Autômatos Finitos

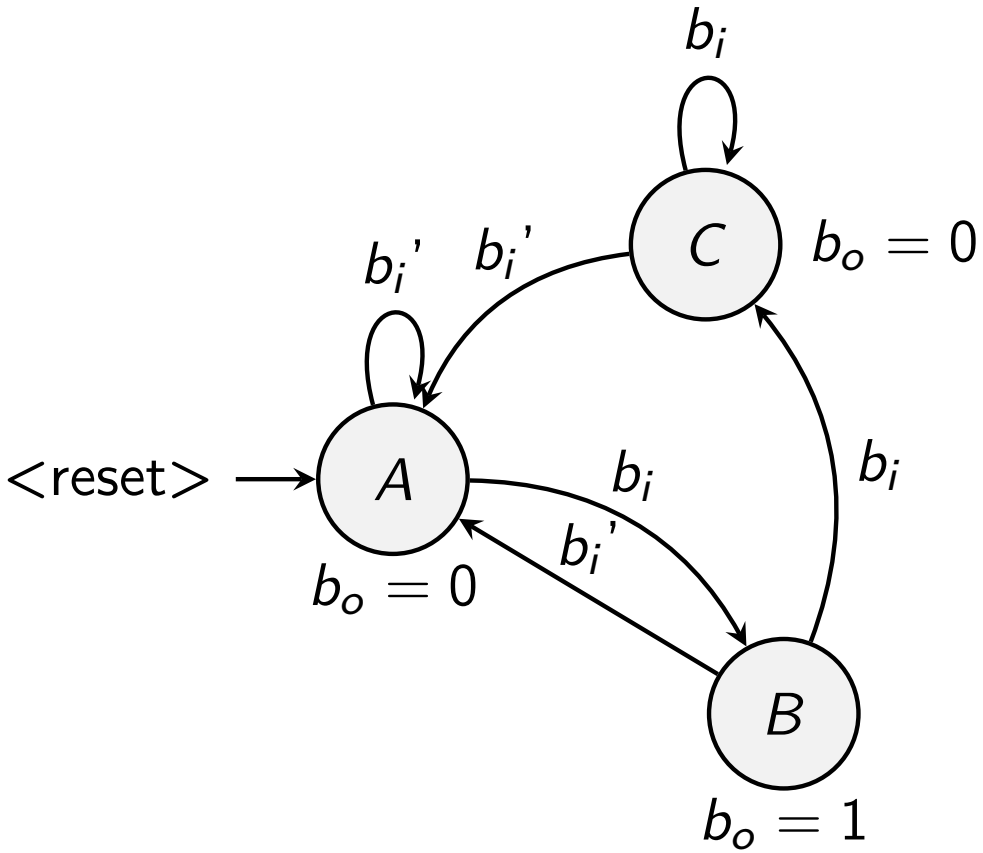
Profª. Drª. Jerusa Marchi
Otto Menegasso Pires

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

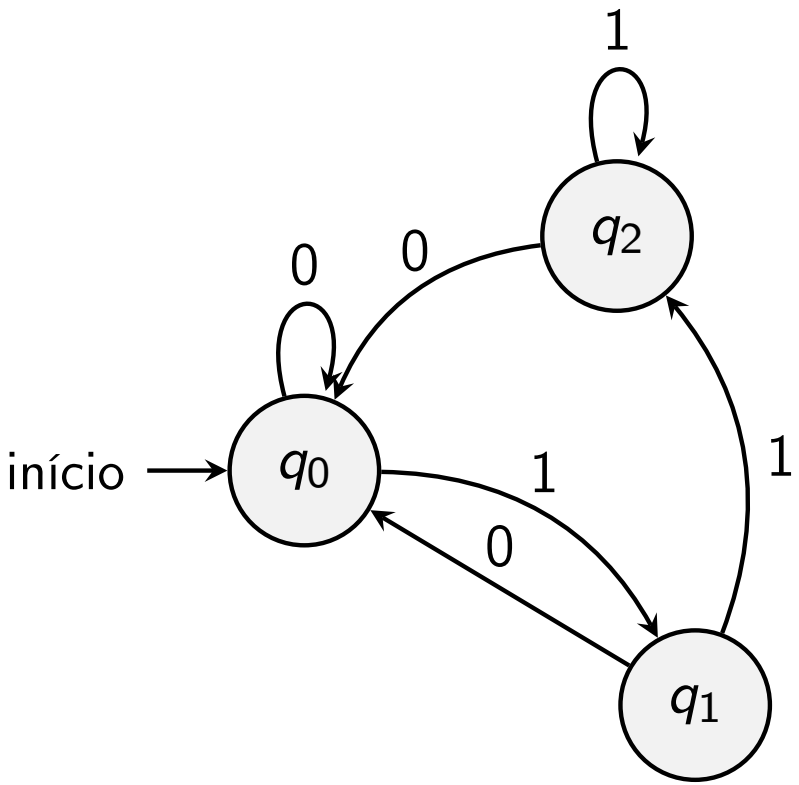
Máquina de Estados Finitos

- Uma Máquina de Estados Finitos (FSM) é uma maneira de descrever o comportamento de um circuito sequencial.
- Possui estados e transições entre estados



Autômatos Finitos

- Um Autômato Finito é um modelo computacional com memória limitada
- Possui estados e transições entre estados

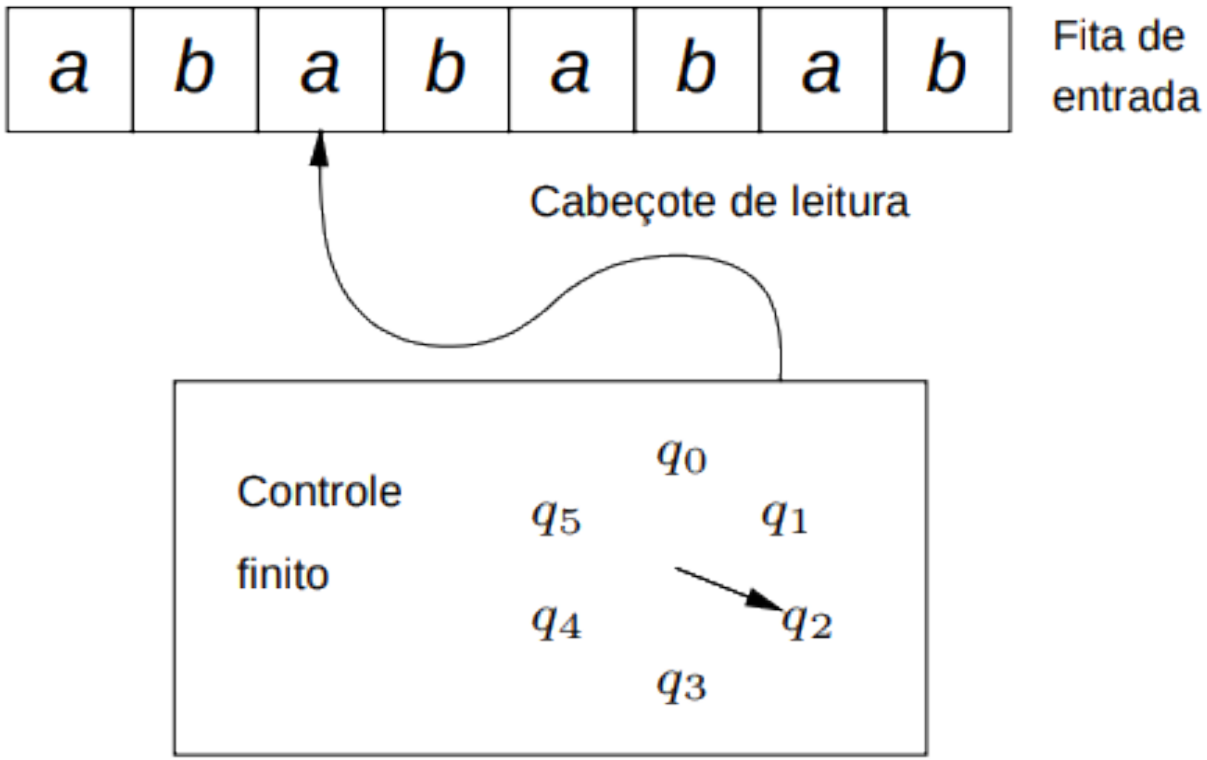


Aplicações

- Implementação de sistemas de controle simples baseados em estados (máquinas de venda automática, elevadores, portas automáticas, etc).
- Análise léxica (compiladores)
- Busca em texto

Descrição

- Fita de entrada
- Cabeçote de leitura
- Unidade central de processamento
- Memória limitada (estados)



Tipos de Autômatos Finitos

- **Transdutores de linguagens:** possuem duas fitas, entrada e saída, e podem escrever qualquer tipo de saída
- **Reconhecedores de linguagens:** possuem uma fita, entrada, e apenas duas saídas possíveis
  - Aceitação
  - Rejeição

Tipos de Autômatos Finitos

- **Determinísticos:** Ao processarem a entrada, a computação possui apenas um caminho possível.
- **Não-determinísticos:** Podem possuir mais de uma possibilidade de caminho para cada entrada. Sua computação se abre em ramos (*branches*).

