

# ***Curso de Engenharia de Computação***

## ***ECM253 – Linguagens Formais, Autômatos e Compiladores***

### **Modelos de computação – Máquinas de estados finitos**



# ***Agenda***

- Máquinas de estados finitos
- Softwares para máquinas de estados
- Máquinas de estados com saídas

# ***Agenda***

- Máquinas de estados finitos
- Softwares para máquinas de estados
- Máquinas de estados com saídas

# Máquinas de estados finitos

## ■ Conceitos

- Um **computador** pode ser entendido de forma **abstrata** como uma **função de transição** do tipo  $T : S \times I \rightarrow S \times O$ , onde  $S$  é o conjunto de **estados**,  $I$  é o conjunto de **símbolos de entrada** e  $O$  é o conjunto de **símbolos de saída**;
- A função  $T$  **mapeia**, de modo genérico, um par contendo um **estado atual** e um certo símbolo de **entrada** para um **novo estado** e uma **saída** a ser emitida neste estado;
- Se o conjunto de estados  $S$  é finito então esta função caracteriza uma **máquina de estados finito**;
- Se o conjunto  $S \times I$  não for muito grande, pode-se **descrever**  $T$  com o auxílio de um **grafo**;
- Normalmente, para serem viáveis, a **capacidade de informação** de máquinas de estado finitos é **pequena**.

# ***Máquinas de estados finitos***

## ■ **Aplicabilidade em Computação:**

- Corretores ortográficos;
- Verificação gramatical;
- Indexação e busca em textos grandes;
- Reconhecimento de voz;
- Transformação de texto com XML e HTML;
- Protocolos de rede;
- ...e muito mais ...

# Máquinas de estados finitos

## ■ Capacidade de informação

- Uma máquina de estados finito realiza computações por transições entre seus estados;
- Se a **capacidade de informação**<sup>1</sup> de tal máquina fosse  $C$  **bits**, ou seja, se cada estado pudesse ser identificado por um único valor de  $C$  bits de tamanho, então tal **máquina** teria, no máximo,  $2^C$  **estados**;
- Então, se fôssemos utilizar máquinas de estados finitos para representar um computador real, com todas as possibilidades de endereçamento de memória e de disco, a quantidade de estados seria tão grande que sua **construção seria impraticável** (por exemplo, uma máquina com 512MB RAM e 60GB de disco possui  $512 \times 2^3 \times 2^{10} \times 2^{10} + 60 \times 2^3 \times 2^{10} \times 2^{10} \times 2^{10}$  bits).

---

<sup>1</sup> Conceito da *Teoria da Informação*, muito importante na teoria de redes de computadores, por exemplo.

# Máquinas de estados finitos

## ■ Capacidade de informação

- **Felizmente**, mesmo para se descrever máquinas reais com máquinas de estados finitos, não é necessário utilizar tantos estados, pois, **na prática**:
  - **Ignoram-se fatores físicos reais** (complexidade de hardware, tempos de transmissão da informação, dissipação de calor);
  - Na descrição do **comportamento** de **máquinas reais**, ignoram-se grande parte dos estados possíveis, pois é apenas uma **pequena parcela dos estados** que são **observáveis** e **desejáveis** pelo projetista – os demais estados podem ser codificados em estados não desejáveis ou de erro e podem ser fortemente simplificados.

# Máquinas de estados finitos

- Exemplo de máquina de estados finitos

- Máquina de vendas

- Considerar uma **máquina de refrigerantes** que deva contemplar os **requisitos** a seguir:
  - i. Inicialmente, a máquina não possui nenhum valor coletado;
  - ii. aceitar apenas moedas de R\$0,25, R\$0,50 e R\$1,00;
  - iii. O preço dos refrigerantes é fixado em R\$2,00;
  - iv. Pode-se depositar quantas moedas quiser, desde que sejam do tipo apresentado. Se o valor depositado fizer com que o valor coletado seja maior que R\$2,00, então a máquina deverá dispensar um troco adequado;
  - v. A máquina possui um único botão que, ao ser pressionado, dispensará um refrigerante, se o valor coletado for R\$2,00;
  - vi. Ao dispensar um refrigerante, a máquina retorna para seu estado inicial e está pronta para receber novos depósitos, repetindo o cenário acima.



# Máquinas de estados finitos

- Exemplo de máquina de estados finitos

- Máquina de vendas

- A máquina descrita anteriormente pode ser assim formulada:

- Conjunto de estados:

$$S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$$

Cada estado representa um acúmulo de dinheiro possível com os valores de moedas apresentado:  $s_0$  é o estado inicial, R\$0,00;  $s_1$  representa R\$0,25 acumulado;  $s_2$  representa R\$0,5 acumulado;  $s_3$  representa R\$0,75 acumulado;  $s_4$  representa R\$1,00 acumulado;  $s_5$  representa R\$1,25 acumulado;  $s_6$  representa R\$1,50 acumulado;  $s_7$  representa R\$1,75 acumulado e  $s_8$  representa R\$2,00 acumulado. O estado  $s_8$  é o estado da máquina do qual se poderá pressionar o botão e retirar o refrigerante.

# Máquinas de estados finitos

## ▪ Exemplo de máquina de estados finitos

### – Máquina de vendas

#### ▪ Descrição da máquina (cont.)

##### – Conjunto de símbolos de entrada:

$$I = \{m_{25}, m_{50}, m_{100}, b\}$$

Cada um dos **símbolos** ( $m_{25}, m_{50}, m_{100}$ ) representam as **moedas aceitas** pela máquina; o **símbolo**  $b$  representa o **botão** que, ao ser pressionado, poderá dispensar ou não o refrigerante.

##### – Conjunto de símbolos de saída:

$$O = \{t_{25}, t_{50}, t_{75}, t_{100}, r, n\}$$

Cada um dos símbolos ( $t_{25}, t_{50}, t_{75}, t_{100}$ ) representam os possíveis **trocos** dispensados pela máquina; o **símbolo**  $r$  representa o **refrigerante** que poderá ser dispensado pela máquina e o **símbolo**  $n$  representa “**nada**”, ou seja, a máquina não emite nem troco e nem refrigerante. Para isto símbolo, poder-se-ia utilizar  $\epsilon$ .

# Máquinas de estados finitos

## ▪ Exemplo de máquina de estados finitos

### – Máquina de vendas

#### ▪ Descrição da máquina (cont.)

##### – Função de transição de estados:

$$T = \{((s_0, b), (s_0, n)), ((s_0, m_{25}), (s_1, n)), ((s_0, m_{50}), (s_2, n)), ((s_0, m_{100}), (s_4, n)),$$

$$((s_1, b), (s_1, n)), ((s_1, m_{25}), (s_2, n)), ((s_1, m_{50}), (s_3, n)), ((s_1, m_{100}), (s_5, n)),$$

$$((s_2, b), (s_2, n)), ((s_2, m_{25}), (s_3, n)), ((s_2, m_{50}), (s_4, n)), ((s_2, m_{100}), (s_6, n)),$$

$$((s_3, b), (s_3, n)), ((s_3, m_{25}), (s_4, n)), ((s_3, m_{50}), (s_5, n)), ((s_3, m_{100}), (s_7, n)),$$

$$((s_4, b), (s_4, n)), ((s_4, m_{25}), (s_5, n)), ((s_4, m_{50}), (s_6, n)), ((s_4, m_{100}), (s_8, n)),$$

$$((s_5, b), (s_5, n)), ((s_5, m_{25}), (s_6, n)), ((s_5, m_{50}), (s_7, n)), ((s_5, m_{100}), (s_8, t_{25})),$$

$$((s_6, b), (s_6, n)), ((s_6, m_{25}), (s_7, n)), ((s_6, m_{50}), (s_8, n)), ((s_6, m_{100}), (s_8, t_{50})),$$

$$((s_7, b), (s_7, n)), ((s_7, m_{25}), (s_8, n)), ((s_7, m_{50}), (s_8, t_{25})), ((s_7, m_{100}), (s_8, t_{75})),$$

$$((s_8, b), (s_0, r)), ((s_8, m_{25}), (s_8, t_{25})), ((s_8, m_{50}), (s_8, t_{50})), ((s_8, m_{100}), (s_8, t_{100}))\}$$

Esta função é melhor apreciada por uma **tabela de estados** ou por um **diagrama de estados**.

# Máquinas de estados finitos

## Exemplo de máquina de estados finitos

### Máquina de vendas

#### Descrição da máquina (cont.)

#### Tabela de estados:

	Próximo estado				Saída			
	Entrada				Entrada			
Estado	<i>b</i>	<i>m</i> <sub>25</sub>	<i>m</i> <sub>50</sub>	<i>m</i> <sub>100</sub>	<i>b</i>	<i>m</i> <sub>25</sub>	<i>m</i> <sub>50</sub>	<i>m</i> <sub>100</sub>
<i>s</i> <sub>0</sub>	<i>s</i> <sub>0</sub>	<i>s</i> <sub>1</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>4</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
<i>s</i> <sub>1</sub>	<i>s</i> <sub>1</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>3</sub>	<i>s</i> <sub>5</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
<i>s</i> <sub>2</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>3</sub>	<i>s</i> <sub>4</sub>	<i>s</i> <sub>6</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
<i>s</i> <sub>3</sub>	<i>s</i> <sub>3</sub>	<i>s</i> <sub>4</sub>	<i>s</i> <sub>5</sub>	<i>s</i> <sub>7</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
<i>s</i> <sub>4</sub>	<i>s</i> <sub>4</sub>	<i>s</i> <sub>5</sub>	<i>s</i> <sub>6</sub>	<i>s</i> <sub>8</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>
<i>s</i> <sub>5</sub>	<i>s</i> <sub>5</sub>	<i>s</i> <sub>6</sub>	<i>s</i> <sub>7</sub>	<i>s</i> <sub>8</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>t</i> <sub>25</sub>
<i>s</i> <sub>6</sub>	<i>s</i> <sub>6</sub>	<i>s</i> <sub>7</sub>	<i>s</i> <sub>8</sub>	<i>s</i> <sub>8</sub>	<i>n</i>	<i>n</i>	<i>n</i>	<i>t</i> <sub>50</sub>
<i>s</i> <sub>7</sub>	<i>s</i> <sub>7</sub>	<i>s</i> <sub>8</sub>	<i>s</i> <sub>8</sub>	<i>s</i> <sub>8</sub>	<i>n</i>	<i>n</i>	<i>t</i> <sub>25</sub>	<i>t</i> <sub>75</sub>
<i>s</i> <sub>8</sub>	<i>s</i> <sub>0</sub>	<i>s</i> <sub>8</sub>	<i>s</i> <sub>8</sub>	<i>s</i> <sub>8</sub>	<i>r</i>	<i>t</i> <sub>25</sub>	<i>t</i> <sub>50</sub>	<i>t</i> <sub>100</sub>

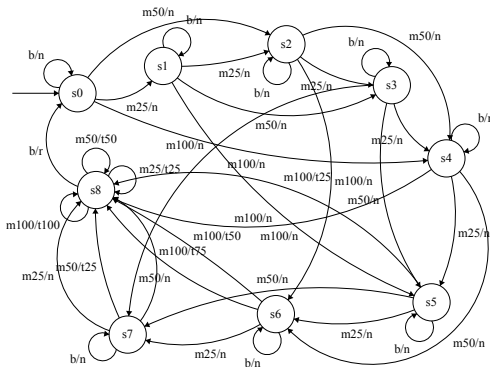
# Máquinas de estados finitos

## Exemplo de máquina de estados finitos

### Máquina de vendas

#### Descrição da máquina (cont.)

#### Diagrama de estados:



# ***Agenda***

- Máquinas de estados finitos
- Softwares para máquinas de estados
- Máquinas de estados com saídas

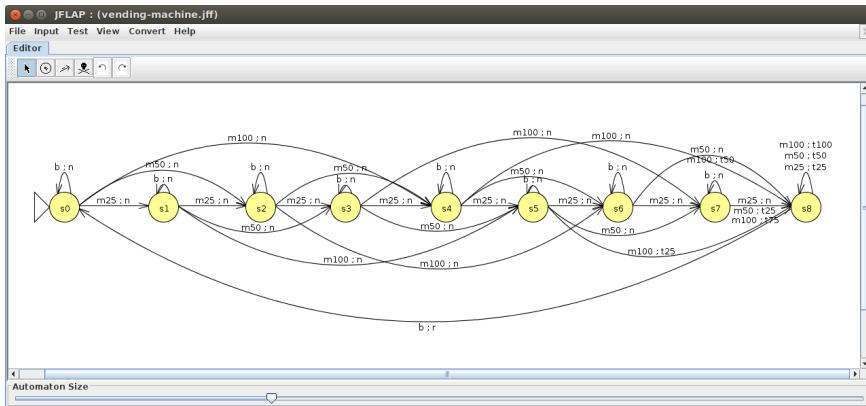
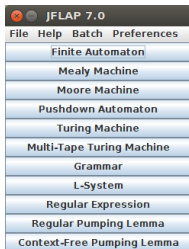
## ***Softwares para máquinas de estados***

- **JFLAP**: atende diversos tipos de máquinas de estados finitos e autômatos.
  - Site: [<http://www.jflap.org/>](http://www.jflap.org/).
  - Programa:  
[<http://www.jflap.org/jflaptmp/may15-2011/withoutSource/JFLAP.jar>](http://www.jflap.org/jflaptmp/may15-2011/withoutSource/JFLAP.jar).
- **JFAST**: específico para autômatos finitos.
  - Site: [<http://jfast-fsm-sim.sourceforge.net/>](http://jfast-fsm-sim.sourceforge.net/).
  - Programa:  
[<http://sourceforge.net/projects/jfast-fsm-sim/files/>](http://sourceforge.net/projects/jfast-fsm-sim/files/).

# Softwares para máquinas de estados

## ■ Exercício

- Testar o software JFLAP editando e simulando a máquina de vendas de refrigerantes apresentada anteriormente.





# ***Agenda***

- Máquinas de estados finitos
- Softwares para máquinas de estados
- Máquinas de estados com saídas

# Máquinas de estados com saídas

## ■ Conceitos

**Definição.** Uma máquina de estados finitos  $M = (S, I, O, f, g, s_0)$  consiste em um **conjunto**  $S$  de **estados**, um **alfabeto finito** de **entrada**, um **alfabeto finito** de **saída**, uma **função de transição**  $f$ , que atribui a cada par de estado e entrada um novo estado, uma **função de saída**  $g$ , que atribui a cada par de estado e entrada uma saída e um **estado inicial**  $s_0$ .

# Máquinas de estados com saídas

## ■ Tipos

- **Máquinas de Mealy:** as saídas ocorrem nas transições entre estados. Todas as máquinas de estados finitos apresentados nesta aula são máquinas de Mealy;
- **Máquinas de Moore:** as saídas ocorrem nos estados. Assim, a modificação conceitual em relação às máquinas de Mealy é que a função  $g$  é definida apenas sobre os estados, e não sobre pares de estados e entradas.
- **Comparação entre os tipos**
  - As máquinas de Mealy tem usualmente menos estados, mas a sua implementação é mais complexa que as máquinas de Moore;
  - As máquinas de Moore são mais seguras de se usar, pois suas saídas ocorrem sempre em um estado e para haver a transição para um estado, é necessário ocorrer um ciclo de relógio (síncrono).

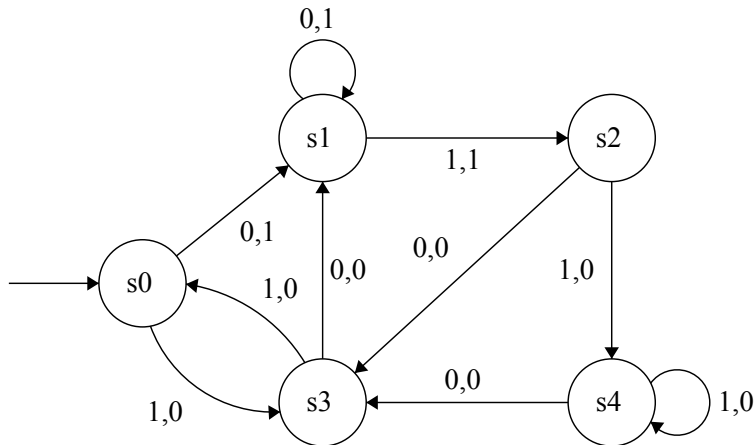
## Teste seus conhecimentos

- (1) Elaborar o diagrama de estados da máquina representada pela tabela de estados a seguir e então simulá-la.

	$f$		$g$	
	Entrada		Entrada	
Estado	0	1	0	1
$s_0$	$s_1$	$s_0$	1	0
$s_1$	$s_3$	$s_0$	1	1
$s_2$	$s_1$	$s_2$	0	1
$s_3$	$s_2$	$s_1$	0	0

## Teste seus conhecimentos

(2) Elaborar uma tabela de estados correspondente ao diagrama de estados a seguir:



## Teste seus conhecimentos

- (3) Simular a máquina do exercício 2 e determinar a cadeia de saída para a seguinte cadeia de entrada 101011.
- (4) Construir um diagrama de estados para a máquina de Moore representada pela tabela de estados a seguir:

	$f$		
	<b>Entrada</b>		
<b>Estado</b>	0	1	$g$
$s_0$	$s_0$	$s_2$	0
$s_1$	$s_3$	$s_0$	1
$s_2$	$s_2$	$s_1$	1
$s_3$	$s_2$	$s_0$	1