

Maquinas de Estado

Steven Villacis, Doménica Salazar, Andrés Montiel

Departamento de Eléctrica y Electrónica

Universidad de las Fuerzas Armadas ESPE

dasalazar14@espe.edu.ec, damontiel@espe.edu.ec, jsvillacis1@espe.edu.ec

Resumen - Una máquina de estados finita es una representación abstracta para manipular símbolos con la finalidad de conocer si una cadena pertenece a un lenguaje o se puede utilizar de forma indistinta. Las máquinas de estados se caracterizan por tener un estado inicial, son capaces de recibir una cadena de símbolos, pueden cambiar de estado por cada elemento leído o pueden permanecer en el mismo estado. Además, tienen un conjunto de estados finales que indican si una cadena pertenece al lenguaje al final de una lectura.

Palabras Claves.- Máquina de estados, representación, estado inicial, estados finales.

Abstract. - A finite state machine is an abstract representation to manipulate symbols in order to know if a string belongs to a language or can be used interchangeably. State machines are characterized by having an initial state, they are capable of receiving a chain of symbols, they can change state for each element read or they can remain in the same state. Also, they have a set of final states that indicate whether a string belongs to the language at the end of a read.

Keywords - State machine, representation, initial state, final states.

I. INTRODUCCIÓN

La máquina de estado se define como un circuito secuencial con un comportamiento reflejado en distintos estados, dependiendo de las entradas y estado actual. Se utiliza para crear aplicaciones donde existen estados distinguibles, donde cada estado puede llevar a uno o varios estados y puede finalizar el flujo del proceso.

Es importante señalar que muchas aplicaciones requieren un estado de "inicialización" seguido de un estado predeterminado, donde se puede realizar muchas acciones diferentes. Estas acciones dependen de las entradas anteriores y actuales, así como de los estados.

Al diseñar máquinas de estados se puede crear un diagrama de estado para representar gráficamente los diferentes estados y cómo interactúan para modelar los algoritmos de control que necesita con estados lógicos discretos.

II. ESTADO DEL ARTE

Design of the Software Architecture for Starcraft Video Game on the Basis of Finite State Machine.

Los investigadores Ivan Y. Smolyakov y Sergey A. Belyaev de la Universidad Electrónica Saint-Petersburg realizaron en

el año 2019 un artículo exponiendo cuál sería el mejor método de resolución del videojuego Starcraft. Los autores del videojuego proponen diferentes enfoques para poder resolver el juego; uno de ellos es la coordinación de agentes radiales independientes, pero también seleccionan el enfoque clásico de las máquinas finitas de estados. Este enfoque viene combinado con la utilización de árboles de decisión. Los resultados mostraron las ventajas y las limitaciones de cada enfoque discutido en el artículo.

(Smolyakov & Belyaev, 2019)

Structural Representation of Synthesis Methods of Finite State Machine with Datapath of Transitions

Roman Babakov y Alexander Barkalov, investigadores de la Universidad de Zielona en Polonia, realizaron en el año 2018 un artículo donde exponen diferentes métodos para representar una máquina de estados finita con una ruta de datos de transiciones. Para una mejor síntesis de la quina de estados finita se debe definir bien las condiciones que se vayan a tomar, esto debido a que las condiciones determinan el diseño. Las componentes de la estructura son elementos del modelo a utilizar para la máquina de estados finita con ruta de datos de transiciones.

(Babakov & Barkalov, 2018)

A Test Generation Method Based on k-Cycle Testing for Finite State Machines

Yuya Kinoshita, Toshinori Hosokawa y Hideo Fujiwara, investigadores de la Universidad Osaka Gakuin en Japón, realizaron en el año de 2019 un artículo que propone un método para evitar las desventajas en las pruebas de escaneo. Estas pruebas requieren un tiempo de aplicación de prueba prolongado y una gran sobrecarga de hardware. El método propuesto es un modelo de expansión de tiempo con restricciones de estado inicial para controladores y un método de generación de pruebas basado en máquinas de estados finitos.

(Kinoshita, Hosokawa, & Fujiwara, 2019)

III. MARCO TEÓRICO

Máquinas de estado Moore y Mealy

Maquina de Mealy

George Mealy escribió un ensayo en el cual entra en profundidad acerca de crear máquinas de estado desde funciones matemáticas, y describe esas salidas de máquinas de estado en términos de sus entradas. Para diagramar la máquina Mealy, la salida está hecha para depender de ambos: el estado actual y la entrada.

La estructura de una máquina de Mealy es:

La estructura de una máquina de Mealy es:

$$MMe = \{Q, Ent, Sal, tran, res, q_0\}$$

Donde:

Q : conjunto de estados

Ent : alfabeto de entrada

Sal : alfabeto de salida

$tran$: función de transición

res : función de respuesta

q_0 : estado inicial

Gráficamente, se lo representa de la siguiente manera:



q_0 es el estado inicial. Si se está en q y llega e entonces se emite $s = res(q, e)$ y se transita a $p = tran(q, e)$.

Maquina de Moore:

Edward Moore escribió un ensayo en 1956 titulado **Gedanken-experiments on Sequential Machines**. En él dice que la salida depende sólo del estado, y el próximo estado es dependiente del estado actual y la entrada.

La estructura de una máquina de Moore se lo representa de la siguiente forma:

$$MMo = \{Q, Ent, Sal, tran, res, q_0\}$$

Donde:

Q : conjunto de estados

Ent : alfabeto de entrada

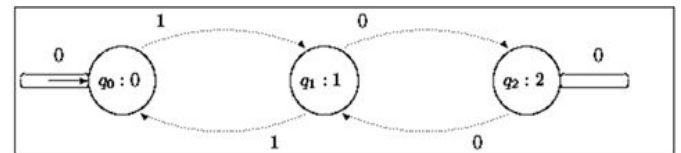
Sal : alfabeto de salida

$tran$: función de transición

res : función de respuesta

q_0 : estado inicial

Gráficamente, se lo representa de la siguiente manera:



Y su tabla sería de la siguiente forma:

<i>tran</i>	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

<i>res</i>	
q_0	0
q_1	1
q_2	2

IV. DESARROLLO DEL PROYECTO

Se desarrolló un código independiente para cada uno de los módulos de Imágenes, Botones, Acelerómetro, Brújula, Sensor de Temperatura y Música que son los módulos de interés.

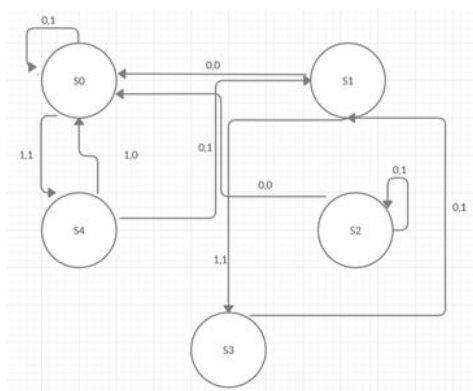
- Dibuje el diagrama de estados para la máquina de estado finito cuya tabla de estados es la siguiente. Partiendo del estado s_0 , calcula la salida para la cadena de entrada 1000110.

Estados	Transición		Salida	
	Entrada		Entrada	
S0	S0	S4	1	1
S1	S0	S3	0	1
S2	S0	S2	0	0
S3	S1	S1	1	1
S4	S1	S0	1	0

Para realizar el diagrama de estados partiendo de la tabla de transiciones, se empieza con estado S0. Se observa el primer valor de entrada en la transición, con ese valor de entrada su siguiente estado es S0, ahora se observa la salida. Con ese mismo valor de entrada su valor de salida será 1.

Terminado el primer valor de entrada se pasa al siguiente valor de entrada. Con ese valor el estado siguiente de S0 será S4, se observa la salida con el mismo valor de entrada, dando como resultado que su salida es 1.

Una vez terminado con el primer estado se procede a realizar el siguiente estado hasta terminar la tabla. el diagrama de transiciones quedaría de la siguiente forma:



Para la entrada 1000110, tan sólo hay que seguir el diagrama de estados, dando como resultado: 1111011.

B. Dibuje el diagrama de estados para la máquina de estado finito cuya tabla de estados es la siguiente. Partiendo del estado inicial s0, calcula la salida para la cadena de entrada abbccc.

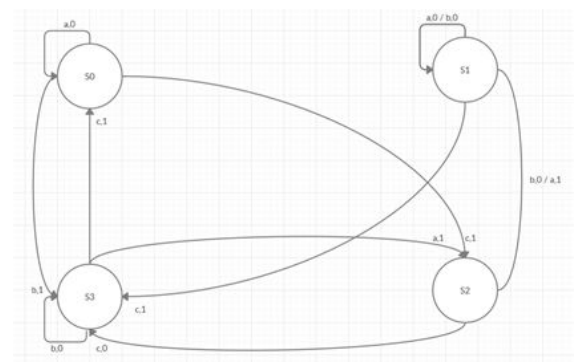
Estados	Transición			Salida		
	a	b	c	a	b	c
S0	S0	S3	S2	0	1	1
S1	S1	S1	S3	0	0	1
S2	S1	S1	S3	1	1	0
S3	S2	S3	S0	1	0	1

Para realizar el diagrama de estados partiendo de la tabla de transiciones, se empieza con estado S0. Se observa el primer valor de entrada en la transición (valor de entrada a), con ese valor de entrada su siguiente estado es S0, ahora se observa la salida. Con ese mismo valor de entrada su valor de salida será 0.

Terminado el primer valor de entrada se pasa al siguiente valor de entrada (valor de entrada b). Con ese valor el estado siguiente de S0 será S3, se observa la salida con el mismo valor de entrada, dando como resultado que su salida es 1.

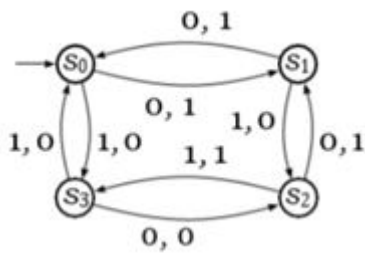
Por último se pasa al último valor de entrada (valor de entrada c), con ese valor en el estado S0 su siguiente estado será S2 y su salida será 1.

Una vez terminado con el primer estado se procede a realizar el siguiente estado hasta terminar la tabla. el diagrama de transiciones quedaría de la siguiente forma:



Para la entrada abbccc, tan sólo hay que seguir el diagrama de estados, dando como resultado: 101101.

C. Halle la tabla de estados para la máquina de estado finito cuyo diagrama de estados es:



Para realizar la tabla de transiciones partiendo del diagrama de estados, se empieza con el estado inicial y con el primer valor de entrada. Se localiza la flecha que tenga el primer valor de entrada y se observa a qué estado se encuentra apuntando. En este caso, la flecha apunta al estado S1 lo que significa que el estado siguiente del estado S0 con una entrada de 0 es S1. El número que se encuentra a lado del valor de entrada, representa el valor de la salida.

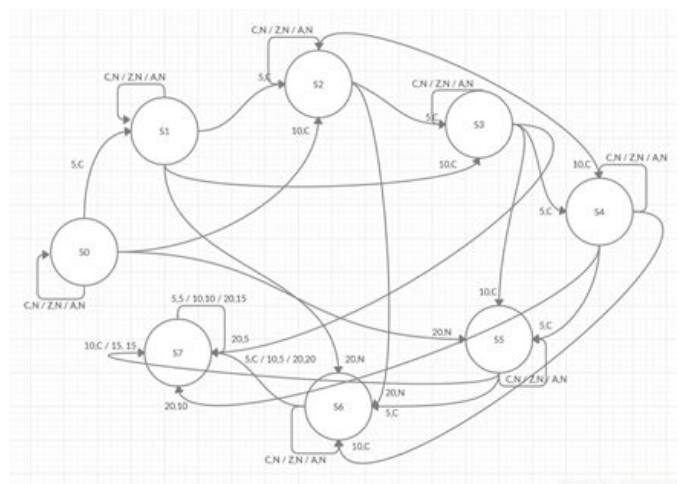
La tabla se llena yendo de izquierda a derecha, poniendo los datos correspondientes que tiene cada uno de los estados en cada una de las entradas. Estos datos se deben poner en las columnas de transiciones y salidas. Quedando de la siguiente manera:

Estados	Transición		Salida	
	Entrada		Entrada	
	0	1	0	1
S0	S1	S3	1	0
S1	S0	S2	1	0
S2	S1	S3	1	1
S3	S2	S0	0	0

- D. Construya una máquina de estado finito que modele una máquina expendedora de bebidas que acepta monedas de 5, 10 y 20 centavos. La máquina acepta monedas hasta que se introducen 25 centavos y devuelve cualquier cantidad que supere los 25 céntimos. Entonces, el cliente puede pulsar los botones y elegir una bebida de cola (C), cerveza (Z) o agua (A).

La máquina expendedora posee 8 posibles estados y 6 entradas que son las monedas que puede recibir la máquina y los productos que se pueden seleccionar. Los tres primeros estados con las entradas de las monedas cambian a los estados superiores, mientras que con las entradas de los productos se produce una retroalimentación. Las salidas de estos estados van a ser las siguientes: en la entrada de las dos primeras monedas su salida será un producto mientras que en el resto de entradas su salida será N de nula.

A partir del cuarto estado con la entrada de la moneda más alta su salida será el valor de la moneda, esto se debe a la acumulación generada por los estados anteriores. Para el último estado, como no hay estados siguientes todas las entradas de monedas darán retroalimentación, mientras que la entrada de los productos pasan a ser estados, esto debido a que para ese momento la acumulación llega a su valor máximo. en cuanto a las salidas del último estado estas tendrán el valor de sus entradas.



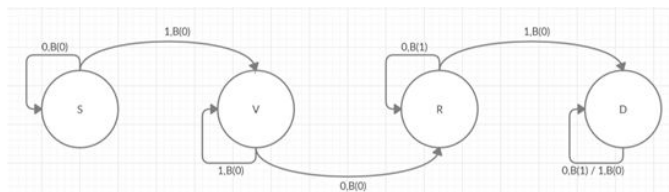
- E. Construya una máquina de estados finito que modele un circuito de riego automático como el mostrado en la figura. El circuito deberá accionar la bomba en las siguientes condiciones: a. El circuito accionará la bomba solamente cuando la tierra esté seca, pero antes debe comprobar las siguientes condiciones: i. Para evitar que la bomba se estropee por funcionar en vacío, nunca se accionará la bomba cuando el depósito de agua esté vacío. ii. Si hay restricciones en el riego (época de verano), sólo se podrá regar de noche. iii. En el resto del año (si no hay restricciones) se podrá regar de día y de noche (si la tierra está seca). b. Para la implementación del circuito se dispone de las siguientes

entradas: i. S: Señal que indica si la tierra está seca: Tierra seca: $S=1$; Tierra húmeda: $S=0$ ii. R: Señal que indica si hay restricciones en el riego (es verano): Hay restricciones: $R=1$ No hay restricciones: $R=0$ iii. D: Señal que indica si es de día o de noche: Día: $D=1$; Noche: $D=0$ iv. V: Señal que indica si el depósito de agua está vacío: Vacío: $V=1$; Hay agua: $V=0$

c. Y la salida B, que accionará la bomba para regar: Bomba funcionando: $B=1$; Bomba apagada $B=0$.

El sistema de riego se realiza se activa únicamente cuando se cumplen todas las condiciones. La máquina de estados comienza comprobando que la tierra se encuentra seca para continuar con la siguiente condición. Después de haber comprobado que la tierra se encuentra seca, se comprueba que el tanque de agua se encuentre lleno, si lo está se pasa a la siguiente condición.

Si el tanque está lleno se comprueba que no haya restricciones de ningún tipo. Si hay restricciones, se comprueba que el momento del día en que sea activada la bomba sea de noche. Si no hay restricciones de ningún tipo, ya no importa comprobar en qué momento del día se tiene que regar, porque de todas formas se activará la bomba.



V. CONCLUSIONES Y RECOMENDACIONES

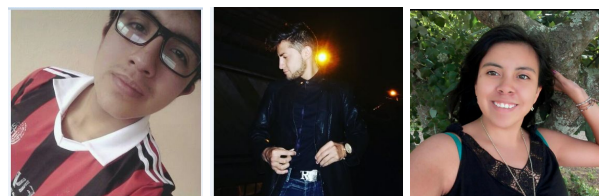
1. Se concluye que mediante la precedencia de una tabla de estados se construyen los grafos de estados a través de la interpretación de la misma.
2. En conclusión, la máquina de estados llega a ser una representación y establece si un suceso depende o no de un estado anterior.
3. Finalmente se determinó la maquina de estados en distintas representaciones utilizando las clasificaciones correspondientes a la de Mealy y Moore.
4. Se recomienda usar diagramas de grafos debido a que es más óptimo representar con 5 estados, a partir de 6 en adelante es más recomendable representar por las tablas.
5. La representación de estados es un inicio para introducir al mundo de la inteligencia artificial ya que cada estado puede ser dependiente del otro o ser independiente, puede inicializarse o no y del mismo

puede derivarse en uno o más estados, sirviendo para mejor organización en la toma de decisiones.

VII. REFERENCIAS O BIBLIOGRAFÍA

- BABAKOV, R., & BARKALOV, A. (2018). STRUCTURAL REPRESENTATION OF SYNTHESIS METHODS OF FINITE STATE MACHINE WITH DATAPATH OF TRANSITIONS. *PROCEEDINGS OF 2018 IEEE 9TH INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS, SERVICES AND TECHNOLOGIES, DESSERT 2018*, 229–233. [HTTPS://DOI.ORG/10.1109/DESSERT.2018.8409134](https://doi.org/10.1109/DESSERT.2018.8409134)
- INTRODUCCIÓN, B., ALEJANDRO, J., & OROZCO, G. (N.D.). *MÁQUINAS DE ESTADOS FINITOS*.
- KINOSHITA, Y., HOSOKAWA, T., & FUJIWARA, H. (2019). A TEST GENERATION METHOD BASED ON K-CYCLE TESTING FOR FINITE STATE MACHINES. *2019 IEEE 25TH INTERNATIONAL SYMPOSIUM ON ON-LINE TESTING AND ROBUST SYSTEM DESIGN, IOLTS 2019*, 232–235. [HTTPS://DOI.ORG/10.1109/IOLTS.2019.8854426](https://doi.org/10.1109/IOLTS.2019.8854426)
- MÁQUINAS DE ESTADO - MCI CAPACITACIÓN. (N.D.). RETRIEVED SEPTEMBER 5, 2020, FROM [HTTPS://CURSOS.MCIELECTRONICS.CL/2019/06/18/MAQUINAS-DE-ESTADO/](https://cursos.mcielectronics.cl/2019/06/18/MAQUINAS-DE-ESTADO/)
- MÁQUINAS DE MEALY. (N.D.). RETRIEVED SEPTEMBER 5, 2020, FROM [HTTP://DELTA.CS.CINVESTAV.MX/~GMORALES/TA/NODE49.HTML](http://delta.cs.cinvestav.mx/~gmorales/ta/node49.html)
- MÁQUINAS DE MOORE. (N.D.). RETRIEVED SEPTEMBER 5, 2020, FROM [HTTP://DELTA.CS.CINVESTAV.MX/~GMORALES/TA/NODE50.HTML](http://delta.cs.cinvestav.mx/~gmorales/ta/node50.html)
- SMOLYAKOV, I. Y., & BELYAEV, S. A. (2019). DESIGN OF THE SOFTWARE ARCHITECTURE FOR STARCRRAFT VIDEO GAME ON THE BASIS OF FINITE STATE MACHINES. *PROCEEDINGS OF THE 2019 IEEE CONFERENCE OF RUSSIAN YOUNG RESEARCHERS IN ELECTRICAL AND ELECTRONIC ENGINEERING, ELCONRUS 2019*, 356–359. [HTTPS://DOI.ORG/10.1109/EIConRus.2019.8656866](https://doi.org/10.1109/EIConRus.2019.8656866)

DATOS GENERALES DE LOS AUTORES



Andrés Montiel, Steven Villacís y Domenica Salazar son estudiantes de la Universidad de las Fuerzas Armadas ESPE, la cual cursan la carrera de Ingeniería en Telecomunicaciones.