $\lambda$ 's podem ser reduzidos para AFD's

# Licença

- ▶ Slides feitos em  $\text{\LaTeX}$  usando beamer e tikz, editados com vim.
- ▶ Licença

*Creative Commons* Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>