



# Automatos Finitos

Sistemas de Estados Finitos  
AF Determinísticos

(H&U, 1979) e (H;M;U, 2001)

## Sistemas de Estados Finitos

- Uma máquina de estados finitos é um modelo matemático de um sistema com entradas e saídas discretas.
- O sistema pode estar em qualquer um de um número finito de estados.
- O estado do sistema sumariza a informação relativa às entradas **passadas** que é necessária para determinar o comportamento do sistema nas próximas entradas.

## Memória Limitada

- AF são exemplos de modelos de computadores com memória limitada.
- Já vimos um exemplo na primeira aula de um AF modelando um interruptor On/Off
  - O interruptor "se lembra" se ele está no estado "on" ou "off" e permite o usuário apertar um botão cujo efeito é diferente, dependendo do seu estado.
- Mas, o que um computador com memória limitada pode fazer????
- **Muitas coisas úteis!**

## Exemplos

- **Controlador de porta automática que abre no meio**
  - Tem um sensor na frente e outro atrás da porta (para não bater em quem está atrás). O controlador está em 1 dos 2 estados: aberto, fechado e as transições seriam:
    - Frente (que faz a porta abrir e segurar aberta)
    - Atrás (que deixa a porta fechada ou mantém aberta)
    - Ambas (pessoas estão na frente atrás)
    - Nenhuma (não há pessoas nem na frente nem atrás)

## Exemplos

- **Mecanismo de controle de um elevador**
  - O mecanismo não se lembra das requisições de serviços anteriores,
    - cada estado sumariza somente as informações de andar atual e direção (para cima ou para baixo).
  - As entradas para o sistema são as chamadas a serem atendidas.

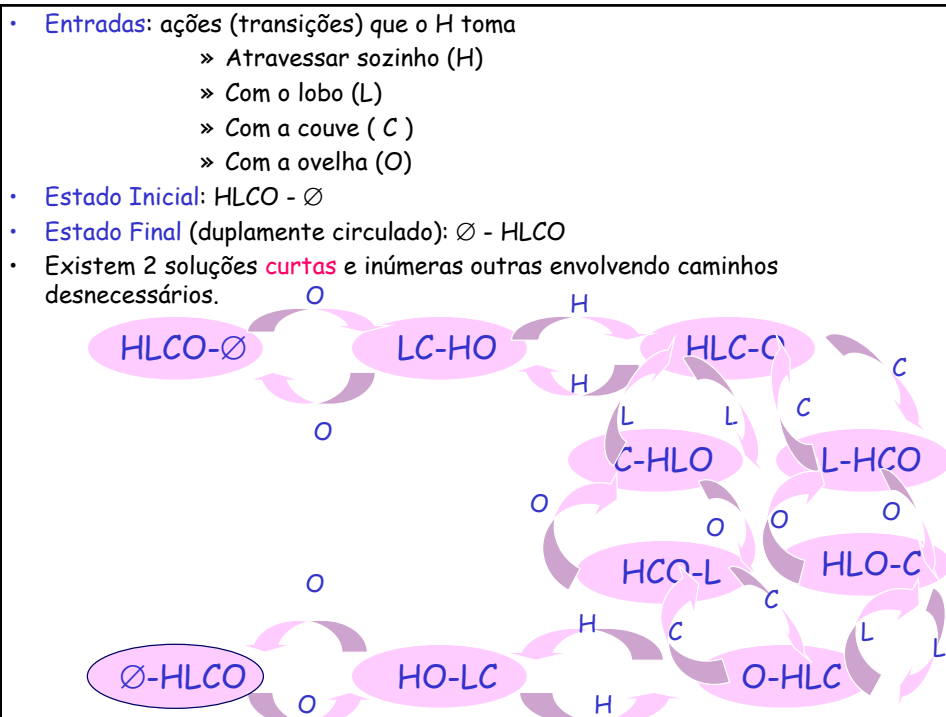
## Exemplos

- **Na computação, um AF é um modelo útil para projetar vários tipos de sistemas:**
  - **Busca de cadeias em editores de textos.** No Word, por exemplo, podemos expressar as cadeias com os caracteres curinga \* e outros especiais.

