

Automatos Finitos

a^n



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

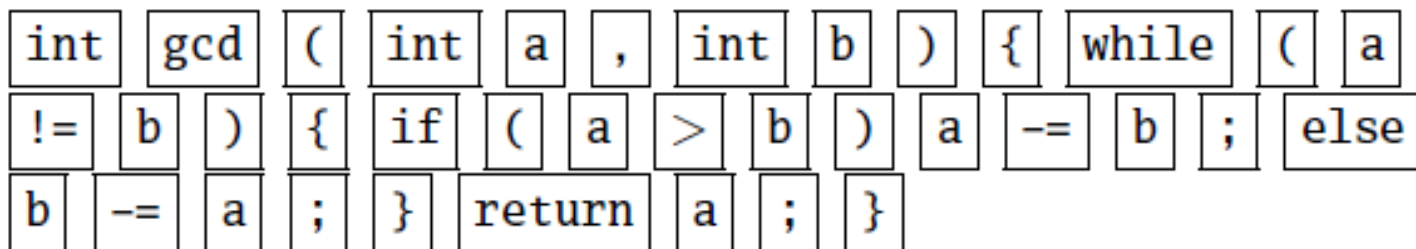
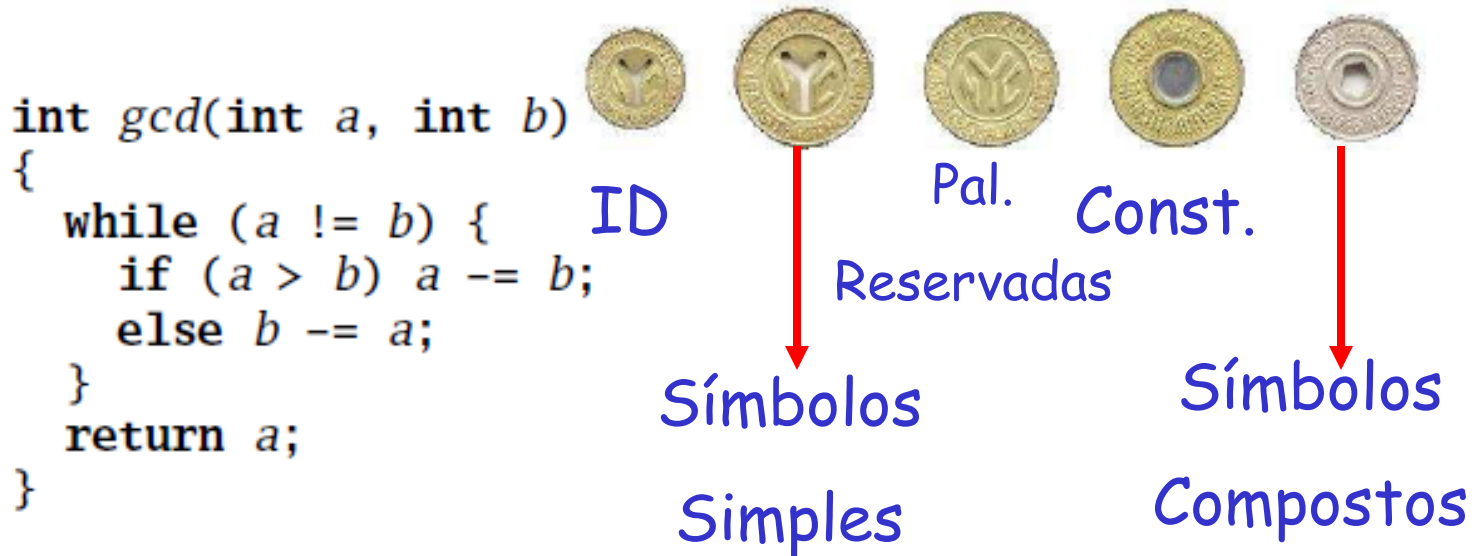
- **Mecanismo de controle de um elevador**
 - O mecanismo não se lembra das requisições de serviços anteriores,
 - cada estado sumariza somente as informações de andar atual e direção (para cima ou para baixo).
 - As entradas para o sistema são as chamadas a serem atendidas.

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.
- Análise Léxica fornece tokens: uma cadeia de tokens sem espaços e comentários para a análise sintática.



