

# **Matemática Discreta**

## *Máquinas de Estado Finito*

Beatriz Dora Waricoda

Guilherme Henrique Gregorini Oliveira

Curso: Engenharia da Computação

Prof. Alisson C. Reinol

Data: 25/11/2018

## Sumário

Introdução .....	3
Constituição .....	4
Regras .....	4
Tipos de máquinas de estado finito .....	5
Representação gráfica .....	5
Tabela de Estado .....	5
Grafo de estado .....	6
Exemplo prático – O Poste .....	6
Implementação.....	7
Eletrônica.....	7
Programação.....	8
Conclusão .....	9
Bibliografia.....	10

## Introdução

Ao se construir soluções na área da engenharia é provável que utilizemos modelos que facilitam em seu desenvolvimento. Um desses modelos são as máquinas de estado finito, na qual se utiliza máquinas cujo armazenamento de *estados* é finito. Por exemplo, se utilizar um software como solução, a memória disponível do computador no qual esse software está sendo executado é o limite (finito) de estados possíveis de se armazenar.

A cada instante, a máquina está em um, e somente um, estado, o chamado *estado atual*.

Um relógio interno da máquina é responsável por criar o *ciclo*, em cada qual rotinas serão executadas periodicamente.

Nessas rotinas é obtida uma *entrada* e a partir dela em conjunto ao *estado atual* da máquina são obtidos um novo estado e uma *saída*. Abordaremos mais sobre essas rotinas posteriormente.

Desta forma, uma sequência de ciclos requer uma sequência de entradas as quais geram uma sequência de estados e saídas.

## Constituição

Uma máquina estado finito é constituída por um **conjunto de estados finito** que ela admite, um **conjunto finito de símbolos (valores) de entrada** possíveis, um **conjunto finito de símbolos (valores) de saída** possíveis, uma **função (rotina) que gera um estado** (próximo estado) a partir de um estado (estado atual) e uma entrada. E por fim, uma **função (rotina) que gera uma saída** a partir de um estado.

Ou seja, uma máquina de estado pode ser definido por  $M = [S, I, O, f_s, f_o]$ , onde:

- $S$  : **conjunto de estados finito**;
- $I$  : **conjunto finito de símbolos (valores) de entrada**;
- $O$  : **conjunto finito de símbolos (valores) de saída**;
- $f_s$  : **função (rotina) que gera um estado**;
- $f_o$  : **função (rotina) que gera uma saída**;

A função  $f_s$  determina o próximo estado a partir do estado atual e de um valor de entrada, ou seja:

$$f_s : S \times I \rightarrow S$$

A função  $f_o$  determina a saída a partir do estado atual, ou seja:

$$f_o : S \rightarrow O$$

A cada ciclo, as rotinas  $f_s$  e  $f_o$  são executadas. Para explicá-las considere uma máquina de  $n$  ciclos no qual  $t_i$  representa cada ciclo, onde  $1 \leq i \leq n$ . Ou seja,  $t_1$  é primeiro ciclo,  $t_2$  é segundo ciclo, assim sucessivamente até  $t_n$  for o último ciclo. E  $t_i$  um ciclo qualquer.

A máquina utiliza  $f_s$  para saber qual será seu próximo estado, ou seja:

$$estado_{t_{i+1}} = f_s(estado_{t_i}, entrada_{t_i})$$

Desta forma o estado atual é atualizado a todo ciclo pela rotina  $f_s$ .

A máquina utiliza  $f_o$  para saber qual é saída do estado atual, ou seja:

$$saída_{t_i} = f_o(estado_{t_i})$$

Desta forma a cada ciclo é gerada uma saída.

## Regras

Algumas regras devem ser respeitadas para o funcionamento da máquina:

1. A máquina sempre é iniciada em um estado inicial;
2. Existem um número finito de estados que a máquina pode alcançar;
3. A cada ciclo a máquina deve estar em um, e somente um, dos seus estados;
4. As rotinas devem ser sincronizadas por ciclos, ou seja, a rotina de um ciclo  $t=1$  não pode acontecer depois da rotina do ciclo  $t=2$ ;
5. Com uma sequência de entradas determinada deve ser possível determinar a sequências das saídas;

6. A máquina pode ser capaz de produzir saídas, as quais são relacionadas diretamente ao estado atual;

### Tipos de máquinas de estado finito

Até agora foi descrito uma **máquina de Moore**, porém existe similarmente a **máquina de Mealy** na qual sua diferença é:

- A função  $f_o$  não só depende de um estado para se obter uma saída, mas também **depende da entrada naquele ciclo**. Desta forma:

$$saída_{t_i} = f_o(estado_{t_i}, entrada_{t_i})$$

Ou

$$f_o: SXI \rightarrow O$$

### Representação gráfica

Uma máquina de estado M pode ser representada graficamente por uma tabela e/ou grafo de estado. Para tanto, é necessário definir seus três conjuntos,  $S, I, O$ , e então suas duas funções  $f_s$  e  $f_o$  pode ser definidas na construção do gráfico em questão.

