



## Aula Aplicada – Fase 2

Esta atividade tem como objetivo enfatizar os conceitos sobre autômatos finitos determinísticos e não determinísticos estudados até a presente aula. As atividades serão realizadas em forma de competição, em que, o participante 🏆 que finalizar todos os exercícios ou ao término do tempo, o maior número, será premiado com brindes 🎁. Podendo haver brindes até a terceira colocação 🥉! Não esqueçam que além da competição, essa é uma atividade avaliativa e que deve se atentar ao prazo e forma de entrega.



Prof. Benevid Felix Silva

### 1. Lista de exercícios sobre AFD e AFND.

Dado as seguintes linguagens sobre os alfabetos  $\Sigma_1 = \{0, 1\}$  ou  $\Sigma_2 = \{a, b\}$ :

1.  $L_1 = \{w \mid w \in \Sigma^* \text{ e } w \text{ começa por } 11 \text{ e termina por } 00\}$
2.  $L_2 = \{10w \mid w \in \Sigma^*\}$
3.  $L_3 = \{x01y \mid x, y \in \{0, 1\}^*\}$
4.  $L_4 = \{b(ab)^n b \mid n \geq 0\}$
5.  $L_5 = \{10^n 10 \mid n \geq 0\}$
6.  $L_6 = \{0^m 1^n \mid m+n \text{ é par}\}$
7.  $L_7 = \{01^m 10(01)^n \mid n \geq 0\}$
8.  $L_8 = \{ab(ab)^n ba \mid n \geq 0\}$
9.  $L_9 = \{ba^n ba \mid n \geq 0\}$
10.  $L_{10} = \{ab^m a^n \mid m, n \geq 0\}$
11.  $\{w \mid w \text{ possui } aaa \text{ como subpalavra}\}$
12.  $\{w \mid \text{o sufixo de } w \text{ é } aa\}$
13.  $\{w \mid w \text{ possui número ímpar de } a \text{ e } b\}$
14.  $\{w \mid w \text{ possui número par de } a \text{ e ímpar de } b \text{ ou } w \text{ possui número par de } b \text{ e ímpar de } a\}$
15.  $\{w \mid \text{o quinto símbolo da direita para a esquerda de } w \text{ é } a\}$

### 2. Atividades

**Passo 1 - Faça o grafo dos autômatos acima. Caso necessario, converta manualmente os AFNDs para AFDs.**

Obs.: Entregar os grafos e conversões por escrito.

## Passo 2 - Construa um algoritmo para cada AFD do exercício 1 anterior em Linguagem C.

Obs.: Mostre que os algoritmos ACEITAM ou REJEITAM um dado conjunto de palavras sobre os  $\Sigma$ .

## Passo 3 - Trabalho da disciplina - Individual

Todos devem entregar até 15/09 os exercícios acima como requisito par obtenção da nota referente ao trabalho principal da disciplina.

Valor 10 pontos.

As atividades devem ser feitas em manuscrito. Portanto, se deve entregar uma arquivo contendo a cópia das folhas escaneadas.

Enviar para o email: benevid@unemat.br

Título do email: Trabalho LFA2019-2

## Exemplos na Linguagem C

### Algoritmo 1 - Palavras da Linguagem $L = (ba)^n, n \geq 1$

```
1
2  #include <stdio.h>
3  #include <string.h>
4
5  /* Função Principal*/
6  int TAM=0;
7
8  void q0(int pos, char w[TAM]);
9  void q1(int pos, char w[TAM]);
10 void qFim(char w[TAM]);
11 void qErro(char w[TAM]);
12
13
14
15 int main(){
16
17     int size = 0;
18     char w[21];
19
20     printf("\n Informe a palavra:");
21     scanf("%s",w);
22
```

```

23     //Calcular o tamanho da palavra
24     while(w[size] != '\0'){
25         size++;
26     }
27     TAM=size;
28     // palavra com tamanho 0
29     if(TAM <= 0){
30         qErro(w);
31     }
32
33     int pos = 0;
34
35     printf("\n TAM: %d", TAM);
36     printf("\n Size: %d", size);
37     printf("\n Palavra: %s", w);
38
39
40     q0(pos,w);
41
42     return 0;
43 }
44
45
46 void q0(int pos, char w[TAM]){
47     if(pos<TAM){
48         if(w[pos] == 'b'){
49             printf("\n q0 lendo %c -> q1", w[pos]);
50             q1(++pos,w);
51         }else{
52             qErro(w);
53         }
54     }else if(pos == TAM){
55         qFim(w);
56     }
57
58 }
59
60 void q1(int pos, char w[TAM]){
61     if(pos < TAM){
62         if(w[pos] == 'a'){
63             printf("\n q1 lendo %c -> q0", w[pos]);
64             q0(++pos,w);
65         }else{
66             qErro(w);
67         }
68     }else if(pos == TAM){
69         qErro(w);
70     }
71 }
72
73 void qErro(char w[TAM]){
74     printf("\n Palavra %s rejeitada pelo automato!\n",w);

```

```

75     }
76
77     void qFim(char w[TAM]){
78         printf("\n Palavra %s Aceita pelo automato!\n",w);
79     }

```

## Algoritmo de Exemplo2

```

1
2     /*Ler as palavras reservadas char, for, float */
3
4
5     #include <stdio.h>
6     #include <string.h>
7     #define TAM 100
8
9     void qInicio ( char palavra[TAM],int size);
10    void q0( int contador, char palavra[TAM],int size);
11    void q1( int contador, char palavra[TAM],int size);
12    void q2( int contador, char palavra[TAM],int size);
13    void q3( int contador, char palavra[TAM],int size);
14    void q4( int contador, char palavra[TAM],int size);
15    void q5( int contador, char palavra[TAM],int size);
16    void q6( int contador, char palavra[TAM],int size);
17    void q7( int contador, char palavra[TAM],int size);
18    void q8( int contador, char palavra[TAM],int size);
19    void q9( int contador, char palavra[TAM],int size);
20    void qFim();
21    void qErro();
22
23    int main(){
24        int size = 0;
25        char palavra[TAM]; //casos de comprovação:
26        fflush(stdin);
27        gets(palavra);
28
29        while(palavra[size] != '\0')
30        {
31            size++;
32        }
33
34        qInicio(palavra,size);
35
36        return 0;
37    }
38
39    void qInicio ( char palavra[TAM],int size ){
40        int contador = 0;
41        q0( contador, palavra,size );
42    }

```

```

43
44 void q0( int contador, char palavra[TAM] ,int size){
45     if ( contador < TAM ){
46         if ( palavra[contador] == 'f' ){
47             q1( ++contador, palavra,size );
48
49             }else if (palavra[contador] == 'c'){
50                 q8( ++contador, palavra ,size);
51
52             }else{
53                 qErro();
54             }
55
56     }
57 }
58
59 void q1( int contador, char palavra[TAM] ,int size){
60     if ( contador < TAM ){
61         if ( palavra[contador] == 'l' ){
62             q2( ++contador, palavra ,size);
63
64             }else if (palavra[contador] == 'o'){
65                 q3( ++contador, palavra,size );
66
67             }else{
68                 qErro();
69             }
70
71     }
72 }
73
74 void q2( int contador, char palavra[TAM] ,int size){
75     if ( palavra[contador] == 'o' ) {
76         q5 ( ++contador, palavra,size );
77
78     } else {
79         qErro();
80     }
81 }
82
83 void q3( int contador, char palavra[TAM] ,int size){
84     if ( palavra[contador] == 'r' ) {
85         q4 ( ++contador, palavra ,size);
86
87     } else {
88         qErro();
89     }
90 }
91
92 void q4( int contador, char palavra[TAM] ,int size){
93     if(contador == size)
94     {

```

```
95     qFim();
96 }
97
98     else
99     {
100         qErro();
101     }
102
103 }
104
105 void q5( int contador, char palavra[TAM] ,int size){
106     if ( palavra[contador] == 'a' ) {
107         q6 ( ++contador, palavra ,size);
108
109     } else {
110         qErro();
111     }
112 }
113
114 void q6( int contador, char palavra[TAM] ,int size){
115     if ( palavra[contador] == 't' ) {
116         q7 ( ++contador, palavra ,size);
117
118     } else {
119         qErro();
120     }
121 }
122
123 void q7( int contador, char palavra[TAM],int size ){
124     if(contador == size)
125     {
126         qFim();
127     }
128
129     else
130     {
131         qErro();
132     }
133 }
134
135 void q8( int contador, char palavra[TAM] ,int size){
136     if ( contador < TAM ){
137         if ( palavra[contador] == 'h' ){
138             q9( ++contador, palavra ,size);
139
140         }else{
141             qErro();
142         }
143
144     }
145 }
146
```

```
147 void q9( int contador, char palavra[TAM] ,int size){
148     if ( palavra[contador] == 'a' ) {
149         q3 ( ++contador, palavra ,size);
150
151     } else {
152         qErro();
153     }
154 }
155
156
157 void qErro(){
158     printf("Palavra rejeitada pelo automata!\n");
159 }
160
161 void qFim(){
162     printf("Palavra Aceita pelo automata!\n");
163 }
```