# TEAT.

# INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE

## REPORTE DE ACTIVIDAD

# **CARRERA:**

INGENIERÍA EN SISTEMASCOMPUTACIONALES

# **ASIGNATURA:**

LENGUAJES Y AUTOMATAS I

# **NOMBRE DE LAACTIVIDAD:**

PROGRAMA: AUTOMATA FINITO DETERMINISTA

# NOMBRE DEL ALUMNO:

**GONZALO MARTINEZ SILVERIO** 

## DOCENTE:

DR. TANIA TURRUBIATES LÓPEZ

# **PERIODO ESCOLAR:**

FEB 2023 – JUN 2023

# **SEMESTRE:**

6°

**GRUPO:** 

6S1A

# ATEA

# INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE

### REPORTE DE ACTIVIDAD

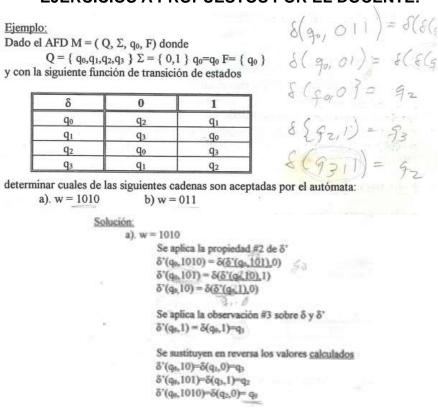
# INTRODUCCIÓN:

Un autómata finito determinista (abreviado AFD) es un autómata finito que además es un sistema determinista; es decir, para cada estado en que se encuentre el autómata, y con cualquier símbolo del alfabeto leído, existe siempre no más de una transición posible desde ese estado y con ese símbolo

Además de que un Autómata Finito Determinista consta de:

- 1. Un conjunto finito de estados, a menudo designado como Q.
- 2. Un conjunto finito de símbolos de entrada, a menudo designado como ∑ (sigma).
- 3. Una función de transición que toma como argumentos un estado y un símbolo de entrada y devuelve un estado. La función de transición se designa habitualmente como  $\delta$  o  $\Delta$  (delta).
- 4. Un estado inicial, uno de los estados de Q.
- 5. Un conjunto de estados finales o de aceptación F. El conjunto F es un subconjunto de Q.

# **EJERCICIOS A PROPUESTOS POR EL DOCENTE:**



# ALL THE REAL PROPERTY OF THE P

# INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE

### REPORTE DE ACTIVIDAD

```
Como \delta'(q_0,w)=q_0 es un estado final, es aceptada por el AFD. !!

b). w=011
Se aplica la propiedad #2 de \delta'
\delta'(q_0,011)=\delta(\underline{\delta'(q_0,01)},1)
\delta'(q_0,01)=\delta(\underline{\delta'(q_0,0)},1)
Se aplica la observación #3 sobre \delta y \delta'
\delta'(q_0,0)=\delta(q_0,0)=q_2
Se sustituyen en reversa los valores <u>calculados</u>
\delta'(q_0,01)=\delta(q_2,1)=q_3
\delta'(q_0,011)=\delta(q_2,1)=q_2
Como \delta'(q_0,w)=q_2 NO es un estado final, No es aceptada por el AFD. !!
```

```
Ejercicio:

1. Para el AFD anterior, validar las siguientes cadenas: w<sub>1</sub> = 11010, w<sub>2</sub>=1111, w<sub>3</sub>=0001, w<sub>4</sub>=101010, w<sub>5</sub>=001100, w<sub>6</sub>=101
```

1. Comenzamos a codificar nuestro AFD, en mi caso lo realizare en java ya que cuento con mas experiencia en este lenguaje, creamos la clase llamada AFD\_GMS e importamos algunas librerías que utilizaremos en nuestro proyecto, creamos variables de la tabla de nuestro autómata, para no hacer muy largo esto, me di la tarea de escribir algunos comentarios el código el cual nos explicara a detalle la función de nuestro programa.

```
//Instituto Tecnologico Superior De Alamo Temapache
2
      //Lenguaje Y Automatas 1
3
      //Alumno: Gonzalo Martinez Silverio
      //Numero de control: 202Z0029
      //Carrera: Ingenieria en sistemas computacionales
6
      //Grupo: 6S1A
                         Semestre: 6°
      //Docente: Dr. Tania Turrubiates Lopez
  import java.util.Arrays;
8
9
      import java.util.HashMap;
10
      import java.io.*;
11
      import java.util.Scanner;
12
13
      public class AFD GMS{
14
15
          public int numColum;
16
          private String ini, edoActual;
17
          private String[] fin,symbols;
          private HashMap<String,String[]> table;
```