Analizador Léxico

Ele é responsável por:

“empacotar” os caracteres do programa e lhes dar um rótulo que será usado pelo analisador sintático montar a árvore sintática.

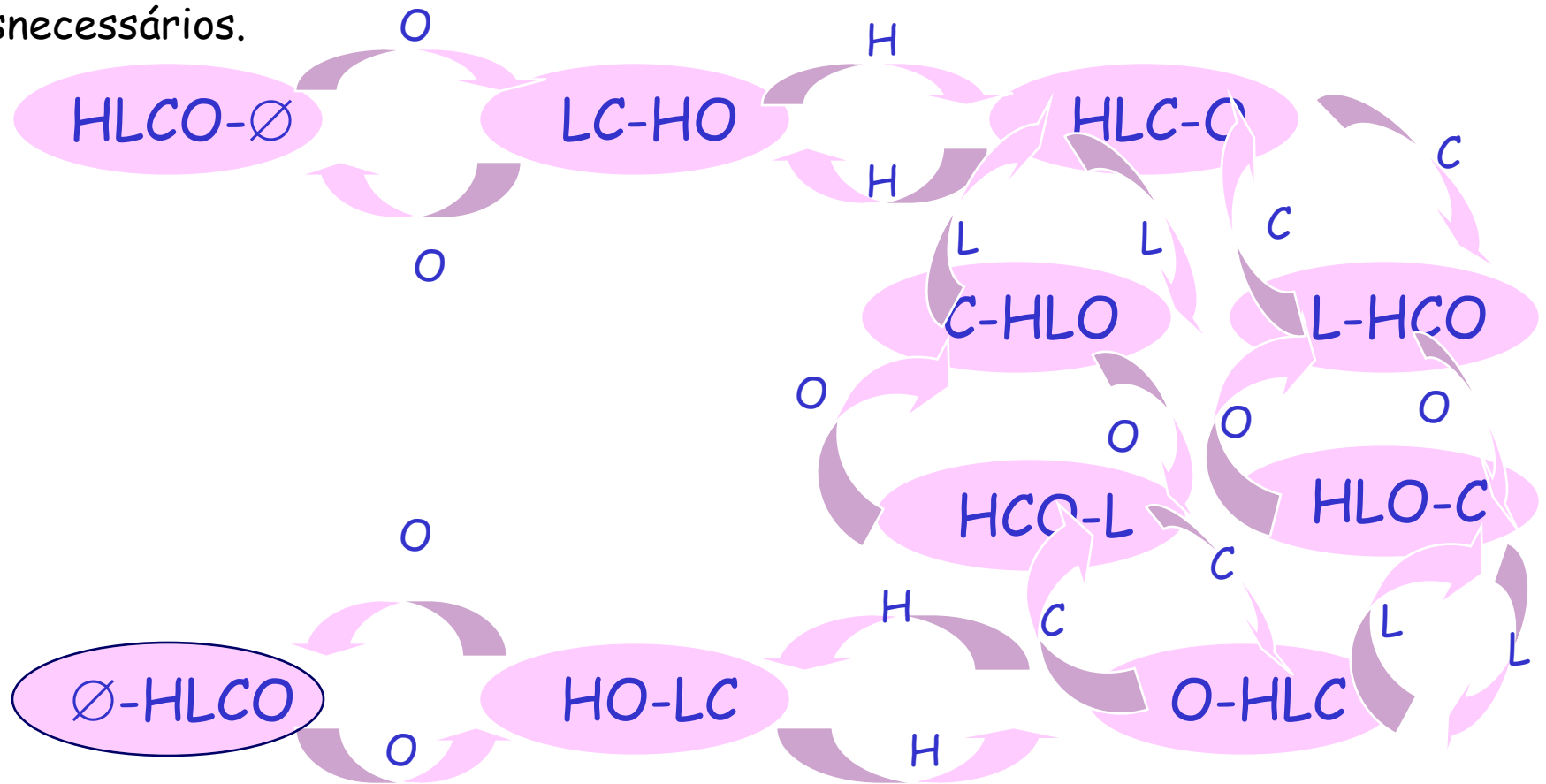
Os rótulos são: **identificadores**, os nomes dos **símbolos simples** (<, =, [,), etc.), dos **compostos** (:= , <>, <=, etc.), **números inteiros**, **números reais**, **cadeias de caracteres**, o nome das **palavras-chaves**.

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).

- Entradas: ações (transições) que o H toma
 - » Atravessar sozinho (H)
 - » Com o lobo (L)
 - » Com a couve (C)
 - » Com a ovelha (O)
- Estado Inicial: HLCO - ∅
- Estado Final (duplamente circulado): ∅ - HLCO
- Existem 2 soluções curtas e inúmeras outras envolvendo caminhos desnecessários.



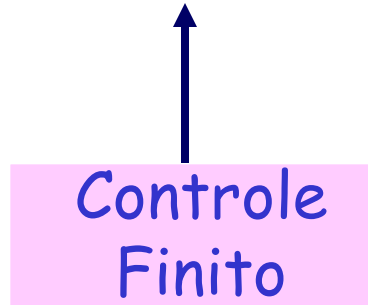
- O sistema é atípico pois:
 - Nele existe só um estado final MAS podem existir vários
 - Para cada transição existe uma reversa, MAS isto não é o caso geral
 - Depois do estado final não existe computação, MAS pode existir

Definição formal de um AF Determinístico

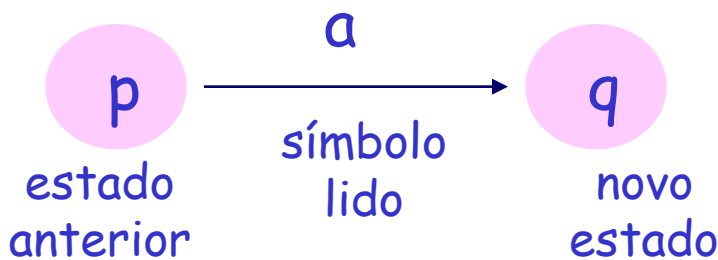
- Um AF Determinístico (AFD) é denotado formalmente por uma **quíntupla** $(Q, \Sigma, \delta, q_0, F)$ onde:
- Q é o conjunto de estados
- Σ é um alfabeto finito
- $q_0 \in Q$ é o símbolo inicial
- $F \subseteq Q$ é o conjunto de símbolos finais
- δ (delta) é a função de transição de estados mapeando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

Esquema da Máquina e Representação usando grafos

Esquema:



Fita com a seqüência de símbolos de Σ



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:

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

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

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

(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 δ no lugar de δ' 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:

λ

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

```
graph LR; q0((q0)) -- 1 --> q1((q1)); q1 -- 1 --> q0; q0 -- 0 --> q2((q2)); q2 -- 0 --> q0; q1 -- 0 --> q3((q3)); q3 -- 0 --> q1; q2 -- 1 --> q3; q3 -- 1 --> q2;
```

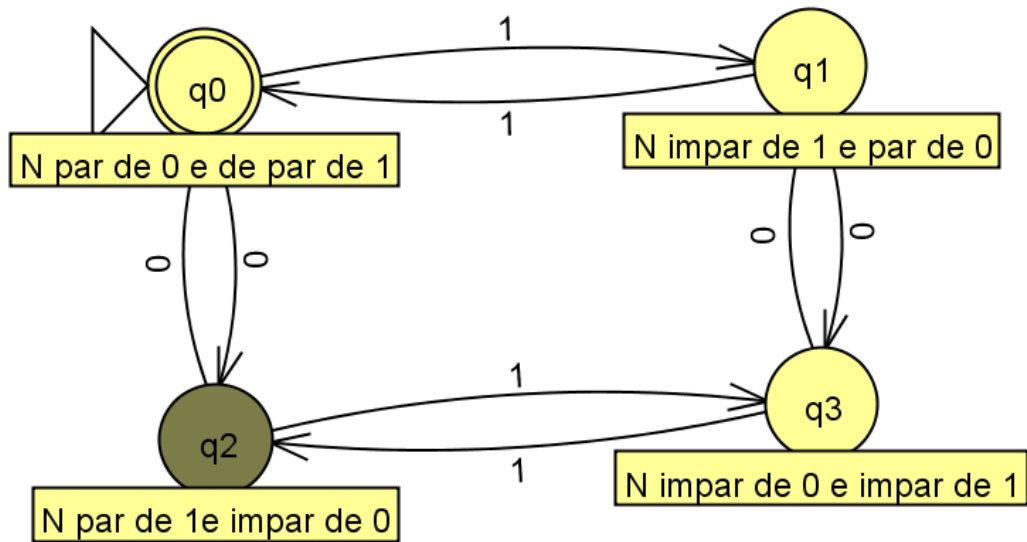
Cadeia aceita:
configuração final
VERDE

Traceback

▶ (q0)	110101
● (q1)	110101
▶ (q0)	110101
● (q2)	110101
● (q3)	110101
● (q1)	110101
▶ (q0)	110101

Step Reset Freeze Thaw Trace Remove

Windows Taskbar: Iniciar, SCE_521_185, automatos.ppt, JFLAP 4.0 Beta Ve..., JFLAP : <untitled1>, Traceback, PT, 15:44



q2

110

Cadeia não aceita:
configuração final
ROSA

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 $L(M)$

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

		entradas	
		0	1
$\delta :$	estados		
	q_0	q_2	q_1
	q_1	q_3	q_0
	q_2	q_0	q_3
q_3	q_1	q_2	

Função de Transição de Estados

$$\delta(q_0, 0) = q_2 \quad \delta(q_0, 1) = q_1$$

$$\delta(q_1, 0) = q_3 \quad \delta(q_1, 1) = q_0$$

$$\delta(q_2, 0) = q_0 \quad \delta(q_2, 1) = q_3$$

$$\delta(q_3, 0) = q_1 \quad \delta(q_3, 1) = q_2$$

Tabela de Transição de Estados

Como projetar um AF?

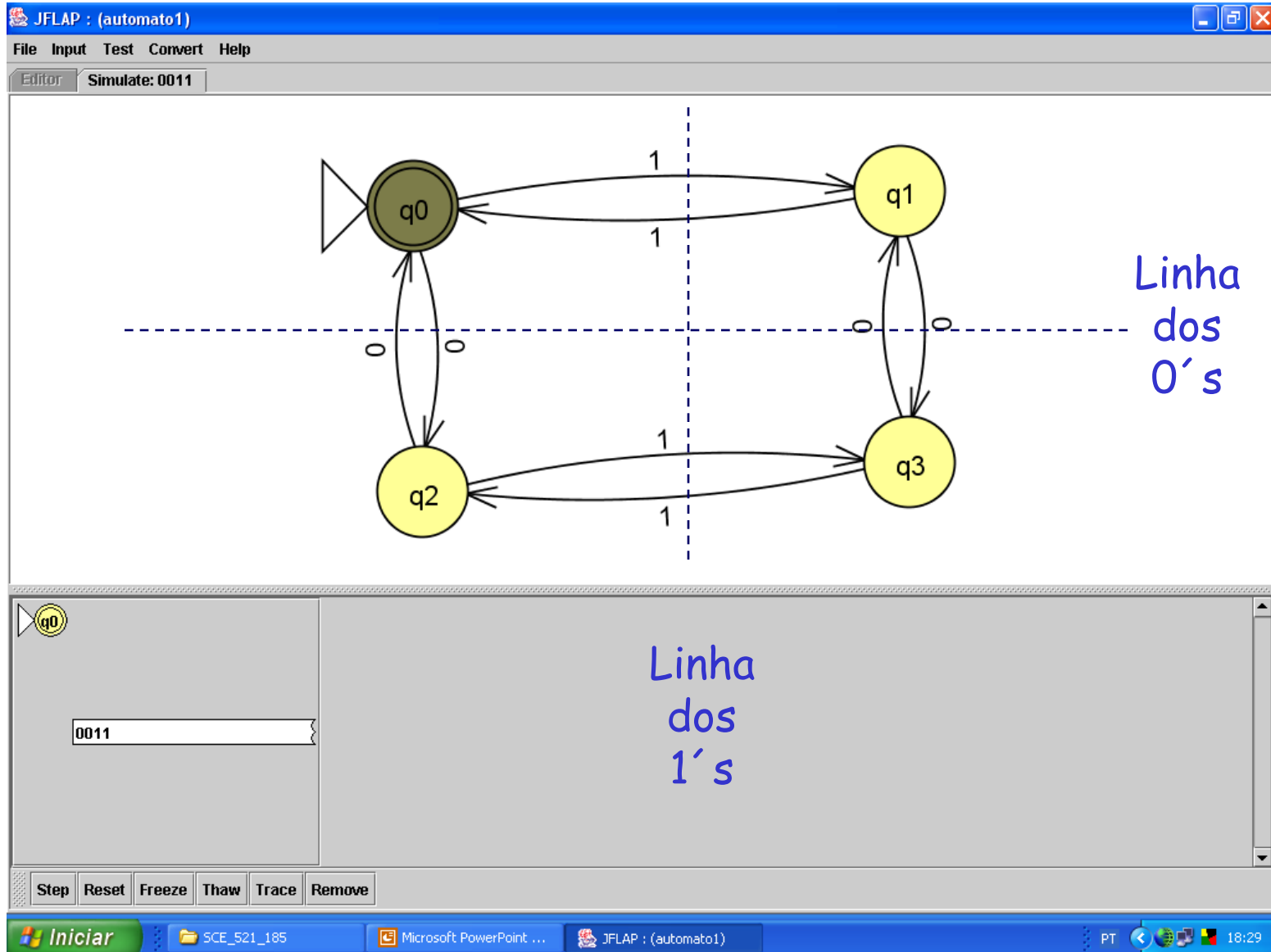
Tendo somente uma memória finita,

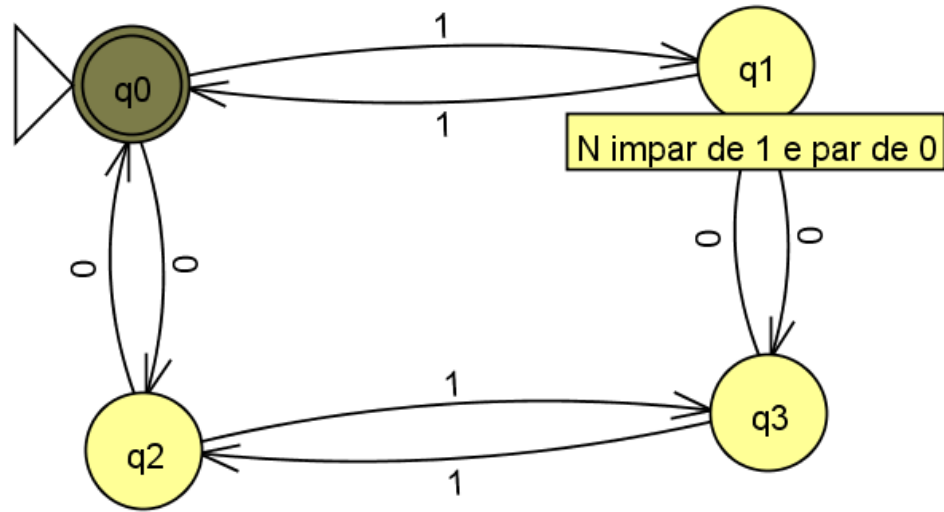
só podemos lembrar as informações importantes e associá-las aos estados

Usamos os estados para armazenar a paridade dos números

e não

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





q0

0011

JFLAP : (automato1)

File Input Test Convert Help

Editor Simulate: 0011

```
graph TD; q0((q0)) -- 1 --> q1((q1)); q1 -- 1 --> q0; q1 -- 0 --> q3((q3)); q3 -- 0 --> q1; q2((q2)) -- 1 --> q3; q3 -- 1 --> q2; q0 -- 0 --> q2; q2 -- 0 --> q0;
```

N impar de 1 e par de 0

N impar de 0 e impar de 1

q0

0011

Step Reset Freeze Thaw Trace Remove

Iniciar SCE_521_185 Microsoft PowerPoint ... JFLAP : (automato1) PT 18:26

JFLAP : (automato1)

File Input Test Convert Help

Editor Simulate: 0011

```
graph LR; q0((q0)) -- 1 --> q1((q1)); q1 -- 1 --> q0; q0 -- 0 --> q2((q2)); q2 -- 0 --> q0; q2 -- 1 --> q3((q3)); q3 -- 1 --> q2; q1 -- 0 --> q3; q3 -- 0 --> q1;
```

N impar de 1 e par de 0

N par de 1 e impar de 0

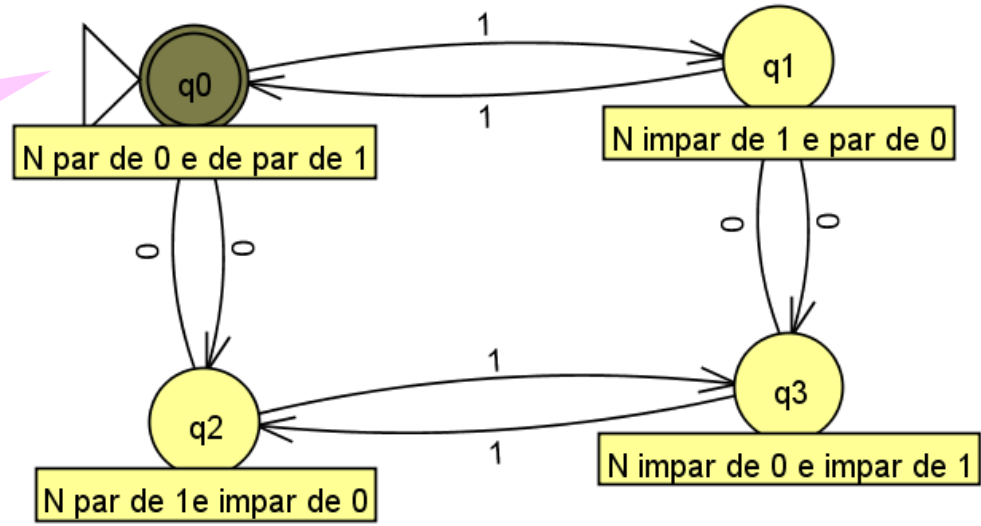
q0

0011

Step Reset Freeze Thaw Trace Remove

Iniciar SCE_521_185 Microsoft PowerPoint ... JFLAP : (automato1) PT 18:25

Quando o estado final é o inicial a cadeia vazia é aceita



Início de uma simulação, entrada em negrito

q0

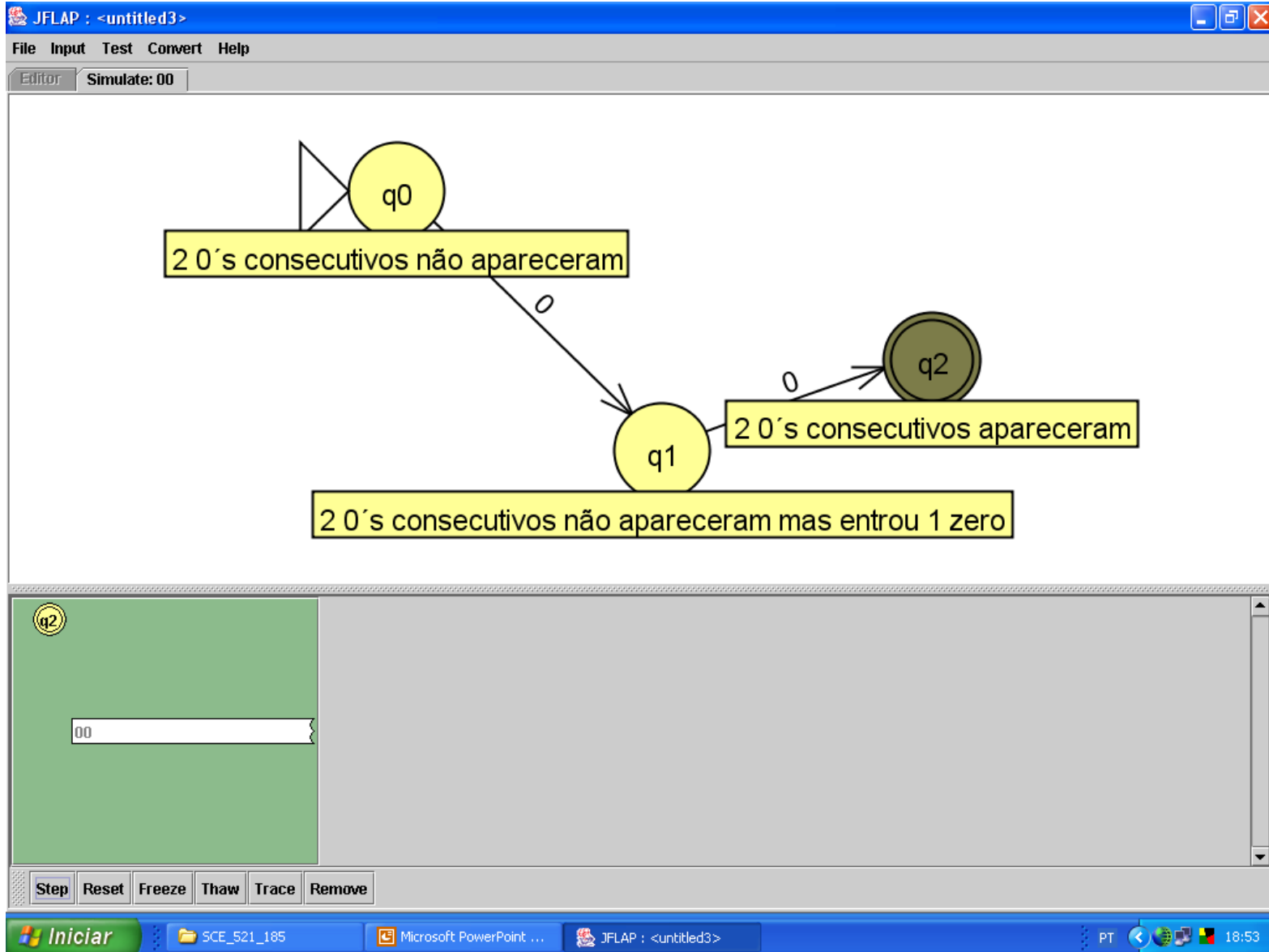
0011

Exemplo 2

Fazer um AFD M que aceita

$L(M) = \{w \in \{0,1\}^* \mid w \text{ possui pelo menos dois } 0\text{'s consecutivos}\}$

Garantindo
a
restrição...



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

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

JFLAP : <untitled3>

File Input Test Convert Help

Editor Simulate: 1110010101

δ :

```
graph LR; q0((q0)) -- 1 --> q0; q0 -- 0 --> q1((q1)); q1 -- 1 --> q0; q1 -- 0 --> q2(((q2))); q2 -- 0 --> q2; q2 -- 1 --> q2;
```

Fim de uma simulação, entrada em cinza

1110010101

Step Reset Freeze Thaw Trace Remove

Iniciar SCE_521_185 Microsoft PowerPoint ... JFLAP : <untitled3> PT 19:00

Exemplo 3

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

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...

The screenshot shows the JFLAP interface. At the top, the title bar reads "JFLAP : <untitled2>". Below it is a menu bar with "File", "Input", "Test", "Convert", and "Help". The main window is divided into two sections. The upper section displays a DFA diagram with two states: q_0 (a yellow circle) and q_1 (a grey circle). q_0 is the start state, indicated by a triangle on its left. q_1 is the final state, indicated by a double circle on its right. There are two transitions between q_0 and q_1 , both labeled with the input symbol '1'. Below q_0 is a yellow box containing the text "Nro par de 1's", and below q_1 is a yellow box containing "nro ímpar de 1's". The lower section is a simulation window with a green background. It shows a small circle with q_1 in the top left corner and a white input field containing the string "111". At the bottom of the simulation window, there is a toolbar with buttons for "Step", "Reset", "Freeze", "Thaw", "Trace", and "Remove". Below the toolbar, a message reads: "Moves existing valid configurations to the next configurations." The Windows taskbar at the bottom shows the "Iniciar" button, several open applications including "Microsoft PowerPoint ..." and "SCE_521_185", and the system tray with the time "19:14".

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

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

JFLAP : <untitled2>

File Input Test Convert Help

Editor Simulate: 001110

δ :

```
graph LR; q0((q0)) -- 0 --> q0; q0 -- 1 --> q1((q1)); q1 -- 0 --> q1; q1 -- 1 --> q0;
```

Aceitação no JFLAP = verde,
com indicação do estado que
parou ou ler a cadeia de
entrada

Step Reset Freeze Thaw Trace Remove

Microsoft PowerPoint ... SCE_521_185 JFLAP : <untitled2> PT 19:17