Autômato Finito Determinístico - Definição Formal

Um Autômato Finito Determinístico M é definido pela quíntupla:

$$M = (K, \Sigma, \delta, q_0, F)$$

Onde:

- K = conjunto finito de estados
- Σ = conjunto finito de símbolos de entrada
- $\delta : K \times \Sigma \rightarrow K$ = função de transição
- q_0 = estado inicial ($q_0 \in K$)
- F = conjunto de estados finais ($F \subseteq K$)

Autômato Finito Determinístico - Definição Formal

O autômato é dito determinístico pois pela definição da função de transição δ, cada par (*estado*, *símbolo*) mapeia para exatamente um estado. Ou seja:

$$\delta(q, a) = p$$

sendo $q, p \in K$ e $a \in \Sigma$

Representação de um Autômato Finito

- Definição Formal
- Tabela de Transição
- Diagrama de Transição

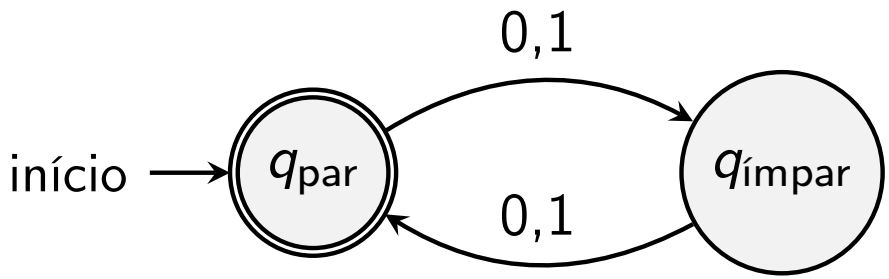
Tabela de Transição

Exemplo: $L_1 = \{w \mid w \in \{0,1\}^* \text{ e } |w| \text{ é par}\}$

	δ	0	1
$\rightarrow *$	q_{par}	$q_{\text{ímpar}}$	$q_{\text{ímpar}}$
	$q_{\text{ímpar}}$	q_{par}	q_{par}

Diagrama de transição

Exemplo: $L_1 = \{w \mid w \in \{0,1\}^* \text{ e } |w| \text{ é par}\}$



Exemplos AFD

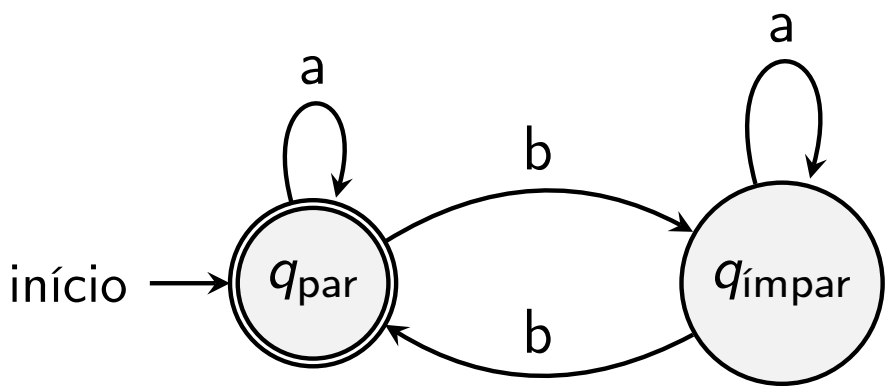
- $L_1 = \{w \mid w \in \{a, b\}^*$ e o número de b's é par$\}$
- $L_2 = \{w \mid w \in \{0, 1\}^*$ e w possui a subsequência 001$\}$
- $L_3 = \{w \mid w \in \{0, 1\}^*$ e w é um número binário múltiplo de 3$\}$

Exemplos AFD

$L_1 = \{w \mid w \in \{a, b\}^*$ e o número de b's é par$\}$

Exemplos AFD

$L_1 = \{w \mid w \in \{a, b\}^*$ e o número de b's é par$\}$

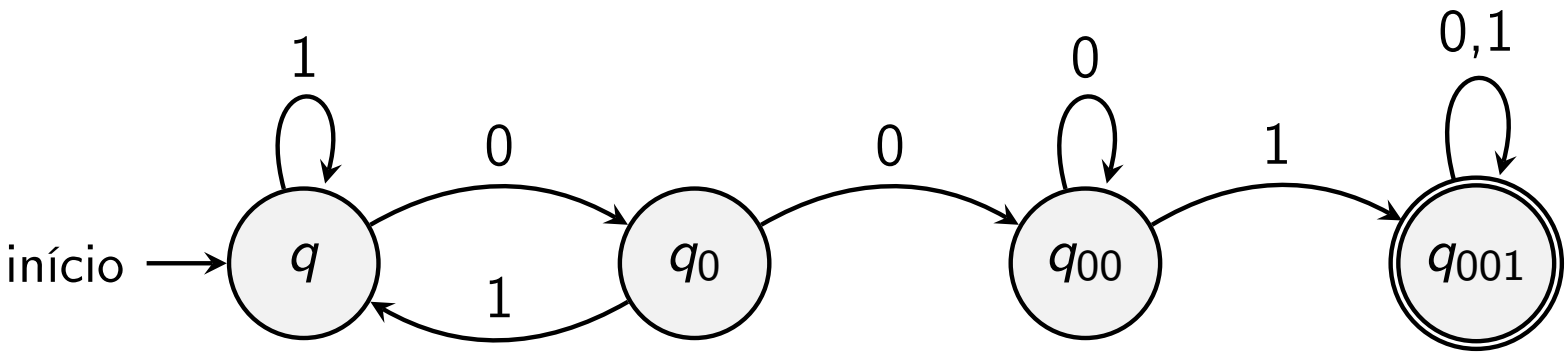


Exemplos AFD

$L_2 = \{w \mid w \in \{0,1\}^*$ e w possui a subsequência $001\}$

Exemplos AFD

$L_2 = \{w \mid w \in \{0,1\}^*$ e w possui a subsequência 001$\}$

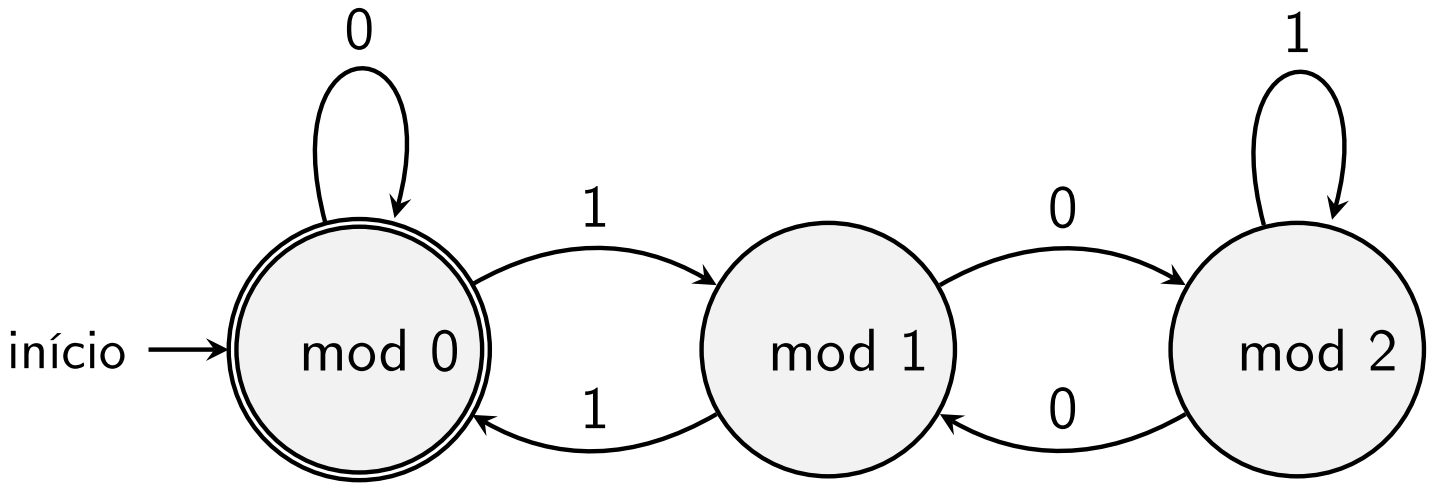


Exemplos AFD

$L_3 = \{w \mid w \in \{0,1\}^*$ e w é um número binário múltiplo de $3\}$

Exemplos AFD

$L_3 = \{w \mid w \in \{0,1\}^*$ e w é um número binário múltiplo de 3$\}$



Definição Formal de Computação

- **Configuração:** uma configuração é determinada pelo estado corrente e pela parte ainda não processada da palavra. Por exemplo,

$$[q_0, abab]$$

representa a configuração inicial de um autômato finito com a entrada $w = abab$

Definição Formal de Computação

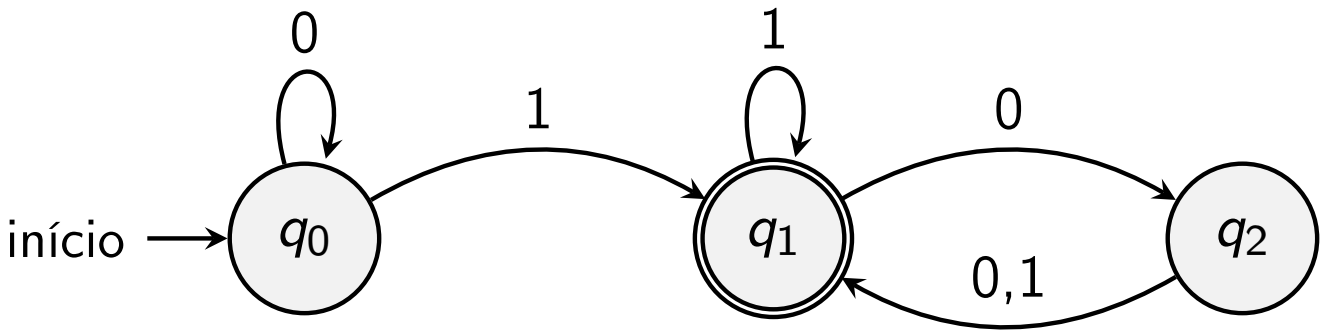
- **Computação:** é uma sequência de configurações. Usa-se a relação $\vdash$ (resulta em) para indicar que a máquina passa de uma configuração à outra. Por exemplo, a relação

$$[q_1, aw] \vdash [q_2, w]$$

existe se e somente se existe uma transição de q_1 para q_2 sob a, onde $a \in \Sigma$ e $w \in \Sigma^*$

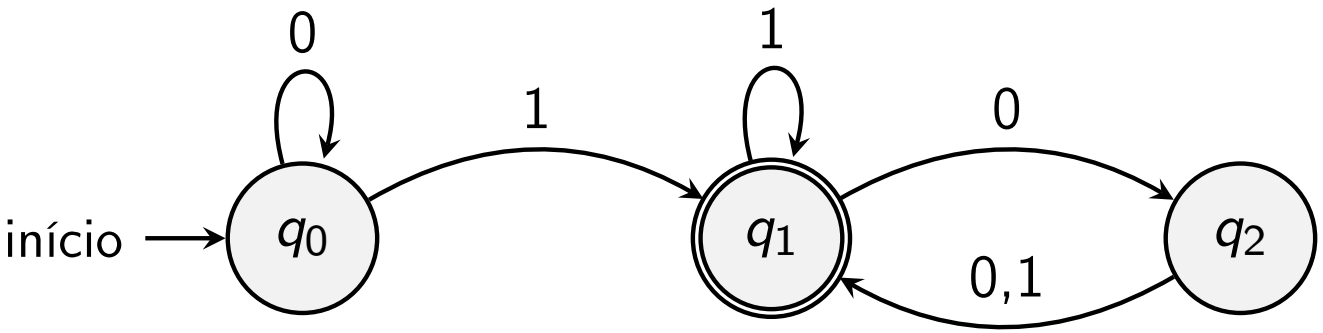
Definição Formal de Computação

Exemplo:



Definição Formal de Computação

Exemplo:



Computação:

$$[q_0, 0101] \vdash [q_0, 101] \vdash [q_1, 01] \vdash [q_2, 1] \vdash [q_1, \epsilon]$$

Definição Formal de Computação

- Uma sentença w é aceita por um autômato finito $M = (K, \Sigma, \delta, q_0, F)$ sse $\hat{\delta}(q_0, w) \rightarrow q$ e $q \in F$, ou seja, há uma computação

$$[q_0, w] \vdash_M^* [q, \epsilon]$$

  onde ϵ representa a palavra vazia

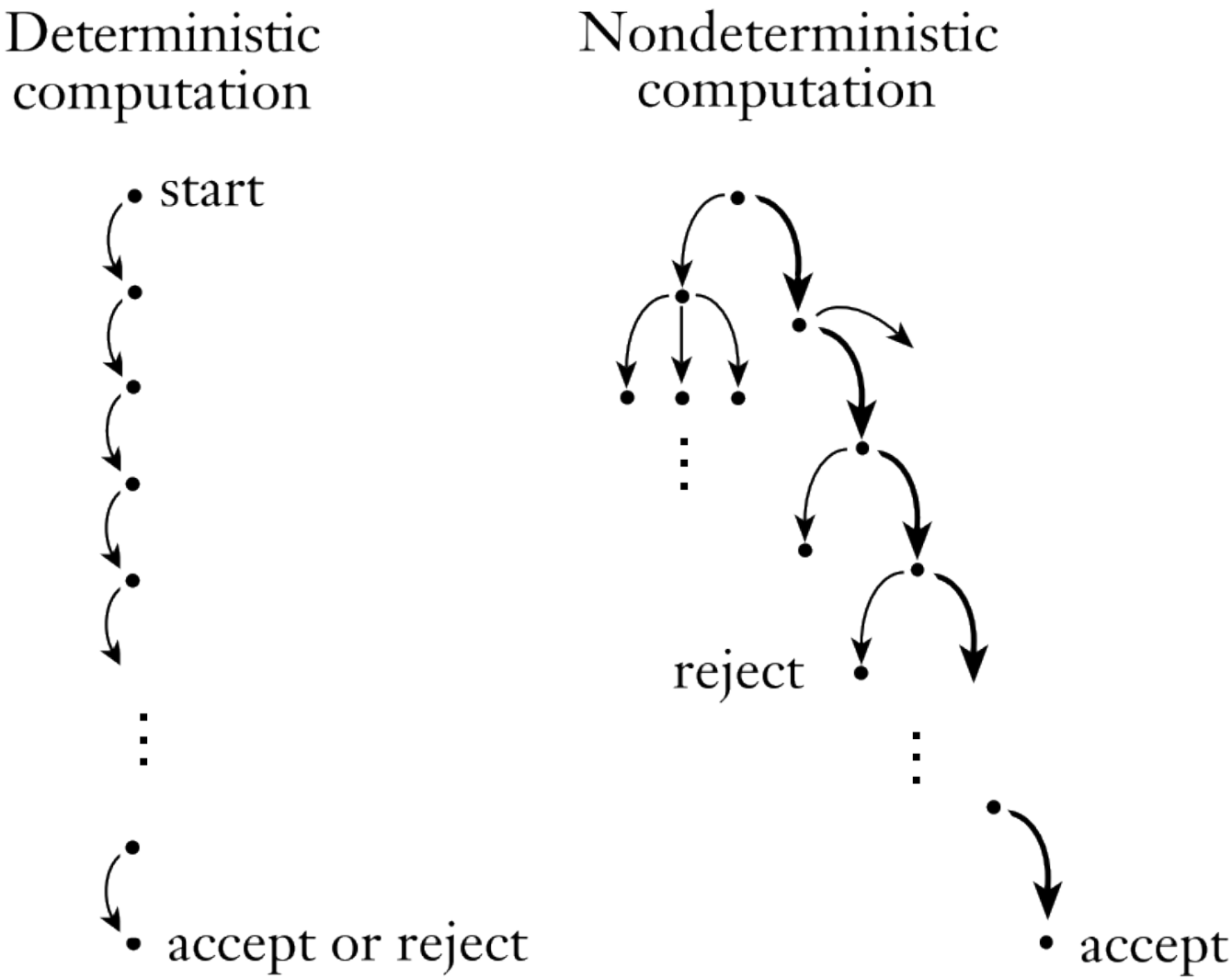
- A linguagem reconhecida por um autômato M é aquela cujo conjunto de sentenças é aceito por M