#### REPORTE DE ACTIVIDAD 19 public AFD GMS(String name) { 20 table = new HashMap<String,String[]>(); 22 try{ 23 24 File inputFile = new File(name); 25 26 // Se crea un lector del archivo y un buffer 27 // que contendrá el texto del archivo FileReader fr = new FileReader(inputFile); 29 BufferedReader br = new BufferedReader(fr); 30

```
// Se lee linea por linea el archivo
32
                  String linea;
33
                  String[] aux;
34
35
                  linea = br.readLine();
                  while(linea != null){
36
37
                       // Se divide la cadena y se guarda en un array
38
                      String dato[] = linea.split(",");
39
                       if(i == 0){
40
                           if(linea.charAt(0) == '@'){
41
                              this.numColum = dato.length;
42
                               symbols = Arrays.copyOfRange(dato, 1, this.numColum);
43
44
                               System.out.println("Archivo no valido");
45
                               System.exit(0);
46
```

```
47
                       }else if(dato[0].equals("Inicio")){
48
                           ini = dato[1];
49
                           edoActual = dato[1];
50
                       }else if(dato[0].equals("Final")){
51
                           fin = Arrays.copyOfRange(dato, 1, dato.length);
52
53
                           aux = Arrays.copyOfRange(dato, 1, this.numColum);
54
                           table.put(dato[0], aux);
55
56
                       linea = br.readLine();
57
                       i++;
58
59
60
                   table.remove(""); // Elimino espacio en blanco
61
```



```
REPORTE DE ACTIVIDAD
                   // Se cierra el lector del archivo
62
63
                  fr.close():
64
65
              }catch(FileNotFoundException e) {
66
                  System.err.println("ArchivoText: " + e);
67
                  System.exit(0);
68
              }catch(IOException e) {
69
                  System.err.println("ArchivoText: " + e);
70
                  System.exit(0);
71
72
73
74
   public void showData() {//Este método muestra la tabla de transiciones
75
              String[] aux;
76
              System.out.print("Alfabeto: ");//Muestra el lenguaje que acepta
77
              for(int i=0;i<this.numColum-1;i++){</pre>
78
                  System.out.print(symbols[i]+" ");
79
80
              System.out.println();
81
              for (String i : table.keySet()) {
82
                  System.out.print("Estado: " + i + "; Transicion: ");
83
84
                  aux = table.get(i);
85
                   for(int j=0; j<this.numColum-1;j++){
86
87
                      System.out.print(aux[j]+",");
88
89
                  System.out.println();
٩n
```

```
92
               System.out.println("Estado inicial: "+ini);
               System.out.print("Estados finales: ");
               for (int i=0; i<this.fin.length; i++) {</pre>
95
                    System.out.print(fin[i]+",");
96
97
               System.out.println();
98
100
101
102
    豆
103
            public void transicion(String[] edos, int index){
104
               this.edoActual = edos[index];
105
106
107
           //Se va a analizar la cadena para ver si es aceptada o no por el automata
            public void analizarCadena() {
108
    阜
109
               String cadena;
110
               Scanner cad = new Scanner(System.in);
111
               System.out.print("Ingrese la cadena: ");
112
               cadena = cad.nextLine();
113
114
               for(int i=0; i<cadena.length(); i++){</pre>
115
116
                   int indAux = -1;
117
118
                    for(int j=0; j<numColum-1; j++){</pre>
119
                         if (cadena.charAt(i) == this.symbols[j].charAt(0)){
120
                            indAux = j;
```



### REPORTE DE ACTIVIDAD

```
124
125
                   if(indAux != -1){
126
          String[] auxEdo = this.table.get(this.edoActual);
127
        //Se van a realizar las transiciones con el indice y el estado auxiliar
128
                   this.transicion(auxEdo,indAux);
129
130
132
                       System.out.println("No pertenece al alfabeto");
133
                       System.exit(0);
134
135
136
        //Si el estado actual de la cadena es igual al estado final, la cadena es compa<mark>t</mark>ible
137
               for (int i=0; i<fin.length; i++) {</pre>
139
                   if(this.edoActual.equals(this.fin[i])){
140
                       System.out.println("Cadena es aceptada");
141
142
                       System.out.println("Estado final: "+this.fin[i]);
143
                        System.exit(0);
144
145
        /Si el estado actual de la cadena no es igual al estado final, la cadena no es compatible
146
147
               System.out.println("Cadena no es aceptada");
148
```

```
public static void main(String[] args) { //Se hace uso de los metodos

AFD_GMS afd = new AFD_GMS(args[0]);

afd.showData();

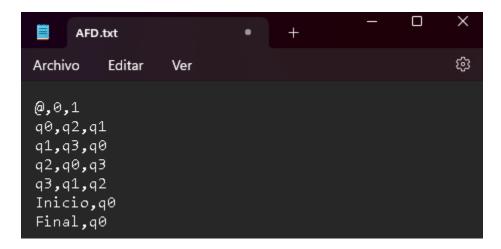
afd.analizarCadena();

}

155 }

156
```

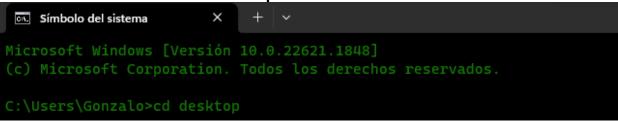
2. Ya en este punto, al haber concluido la codificación del programa vamos a crear un archivo de texto llamado AFD el cual contendrá los datos de la tabla de nuestro Autómata Finito Determinista del ejercicio propuesto por el docente.





### REPORTE DE ACTIVIDAD

3. Ahora es momento de ejecutar nuestro programa, tendremos que abrir nuestra terminal y vamos a escribir la dirección donde esta nuestro programa, en mi caso esta en el escritorio: **cd desktop** 



 Vamos a copilar nuestro programa escribiendo el comando: javac AFD\_GMS.java

C:\Users\Gonzalo\Desktop>javac AFD\_GMS.java

5. En este punto ejecutaremos nuestro código con el comando: java AFD\_GMS

C:\Users\Gonzalo\Desktop>java AFD\_GMS
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
 at AFD\_GMS.main(AFD\_GMS.java:158)

- 6. Una vez terminado de ejecutar el programa vamos a abrir el archivo de texto creado en el punto 2 con el siguiente comando: java AFD\_GMS AFD.txt
- 7. Como verán nos pide introducir la cadena, vamos a introducir las cadenas del ejercicio propuesto por el docente las cuales son las siguientes:

No.	Cadenas:
1	1010
2	011
3	11010
4	1111
5	0001
6	101010
7	001100
8	101

### REPORTE DE ACTIVIDAD

8. Cadena: 1010 → Resultado: cadena es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 1010
Cadena es aceptada
Estado final: q0
```

9. Cadena: 011 → Resultado: cadena no es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 011
Cadena no es aceptada
```

10. Cadena: 11010 → Resultado: cadena no es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 11010
Cadena no es aceptada
```

### REPORTE DE ACTIVIDAD

11. Cadena: 1111 → Resultado: cadena es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 1111
Cadena es aceptada
Estado final: q0
```

12. Cadena: 0001 → Resultado: cadena no es aceptada.

```
C:\Usemparanzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 0001
Cadena no es aceptada
```

13. Cadena: 101010 → Resultado: cadena no es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 101010
Cadena no es aceptada
```

### REPORTE DE ACTIVIDAD

14. Cadena: 001100 → Resultado: cadena es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 001100
Cadena es aceptada
Estado final: q0
```

15. Cadena: 101 → Resultado: cadena no es aceptada.

```
C:\Users\Gonzalo\Desktop>java AFD_GMS AFD.txt
Alfabeto: 0 1
Estado: q1; Transicion: q3,q0,
Estado: q2; Transicion: q0,q3,
Estado: q3; Transicion: q1,q2,
Estado: q0; Transicion: q2,q1,
Estado inicial: q0
Estados finales: q0,
Ingrese la cadena: 101
Cadena no es aceptada
C:\Users\Gonzalo\Desktop>
```

### REPORTE DE ACTIVIDAD

# **CONCLUSIÓN:**

En conclusión, los autómatas resueltos anteriormente son ejercicios realizados en la en clase y nos damos cuenta de que han quedado completamente resueltos de manera correcta ya que al codificar nuestro programa y ejecutarlo correctamente nos dan los mismos resultados.

Ahora puedo decir con exactitud y comprendo que el AFD contiene dos tipos de elementos los cuales son:

Alfabeto es un conjunto de símbolos y normalmente se denota con la letra Σ.
 Ejemplos:

$$\Sigma = \{a,b,c,...z\} \Sigma = \{1,2,3,...9\} \Sigma = \{0,1\} \Sigma = \{a,b\}$$

• Cadena o palabra es un conjunto de símbolos de algún alfabeto  $\Sigma$  concatenados entre sí, es decir uno enseguida del otro.

Ahora puedo concluir que un los Autómatas Finitos Deterministas (que son parte de los lenguajes regulares) los cuales son abstracciones de las máquinas, sin tomar en cuenta ni la forma de la máquina, ni sus dimensiones, sino que se enfoca a entender cómo funciona, es decir capturan solamente el aspecto referente a las secuencias de eventos que ocurren.

# **BIBLIOGRAFÍA:**

- Apuntes proporcionados por el docente.
- Dean K. (1995). "Teoría de Autómatas y Lenguajes Formales". Edit. Prentice Hall, España.
- Hopcroft J. E., Ullman J.D. (2007). "Introducción a la teoría de autómatas, lenguajes y computación". 3ª ed. Edit. Pearson Educación, Madrid.
- Linz P. (2001) "An Introduction to Formal Languages and Automata", 3rd Edition, J.A. Bartlett.
- Martin J. (2004). "Lenguajes Formales y Teoría de la computación". 3ª ed. Edit. MacGraw-Hill Interamericana de México

# LINK DEL AUTÓMATA Y EL CÓDIGO FUENTE:

https://drive.google.com/drive/folders/1PkNltBdXWyAZxd24QWmWwunmF913MSEv?usp=drive link