LENGUAJES FORMALES Y DE PROGRAMACION

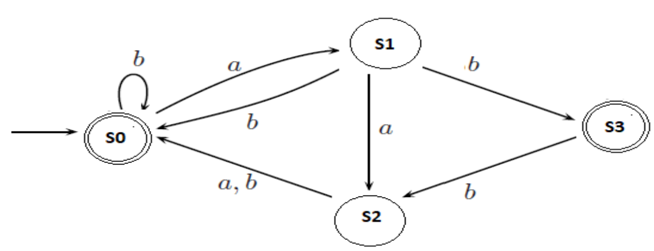
SECCION A+

Ing. Otto A. Rodriguez A.

EJERCICIOS Y EJEMPLOS

Realizar en cada ejercicio los autómatas finitos que reconozcan los siguientes lenguajes:

1. Debe iniciar con las letras abb seguido de cero o más a's o b's.
   L = {p | p inicie con abb seguido de cero o más a's o b's}
2. El lenguaje debe tener cualquier cantidad de a's o b's, pero deber permitir solamente cero o número par de a's. L = {P |P tiene un numero par de a's}
3. El lenguaje debe tener cualquier cantidad de a's o b's, pero deber permitir solamente número impar de a's. L = {P |P tiene un numero impar de a's}
4. Lenguaje que acepte x's y y's pero que después de una x no puede venir otra x, después de una y no puede venir otra y. L = {P | P se x o y pero p no puede ser xx o yy}
5. Lenguaje que acepte x's y y's pero por cada x deben haber dos y's como L = {P | P se x o y pero debe ser de la forma xyy }
6. Realizar un autómata que dado una secuencia de números binarios {1,0}. 1's y 0's al leer cada digito de izquierda a derecha el número debe ser múltiplo de 3.
   L = {P | P es número binario es múltiplo de 3}
7. Realizar un autómata que reconozca las palabras formadas por a's y b's donde toda a este entre 2 b's. L = {p | p sea a o b y toda a en p está entre dos b's}
8. Realizar un autómata formado por a's y b's donde no hay dos a's consecutivas.
   L = {w| no hay dos a's consecutivas en w}
9. Realizar un autómata finito con a's y b's que contengan ab o ba
   L = {P | P contiene un ab o ba }
10. Dado el siguiente diagrama de transición



Determinar las cadenas que son aceptadas o no por el autómata. Colocar la secuencia seguida para realizar la evaluación. Por ejemplo, So → a → S1 → b → S2 se lee:

de So con una **a** se mueve a S1, de S1 con una b se mueve a S2.

para baabaaa la secuencia seria:

So → b → So → a → S1 → a → S2 → b → So → a → S1 →a →s2 →So (se acepta)

Indicar si las siguientes expresiones es aceptan.

a) bab	b) aaba	c) aaaaaab	d) babababab
e) aba	f) abab	g) aabbaa	h) ab
i) aaaaaa	j) bbbbbba	k)aaa	l) ababaabb

Método del árbol

Realizar el Árbol de las siguientes expresiones regulares

1. (a|b)a*(b|ε)	2. (x|y|z|w) (x|y|z|w)*	3. abc(a|b)*
4. (ab)*(a|b)*(a|b| ε)	5. (abc)|a|b|c|ε)	6. L(L|D)*
7. (a|b||ε)x x* w x x*	8. ((ab)*(ba)*) |ε)	9. (a|ε)|(a*)|(a|ε)*
10. (((ab)*)|((ba)*)) |ε)		

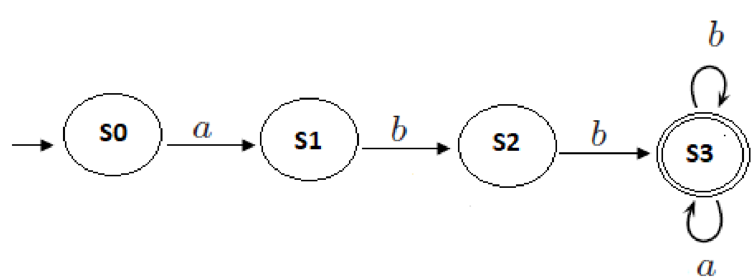
Encontrar el DFA mínimo de las expresiones regulares anteriores.

Soluciones

1. Debe iniciar con las letras abb seguido de cero o más a's o b's.
L = {p | p inicie con abb seguido de cero o más a's o b's}

Análisis: abb es una secuencia de letras que debe venir uno después de otro. Cada letra define un estado. Si no viene en ese orden especifico no pertenece al lenguaje. Luego pueden venir cualquier letra (a o b).

Respuesta:



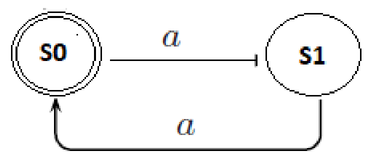
Tarea extra: Hacer un autómata finito que reconozca el mismo lenguaje, pero con menos estados.

2. El lenguaje debe tener cualquier cantidad de a's o b's, pero deber permitir solamente cero o número par de a's. L = {P |P tiene un numero par de a's}

<u>Análisis</u>

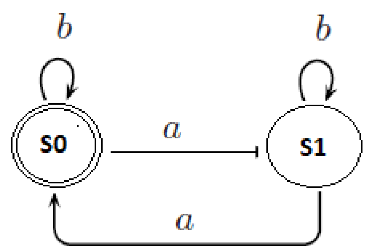
No importan cuantas b's vengan, pero por cada a que venga debe venir otra a. Esto se vuelve un ciclo que debe repetirse para aceptar todas las a's que vengan. De esta manera siempre tendrá cero o número par: 2, 4, 6…

Por ejemplo, abba valido (no importa cuantas b's)
bbabbabaa valido (las b's pueden estar al inicio o entre las a's)



Para cada a que pasa del estado So al estado S1 debe haber otra a que regrese para asegurar que son pares. El estado de aceptación debe estar en So pero no en S1

Respuesta:



Las b´s pueden repetirse en los dos estados. Con esto el Autómata finito permite número de a's pares.

Tarea extra: Realizar la expresión regular y la gramática para este autómata finito.

3. El lenguaje debe tener cualquier cantidad de a's o b's, pero deber permitir solamente un número impar de a's

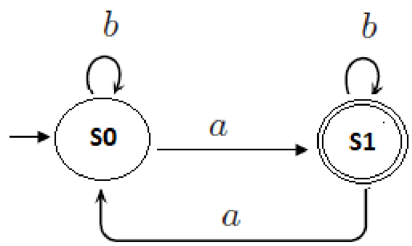
<u>Análisis</u>

No importan cuantas b's vengan, pero con una a debe aceptarse. Si viene otra a entonces debe regresar al ciclo y volver con otra a para que se acepte. De esta manera si hay un umero par de a's debe venir otra para aceptarse. Las b's deben repetirse o más veces. Esto es muy parecido al ejemplo anterior.

Por ejemplo, ababa es válida (no importa cuantas b's)
bbabbaba es válida (las b's pueden estar al inicio o entre las a's)
a es válida (solo una a es impar y se debe aceptar)

Lo que cambia ahora es el estado de aceptación.

Respuesta:

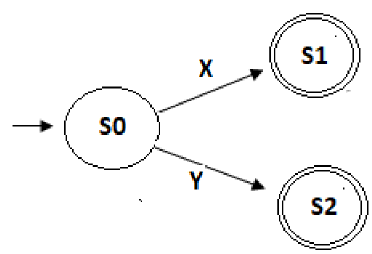


Tarea extra: Realizar la expresión regular y la gramática para este autómata finito.

4. Leguaje que acepte x's y y's pero que después de una x no puede venir otra x, después de una y no puede venir otra y. L = {P | P se x o y pero p no puede ser xx o yy }

   Análisis:

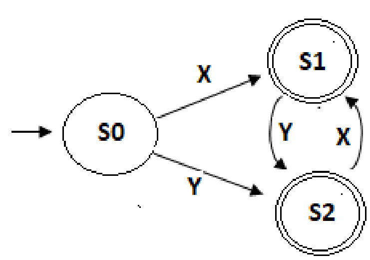
   a. El autómata puede iniciar con una x o una y. Eso es válido y se acepta.



   b. Si pasa del estado inicial al siguiente estado con una x entonces no puede venir una x por lo que debe cambiar de estado con una y. Debe ir al estado de y.

   c. Si pasa del estado inicial al siguiente estado con una y entonces no puede venir una y por lo que debe cambiar de estado con una y. Deber ir al estado de x.

   Respuesta:

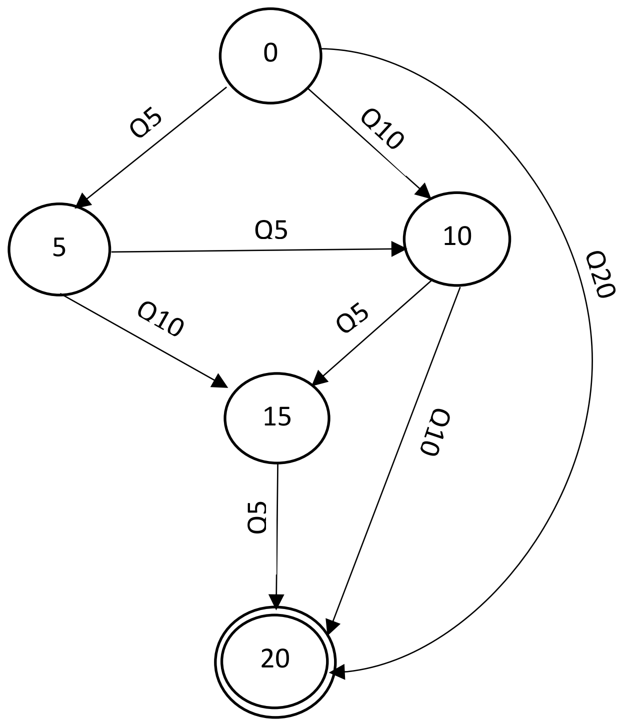


Tarea extra: Realizar la expresión regular y la gramática de este autómata finito.

5. Una empresa de parqueos en la ciudad de Guatemala ha comprado una máquina para el cobro del parqueo y le ha solicitado que la programe para que pueda cobrar y abrir la puerta automáticamente. La maquina solamente acepta billetes de Q. 5, Q. 10 y Q. 20. Además, no da

vuelto en ningún momento y acepta pagos mayores a lo que se debe. Cuando se haya pagado Q20 la puerta debe abrirse. Realizar el autómata para programar la máquina.

Solución: El valor de cada billete es una transición entre estados. Cada estado representa lo que ha pagado hasta ese momento (con cada billete). El estado inicial es el estado de 0 quetzales pagados, dado que no se ha pagado nada. El estado de Q 20. Es el estado de aceptación. De cada estado debe haber tres transiciones con cada billete, para llegar a Q. 20, una de Q. 5, una de Q. 10 y una de Q. 20.



6. Realizar un autómata que dado una secuencia de números binarios {1,0}. 1's y 0's al leer cada digito de izquierda a derecha el número debe ser múltiplo de 3.

   L = {P | P es número binario es múltiplo de 3}

Un número es múltiplo de 3 si su residuo es 0. Es decir, si tenemos un numero M y lo dividimos dentro de 3 el residuo debe quedar 0. Esta operación se le llama modulo. Entonces M es múltiplo de tres si M Modulo 3 = 0.
Cuando se realiza la operación M modulo 3 solo se pueden obtener tres resultados en el residuo 0, 1 o 2.
Así se tiene:

M	M mod 3	M en binario	M	M mod 3	M en binario	M	M mod 3	M en binario	M	M mod 3	M en binario
0	0	0	12	0	1100	24	0	11000	24	0	11000
1	1	1	13	1	1101	25	1	11001	25	1	11001
2	2	10	14	2	1110	26	2	11010	26	2	11010
3	0	11	15	0	1111	27	0	11011	27	0	11011
4	1	100	16	1	10000	28	1	11100	28	1	11100
5	2	101	17	2	10001	29	2	11101	29	2	11101
6	0	110	18	0	10010	30	0	11110	30	0	11110
7	1	111	19	1	10011	31	1	11111	31	1	11111
8	2	1000	20	2	10100	32	2	100000	32	2	100000
9	0	1001	21	0	10101	33	0	100001	33	0	100001
10	1	1010	22	1	10110	34	1	100010	34	1	100010
11	2	1011	23	2	10111	35	2	100011	35	2	100011

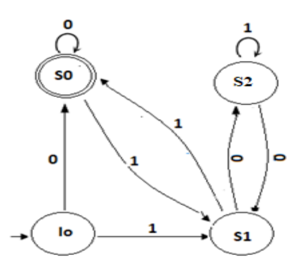
Si se representan los módulos posibles (0, 1 y 2) mediante los estados (So, S1 y S2), y los números binarios 0 y 1 como símbolos de entrada, tendremos la siguiente tabla de transición:

	0	1
So	So	S1
S1	S2	S0
S2	S1	S2

Del estado So con un 0 se va al estado So. Del estado So con un 1 se va al estado S1.
Del estado S1 con un 0 se va al estado S2. Del estado S1 con un 1 se va al estado So.
Del estado S2 con un 0 se va al estado S1. Del estado S2 con un 1 se va al estado S2.

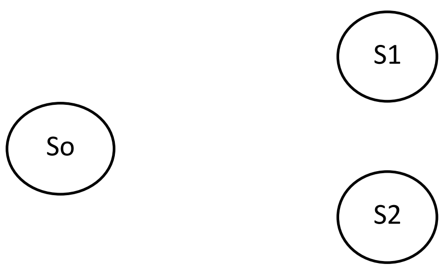
Para evitar que acepte una cadena vacía de entrada (es decir, si no viene ningún número binario) dado que el estado So debe ser el estado de aceptación (se agrega un estado Inicial (Io). Si viene un 0 Debe ir al estado So y si viene un 1 debe ir al estado S1.

Respuesta:

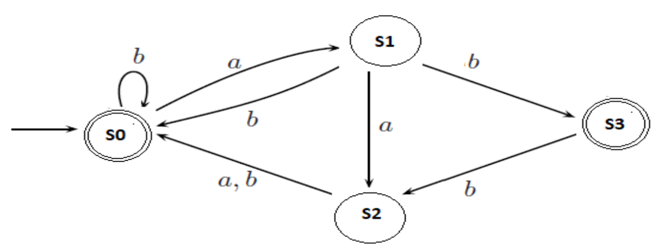


Tarea extra: Evaluar el autómata finto con 3 números que si cumplen con ser múltiplos y 2 número que no cumplen.

7. **Realizar** un autómata que reconozca las palabras formadas por a's y b's donde toda a este entre 2 b's. L = {p | p sea a o b y toda a en p está entre dos b's}
8. Realizar un autómata formado por a's y b's donde no hay dos a's consecutivas.
   L = {w| no hay dos a's consecutivas en w}
9. Realizar un autómata finito con a's y b's que contengan ab o ba con cualquier cantidad de a's o b's al inicio y al final. L = {P | P contiene un ab o ba }



10. Dado el siguiente diagrama de transición



Determinar las cadenas que son aceptadas o no por el autómata. Colocar la secuencia seguida para realizar la evaluación. Por ejemplo,

So → a → S1 → b → S2 se lee:

de So con una **a** se mueve a S1, de S1 con una b se mueve a S2.

para baabaaa la secuencia seria:

So → b → So → a → S1 → a → S2 → b → So → a → S1 →a →s2 →So (se acepta)

Indicar si las siguientes expresiones son aceptan.

a) bab	b) aaba	c) aaaaaab	d) babababab
e) aba	f) abab	g) aabbaa	h) ab
i) aaaaaa	j) bbbbbba	k)aaa	l) ababaabb

a) valida	b) no valida	c) valida	d) valida

e) no valida	f) valida	g) no valida	h) valida
i) valida	j) no valida	k)valida	l) valida