$$L(M) = \{w \mid w \in \Sigma^*,\ \hat{\delta}(q_0, w) \rightarrow q \text{ e } q \in F\}$$

Definição Formal de Computação

- Dois autômatos finitos M_1 e M_2 são ditos equivalentes sse $L(M_1) = L(M_2)$
- Uma linguagem é **regular** sse ela for reconhecida por um autômato finito

Não-determinismo



Não Determinismo

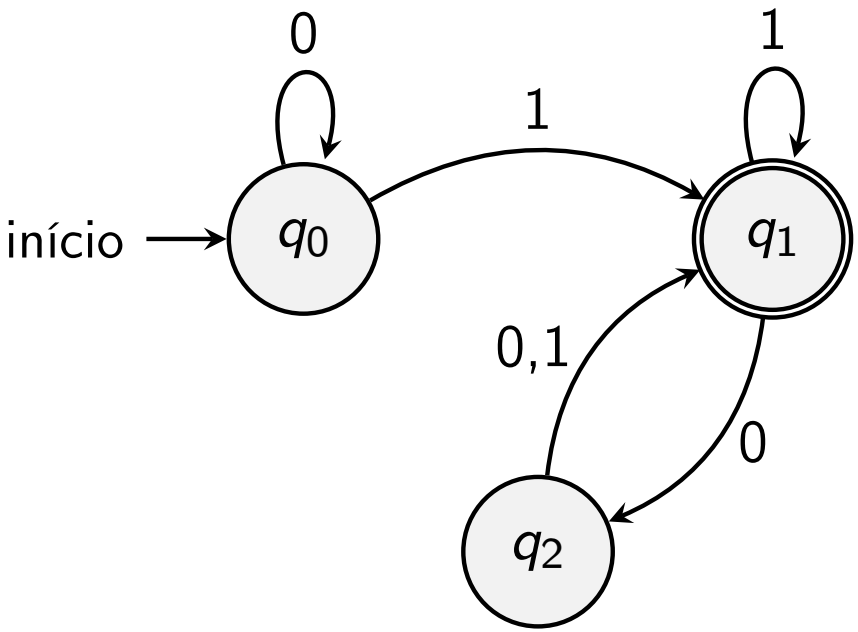
- Mais transições para um mesmo símbolo
- **ϵ-transição:** Uma ϵ-transição é uma transição que não consome nenhum elemento da entrada.

Não-determinismo

Para que uma computação não-determinística *aceite* uma palavra, pelo menos um ramo da computação tem que terminar em um estado de aceitação

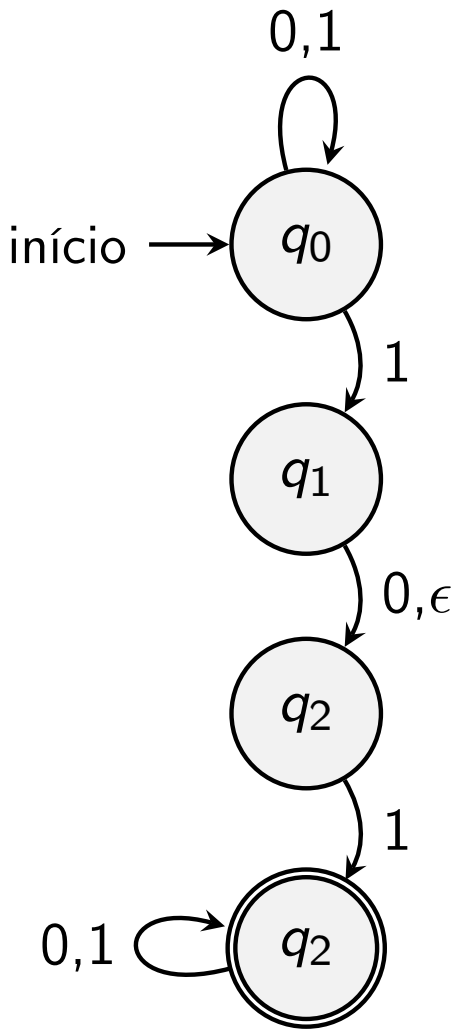
Para que uma computação não-determinística *rejeite* uma palavra, todos os ramos da computação têm que terminar em um estado de rejeição.

Computação Determinística



Entrada:

Computação Não-determinística



Entrada:

Computação - Análise

Computação Determinística

A profundidade da árvore de computação cresce linearmente com relação ao tamanho da entrada.

Computação Não-determinística

A profundidade da árvore de computação cresce linearmente com relação ao tamanho da entrada, porém a largura pode crescer de maneira exponencial em relação ao tamanho da entrada.

Autômato Finito Não-determinístico - Definição Formal

Um Autômato Finito Não-determinístico (AFND) M é definido pela quíntupla:

$$M = (K, \Sigma, \delta, q_0, F)$$

Onde:

- K = conjunto finito de estados
- Σ = conjunto finito de símbolos de entrada
- $\delta : K \times (\Sigma \cup \epsilon) \rightarrow 2^K$ = função de transição
- q_0 = estado inicial ($q_0 \in K$)
- F = conjunto de estados finais ($F \subseteq K$)

Autômatos Finitos Não-determinísticos

O autômato é dito não-determinístico se há pelo menos uma transição δ, para um par (estado, símbolo) que mapeia para um subconjunto de estados. Ou seja:

