

# Trabalho de Linguagens Formais e Autômatos

Ariel Nogueira Kovaljski  
<arielnogueirak@gmail.com>

*Computer Engineering Course,  
Instituto Politécnico (IPRJ) — Rio de Janeiro State University,  
Rua Bonfim 25, Nova Friburgo, RJ 28625-570, Brazil*

<dia-atual> de maio de 2021

---

## Abstract

In this assignment we build and analyze the behavior and output of a Finite State Machine (FSM) and a Turing Machine (TM) for a given set of inputs.

*Keywords:* finite state machine, turing machine, finite automata

## Resumo

Neste trabalho nós construímos e analisamos o comportamento e a saída de uma Máquina de Estado Finito (MEF) e uma Máquina de Turing (MT) para um dado conjunto de entradas.

*Palavras-chave:* máquina de estado finito, máquina de turing, autômatos finitos

## 1. Introdução

A teoria de autômatos trata do estudo de máquinas abstratas que seguem instruções pré-determinadas automaticamente.

A partir da hierarquia de Chomsky de gramáticas formais, é possível definir autômatos que são capazes de reconhecer linguagens formais pertencentes a distintas classes de gramática. Sendo assim, para uma linguagem formal possivelmente infinita, define-se um autômato como uma representação finita desta linguagem.

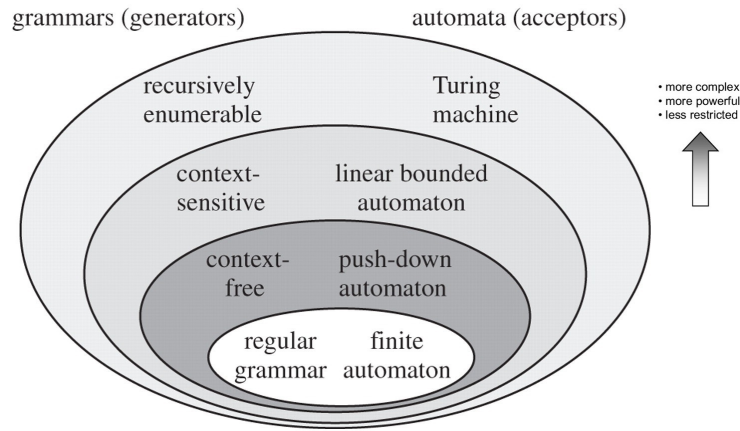


Figura 1: Hierarquia de Chomsky

Dentre os tipos de autômatos existentes, neste trabalho iremos abordar a construção e o funcionamento das *máquinas de estado finito* e das *máquinas de Turing*. Estes são responsáveis pelo reconhecimento das *gramáticas regulares* e *recursivamente enumeráveis*, respectivamente.

## 2. Desenvolvimento

Iremos nos aprofundar no desenvolvimento destes autômatos. <expandir>

### 2.1. Máquina de Estado Finito

<falar sobre memória>

Uma máquina de estado finito (MEF) pode ser considerada como um modelo simplificado do funcionamento de um computador. Esta pode ser definida como uma quintupla ordenada  $M = (S, I, O, f_S, f_O)$  onde:

- $S$  é o conjunto finito de estados;
- $I$  é o conjunto finito de símbolos de entrada (alfabeto de entrada);
- $O$  é o conjunto finito de símbolos de saída (alfabeto de saída);
- $f_S : S \times I \rightarrow S$  é uma função que retorna o próximo estado  $s_{t_{k+1}} \in S$  dado o estado anterior  $s_{t_k} \in S$  e um símbolo de entrada  $i_{t_k} \in I$ ;
- $f_O : S \rightarrow O$  é a função *output*, que retorna o símbolo de saída  $o_{t_k} \in O$  do estado atual  $s_{t_k} \in S$ .

As operações da MEF são sincronizadas por pulsos discretos de um relógio (*clock*) representados por  $t_k$ , onde  $k \in \mathbb{N}_0$  representa o ciclo de *clock* atual. Partindo de um estado inicial  $s_0 = s_{t_0}$ , após um pulso de *clock*, a função  $f_S$  retornará um novo estado  $s_{t_1}$  dado a entrada anterior  $i_{t_0}$  e o estado anterior  $s_0$ . Generalizando para qualquer pulso de *clock*, é possível afirmar que a MEF possui um comportamento determinístico, ou seja, o novo estado sempre dependerá do estado e entrada anteriores. Para cada um dos estados, há um símbolo de saída, que é retornado pela função *output*  $f_O$ .

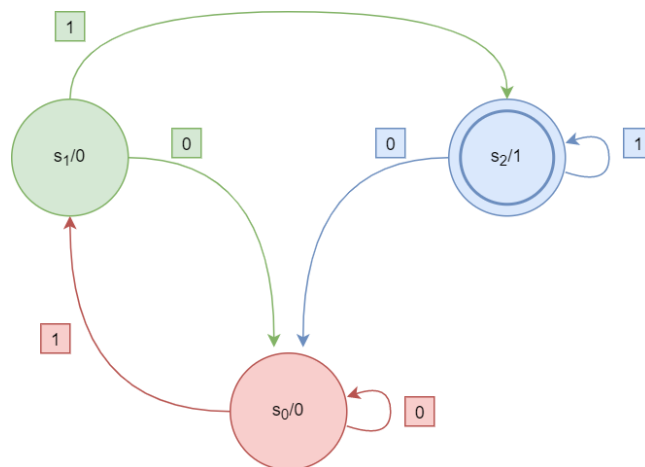


Figura 2: Exemplo de máquina de estado finito

Tomando como exemplo a MEF acima, começamos no estado  $s_0$  e temos como saída o símbolo **0**. Aqui, há duas possibilidades: enquanto a entrada for **0**, permanecemos neste estado; caso a entrada seja **1**, seguimos para o estado  $s_1$ . Ao seguirmos para o estado  $s_1$  temos como saída o símbolo **0**. Novamente, há duas possíveis entradas: caso a entrada seja **0**, retornamos ao estado  $s_0$ ; caso a entrada seja **1**, seguimos para o estado  $s_2$ . Por fim, ao seguirmos para o estado  $s_2$  temos como saída o símbolo **1**. As alternativas são: caso a entrada seja **0**, retornarmos ao estado  $s_0$ ; enquanto a entrada seja **1**, permanecemos neste estado.

Esta forma um tanto verbosa de se descrever uma MEF pode ser colocada em uma tabela, com o estado atual, possíveis entradas e saídas como colunas da mesma.

Estado Atual	Próximo Estado		Saída
	<b>0</b>	<b>1</b>	
$s_0$	$s_0$	$s_1$	<b>0</b>
$s_1$	$s_0$	$s_2$	<b>0</b>
$s_2$	$s_0$	$s_2$	<b>1</b>

Nota-se que dentre todas as possíveis entradas, esta MEF só aceitará, isto é, terminará no estado final  $s_2$ , para *strings* de entrada que terminem com dois ou mais **1** (**11**, **111**, **1111**, ...).

## 2.2. Máquina de Turing

<expandir>

## 3. Conclusão

Aqui está a conclusão.

#### 4. Referências Bibliográficas

**Figura 1:** Fitch, Tecumseh. 2014. *“Toward a computational framework for cognitive biology: Unifying approaches from cognitive neuroscience and comparative cognition”*