## Exemplos

- **Analísadores léxicos (AL) de compiladores.** O AL é a interface entre o programa fonte e o resto do compilador. Ele é responsável por:
  - "empacotar" os caracteres do programa e lhes dar um rótulo que será usado pelo analisador sintático montar a árvore sintática.
  - Os rótulos são: **identificadores**, os nomes dos **símbolos simples** (<, =, [, ), etc.), dos **compostos** (:=, <>, <=, etc.), **números inteiros**, **números reais**, **cadeias de caracteres**, o nome das **palavras-chaves**.
  - Eliminar caracteres brancos e comentários e, se for pedido, reeditar o texto fonte acrescentando número de linhas, indentação, etc.
  - Reconhecer alguns erros léxicos como má-formação de inteiros, de reais, de cadeias de caracteres, comentário que não fecha, caracteres não reconhecidos pela linguagem.

- **Problemas de IA** →
  - **Descrição do Problema:** um homem (H) com uma couve (C), um lobo (L) e uma ovelha (O) estão na margem esquerda de um rio. Existe um barco com **2 lugares** para carregar o homem com um dos outros três.
  - Se o H deixa a O com a C ela a come ou o L com a O ele a come.
  - **É possível os quatro atravessarem o rio sem a ovelha ou a couve serem comidas?**
  - O problema pode ser modelado analisando os ocupantes da margem esquerda e direita a cada travessia. Cada um é um **estado** menos os fatais (OC - HL, por exemplo).



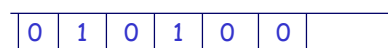
- O sistema é atípico pois:
  - Nele existe só um estado final MAS podem existir vários
  - Para cada transição existe uma reversa, MAS isto não é o caso geral
  - Depois do estado final não existe computação, MAS pode existir

## Definição formal de um AF Determinístico

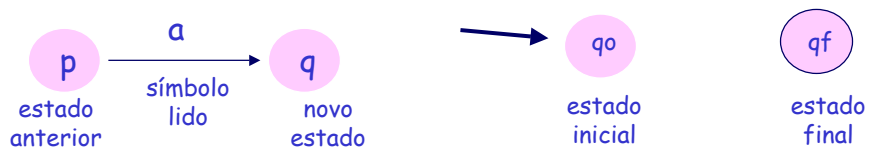
- Um AF Determinístico (AFD) é denotado formalmente por uma **quíntupla**  $(Q, \Sigma, \delta, q_0, F)$  onde:
- $Q$  é o conjunto de estados
- $\Sigma$  é um alfabeto finito
- $q_0 \in Q$  é o símbolo inicial
- $F \subseteq Q$  é o conjunto de símbolos finais
- $\delta$  (delta) é a função de transição de estados mapeando  $Q \times \Sigma \rightarrow Q$ .
- $\delta(q,a)$  é um estado para cada estado  $q$  e entrada  $a$

## Esquema da Máquina e Representação usando grafos

Esquema:



Fita com a sequência de símbolos de  $\Sigma$



### Aceitando cadeias - função de transição estendida

Estendemos a função de transição para aceitar cadeias:

$$\delta': Q \times \Sigma^* \rightarrow Q$$

$\delta'(q,w)$  é um estado  $p$  que o AF vai estar depois de ler a cadeia  $w$ , começando do estado  $q$ . Isto é, existe um caminho no diagrama de transições de  $q$  para  $p$  denominado  $w$

Definimos:

1)  $\delta'(q,\lambda) = q$

(sem ler um símbolo de entrada o AF não pode mudar de estado)

2)  $\delta'(q,wa) = \delta(\delta'(q,w),a)$  para  $a$  como símbolo de entrada e  $w$  como cadeia de entrada

(Diz como achar o estado depois de ler uma cadeia de entrada  $wa$  não vazia. Isto é, encontre o estado  $p = \delta'(q,w)$  depois de ler  $w$ . Então calcule  $\delta(p,a)$ )

Usaremos  $\delta$  no lugar de  $\delta'$  pois  $\delta(q,a) = \delta(\delta'(q, \lambda),a) = \delta'(q,a)$

### Linguagem aceita por um AF M

- Uma cadeia  $x$  é dita ser aceita pelo AFD  $M = (Q, \Sigma, \delta, q_0, F)$  se  $\delta(q_0,x) = p$  para algum  $p \in F$ .