$$\delta(q, a) = \{p, r\}$$

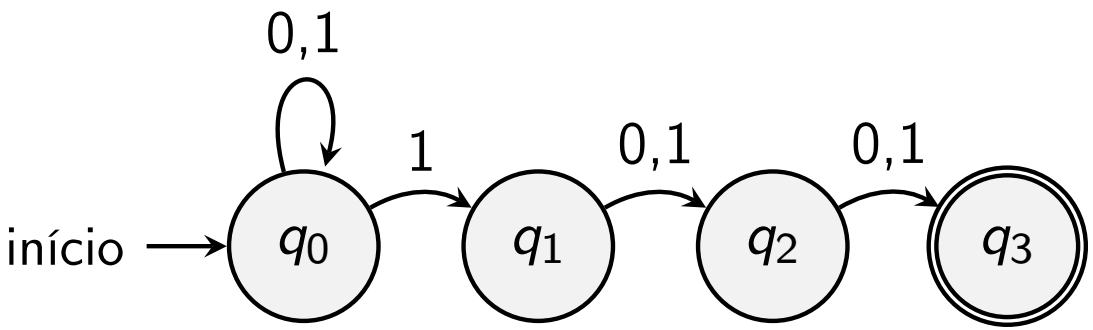
sendo $q, p, r \in K$ e $a \in \Sigma$

Exemplos - AFND

$L_1 = \{w \mid w \in \{0,1\}^*$ e w possui 1 na antepenúltima posição$\}$

Exemplos - AFND

$L_1 = \{w \mid w \in \{0,1\}^* \text{ e w possui 1 na antepenúltima posição}\}$

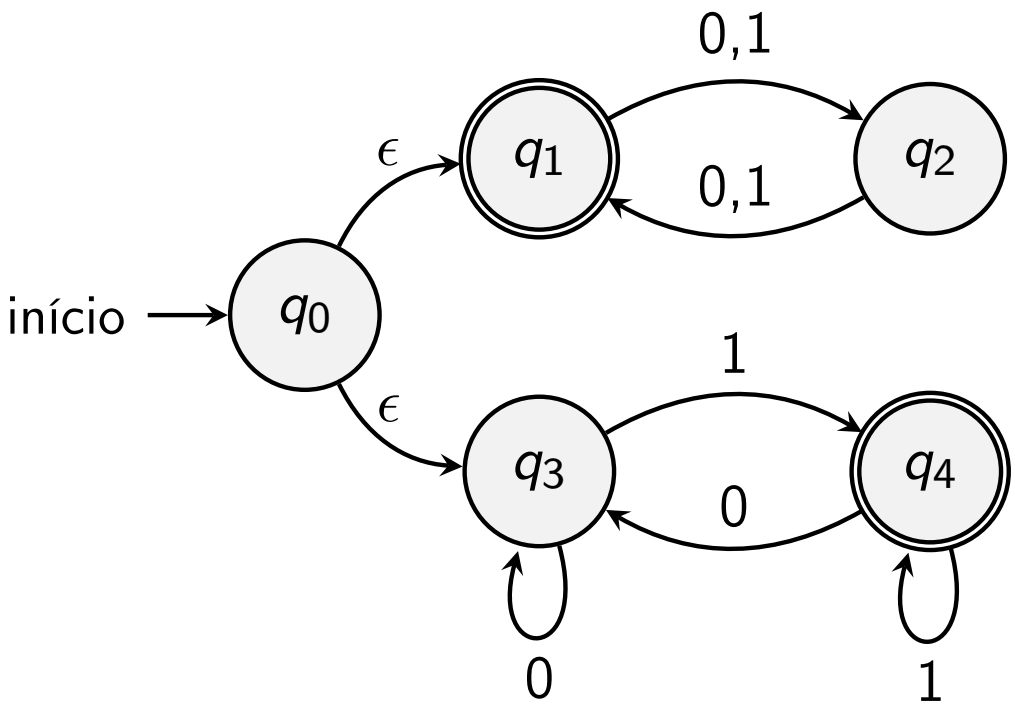


Exemplos - AFND-ϵ

- $L_2 = \{w \mid w \in \{0,1\}^*$ e $|w|$ é par ou w termina em $1\}$

Exemplos - AFND-ϵ

- $L_2 = \{w \mid w \in \{0,1\}^*$ e $|w|$ é par ou w termina em $1\}$



Equivalência AFD e AFND

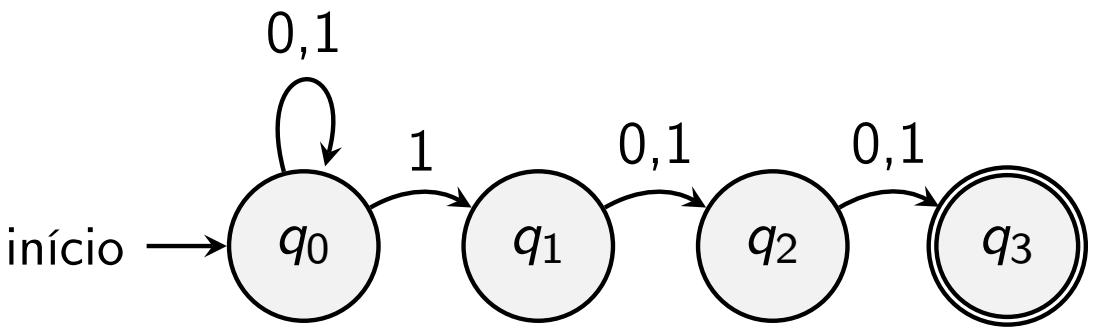
Teorema

Todo Autômato Finito Não-determinístico possui um Autômato Finito Determinístico equivalente.

Lembrando que: dois autômatos finitos M_1 e M_2 são ditos equivalentes sse $L(M_1) = L(M_2)$

Equivalência AFD e AFND - Prova

Faça um AFD que aceite a linguagem
$L = \{w \mid w \in \{0,1\}^* \text{ e w possui 1 na antepenúltima posição}\}$



Equivalência AFD e AFND - Prova

Faça um AFD que aceite a linguagem
$L = \{w \mid w \in \{0,1\}^*$ e w possui 1 na antepenúltima posição$\}$

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
q_2	q_3	q_3
$*q_3$	$-$	$-$

Equivalência AFD e AFND - Prova

Faça um AFD que aceite a linguagem
$L = \{w \mid w \in \{0,1\}^*$ e w possui 1 na antepenúltima posição$\}$

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
q_2	q_3	q_3
$*q_3$	–	–
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$*\{q_0, q_3\}$	q_0	$\{q_0, q_1\}$
$*\{q_0, q_1, q_3\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$*\{q_0, q_2, q_3\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$*\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

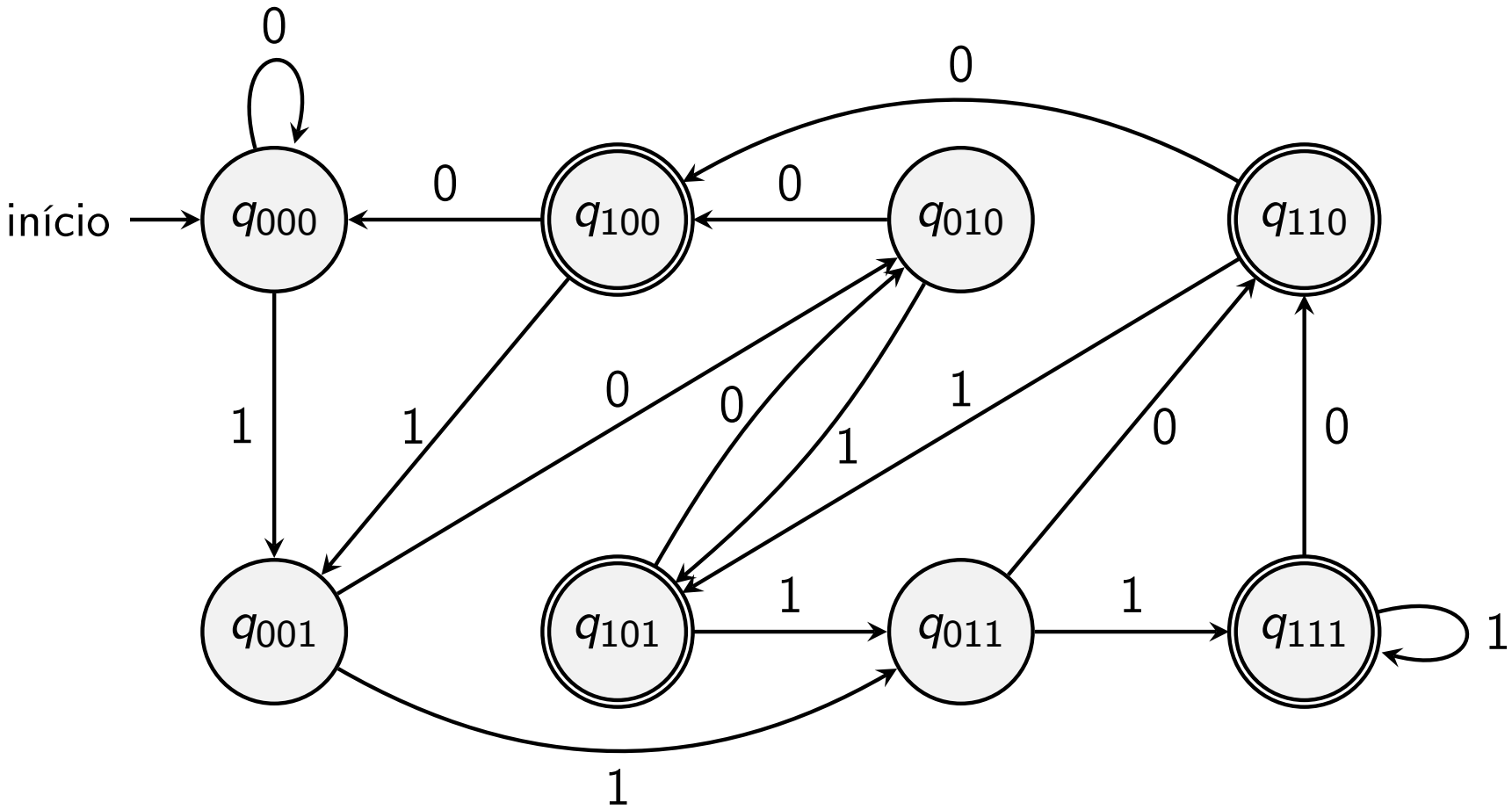
Equivalência AFD e AFND - Prova

Faça um AFD que aceite a linguagem
$L = \{w \mid w \in \{0,1\}^*$ e w possui 1 na antepenúltima posição$\}$

