Máquinas de estado finito

3 de septiembre de 2020

1. Preliminares

Definición 1 *Una n-ada ordenada es un "conjunto ordenado" con n elementos.*
Nota: En específico el 1º,2º,...,(n-1)-avo elemento del conjunto ordenado son los n-1 elementos de la primera componente de la n-ada ordenada que, se repita, es una (n-1)-ada ordenada, y el n-ésimo elemento del conjunto ordenado es la segunda componente de la n-ada ordeneada. Así, para una n-ada ordenada, se puede referir al primero, segundo y n-esimo elemento (componente) de la n-ada ordenada. En la práctica, con frecuencia se relaja la notación y se usa (a,b,c) o abc para representa la terna ordenada ((a,b),c); se usa (a,b,c,d) o abcd para representar la cuarteta ordenada (((a,b),c),d), y así sucesivamente.

Ejemplo: Sea A={a,b,c,d,...,x,y,z} las 27 letras del alfabeto español. Una palabra de n letras es una n-ada ordenada de letras del alfabeto.
Nota: En el contexto de lenguajes frecuentemente se usa los términos sucesión, cadena o enunciados (de letras) de forma intercambiable con el término n-adas ordenadas (de letras).

Nota: Se usa la notación A^n o $\{a,b,c,d,...,x,y,z\}^n$ para denotar el conjunto de todas las sucesiones de n letras del conjunto A o {a,b,c,d,...,x,y,z}. También, se utilizará la notación A^+ o $\{a,b,c,d,...,x,y,z\}^+$ para denotar el conjunto de todas las sucesiones de una letra, de dos letras y de n letras.

Ejemplo:
El conjunto de los nombres de un directorio telefónico es un subconjunto de A^+.
El conjunto de los nombres de 5 letras del directorio es un subconjunto de A^5.

Ejemplo: Sea B = { a,b,...,y,z,A,B,...,Y,Z,., ,,;,:,!,?,_} (donde el guión denota un espacio en blanco). Un enunciado del español es una sucesión de B^+. De forma análoga, una afirmación de cualquier lenguaje de programación es una sucesión de C^+, donde C={A,B,...,Y,Z,0,1,2,...,8,9,+,-,*,/,;,.,=}

Definición 2 *Sea A un conjunto finito, que es el alfabeto del lenguaje. Un lenguaje (sobre el alfabeto A) es un subconjunto del conjunto A^+.*
Nota: No se usa el concepto de "sucesión nula"(una sucesión que no contiene letras). En general, un lenguaje puede contener la sucesión nula como una de sus sucesiones.

Ejemplo: Sea A={a,b,c}. Los siguientes son lenguajes sobre el alfabeto A:
L_1={a,aa,ab,ac,abc,cab}
L_2={aba,aabaa}
L_3={ }
L_4={ $a^i c b^i$ | $i \geq 1$}

Nota: Los lenguajes se definen como conjunto de cadenas; por lo tanto, se pueden aplicar todas las operaciones de conjunto a los lenguajes.

Ejemplo: Sean L_1 y L_2 dos lenguajes.
$L_1 \cup L_2$ es un lenguaje, que contiene los enunciados que están en L1 o en L2.
$L_1 \cap L_2$ es un lenguaje, que contiene los enunciados que están en L1 y en L2.
$L_1 \setminus L_2$ es un lenguaje, que contiene los enunciados que están en L1 y no están en L2.

Definición 3 *Una gramática de estructuras de frases para especificar un lenguaje consta de cuatro elementos:*
1. Un conjunto de terminales T.
2. Un conjunto de no terminales N.
3. Un conjunto de producciones P.
4. Un no terminal especial en N que es el símbolo inicial.
Nota: Esto significa:
1. Las terminales en T son símbolos usados para formar enunciados en el lenguaje.
2. Las no terminales en N son los símbolos intermedios usados para describir la estructura de los enunciados.
3. Las producciones son reglas gramaticales que especifican cómo se pueden construir enunciados en el lenguaje. Una producción es de la forma $\alpha \longrightarrow \beta$, donde α y β son cadenas de terminales y no terminales. Una producción específica que la cadena α puede transformarse en la cadena β.
4- El símbolo inicial es un no terminal especial que comienza la generación de cualquier enunciado del lenguaje.

Nota: Dada una gramática, se puede generar los enunciados del lenguaje como:
1. Comenzar con el símbolo inicial como la cadena actual de terminales y no terminales.
2. Si cualquier porción de la cadena actual de terminales y no terminales coincide con el lado izquierdo de la producción, remplazar esa porción de la cadena por el lado derecho de la producción. Denote α la cadena actual de terminales y no terminales. Si $\alpha = \alpha_1 \alpha_2 \alpha_3$

y existe una producción $\alpha_2 \longrightarrow \beta$, entonces se puede remplazar la subcadena α_2 en α por β para obtener la cadena $\alpha_1\beta\alpha_3$. Se usa la notación $\alpha \Longrightarrow \alpha_1\beta\alpha_3$ para indicar que la cadena α se puede tranformar en la cadena $\alpha_1\beta\alpha_3$ remplazando una porción de la cadena α de acuerdo con una de las producciones de la gramática.
3. Cualquier cadena de terminales que se obtenga repitiendo el paso 2 es un enunciado del lenguaje. En el paso 2 existe la posibilidad de aplicar más de una producción para transformar la cadena actual de terminales y no terminales; en este caso se puede seleccionar cualquiera de las producciones. Además, si en el paso 2 se llega a una cadena de terminales y no terminales para la cual no se puede aplicar una producción, entonces se ha llegado a un callejón sin salida y se debe comenzar de nuevo con el símbolo inicial para obtener un enunciado del lenguaje.
El proceso de generar un enunciado como el descrito se conoce como derivación.

Ejemplo: El enunciado "un perro corre lentamente"se deriva como: **enunciado** $\Longrightarrow$ **frase-nominal frase-de-verbo-intranistivo** $\Longrightarrow$ **frase-nominal verbo-intransitivo adverbio** $\Longrightarrow$ **frase-nominal verbo-intransitivo** lentamente $\Longrightarrow$ **frase-nominal** corre lentamente $\Longrightarrow$ **artículo sustantivo** corre lentamente $\Longrightarrow$ **artículo** perro corre lentamente $\Longrightarrow$ un perro corre lentamente

Ejemplo: Construir una gramática para el lenguaje L={x | x∈{a,b}$^+$, el número de letras a en x es múltiplo de 3}.
Sean T={a,b} y N={S,A,B}, con S como símbolo inicial. Además, P={ S⟶bS, S⟶b, S⟶aA, A⟶bA, A⟶aB, B⟶bB, B⟶aS, B⟶a}.
La cadena bbababbab es derivada.
S $\Longrightarrow$ bS $\Longrightarrow$ bbS $\Longrightarrow$ bbaA $\Longrightarrow$ bbabA $\Longrightarrow$ bbabaB $\Longrightarrow$ bbababB $\Longrightarrow$ bbababbB $\Longrightarrow$ bbababbaS $\Longrightarrow$ bbababbab.

Definición 4 *Una gramática es una gramática de tipo 3 si todas las producciones de la gramática son de las formas A ⟶a o A⟶aB o, de manera equivalente, de las formas A⟶a o A ⟶Ba.*

2. Máquina de estado finito

Definición 5 *Una máquina de estado finito está especificada por:*
1. Un conjunto finito de estados S={s_0, s_1, s_2,...}.
2. Un elemento especial del conjunto S, s_0, conocido como estado inicial.
3. Un conjunto finito de caracteres de entrada I={i_1, i_2,...}.
4. Un conjunto finito de caracteres de salida O={o_1, o_2,... }.
5. Una función f de S×I a S, conocida como función de transición.
6. Una función g de S a O, conocida como función de salida.
Nota: En cualquier instante, una máquina de estado finito se encuentra en uno de sus

estados. Al llegar un caracter de entrada, la máquina pasará a otro estado de acuerdo con la función de transición. Además, en cada estado la máquina produce un caracter de salida según la función de salida. Al principio, la máquina se encuentra en su estado inicial.

Ejemplo: Sea S={s_0, s_1, ..., s_6}, I={a,b,c}, O={0,1} y s_0 estado inicial.

		Entrada		
	Estado	a	b	c
$\Rightarrow$	s_0	s_1	s_2	s_5
	s_1	s_2	s_3	s_6
	s_2	s_3	s_4	s_6
	s_3	s_4	s_5	s_6
	s_4	s_5	s_6	s_6
	s_5	s_6	s_6	s_6
	s_6	s_1	s_2	s_5

Cuadro 1: Especificación de f

Estado	Salida
s_0	0
s_1	0
s_2	0
s_3	0
s_4	0
s_5	0
s_6	1

Cuadro 2: Especificación de g

Nota: f es especificada como una tabla.
Significado de máquina vendedora, donde los estados s_0, s_1, s_2, s_3, s_4, s_5, s_6 corresponden a 0, 5, 10, 15, 20, 25 y 30 o más centavos. Los caracteres a, b y c corresponden a las monedas de 5, 10 y 25 centavos y los caracteres de salida 0 y 1 corresponden al momento en que la máquina entregará NADA o GOMA DE MASCAR.

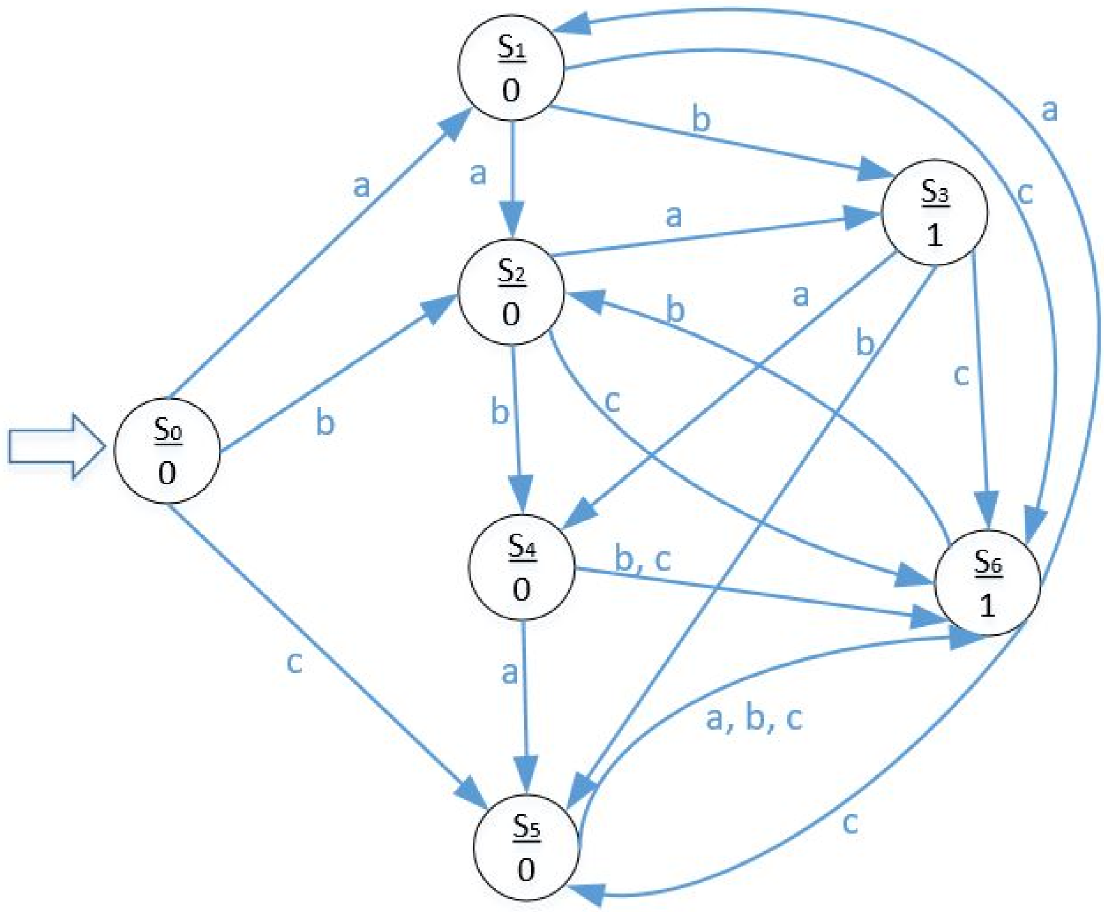

Figura 1: Máquina de estado finito

Nota: Una máquina de estado finito pude ser útil para modelar un sistema físico.

Ejemplo: Diseñar un contador módulo 3 que reciba una sucesión de número 0, 1 y 2 como entrada y produzca una sucesión de números 0, 1 y 2 como salida, de manera que la salida sea igual al módulo 3 de la suma de los dígitos de la sucesión de entrada.

		Entrada			
	Estado	0	1	2	Salida
⇒	A	A	B	C	0
	B	B	C	A	1
	C	C	A	B	2

Cuadro 3: Especificación de f y g

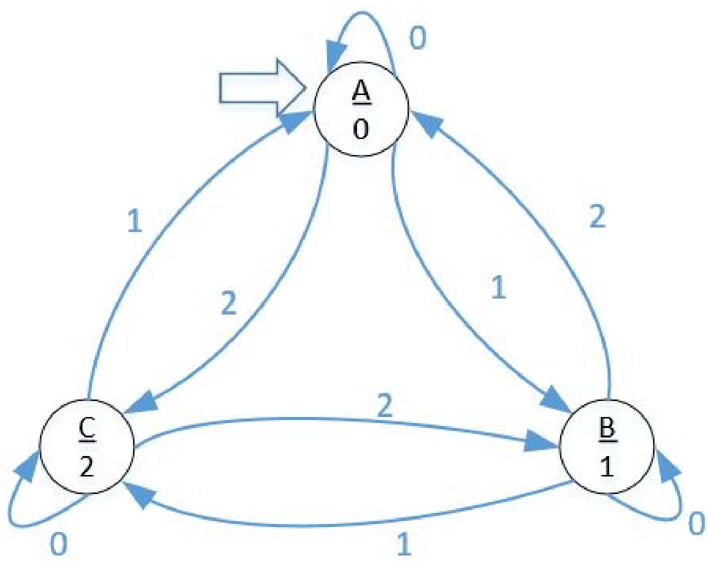

Figura 2: Máquina de estado finito

Nota; A es un estado correspondiente a la situación en que el módulo 3 de la suma de los dígitos es 0; B, corresponde a la situación en que el módulo 3 de la suma de los dígitos de entrada es 1 y C, corresponde a la situación que el módulo 3 es la suma de los dígitos de entrada es 2.

Ejemplo: Diseño de un dispositivo que compara 2 números binarios para determinar si son iguales o cuál de los dos es mayor. Supóngase que los dígitos de los 2 números llegan uno por uno, empezando con los digitos de orden inferior. Así, el alfabeto de entrada es {00. 01. 10. 11}, donde los 2 dígitos de cada pareja son los digitos correspondiente en los números a ser comparados. El alfabeto de salida es {=,>,<}

		Entrada				
	Estado	00	01	10	11	Salida
⇒	A	A	C	B	A	=
	B	B	C	B	B	>
	C	C	C	B	C	<

Cuadro 4: Especificación de f y g

Ejemplo: Modela el comportamiento de una persona. El conjunto de estados es { Feliz, Enojado, Deprimido}, el conjunto de entradas es { tarea, fiesta, mal situación} y el conjunto de salida es { cantar, mal decir, dormir}.

	Estado	Entrada			
		tarea	fiesta	mal situación	Salida
⇒	A (Feliz)	A	A	B	cantar
	B (Enojado)	C	A	B	mal decir
	C (Deprimido)	C	A	C	dormir

Cuadro 5: Especificación de f y g

3. Máquinas equivalentes

Definición 6 *Dos máquinas de estado finito son equivalentes si, a partir de sus respectivos estados iniciales, producen la misma sucesión de salida cuando se les da la misma sucesión de entrada.*
Nota: Máquinas equivalentes poseen comportamientos terminales idénticos aunque sus estructuras internas pueden ser diferentes.

Ejemplo: Las dos máquinas representadas por las siguientes tablas son equivalentes.

	Estado	Entrada		
		1	2	Salida
⇒	A	B	C	0
	B	F	D	0
	C	G	E	0
	D	H	B	0
	E	B	F	1
	F	D	H	0
	G	E	B	0
	H	B	C	1

Cuadro 6: Especificación de f y g

	Estado	Entrada		
		1	2	Salida
⇒	A	B	C	0
	B	C	D	0
	C	D	E	0
	D	E	B	0
	E	B	C	1

Cuadro 7: Especificación de f y g

Ejemplo: Para la sucesión de entradas 1122212212. Determine su salida para ambas

máquina del ejemplo anterior.
00010100001 00010100001

Definición 7 *En una máquina de estado finito, los estados s_i y s_j son equivalentes si para cualquier sucesión de entrada la máquina producirá la misma sucesión de salida, independientemente de que ésta inicie en s_i o en s_j.*
Nota: Dos estados equivalente pueden comportarse en uno sólo sin cambiar el comportamiento terminal de la máquina. Si s_i y s_j son estados equivalente, se pueden modificar la función de transición al eliminar el estado s_j y al remplazar todas las transiciones hacia el estado s_j con otras hacia s_i.

Ejemplo: Como los estados C y F, D y G y E y H son estados equivalentes, se puede eliminar F, G y H para obtener la máquina de estados.

Nota: El estado de una máquina representa el resumen de la historia de la máquina. Así, 2 estados son equivalentes si representan resúmenes equivalentes en lo que al comportamineto terminal de la máquina se refiere.

Ejemplo: La máquina del cuadro 6 acepta una sucesión de números 1 y 2 como entrada, y producirá un 1 si la suma de los dígitos que ha recibido es divisible por 4. Los estados E y H representan que la suma de los dígitos que la máquina ha recibido sea un múltiplo de 4 (tiene 1 en la salida), los estados C y F representan que la suma de los dígitos que la máquina ha recibido sea un múltiplo de 4 más 2, los estados D y G representan que la suma de los dígitos que la máquina ha recibido sea múltiplo de 4 más 3 y el estado B representa que la suma de dígitos que la máquina ha recibido sea múltiplo de 4 más 1. Por lo tanto, se puede combinar estados equivalentes sin cambiar el comportamiento terminal de la máquina.

Nota: ¿Cómo determinar cuando 2 estados son equivalentes? Según definición es necesario un examen exhaustivo de todas las posibles sucesiones de entrada de longitud arbitraria.

Definición 8 *Dos estados son 0-equivalentes si tienen la misma salida; dos estados son 1-equivalentes si tienen la misma salida y si, para todo caracter de entrada, sus sucesores son 0-equivalente. En general, dos estados son k-equivalentes si tienen la misma salida y si, para cada caracter de entrada, sus sucesores son (k-1)-equivalentes.*
Nota: Se obtiene que 2 estados s_j o s_i son k-equivalentes, entonces para cualquier sucesión de entrada de longitud k o menor, la máquina producirá sucesiones de salida idénticas sin importar si está en el estado s_j o s_i. Además, dos estados son equivalentes si son k-equivalentes para toda k.

Ejemplo: Del cuadro 8 se observa que:

	Entrada		
Estado	0	1	Salida
A	B	F	0
B	A	F	0
C	G	A	0
D	H	B	0
E	A	G	0
F	H	C	1
G	A	D	1
H	A	C	1

Cuadro 8: Especificación de f y g

a) Los estados A, C son 0-equivalentes.
b) Los estados G y H son 1-equivalentes.
π_0={ { A,B,C,D,E }, { F,G,H } }
π_1={ { A,B,E }, { C,D }, { F }, { G,H } }
π_2={ { A,B }, { E }, { C,D }, { F }, { G,H } }

Nota: Si 2 estados s_i y s_j son k-equivalentes y s_i y s_h son k-equivalentes, entonces s_j y s_h también son k-equivalentes. Se puede definir una relación de equivalencia sobre el conjunto de estados de forma que 2 estados están relacionados si son k-equivalentes. Por lo tanto, está relación induce a una partición sobre el conjunto de estados. Se denota mediate π_k.

Teorema 1 *Dos estados están en el mismo bloque (clase de equivalencia)* $\pi_k \iff$ *estos están en el mismo bloque* π_{k-1} *y para todo caracter de entrada, sus sucesores están en el mismo bloque* π_{k-1}.
Nota: Sugiere un procedimiento para calcular las particiones π_0. π_1. π_2, ... , π_k *en forma sucesiva.*

Ejemplo: Como A,B,C,D y E tienen la misma salida (0) y F,G y H tienen la misma salida (1). Entonces, π_0={ { A,B,C,D,E }, { F,G,H } }.
Se obtiene: π_1={ { A,B,E }, { C,D }, { F }, { G,H } }.
De igual manera, π_2={ { A,B }, { E }, { C,D }, { F }, { G,H } } = π_3

Nota:
1. Si π_k es igual a π_{k-1}, entonces π_m=π_{k-1}, $\forall$ m$\geq$k.
2. π_k es un refinamiento de π_{k-1} .
El punto 1 permite terminar el procedimiento de construcción siempre que se alcance dos particiones sucesivas idénticas. En este caso, todos los estados sean k-equivalentes son equivalentes.

El punto 2 garantiza que el procedimiento de construcción no irá más allá de π_{n-2}, donde n es el número de estados , dado que π_0 tiene al menos 2 bloques y π_{n-2} tendrá n bloques si el procedimeinto de construcción no se termina antes.

Ejemplo: Como $\pi_2 = \pi_3$, se concluye que A y B son equivalentes, C y D son equivalentes y G y H son equivalentes.

Nota: Se puede emplear el procedimiento anterior para determinar los estados equivalentes y obtener una máquina equivalentes que podría tener menos estados.

4. Máquinas de estado finito como reconocedores de lenguaje

Nota: Sea O={0,1} el alfabeto de salida de una máquina de estado finito.

Definición 9 *Un estado es de aceptación si su salida es 1 y un estado es de rechazo si su salida es 0.*

Definición 10 *Una sucesión de entrada es aceptada por una máquina de estado finito, si la sucesión conduce a la máquina desde el estado inicial hasta un estado de aceptación.*

Definición 11 *Una sucesión de entrada es rechazada por una máquina de estado finito, si la sucesión conduce a la máquina desde el estado inicial hasta un estado de rechazo.*

Ejemplo: Una máquina de estado finito que acepta todas las sucesiones binarias que terminan con los dígitos 011.

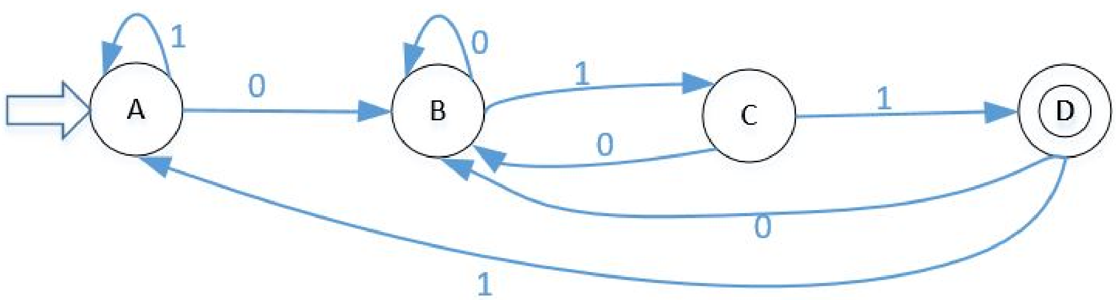

Figura 3: Máquina de estado finito

Ejemplo: Una máquina de estado finito que acepta todas las sucesiones binarias que tiene la forma de cualquier cantidad de números 0, seguidos por un 1 o más números 1, seguido por un 0 o más números 0, seguidos por un 1, seguido por cualquier cantidad de números 0, seguido por 1 y entonces seguidos por cualquier combinación de 0's y 1's.

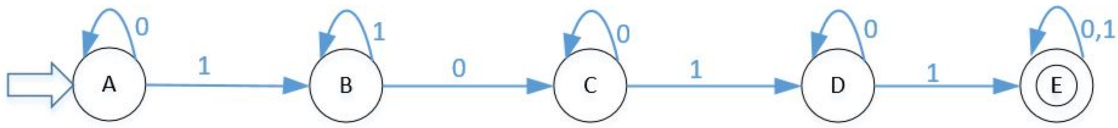

Figura 4: Máquina de estado finito

Definición 12 *Un lenguaje es un lenguaje de estado finito si existe una máquina de estado finito que acepta exactamente todos los enunciados del lenguaje.*

Ejemplo: El lenguaje que consiste de todas las sucesiones binarias que terminan con 011 es un lenguaje de estado finito.

Ejemplo: L= { $a^k b^k$ | k≥1} no es un lenguaje estado finito.

Ejemplo: L= { a^k | k=i^2,i≥1} no es un lenguaje estado finito.

Teorema 2 (Lema de bombeo) *Sea L un lenguaje de estado finito aceptado por un máquina de estado finito con N estados. Para cualquier sucesión α en el lenguaje, cuya longitud es mayor o igual a N, α puede escribirse como uvw de manera que v es no vacía y uv^iw también está en el lenguaje para i≥0, donde v^i denota la concatenación de i copias de la sucesión v.*

5. Lenguajes de estado finito y leguajes tipo-3

Definición 13 *Una máquina de estado finito no determinista es:*
1. Un conjunto finito de estados S={ s_0, s_1, s_2, ...}.
2. Un elemento especial del conjunto S, s_0, estado inicial.
3. Un conjunto finito de caracteres de entrada I={ i_1, i_2, ...}.
4. Un conjunto finito de caracteres de salida O={ o_1, o_2, ...}.
5. Una función F desde S×I a P(S), función de transición.
6. Una función g desde S hacia O, función de salida.
Nota: Es un modelo abstracto útil como reconocedor de lenguajes.

Ejemplo: Dada una máquina de estado finito. Muestre los estados de la máquina correspondiente a la solución de la entradas 000100001.

		Entrada		
	Estado	0	1	Salida
⇒	A	B	B,C	0
	B	A,C	C	0
	C	A	B,C	1

Cuadro 9: Especificación de f y g de la máquina de estado no-determinista

	0	0	0	1	0	0	0	0	1
A	B	A	A	B	A	A	A	A	B
		C	B	C	C	B	B	B	C
							C	C	

Cuadro 10: Ejecución de la cadena

Definición 14 *Una sucesión es aceptada por una máquina de estado finito no determinista si al comenzar desde el estado inicial, de entre todos los estados finales a los que la sucesión conducirá a la máquina, uno de ellos es un estado de aceptación.*

Ejemplo: Para la máquina de estado finito no-determinista del cuadro 11, la sucesión 0001 es aceptada y la sucesión 11100 no lo es.

Teorema 3 *Para cualquier máquina de estado finito no determinista existe una máquina de estado finito determinista que acepta el mismo lenguaje.*

Ejemplo: Para la máquina de estado finito no-determinista del cuadro 11, se muestra la máquina de estado finito determinista correspondiente en el cuadro 12.

		Entrada		
	Estado	0	1	Salida
⇒	A	B	B,C	0
	B	A,C	–	0
	C	A	B,C	1

Cuadro 11: Especificación de f y g de la máquina de estado no-determinista

		Entrada		
	Estado	0	1	Salida
$\Rightarrow$	{A}	{B}	{B,C}	0
	{B}	{A,C}	{ }	0
	{C}	{A}	{B,C}	1
	{A,B}	{A,B,C}	{B,C}	0
	{A,C}	{A,B}	{B,C}	1
	{B,C}	{A,C}	{B,C}	1
	{A,B,C}	{A,B,C}	{B,C}	1
	{ }	{ }	{ }	0

Cuadro 12: Especificación de f y g de la máquina de estado determinista

Ejemplo: Para la máquina de estado finito, se construye una gramática que especifica el lenguaje aceptado por la máquina.

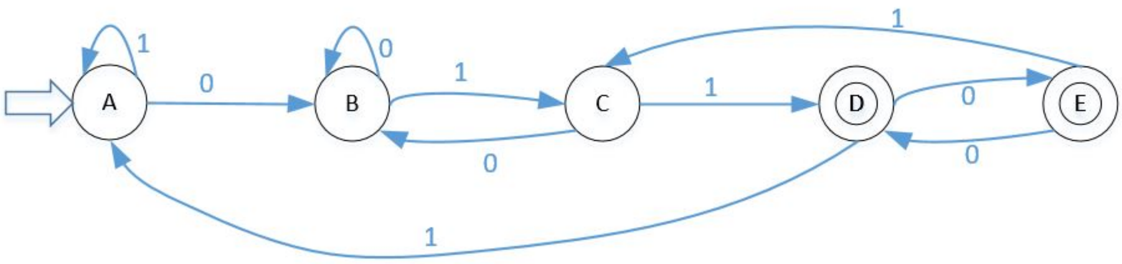

Figura 5: Máquina de estado finito

1. Sea {0,1}, el conjunto de caracteres de entrada, el conjunto de terminales.
2. Sea { A, B, C, D, E}, el conjunto de estados, el conjunto de no terminales.
3. Sea A, el estado inicial, el símbolo inicial.
4. Correspondiente a la transición f(s_p,i_q)=s_k, existe una producción $s_p \longrightarrow i_q s_k$ si s_k no es un estado de aceptación, y existen 2 producciones $s_p \longrightarrow i_q s_k$ y $s_p \longrightarrow i_q$ si s_k es un estado de aceptación.

Luego,
A $\longrightarrow$ 0B
A $\longrightarrow$ 1A
B $\longrightarrow$ 0B
C $\longrightarrow$ 0B
C $\longrightarrow$ 1D
C $\longrightarrow$ 1
D $\longrightarrow$ 0E
D $\longrightarrow$ 0
D $\longrightarrow$ 1A

$E \longrightarrow 0D$
$E \longrightarrow 0$
$E \longrightarrow 1C$
Nota: Cada no terminal en la gramática representa un conjunto de sucesiones que conducen a la máquina de estado finito, desde el estado correspondiente hasta un estado de aceptación.

Ejemplo: Para la gramática se construye una máquina de estado finito que acepta los enunciados del lenguaje específicado por la gramática.
$A \longrightarrow 0A$
$A \longrightarrow 1B$
$B \longrightarrow 0C$
$B \longrightarrow 0D$
$C \longrightarrow 0$
$C \longrightarrow 1B$
$C \longrightarrow 1D$
$D \longrightarrow 1$
$D \longrightarrow 1A$
1. Sea $\{0,1\}$, el conjunto de terminales, el conjunto de caracteres de entrada.
2. Existe un estado que correponde a cada no terminal, con el estado correspondiente al símbolo de inicio como estado inicial. Existe un estado adicional, el cual es un estado de aceptación. Además, existe otro estado, que es llamado un estado de atrapamiento. Siempre que una máquina entra en un estado de atrapamiento, permacerá en dicho estado y no podrá pasar a ningún otro estado. Así, si una sucesión de entrada conduce a la máquina de estado finito hacia un estado de atrapamiento, posiblemente tal sucesión de entrada no puede ser una porción de un enunciado en el lenguaje.
3. Para una producción de la forma $N_p \longrightarrow i_q N_k$ existe una transición del estado N_p hasta el estado N_k cuando la entrada es i_q. En el caso de una producción $N_p \longrightarrow i_q$, existe una transición del estado N_p al estado de aceptación cuando la entrada es i_q. Por otro lado, si no hay una producción de la forma $N_p \longrightarrow i_q N_k$ o $N_p \longrightarrow i_q$ para el estado N_p y la entrada i_q, existe una transición desde N_p hasta el estado de atrapamiento cuando la entrada es i_q. Por último para cualquier caracter de entrada, existe una transición desde el estado de aceptación hasta el estado de atrapamiento.

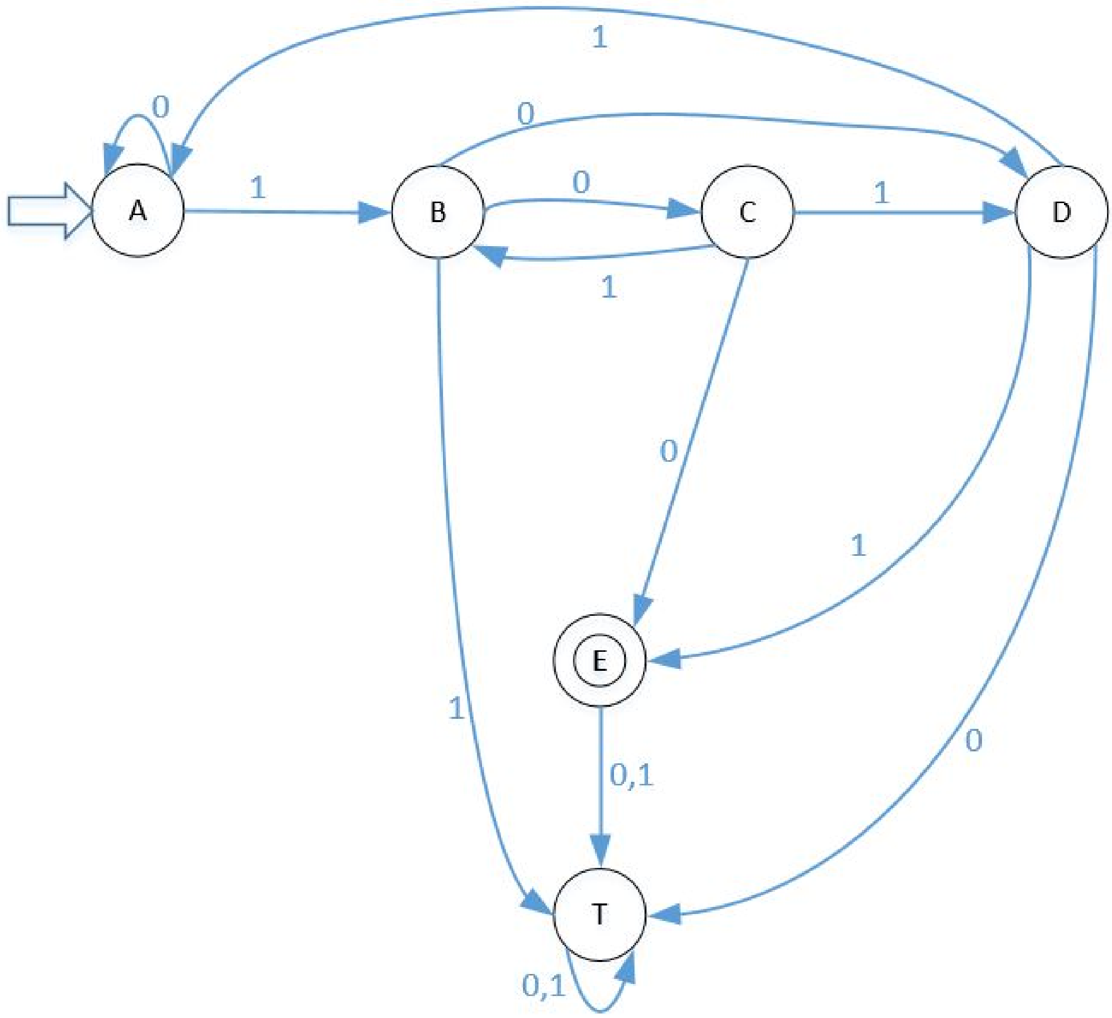

Figura 6: Máquina de estado finito

Nota: Cualquier lenguaje que puede ser aceptado por una máquina de estado finito no determinista, también puede ser aceptado por una máquina de estado finito determinista.

Nota: La clase de lenguajes de estado finito es la clase de lenguajes tipo-3.

6. Bibliografía

- Grimaldi, R. (1998), Matemáticas Discretas y Combinatoria. Addison Wesley Longman. México.

- Liu, C. (1995), Elementos de Matemáticas Discretas. McGraw-Hill. México.