

# Trabalho 1: Autômatos finitos

Laura Cruz Quispe  
Jarlinton Moreno Zea  
SCC5832: Teoria da Computação  
Prof. Diego Raphael Amancio

September 25, 2017

## 1 Definição

Um autômato finito (AF) é uma quintupla  $M = (Q, \Sigma, \delta, q_0, F)$  onde:

- $Q$  é um conjunto finito de estados.
- $\Sigma$  é um alfabeto finito chamado **alfabeto de entrada**.
- $\delta$  é a função de transição de estados, com um mapeamento:

$$Q \times \Sigma \Rightarrow Q$$

- $q_0 \in Q$  é o estado inicial.
- $F$  é o conjunto de estados finais  $F \subseteq Q$

O Autômato não Determinístico aceita cadeia vazia  $\lambda$ , e tem um conjunto de possíveis estados para cada transição.

### 1.1 Interpretação de transições ( $\delta$ )

Na função de transições ela aceita o formato determinista:

$$\delta(q_i, a) \Rightarrow q_j$$

Onde  $q_i, q_j \in Q, a \in \Sigma$

neste caso particular o autômato é determinista.

## 2 Implementação

Após de apresentar uma definição de autômato finito, descrevemos o processo de implementação. Escolhimos a linguagem de programação PYTHON, para a implementação do autômato finito, a seguir descrevemos os passos na execução do programa.

Na leitura o nome do arquivo de entrada, é passado o nome do mesmo pelo formato de entrada da consola, no formato; "python app.py nome\_arquivo.txt".

Depois de aceitar o arquivo de entrada, chamamos o método "process\_input\_file", que recebe o arquivo de entrada, este método define a estrutura do autômato como resultado é definido cada variável do autômato ( $Q, \Sigma, \delta, q_0, F$ )

Finalmente após a definição dos elementos do autômato é chamado o método "evaluate\_automatus(AF, cadeia)" o método é apresentado no algoritmo 1 para autômato determinístico e algoritmo 2 para autômato não determinístico. Cada método recebe o conjunto de transições os estados iniciais, de aceitação e a cadeia do caso de teste, ele retorna a cadeia recebida e a mensagem "aceita" ou "rejeita".

**Data:**  $M = (Q, \Sigma, q_0, F)$ ,  $s$ : cadeia de entrada  
**Result:** Aceita/Rejeita cadeia de entrada  
 Inicialização;  
 DEFINE estado atual  $p$  como  $q_0$ ;  
**for** cada símbolo  $a$  na cadeia **do**  
     **if** há uma regra de transições em  $\delta$  tal que  $\delta(p, a) = (q)$  **then**  
         | UPDATE atual de estado  $p$  com o novo estado  $q$   
     **end**  
**end**  
**if** atual estado  $p$  está em estados finais ( $F$ ) **then**  
     | Aceita cadeia.  
**else**  
     | Rejeita cadeia.  
**end**

**Algorithm 1:** Algoritmo Proposto: Automata Finito Determinístico

**Data:**  $M = (Q, \Sigma, q_0, F)$ ,  $s$ : cadeia de entrada  
**Result:** Aceita/Rejeita cadeia de entrada  
 Inicialização;  
 DEFINE um conjunto de estado atuais  $A$  como  $q_0$ ;  
**for** cada símbolo  $a$  na cadeia **do**  
     **for** cada  $p_i$  no conjunto  $A$  **do**  
         **if** há uma regra de transições em  $\delta$  tal que  $\delta(p_i, a) = (q)$  **then**  
             | UPDATE estados atuais  $A$  com o novo estado  $q$   
         **else**  
             | Se não ha uma transição para o simbolo  $a$ , busca transições com cadeia =  $\lambda$ .  
             | UPDATE estados atuais  $A$  com o novo estado  $q$ ;  
             | Volta a leer o mesmo simbolo  $a$   
         **end**  
     **end**  
**end**  
**if** atual estado  $p$  está em estados finais ( $F$ ) **then**  
     | Aceita cadeia.  
**else**  
     | Rejeita cadeia.  
**end**

**Algorithm 2:** Algoritmo Proposto: Automata Finito não Determinístico

## 2.1 Provas

O linguagem utilizada que prova o Autômato Determinístico é:  $(a+b)a^*bb^*(a(a+b)a^*bb^*)^*$  descrito no arquivo 'afd\_1.txt', e o linguagem regular utilizada que prova o Autômato Não Determinístico é  $a^*|b^*$  descrito no arquivo 'afnd\_1.txt'. Para testar o código AFD: 'python automata\_finito.py' e AFND: 'python automata\_finito\_non.py', ambos carregam por defeto os arquivos das provas.

## 3 Conclusões

Nossa proposta esta baseado num algoritmo iterativo, tem complexidade  $O = (n)$  para Autômato Determinístico onde  $n$  é o comprimento da cadeia, e o acesso ao mapa de transições é constante já que fazemos uso de um mapa. Por otro lado o Autômato não Determinístico tem complexidade de  $O = (n * m)$  devido a que tem um conjunto de estados posiveis em cada transição como máximo comprimento  $m$ ,  $m$  é o número de estados totales.