δ	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$*\{q_0, q_3\}$	q_0	$\{q_0, q_1\}$
$*\{q_0, q_1, q_3\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$*\{q_0, q_2, q_3\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$
$*\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

Equivalência AFD e AFND - Prova

$L = \{w \mid w \in \{0,1\}^*$ e w possui 1 na antepenúltima posição$\}$



Equivalência AFD e AFND - Prova

Importante

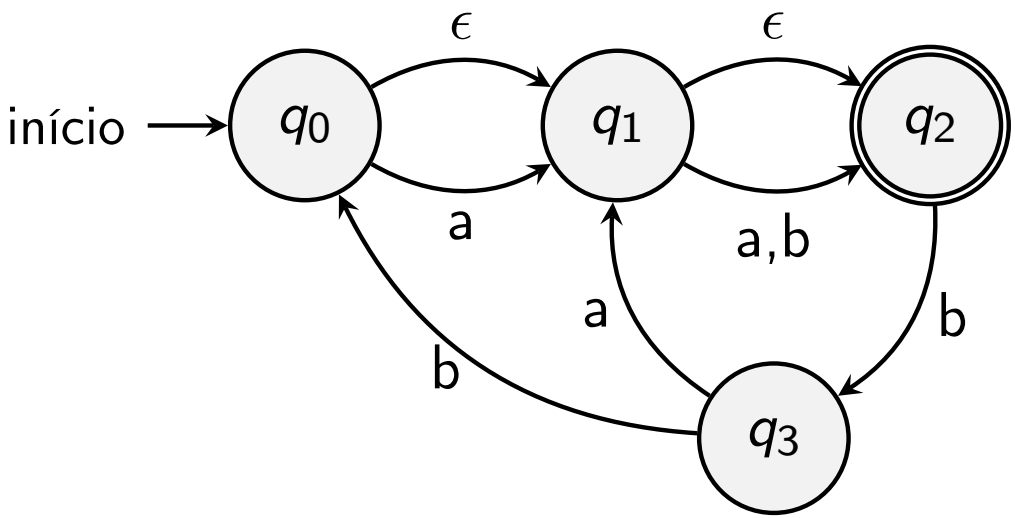
Um AFND com n estados pode gerar um AFD de até 2^n estados após ser determinizado.

Equivalência AFD e AFND-ϵ

Igual a equivalência anterior, é preciso construir um AFD capaz de reconhecer a mesma linguagem.

É preciso considerar o ϵ-fecho dos estados, isto é, todos os estados alcançáveis por esse estado a partir de ϵ transições.

Equivalência AFD e AFND-ϵ - Prova



Equivalência AFD e AFND-ϵ - Prova

	δ	a	b	ϵ
$\rightarrow$	q_0	q_1	$-$	q_1
	q_1	q_2	q_2	q_2
$*$	q_2	$-$	q_3	$-$
	q_3	q_1	q_0	$-$

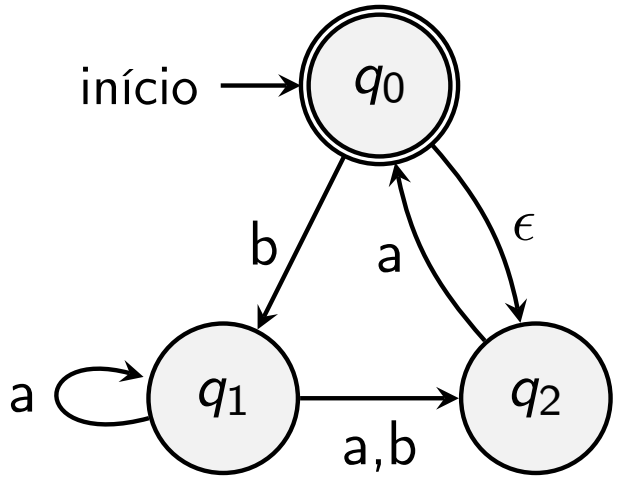
Equivalência AFD e AFND-ϵ - Prova

	δ	a	b	ϵ	ϵ-fecho
$\rightarrow$	q_0	q_1	$-$	q_1	$\{q_0, q_1, q_2\}$
	q_1	q_2	q_2	q_2	$\{q_1, q_2\}$
$*$	q_2	$-$	q_3	$-$	q_2
	q_3	q_1	q_0	$-$	q_3

Equivalência AFD e AFND-ϵ - Prova

	δ	a	b
$\rightarrow *\{q_0, q_1, q_2\}$		$\{q_1, q_2\}$	$\{q_2, q_3\}$
$*\{q_1, q_2\}$		q_2	$\{q_2, q_3\}$
$*\{q_2, q_3\}$		$\{q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
$*\{q_0, q_1, q_2, q_3\}$		$\{q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
$*q_2$		$-$	q_3
q_3		$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$

Equivalência AFD e AFND-ϵ - Prova



Equivalência AFD e AFND-ϵ - Prova

	δ	a	b	ϵ
$* \rightarrow$	q_0	$-$	q_1	q_2
	q_1	$\{q_1, q_2\}$	q_2	$-$
	q_2	q_0	$-$	$-$

Equivalência AFD e AFND-ϵ - Prova

	δ	a	b	ϵ	ϵ-fecho
$* \rightarrow$	q_0	$-$	q_1	q_2	$\{q_0, q_2\}$
	q_1	$\{q_1, q_2\}$	q_2	$-$	q_1
	q_2	q_0	$-$	$-$	q_2

Equivalência AFD e AFND-ϵ - Prova

	δ	a	b
$* \rightarrow$	$\{q_0, q_2\}$	$\{q_0, q_2\}$	q_1
	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$	q_2
$*$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
	q_1	$\{q_1, q_2\}$	q_2
	q_2	$\{q_0, q_2\}$	$-$

Operações Regulares

Teorema

A classe das linguagens regulares é fechada nas operações de:

- União
- Concatenação
- Estrela

Isso quer dizer que ao aplicarmos essas operações em linguagens regulares, o resultado também é uma linguagem regular.

União

Teorema

A classe das linguagens regulares é fechada na operação de união.

Isso quer dizer que dadas duas linguagens regulares, L_1 e L_2, então $L_1 \cup L_2$ também é uma linguagem regular.

União

Teorema

A classe das linguagens regulares é fechada na operação de união.

Para se provar isso é preciso demonstrar um autômato finito M capaz de reconhecer $L_1 \cup L_2$.