Para exemplificação a seguir, considere os três conjuntos da máquina de estado M:

$$S = \{NOITE, DIA\}$$

$$I = \{0,1\}$$

$$O = \{0,1\}$$

### Tabela de Estado

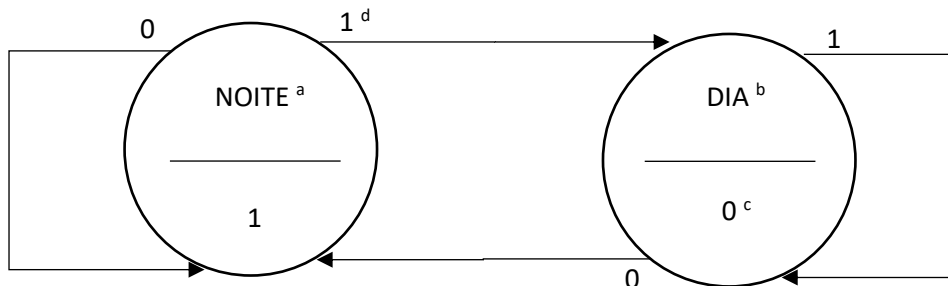
Estado atual	Próximo estado		Saída <sup>d</sup>
	Entrada atual		
	0	1 <sup>b</sup>	
NOITE <sup>a</sup>	NOITE	DIA <sup>c</sup>	1
DIA <sup>e</sup>	NOITE	DIA	0 <sup>f</sup>

Tabela 1 - Tabela de estado

É importante perceber na tabela de estado acima que as  $f_s$  e  $f_o$  foram definidas. Pois, como visto anteriormente  $f_s$  define qual será o próximo estado partindo de um estado atual e uma entrada atual, por exemplo, no estado atual *NOITE* <sup>a</sup> quando a entrada atual é 1 <sup>b</sup>, o próximo estado será *DIA* <sup>c</sup>. Na última coluna (*Saída* <sup>d</sup>) é possível observar  $f_o$ , pois esta define qual será a saída de cada estado, por exemplo, se o estado atual for *DIA* <sup>e</sup>, então a saída será 0 <sup>f</sup>.

## Grafo de estado

Um grafo direcionado cujo vértices representa cada estado, arestas definem  $f_s$  os rótulos de seus vértices definem  $f_o$  é chamada de **grafo de estado**. Veja o grafo de estado abaixo para M:



No grafo de estado acima, é possível perceber a relação  $f_s$  através das arestas direcionadas cuja início tem um valor de entrada. Por exemplo, no estado *NOITE*<sup>a</sup> há uma aresta cujo início tem valor  $1^d$  com fim conectado ao vértice *DIA*<sup>b</sup>, isso significa que se o valor de entrada for  $1$  e a entrada atual for *NOITE* para máquina M, então o próximo estado será *DIA*. Também é possível observar a  $f_o$ , por exemplo o estado *DIA*<sup>b</sup> tem com saída  $0^c$ .

## Exemplo prático – O Poste

Um exemplo prático pode ser elaborado em cima de um problema simples, o poste. Um poste de luz, desses que tem na rua, apaga durante a noite e acende durante o dia. Abstraindo dessa problemática para uma máquina de estados finitos, temos dois conjuntos:

- $S = \{NOITE, DIA\}$ , que representa os **estados possíveis do poste**
- $O = \{0,1\}$ , que representa o nível lógico que a luz do poste pode assumir (0 para luz apagada e 1 para a luz acesa), ou seja, **conjunto de saída**

Nesse exemplo usaremos um *LDR (Light Dependent Resistor)* como sensor da luz solar. A saída desse sensor é analógica com resolução de 10bits, ou seja, conseguimos fazer uma leitura desse sensor com um conjunto de números inteiros de 0 a 1024. Quanto **maior luminosidade o LDR recebe, menor é o valor lido**. Obviamente, quanto mais escuro, maior será o valor lido. Daí temos um novo conjunto para nossa máquina de estado finito:

- $I = \{x \in \mathbb{N} / 0 \leq x \leq 1024\}$ , que representa o **conjunto de entrada**

Para concluir a nossa máquina temos que definir suas  $f_s$  e  $f_o$ . Como o valor de entrada vai de 0 a 1024, temos q determina uma **constante de sensibilidade** que servirá como indicador de luminosidade, ou seja, se o **valor de entrada for menor que ela**, então quer dizer que **está de dia**, do mesmo modo ao contrário, se o **valor de entrada**

**for maior que ela, então estamos de noite**, daí temos a  $f_s$ . Pensando na  $f_o$ , como quando está de **noite**, o poste deve **acender**, então traduzindo para máquinas de estado finito, temos que **a saída do estado NOITE é 1**, o mesmo vale ao contrário, **quando o estado é DIA, a saída é 0 para a luz apagar**.

Para melhor compreensão, veja a tabela de estado abaixo:

Estado atual	Próximo estado		Saída
	Entrada atual		
	$> C.de Sensibilidade$	$< C.de Sensibilidade$	
NOITE	NOITE	DIA	1
DIA	NOITE	DIA	0

*Tabela 2 - Tabela de estado do poste*

### Implementação

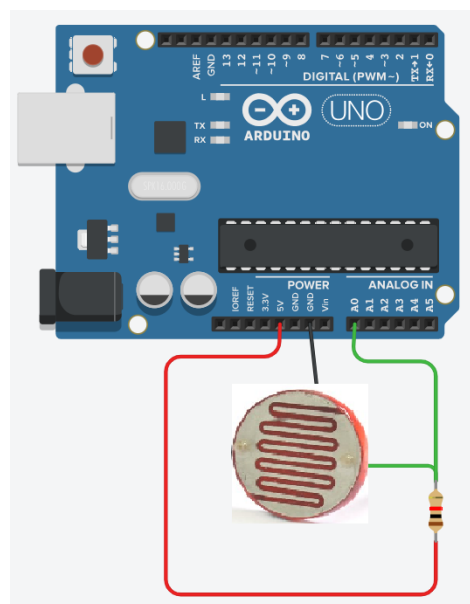
Para implementar nossa máquina, utilizaremos um *Arduino UNO* que é uma plataforma de prototipagem eletrônica de hardware, pois tem fácil implementação e boa didática para esse exemplo.

### Eletrônica

Listagem de componentes:

- 1 - Arduino Uno
- 1 - LDR de 20K $\Omega$
- 1 - Resistor de 1K $\Omega$
- Pedacos de fio de cobre (jumpers)

Sendo que cada componente está conectado da seguinte forma:



*Figura 1 - Esquematização eletrônica na plataforma do Arduino*

## Programação

O seguinte código na linguagem **C** utilizando a **IDE do Arduino** foi compilado no Arduino.

```
1 typedef enum { NOITE,DIA } S;
2
3 S estadoAtual;
4 int entradaAtual;
5 int saida;
6
7 int cSensibilidade = 750;
8
9 void setup() {
10     // Inicia serial à 9600bps
11     Serial.begin(9600);
12
13     //Inicia estado atual como NOITE (S0)
14     estadoAtual = NOITE;
15
16     pinMode(LED_BUILTIN, OUTPUT);
17 }
18
19 void loop() {
20     //Recebe entradaAtual da leitura analogica do sensor LDR
21     entradaAtual = analogRead(A0);
22
23     //atualizamos o novo estado com fs
24     estadoAtual = fs(estadoAtual, entradaAtual);
25
26     //pegamos a saida com fo
27     saida = fo(estadoAtual);
28
29     //enviamos a saida na serial
30     Serial.print(saida);
31     digitalWrite(LED_BUILTIN, saida);
32
33     delay(10); // delay pra estabilidade
34 }
35
36 S fs(S estado, int entrada) {
37     if(entrada<cSensibilidade){
38         return DIA;
39     } else {
40         return NOITE;
41     }
42 }
43
44 int fo(S estado){
45     if(estado==NOITE) {
46         return 1;
47     }
48
49     if(estado==DIA) {
50         return 0;
51     }
52 }
```



## Conclusão

Máquinas de estado finito operam de forma síncrona e determinística com capacidade de memória limitada. Porém esse modelo consegue resolver muitos dos problemas na engenharia, pois ele processa entradas junto com o estado e produz uma saída, além de que tendo uma máquina especificada com entradas é possível prever suas saídas, o que auxilia na construção de soluções como visto no exemplo do poste.

## Bibliografia

Bonato, V. **Elementos de Lógica Digital II**. Disponível em:

[http://wiki.icmc.usp.br/images/7/7d/Aula\\_5\\_-\\_StateMachine.pdf/](http://wiki.icmc.usp.br/images/7/7d/Aula_5_-_StateMachine.pdf/) . Acesso: 26 de nov. 2018.

GERSTING, J.L. **Fundamentos Matemáticos para a Ciência da Computação**. 3 ed. LTC, 2010.

**Máquinas de Estado**. Disponível em:

<https://fenix.tecnico.ulisboa.pt/downloadFile/3779571788876/> . Acesso: 25 de nov. 2018.

Vieira, N.J. **Introdução aos fundamentos da computação**, Editora Thomson. Cap. 2. Disponível em:

<https://homepages.dcc.ufmg.br/~nvieira/cursos/tl/a17s2/livro/cap2.pdf> . Acesso: 25 de nov. 2018.