

## 2. Linguagens Regulares

Chomsky define uma Hierarquia para as Linguagens Formais. As *linguagens regulares* ou *tipo 3*, nessa hierarquia, identificam a classe das linguagens mais simples e são abordadas através de 3 formalismos:

- Operacional ou Reconhecedor
- Axiomático ou Gerador
- Denotacional

### AUTÔMATO FINITO DETERMINÍSTICO

Um Sistema de Estados Finitos é um modelo matemático de sistema com entradas e saídas discretas. Esse sistema pode assumir um número finito e conhecido de estados. Cada estado mantém somente as informações passadas necessárias para determinar a ação para a próxima entrada.

#### **Definição 1: Autômato Finito Determinístico (AFD)**

Um **Autômato Finito Determinístico** é uma máquina composta de 3 partes:

**a) Fita:** Dispositivo de entrada que contém a informação a ser processada. A fita é finita à esquerda e infinita à direita. Cada célula da fita armazena um símbolo pertencente a um alfabeto de entrada. Não é possível gravar sobre a fita. Inicialmente, a palavra de entrada ocupa toda fita.

**b) Unidade de Controle:** Reflete o estado corrente da máquina (há um número finito e pré-definido de estados.). Possui uma unidade de leitura (cabeça da fita) que acessa uma célula da fita de cada vez e movimenta-se exclusivamente para a direita (uma célula). É uma unidade de leitura. Inicia sempre à esquerda da fita.

**c) Programa ou Função de Transição:** comanda as leituras e define o estado da máquina. É uma função parcial (cada elemento do domínio está relacionado com no máximo um elemento do contradomínio - estado e símbolo). Determina o novo estado do autômato.

Um **Autômato finito** não possui memória de trabalho; para armazenar informações, usa o conceito de estados.

**Formalmente**, um Autômato Finito Determinístico (AFD) ou Autômato Finito é uma 5-upla:

$$M = (\Sigma, Q, \delta, q_0, F)$$

onde:

$\Sigma$  é o alfabeto de entrada

$Q$  é o conjunto de estados possíveis do autômato. É um conjunto finito.

$\delta$  é a função programa ou função de transição.  $\delta: Q \times \Sigma \rightarrow Q$  é uma função parcial

$q_0$  é o estado inicial.  $q_0 \in Q$

$F$  é o conjunto de estados finais.

A Função Programa pode ser representada por um grafo finito direto:



Um autômato finito sempre para ao processar qualquer palavra. A parada de um autômato pode ser de duas maneiras: aceitando ou rejeitando a palavra w.

**Condições de Parada:**

- a) Processa o último símbolo da fita e assume um estado final (Aceita)
- b) Processa o último símbolo da fita e assume um estado não final (Rejeita)
- c) Função programa indefinida para o argumento (Rejeita)

**Definição 2: Função Programa Estendida**

É a Função Programa acrescida de argumento: estado e palavra. Seja  $M = (\Sigma, Q, \delta, q_0, F)$  um AFD,  $\underline{\delta}: Q \times \Sigma^* \rightarrow Q$  é a Função Programa Estendida e definida indutivamente como segue:

$$\underline{\delta}(q, \varepsilon) = q$$

$$\underline{\delta}(q, aw) = \underline{\delta}(\delta(q, a), w)$$

**Exemplo:**  $w = abba$

**Notação:** Seja o AFD M,

ACEITA(M) ou L(M): conjunto de todas as palavras  $\in \Sigma^*$  aceitas por M

REJEITA(M): conjunto de todas as palavras  $\in \Sigma^*$  rejeitadas por M

- $ACEITA(M) \cap REJEITA(M) =$
- $ACEITA(M) \cup REJEITA(M) =$
- $ACEITA(M)' =$
- $REJEITA(M)' =$

**Definição 3: AFD Equivalentes**

AFD M1 e M2 são equivalentes se e somente se  $ACEITA(M1) = ACEITA(M2)$

**Definição 4: Linguagem Regular ou Tipo 3**

É uma linguagem aceita por um AFD.

**Exercícios – Considere  $\Sigma = \{a, b\}$ . Desenvolva os AFDs para as seguintes linguagens:**

- a)  $L_1 = \{w \text{ possui prefixo } a\}$
- b)  $L_2 = \{w \text{ possui sufixo } bbb\}$
- c)  $L_3 = \{w \text{ possui subpalavra } ab\}$
- d)  $L_4 = \{w \text{ possui número par de } b's\}$
- e)  $L_5 = \{w \text{ possui } aa \text{ ou } bb \text{ como subpalavra}\}$