União - Prova por Produto Cartesiano

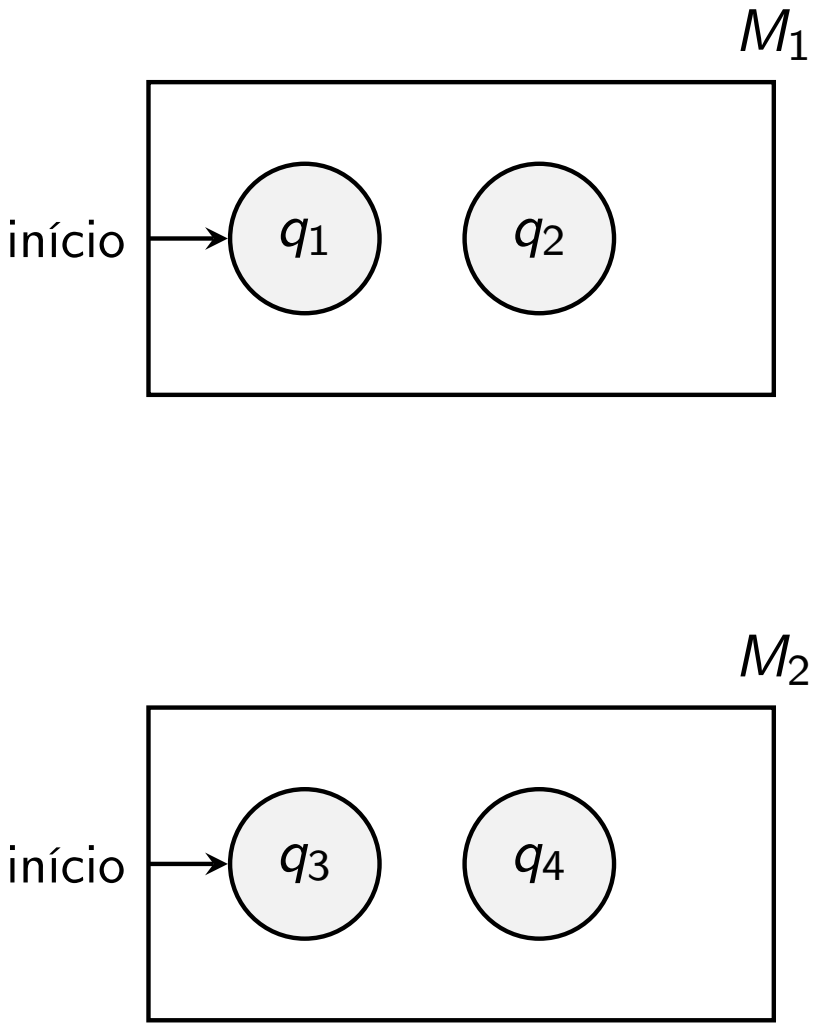
Dado um autômato finito $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ que reconheça L_1, e um autômato finito $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que reconheça L_2, construa M capaz de reconhecer $L_1 \cup L_2$ onde $M = (Q, \Sigma, \delta, q_0, F)$.

1. $Q = \{(r_1, r_2) | r_1 \in Q_1 \text{ e } r_2 \in Q_2\}$.
2. Σ, O alfabeto é o mesmo para ambos M_1 e M_2.
3. δ, para cada $(r_1, r_2) \in Q$ e cada $a \in \Sigma$,

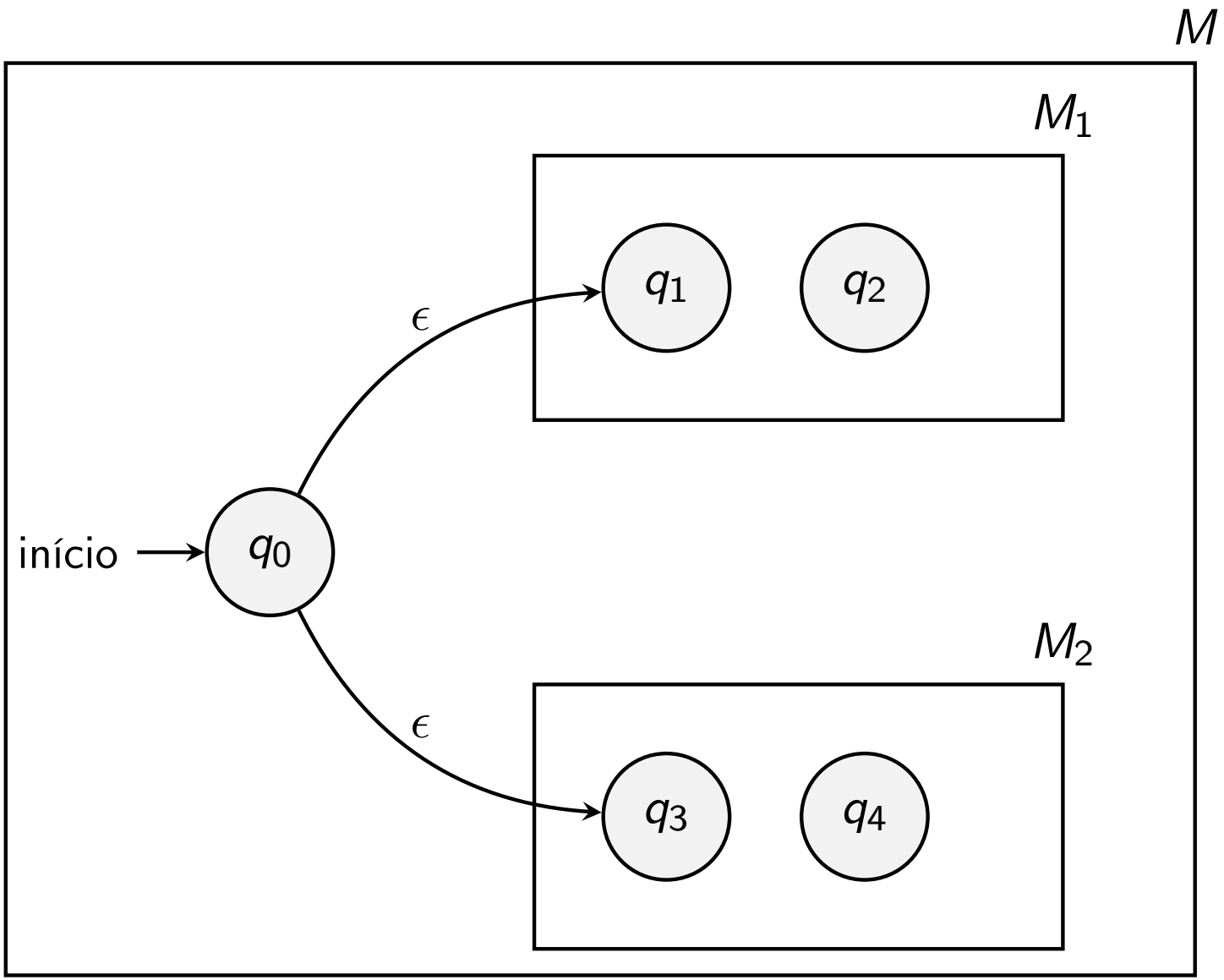
$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)).$$

4. q_0 é o par (q_1, q_2).
5. $F = \{(r_1, r_2) | r_1 \in F_1 \text{ ou } r_2 \in F_2\}$.

União - Prova por Não-determinismo



União - Prova por Não-determinismo



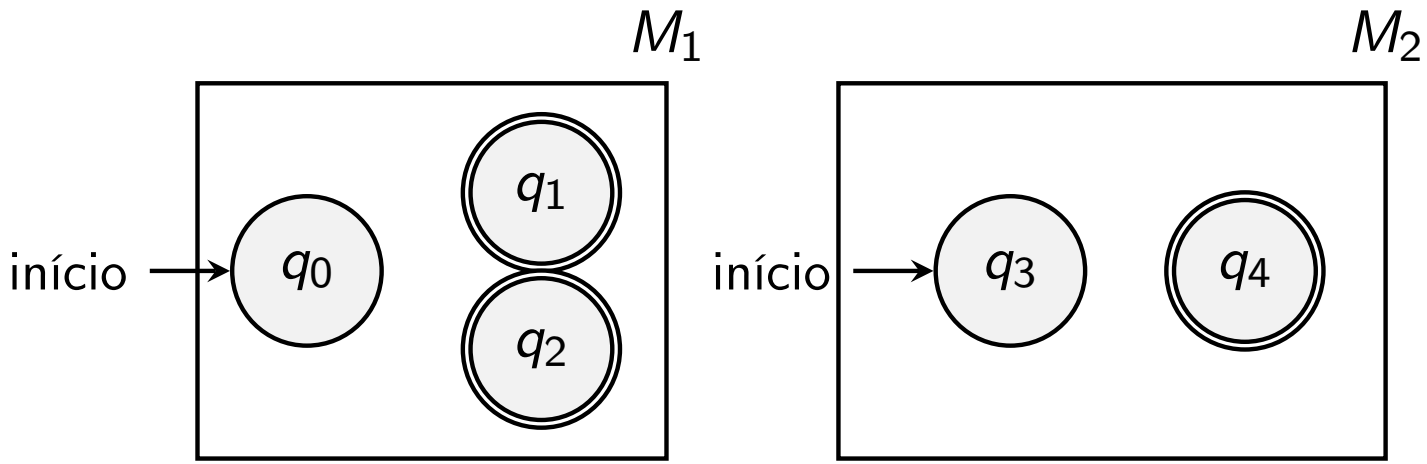
Concatenação

Teorema

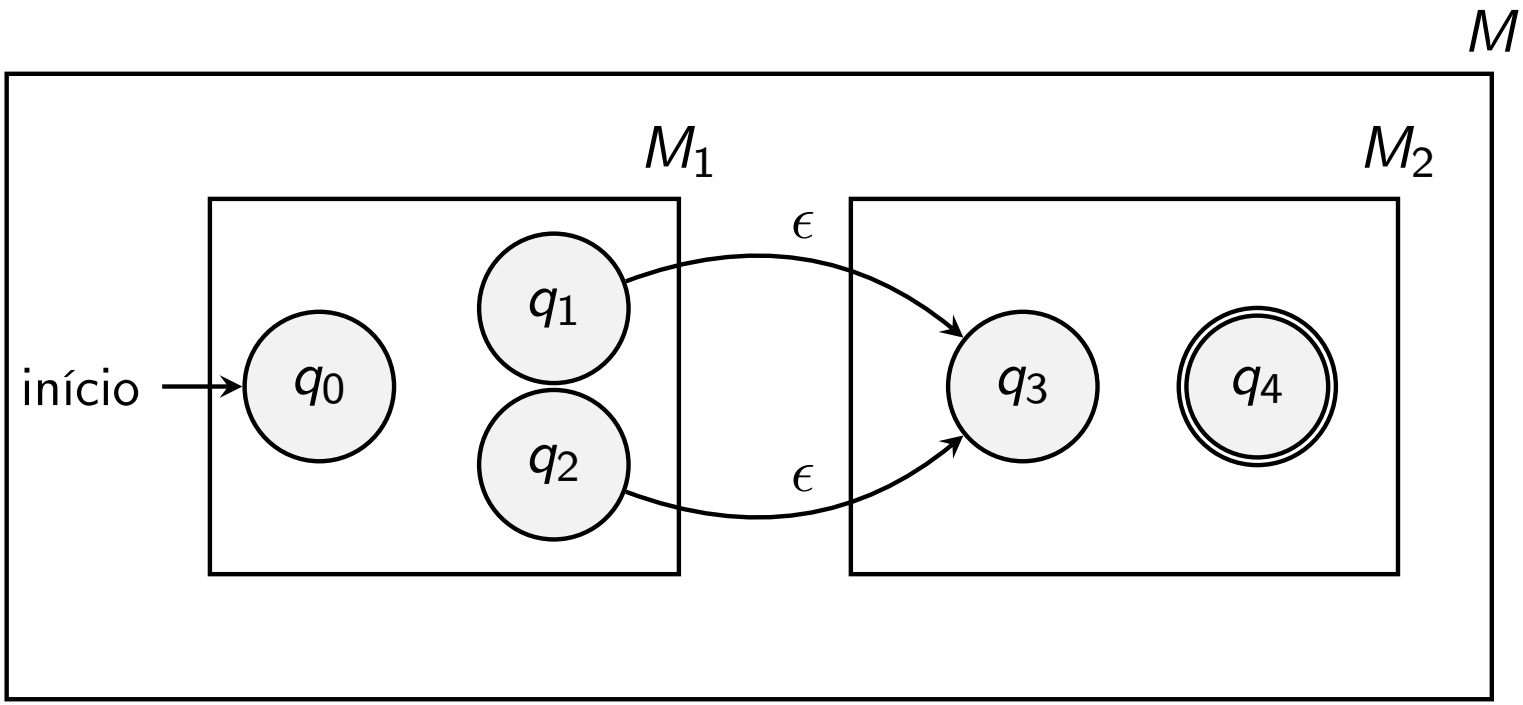
A classe das linguagens regulares é fechada na operação de concatenação.

Para provar o teorema é preciso projetar um autômato finito M capaz de reconhecer $L_1 \circ L_2$.

Concatenação - Prova por Não-determinismo



Concatenação - Prova por Não-determinismo



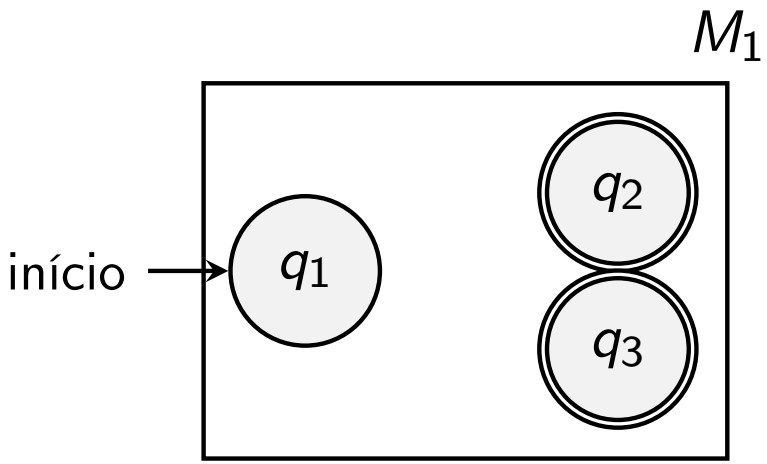
Estrela

Teorema

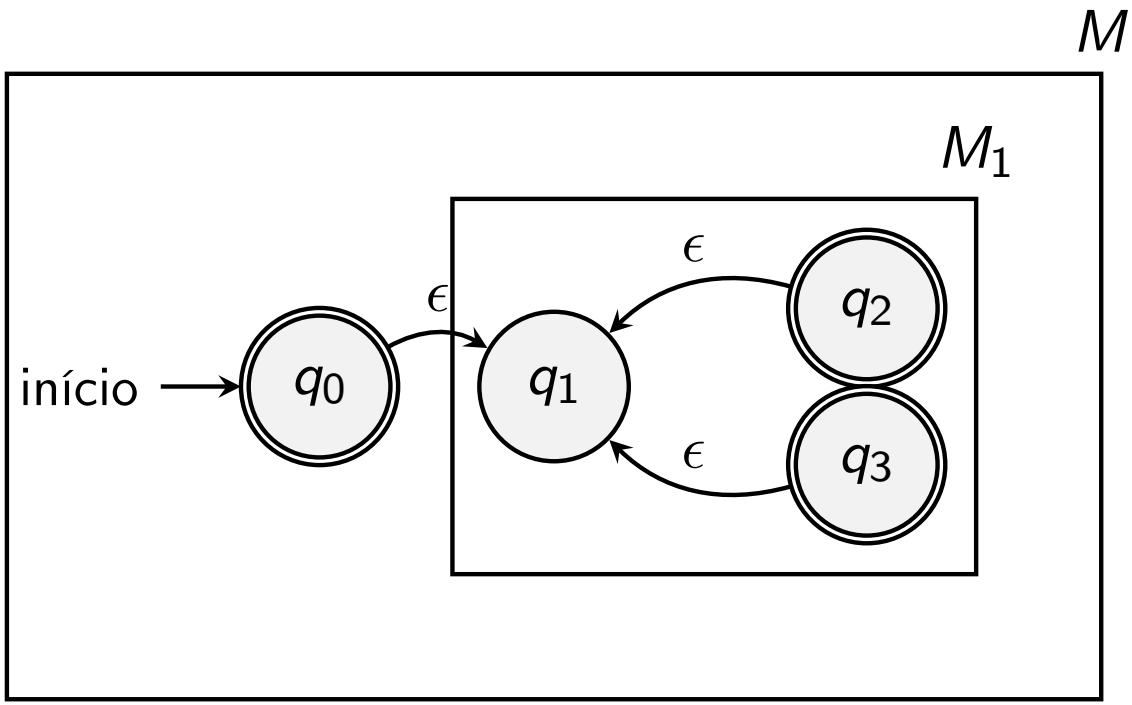
A classe das linguagens regulares é fechada na operação de Estrela.

Para provar o teorema é preciso projetar um autômato finito M capaz de reconhecer L^*.

Estrela - Prova por Não-Determinismo



Estrela - Prova por Não-Determinismo



Autômatos Finitos

Profª. Drª. Jerusa Marchi
Otto Menegasso Pires

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina