Modelos de Computación - Práctica 3

Alumno: Raúl Rodríguez Pérez

Grupo: 12

1: Para la realización de este ejercicio primero debemos realizar un pequeño análisis. Debemos ver los números en mod 5, para saber como podemos sacar nuestro AFD. Lo primero que nos damos cuenta es que todos los números múltiplos de 5 (incluyendo el Ø) en mod 5 son Ø, lo que significa que tenemos 5 casos posibles, o lo que es lo mismo, 5 estados.

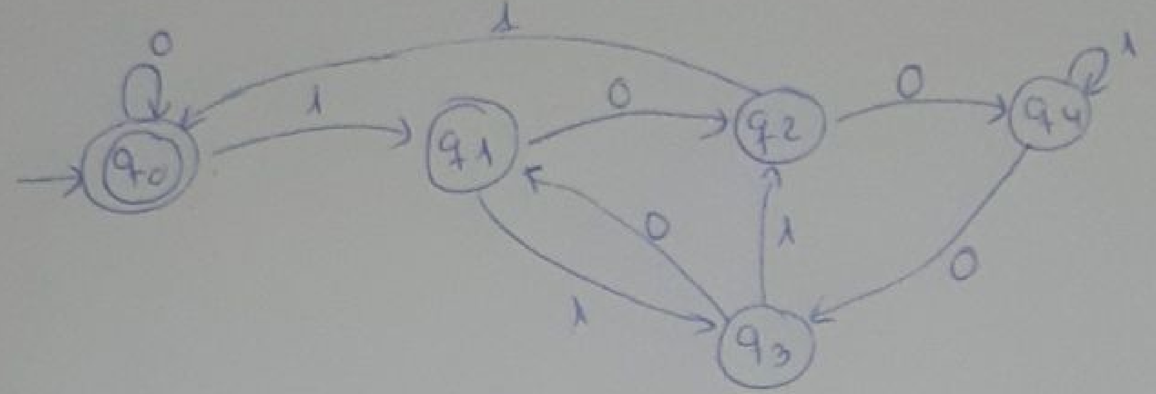
Del mismo modo, sabemos gracias a ejercicios de la relación del tema 2 que; en números binarios, la agregación de un 'Ø' es igual a multiplicarlo por 2, y que, añadir un 1, se traduce en multiplicar por 2 y sumarle 1:

$\underset{(5)}{101} \xrightarrow{x2} \underset{(10)}{1010} \xrightarrow{x2} \underset{(20)}{10100}$ / $\underset{(5)}{101} \xrightarrow{x2+1} \underset{(11)}{1011} \xrightarrow{x2+1} \underset{(23)}{10111}$

Por lo que, una vez sabiendo esto, podemos saber lo que ocurrirá con nuestros 5 estados:

		Añadimos un Ø	añadimos un 1
101 ←	0 mod 5	(1) 1011 (11)	(0) 1010 (10)
001 ←	1 mod 5	(3) 011 (3)	(2) 010 (2)
010 ←	2 mod 5	(0) 101 (5)	(4) 100 (4)
011 ←	3 mod 5	(2) 111 (7)	(1) 110 (6)
100 ←	4 mod 5	(4) 1001 (9)	(3) 1000 (8)

Ahora, con toda la información que tenemos podemos sacar el AFD y la gramática regular por la izda:



$$
\text{GR izda}
\begin{cases}
0 & S \longrightarrow Y1 \mid S0 \mid \varepsilon \\
1 & X \longrightarrow S1 \mid Z0 \\
2 & Y \longrightarrow X0 \mid Z1 \\
3 & Z \longrightarrow W0 \mid X1 \\
4 & W \longrightarrow Y0 \mid W1
\end{cases}
$$

* Leyenda de la gramática

$q_0 \rightarrow S$ $\quad q_3 \rightarrow Z$

$q_1 \rightarrow X$ $\quad q_4 \rightarrow W$

$q_2 \rightarrow Y$

* Destacar que la gramática regular la he sacado invirtiendo el autómata, ya que la gramática que se genera (de) mi autómata propuesto da lugar a una gramática regular por la derecha. Esta inversión se realiza invirtiendo las flechas de los estados.

- Vamos a proceder a sacar la expresión regular por medio de Lema de Arden:

$$
\begin{cases}
q_0 = 0q_0 + 1q_1 + \varepsilon \quad ⊛ \\
q_1 = 0q_2 + 1q_3 \longrightarrow q_1 = 0(01^*01)^*01^*00q_1 + 0(01^*01)^*1q_0 + 10q_1 + 11(01^*01)^*01^*00q_1 + 01^*01q_2 + 1q_0 \ldots \\
q_2 = 0q_4 + 1q_0 \longrightarrow 01^*(0q_3) + 1q_0 \longrightarrow q_2 = 01^*0(0q_1 + 1q_2) + 1q_0 = 01^*00q_1 + 01^*01q_2 + 1q_0 \xrightarrow{LA} q_2 = (01^*01)^*(01^*00q_1 + 1q_0) \\
q_3 = 0q_1 + 1q_2 \longrightarrow q_3 = 0q_1 + 1(01^*01)^*(01^*00q_1 + 1q_0) \\
q_4 = 0q_3 + 1q_4 \xrightarrow{LA} q_4 = 1^*(0q_3)
\end{cases}
$$

$\longrightarrow q_1 = (0(01^*01)^*01^*00 + 11(01^*01)^*01^*00)q_1 + 0(01^*01)^*1q_0 + 11(01^*01)^*1q_0 \ldots$

⊛ $+ 11(01^*01)^*1q_0 \rightarrow q_1 = (0(01^*01)^*01^*00 + 11(01^*01)^*01^*00)q_1 + 0(01^*01)^*1q_0 + 11(01^*01)^*1q_0) \rightarrow ((0+11)$

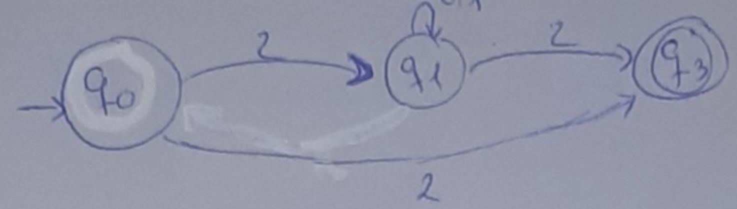
$\xrightarrow{LA} q_1 = (0(01^*01)^*01^*00 + 11(01^*01)^*01^*00)^*(0(01^*01)^*1q_0 + 11(01^*01)^*1q_0)$

$(01^*01)^*01^*00))^*((0+11)(01^*01)^*1)q_0 + \varepsilon \xrightarrow{LA}$ Pag siguiente

⊛ $q_0 = 0q_0 + 1((0+11)((01^*01)^*01^*00))^*((0+11)(01^*01)^*1)q_0 + \varepsilon$

$$\boxed{q_0 = (0 + 1((0+11)((01^*01)^*01^*00))^*((0+11)(01^*01)^*1))^*}$$

2.-



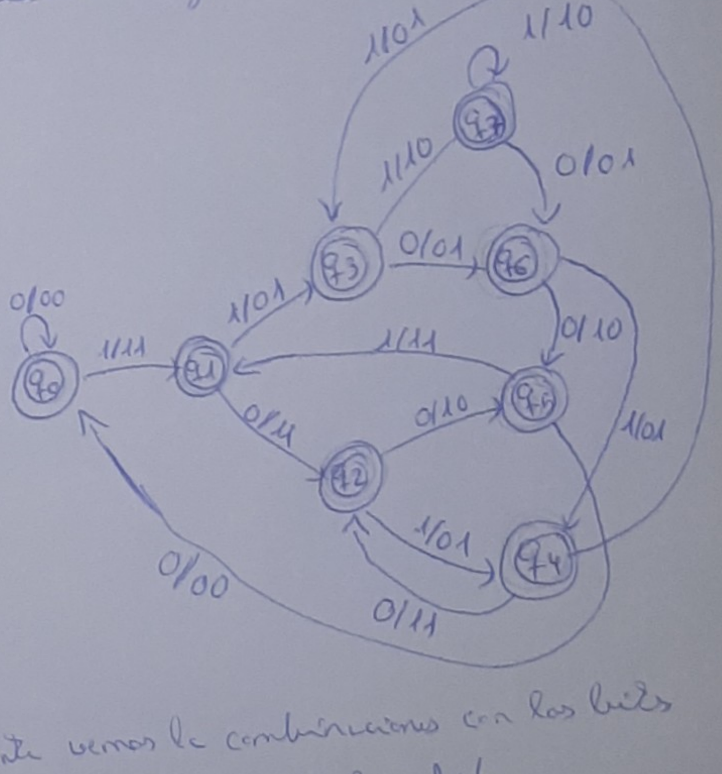
No hay mucho que añadir en este ejercicio, forzamos a que lo primero que se escriba sea un 2, con el paso de $q_0 \to q_1$. Tras esto en q_1 podemos añadir la combinación de 0 y 1 que queramos, pero si añadimos un 2, pasaremos a q_3 y finalizaremos. Además, he añadido la transición de $q_0 \to q_3$ donde añade un solo 2 y termina, puesto que pienso que es una palabra del lenguaje ya que empieza/termina por 2.

3.- En primer lugar, para realizar este ejercicio, primero voy a completar la tabla donde reflejamos la codificación según los dos bits leídos anteriormente y el actual:

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

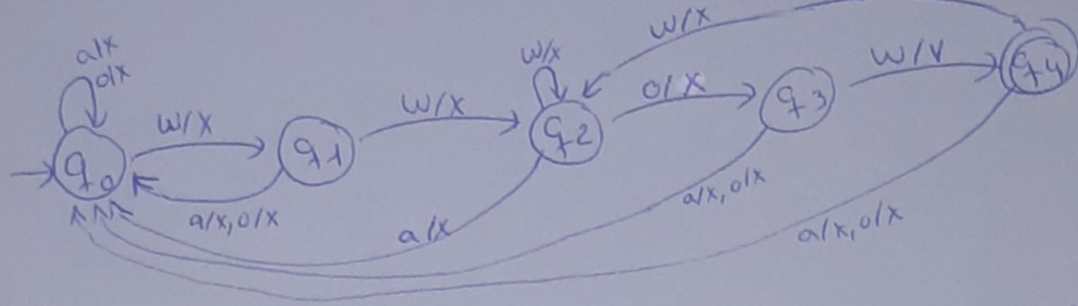
Bit actual

estado	0	1
q_0	00	11
q_1	11	00
q_2	10	01
q_3	01	10
q_4	00	11
q_5	11	00
q_6	10	01
q_7	01	10



Para crear la máquina de Mealy, simplemente vemos la combinaciones con los bits anteriormente leídos y el actual, y fijármos en la tabla para ver la salida.

4.- He decidido diseñar una máquina de Mealy para resolver este ejercicio. Primero antes que nada, debemos entender bien lo que piden; el enunciado nos dice que hay que tener una 'V' como salida siempre y cuando se lea la cadena 'wwow'. Pero incluso, dicha secuencia puede estar solapada, es decir, si se lee la cadena wwowwow, se debería de mostrar el estado encendido en la 3ª y en la última 'w'. Por lo tanto mi máquina propuesta es:



Para explicarlo brevemente, (~~el act~~) la máquina de mealy consigue solventar todas las combinaciones en las que no se escriba la cadena 'wwow'. Es decir, la transiciones de q1-q0, q2-q0, q3-q0 y q4-q0, representan las excepciones en donde ~~cuando~~ se esta escribiendo la ~~pala~~ cadena 'wwow' pero añadimos otra letra que jode dicha cadena, por lo que volvemos a empezar sin encender nada. Además cabe destacar que la transición q4-q2, ~~se~~ se rediseña para explicar el solapamiento descrito previamente, es decir, si ya hemos leído la cadena 'wwow', solo si añadimos una 'w', ~~pero~~ solo nos hace falta escribir la cadena 'ow' para volver a encender el estado.