Ou

- $L(M) = \{x \mid \delta(q_0,x) \in F\}$
- Def 1: Uma linguagem aceita por um AFD é uma linguagem regular (ou do tipo 3)
- Def 2: Dois AF  $M_1$  e  $M_2$  são equivalentes se  $L(M_1) = L(M_2)$

## Exercício

Fazer um AFD  $M$  que aceita  $L(M) = \{w \mid w \text{ possui um nro par de } 0\text{'s e de } 1\text{'s}\}$

Exemplos de cadeias da linguagem:

Lambda

00

11

1010

0101

...

## Exemplo 1

Fazer um AFD  $M$  que aceita  $L(M) = \{w \mid w \text{ possui um nro par de } 0\text{'s e de } 1\text{'s}\}$

Diagrama de Transição de Estados

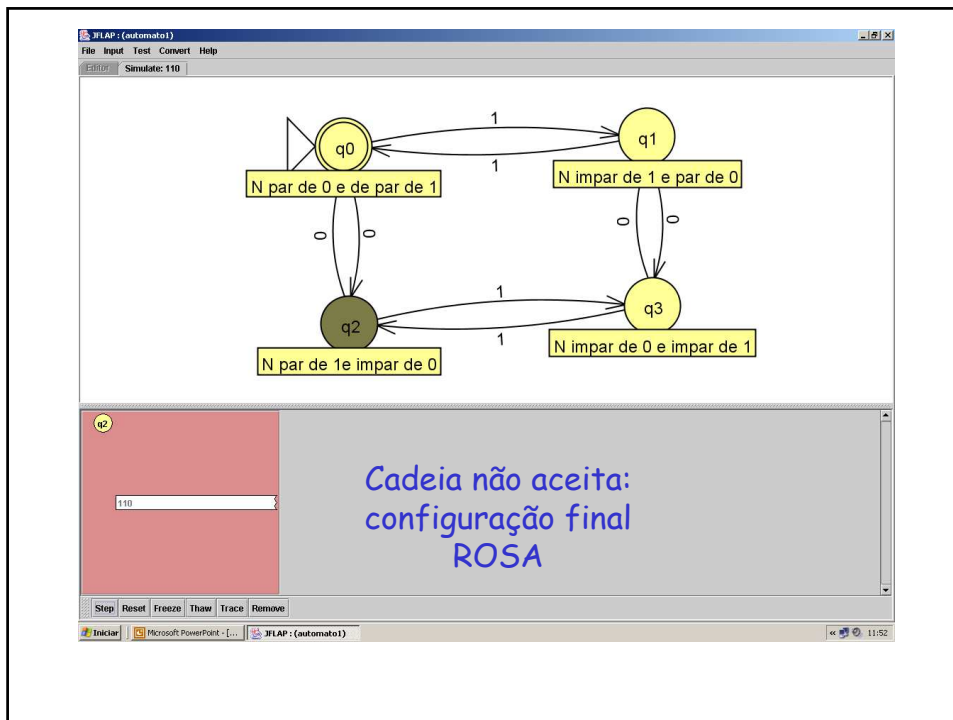
Cadeia aceita: configuração final VERDE

Traceback

q0	110101
q1	110101
q0	110101
q2	110101
q3	110101
q1	110101
q0	110101

Step Reset Freeze Thaw Trace Remove





### Reconhecimento de 110101

$$\delta(q_0, 1) = q_1 \text{ e } \delta(q_1, 1) = q_0$$

$$\text{Assim: } \delta(q_0, 11) = \delta(\delta(q_0, 1), 1) = \delta(q_1, 1) = q_0 \dots \delta(q_0, 110101) = q_0$$

Portanto 110101 está na  $L(M)$

$$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

$\delta$ : estados	entradas		Função de Transição de Estados
	0	1	
q0	q2	q1	$\delta(q_0, 0) = q_2$ $\delta(q_0, 1) = q_1$
q1	q3	q0	$\delta(q_1, 0) = q_3$ $\delta(q_1, 1) = q_0$
q2	q0	q3	$\delta(q_2, 0) = q_0$ $\delta(q_2, 1) = q_3$
q3	q1	q2	$\delta(q_3, 0) = q_1$ $\delta(q_3, 1) = q_2$

