3º curso / 1º cuatr.
Grado Ing. Inform.
Doble Grado Ing.
Inform. y Mat.

Modelos de Computación (MC)

Entrega de ejercicios.
Tema 3

Estudiante (nombre y apellidos): Miguel Lentisco Ballesteros
Grupo de prácticas: A1
Fecha de entrega: 03/11/2017

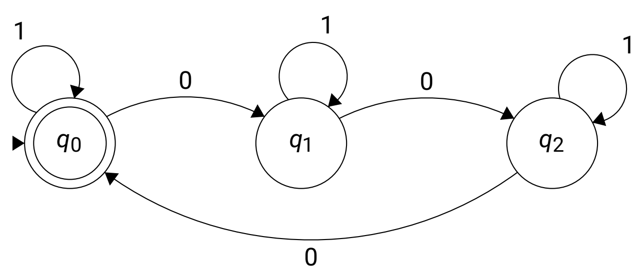
Ejercicios Tema 3

Ejercicio 1. Construir expresiones regulares para los siguientes lenguajes sobre el alfabeto $\{0,1\}$:

1. Palabras en las que el número de símbolos 0 es múltiplo de 3.
2. Palabras que contienencomo subcadena a 1100 ó a 00110.
3. Palabras en las que cada cero forma parte de una subcadena de 2 ceros y cada 1 forma parte de una subcadena de 3 unos.
4. Palabras en las que el número de ocurrencias de la subcadena 011 es menor o igual que el de ocurrencias de la subcadena 110.

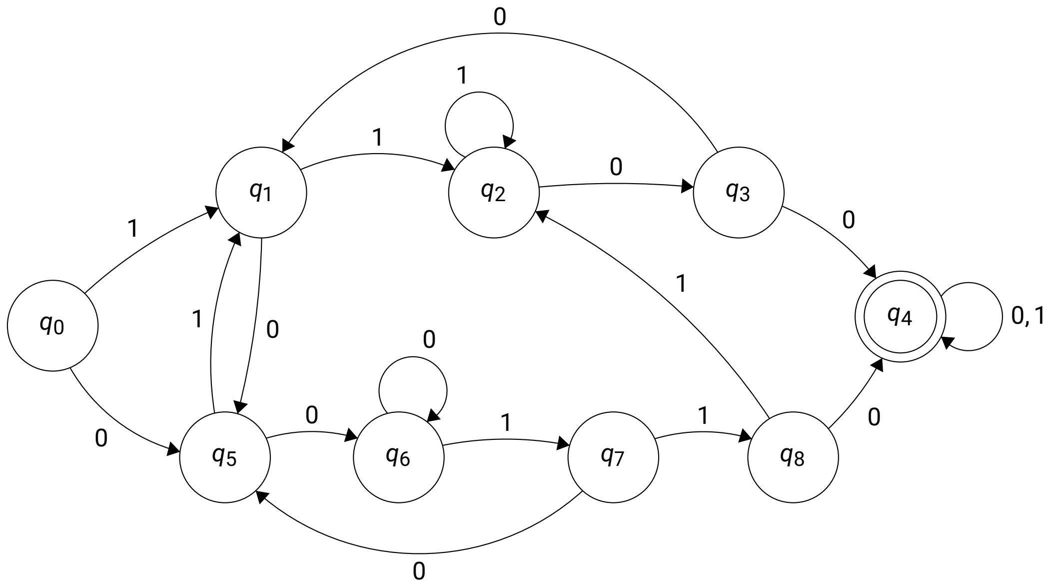
Respuesta.

1. Hacemos el autómata:



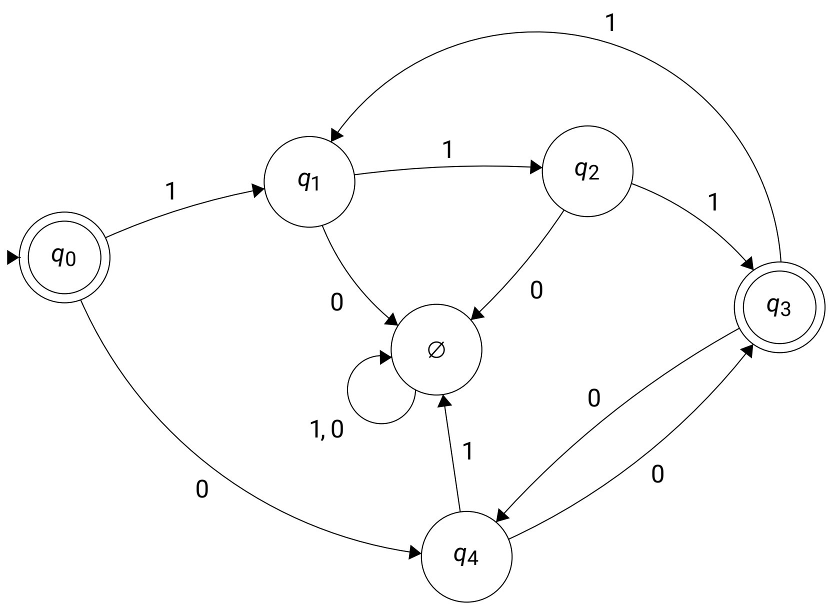
Obteniendo la expresión regular: $(1^*01^*01^*0)^*$.

2. Hacemos el autómata:



La expresión es difícil de sacar, pero a ojo se ve: $(0 + 1)^*(1100 + 00110)(0 + 1)^*$.

3. Autómata:



Expresión regular: $(00 + 111)^*$.

4. Como el autómata es más confuso de obtener, he decidido sacarlo a ojo: $((011(0 + 1)^*110) + (110(0 + 1)^*011) + (0 + 1)^*)^*(110 + (0 + 1)^*)^*$

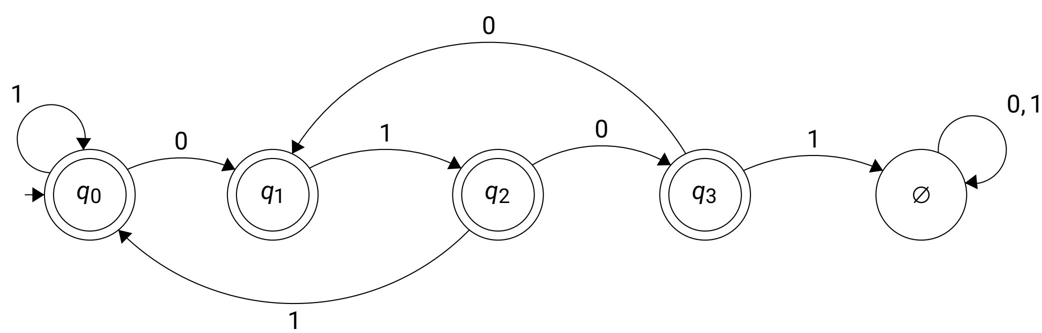
Ejercicio 2. Encuentra para cada uno de los siguientes lenguajes una gramática de tipo 3 que lo genere o un autómata finito que lo reconozca:

- $L_1 = \{u \in \{0,1\}^* : u$ no contiene la subcadena 0101$\}$.
- $L_2 = \{0^i 1^j 0^k : i \geq 1, k \geq 0, i$ impar, k múltiplo de 3, y $j \geq 2\}$.

Diseña el AFD mínimo que reconoce el lenguaje $(L_2 \cap L_1)$

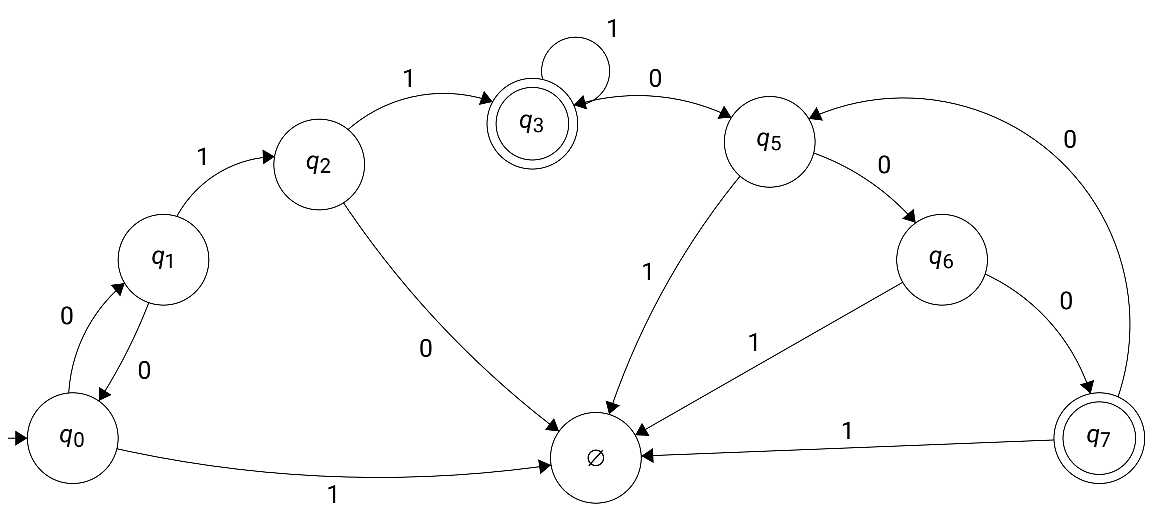
Respuesta. Para L_1 el enunciado estaba mal así que he supuesto que $a = 0, b = 1$. Los autómatas correspondientes serían:

- L_1:



  Y por el algoritmo para hacerlo minimal, vemos que ya es minimal.

- L_2:



  Aplicamos el algoritmo para hacerlo minimal y vemos que ya es minimal.

- Ahora bien, como se ve claramente en L_2 no se aceptan palabras del tipo 0101 ó 1010, por lo tanto $L_2 \subset L_1$ y por tanto $(L_2 \cap L_1) = L_2)$ y como L_2 ya es minimal ya tenemos el AFD minimal de la intersección.

Ejercicio 3. Sea el alfabeto $A = \{0, 1, +, =\}$, demostrar que el lenguaje

$$ADD = \{x = y + z : x, y, z \text{ son números en binario, y } x \text{ es la suma de } y \text{ y } z\}$$

no es regular.

Respuesta. Vamos a ver que tiene infinitas clase de equivalencia, está claro que si fijamos un 1^i = por mucho que yo quiera, lo que le añada al final tiene que ser la suma de dos números que tienen que dar ese 1^i para dos i distintos, las clases de equivalencia son distintas, y por tanto como mínimo tenemos que:

$$\forall i \in \mathbb{N}, \exists [1^i] : [1^i] \neq [1^{1+i}]$$

Ejercicio 4. Si L_1, L_2 son lenguajes sobre el alfabeto A, entonces la mezcla perfecta de estos lenguajes se define como

$$\{w : w = a_1 b_1 \cdots a_k b_k \text{ donde } a_1 \cdots a_k \in L_1, b_1 \cdots b_k \in L_2, a_i, b_i \in A\}$$

Demostrar que si L_1 y L_2 son regulares, entonces la mezcla perfecta de L_1 y L_2 es regular.

Respuesta. Como ambos lenguajes son regulares, podemos crear AFD para cada uno que acepten el lenguaje respectivo. Por tanto la idea es construir otro autómata con la mezcla de los dos, de tal manera que cada vez que introduzco un a_i voy al estado correspondiente según el AFD de L_1, que lo llamaremos q_{a_i}, aqui pasaremos al introducir un b_i al estado correspondiente del AFD de L_2 llamado q_{b_i}, hasta llegar a b_k que llegamos al estado final q_{b_k} que es el que se acepta.

Es decir, la idea es que si $a_1 \cdots a_k$ se acepta en L_1 y $b_1 \cdots b_k$ se acepta en L_2, la idea es ir moviendome del estado al que iría en L_1 y después al estado al que iría en L_2 con b_1, por tanto si llegamos al mismo estado final con el que se reconoce $a_1 \cdots a_k$ y $b_1 \cdots b_k$ entonces la palabra compuesta se acepta en el nuevo autómata, por tanto el lenguaje aceptado mezcla perfecta es regular.

Ejercicio 5. Minimizar el autómata que se da.

Respuesta. Facilmente con el algoritmo para hacerlo minimal vemos que los siguientes estados son indistinguibles: $q_1 \equiv q_4$, $q_2 \equiv q_5$ y $q_3 \equiv q_6$, por tanto el autómata nos quedaría de la siguiente forma: