



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE ACTIVIDAD

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:

LENGUAJES Y AUTOMATAS I

NOMBRE DE LA ACTIVIDAD:

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



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 Σ (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 δ o Δ (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:

Ejemplo:

Dado el AFD $M = (Q, \Sigma, q_0, F)$ donde

$Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{0, 1\}$ $q_0 = q_0$ $F = \{q_0\}$

y con la siguiente función de transición de estados

δ	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

$$\begin{aligned}\delta(q_0, 011) &= \delta(\delta(q_0, 0), 11) \\ \delta(q_0, 01) &= \delta(\delta(q_0, 0), 1) \\ \delta(q_0, 0) &= q_2 \\ \delta(q_2, 1) &= q_3 \\ \delta(q_3, 1) &= q_2\end{aligned}$$

determinar cuales de las siguientes cadenas son aceptadas por el autómata:

a). $w = 1010$

b). $w = 011$

Solución:

a). $w = 1010$

Se aplica la propiedad #2 de δ^*

$$\delta^*(q_0, 1010) = \delta(\delta^*(q_0, 101), 0)$$

$$\delta^*(q_0, 101) = \delta(\delta^*(q_0, 10), 1)$$

$$\delta^*(q_0, 10) = \delta(\delta^*(q_0, 1), 0)$$

Se aplica la observación #3 sobre δ y δ^*

$$\delta^*(q_0, 1) = \delta(q_0, 1) = q_1$$

Se sustituyen en reversa los valores calculados

$$\delta^*(q_0, 10) = \delta(q_1, 0) = q_3$$

$$\delta^*(q_0, 101) = \delta(q_3, 1) = q_2$$

$$\delta^*(q_0, 1010) = \delta(q_2, 0) = q_0$$



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 δ^*

$$\delta^*(q_0, 011) = \delta(\delta^*(q_0, 01), 1)$$

$$\delta^*(q_0, 01) = \delta(\delta^*(q_0, 0), 1)$$

Se aplica la observación #3 sobre δ y δ^*

$$\delta^*(q_0, 0) = \delta(q_0, 0) = q_2$$

Se sustituyen en reversa los valores calculados

$$\delta^*(q_0, 01) = \delta(q_2, 1) = q_3$$

$$\delta^*(q_0, 011) = \delta(q_3, 1) = q_2$$

Como $\delta^*(q_0, w) = q_2$ NO es un estado final, No es aceptada por el AFD. !!

Ejercicio: 2

1. Para el AFD anterior, validar las siguientes cadenas: $w_1 = 11010$, $w_2 = 1111$, $w_3 = 0001$, $w_4 = 101010$, $w_5 = 001100$, $w_6 = 101$

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.

```
1 //Instituto Tecnologico Superior De Alamo Temapache
2 //Lenguaje Y Automatas 1
3 //Alumno: Gonzalo Martinez Silverio
4 //Numero de control: 20220029
5 //Carrera: Ingenieria en sistemas computacionales
6 //Grupo: 6SiA Semestre: 6°
7 //Docente: Dr. Tania Turrubiates Lopez
8 import java.util.Arrays;
9 import java.util.HashMap;
10 import java.io.*;
11 import java.util.Scanner;
12
13 public class AFD_GMS{
14
15     public int numColumn;
16     private String ini, edoActual;
17     private String[] fin,symbols;
18     private HashMap<String,String[]> table;
```



REPORTE DE ACTIVIDAD

```
19
20 public AFD_GMS(String name) {
21     table = new HashMap<String,String[]>();
22     try{
23         // Se crea y se abre un fichero.
24         File inputFile = new File(name);
25
26         // Se crea un lector del archivo y un buffer
27         // que contendrá el texto del archivo
28         FileReader fr = new FileReader(inputFile);
29         BufferedReader br = new BufferedReader(fr);
30
```

```
31         // Se lee línea por línea el archivo
32         String linea;
33         String[] aux;
34         int i = 0;
35         linea = br.readLine();
36         while(linea != null){
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) == '0'){
41                     this.numColumn = dato.length;
42                     symbols = Arrays.copyOfRange(dato, 1, this.numColumn);
43                 }else{
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             }else{
53                 aux = Arrays.copyOfRange(dato, 1, this.numColumn);
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

```
122         }
123     }
124     //Si el indice auxiliar es diferente a -1
125     if(indAux != -1){
126         //Se va a instanciar un estado auxiliar
127         String[] auxEdo = this.table.get(this.edoActual);
128         //Se van a realizar las transiciones con el indice y el estado auxiliar
129         this.transicion(auxEdo, indAux);
130     }else{
131         //De otra forma, como no pertenece al alfabeto, termina el programa
132         System.out.println("No pertenece al alfabeto");
133         System.exit(0);
134     }
135 }
136 }
137 //Si el estado actual de la cadena es igual al estado final, la cadena es compatible
138 for(int i=0; i<fin.length; i++){
139     if(this.edoActual.equals(this.fin[i])){
140         System.out.println("Cadena es aceptada");
141         //imprime el estado final
142         System.out.println("Estado final: "+this.fin[i]);
143         System.exit(0);
144     }
145 }
146 //Si el estado actual de la cadena no es igual al estado final, la cadena no es compatible
147 System.out.println("Cadena no es aceptada");
148 }
149 }
```

```
150 public static void main(String[] args) { //Se hace uso de los metodos
151     AFD_GMS afd = new AFD_GMS(args[0]);
152     afd.showData();
153     afd.analizarCadena();
154 }
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.

```
@,0,1
q0,q2,q1
q1,q3,q0
q2,q0,q3
q3,q1,q2
Inicio,q0
Final,q0
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE ACTIVIDAD

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

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.1848]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Gonzalo>cd desktop
```

- Vamos a copilar nuestro programa escribiendo el comando: **javac AFD_GMS.java**

```
C:\Users\Gonzalo\Desktop>javac AFD_GMS.java
```

- 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)
```

- 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**
- 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



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

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




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:\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: 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
```



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>
```



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:
 $\Sigma = \{a,b,c,\dots,z\}$ $\Sigma = \{1,2,3,\dots,9\}$ $\Sigma = \{0,1\}$ $\Sigma = \{a,b\}$
- Cadena o palabra es un conjunto de símbolos de algún alfabeto Σ 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