Tabela de Transição de Estados

## Como projetar um AF?

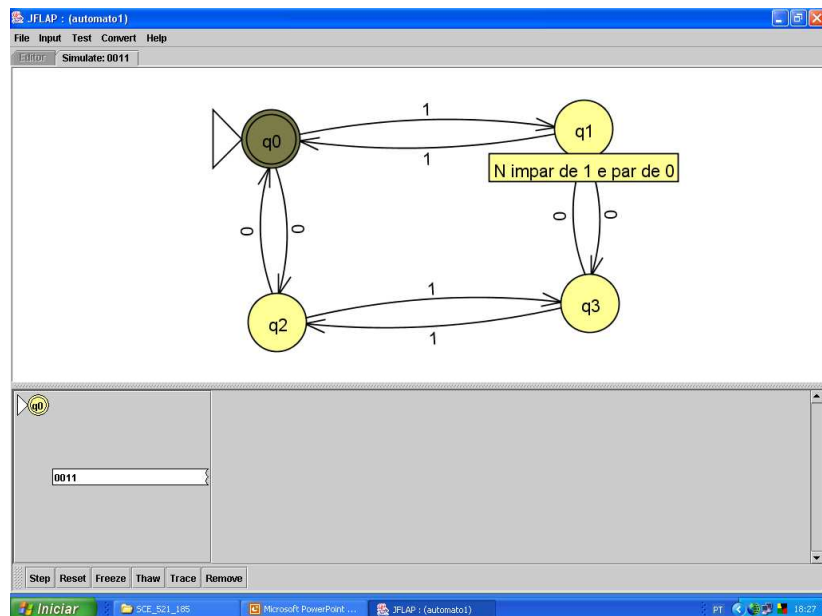
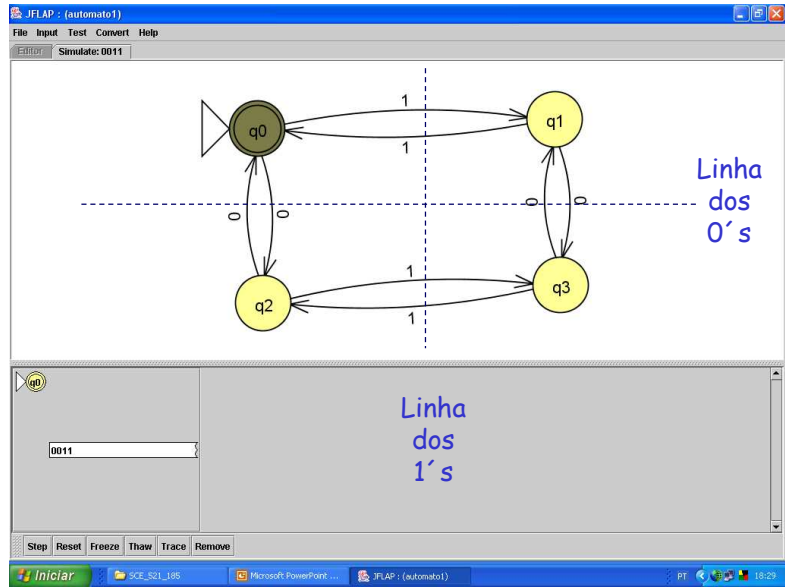
Tendo somente uma memória finita,

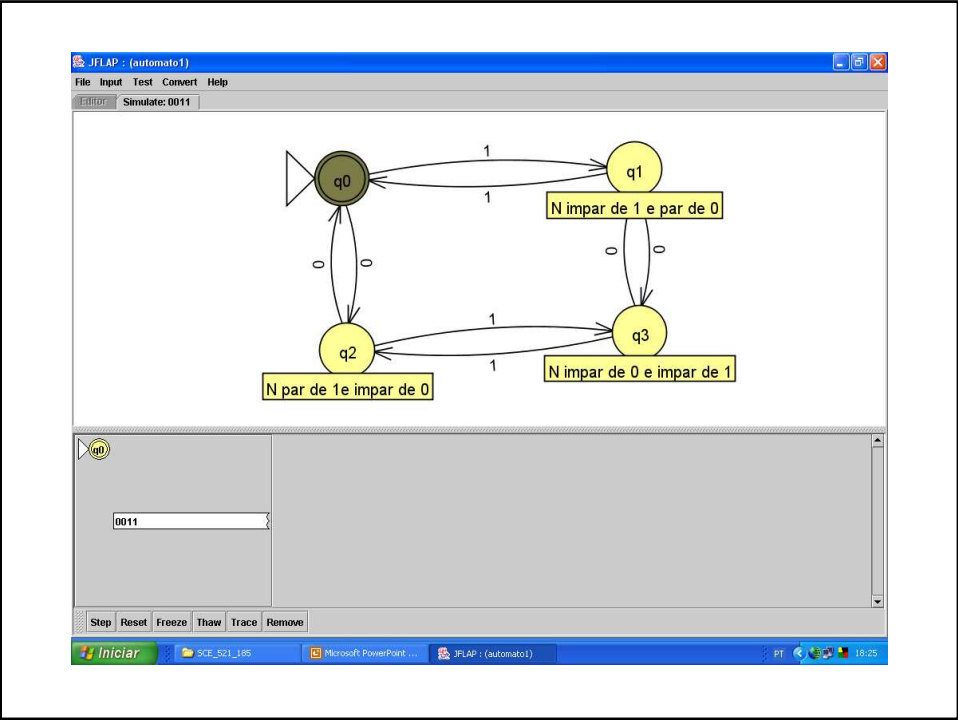
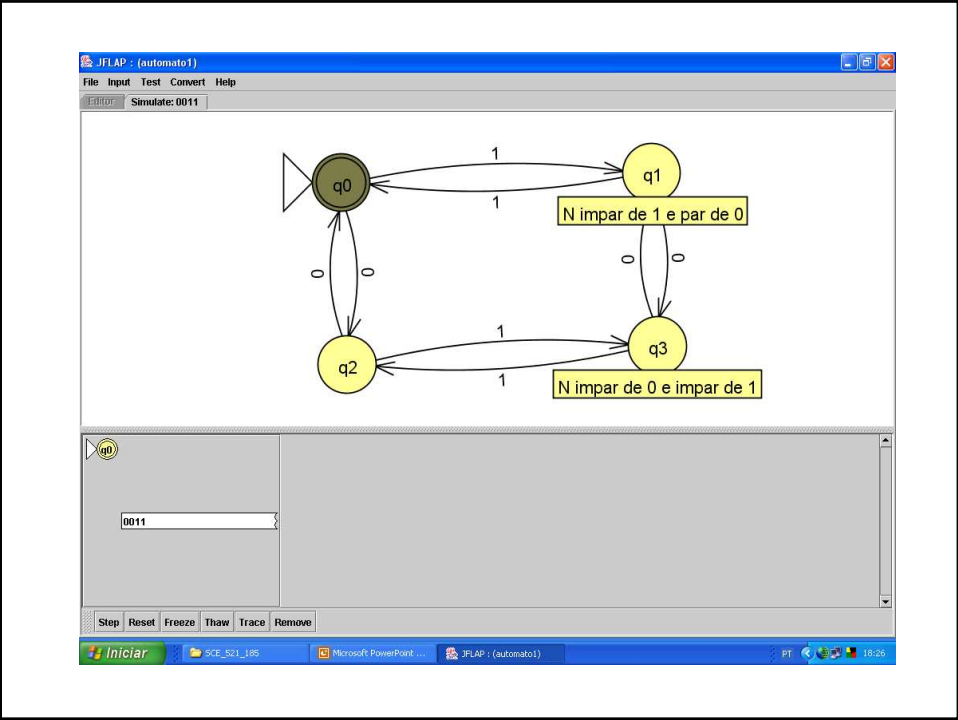
só podemos lembrar as informações importantes e associá-las aos estados

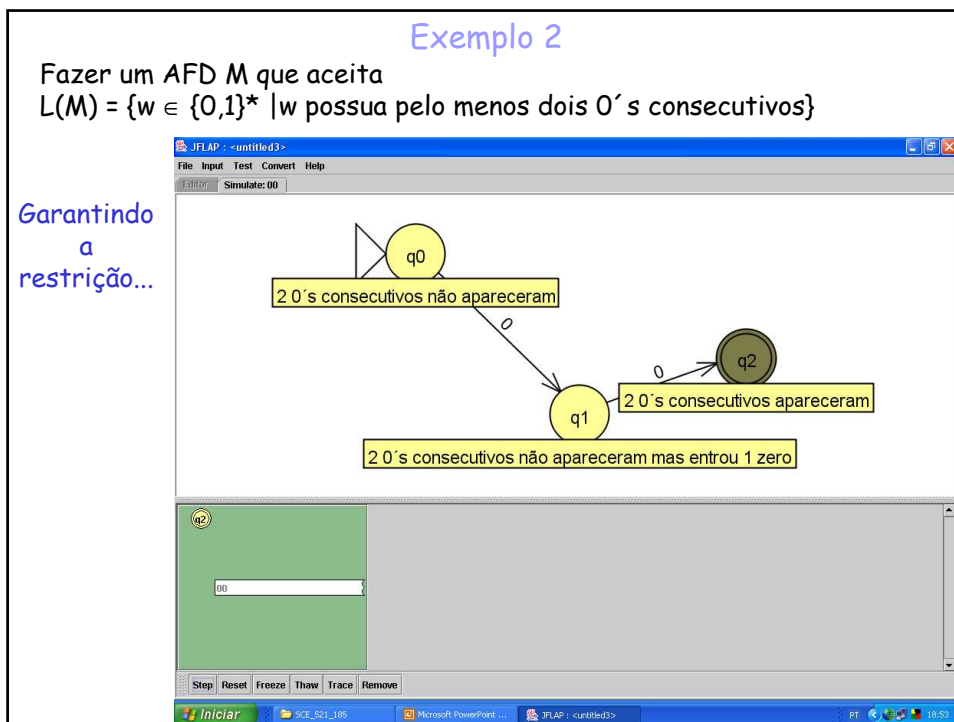
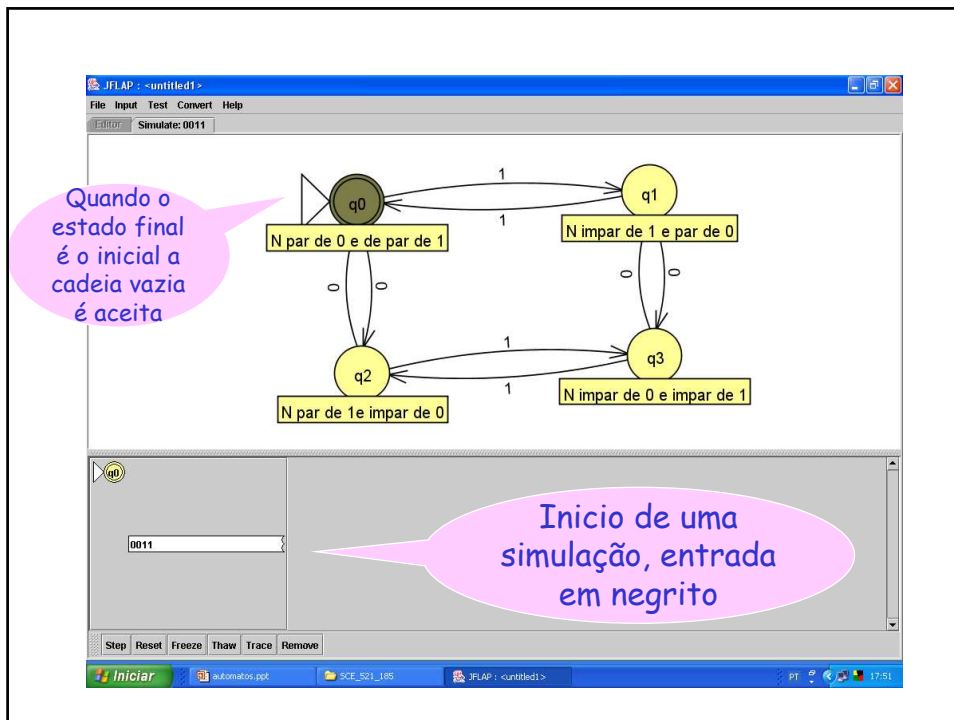
Usamos os estados para armazenar a paridade dos números

e não

os números o que exigiria um número infinito de estados (memória infinita).







Completando o AFD M e reconhecendo uma cadeia de L(M)

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta$ :

Fim de uma simulação, entrada em cinza

### Exemplo 3

Fazer um AFD M que aceita  $L(M) = \{w \in \{0,1\}^* \mid w \text{ possui um número ímpar de } 1\text{'s}\}$

Garantindo a restrição...

Nro par de 1's

nro ímpar de 1's

Completando o AFD M e reconhecendo uma cadeia de L(M)

$$M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

