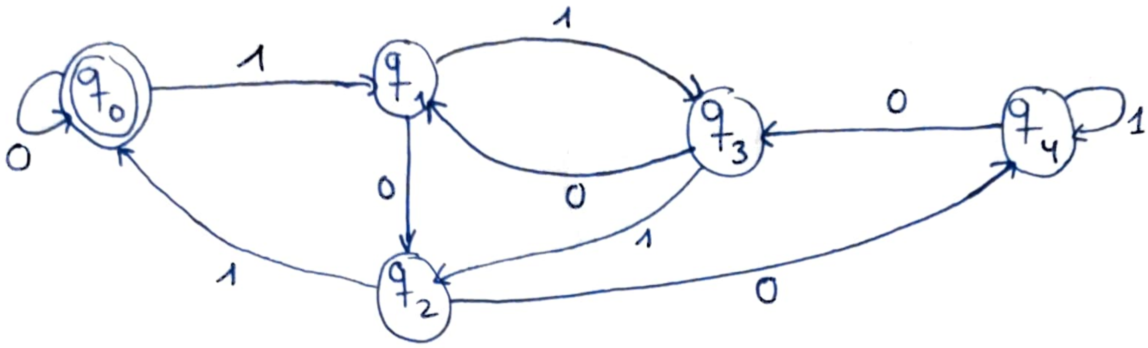
RAÚL CASTRO MORENO 3°A - A2

Ejercicio 1



La idea para realizar ~~ese~~ este autómata, es que el estado terminal es $0 \pmod 5$, es decir, que cada número está en un miembro de $\mathbb{Z}_5$.

En el autómata tenemos 5 estados, q_0, q_1, q_2, q_3, q_4; que representan cada uno el miembro de $\mathbb{Z}_5$ en el que se encuentra, sabiendo que van desde $0 \pmod 5$ a $4 \pmod 5$ respectivamente.

También es necesario saber que añadir un 0 al final en binario, es equivalente a multiplicar por 2, y añadir un 1, es equivalente a multiplicar por 2 y sumar 1.

Si estamos en q_0 ($0 \pmod 5$), al añadir 0 multiplicaríamos por 2 y por lo tanto, $2 \cdot 0 \pmod 5 = 0 \pmod 5$, siguiendo en q_0, y si añadimos un 1, tendríamos $2 \cdot 0 + 1 \pmod 5 = 1 \bmod 5$ pasando así al estado q_1.

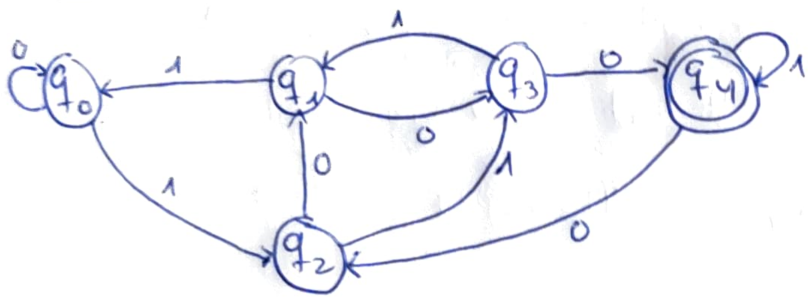
Ahora es aplicar lo mismo con todos, con q_1 por ejemplo tendríamos al añadir 0, $2 \cdot 1 \pmod 5 = 2 \pmod 5$ pasando a q_2 y si añadiéramos un 1, $2 \cdot 1 + 1 \pmod 5 = 3 \pmod 5$, pasando al estado q_3. Con q_2 como se ve en el autómata, pasaríamos a q_0 al añadir 1 debido a $2 \cdot 2 + 1 \pmod 5 = 0 \bmod 5$, y con el 0, $2 \cdot 2 \pmod 5 = 4 \pmod 5$, siendo el estado q_4 ahora.

En el estado q_3, añadir 0 supondría $2\cdot 3 \ (\mathrm{mod}\ 5) = 6 \ (\mathrm{mod}\ 5) = 1 \ (\mathrm{mod}\ 5)$, por ello pasa al estado q_1, y al añadir 1, $2\cdot 3+1 \ (\mathrm{mod}\ 5) = 7 \ (\mathrm{mod}\ 5) = 2 \ (\mathrm{mod}\ 5)$, llegando así al estado q_2.

Por último, en q_4, añadir un 0 sería $2\cdot 4 \ (\mathrm{mod}\ 5) = 8 \ (\mathrm{mod}\ 5) = 3 \ (\mathrm{mod}\ 5)$ pasando así al estado q_3, y al añadir un 1, se mantendría en el estado q_4 debido a que $2\cdot 4+1 \ (\mathrm{mod}\ 5) = 9 \ (\mathrm{mod}\ 5) = 4 \ (\mathrm{mod}\ 5)$.

Para obtener la gramática regular por la izquierda, aplicamos los siguientes pasos:

Paso 1: Revertir autómata



Paso 2: Sacamos GLD

$q_0 \to 0q_0 \mid 1q_2$

$q_1 \to 1q_0 \mid 0q_3$

$q_2 \to 0q_1 \mid 1q_3$

$q_3 \to 0q_4 \mid 1q_1$

$q_4 \to 1q_4 \mid 0q_2 \mid 1$

Paso 3: Invertimos transiciones

$q_0 \to q_0 0 \mid q_2 1$

$q_1 \to q_0 1 \mid q_3 0$

$q_2 \to q_1 0 \mid q_3 1$

$q_3 \to q_4 0 \mid q_1 1$

$q_4 \to q_4 1 \mid q_2 0 \mid 1$

(Aquí comprobamos que en el estado terminal, no termina con la cadena vacía, sino con 1 (en el autómata normal con 0) debido a que la cadena vacía no es múltiplo de 5)

Para obtener la expresión regular; primero sacamos el sistema de ecuaciones.

$$\begin{cases} q_0 = 0q_0 + 1q_1 + 0 \\ q_1 = 1q_3 + 0q_2 \\ q_2 = 1q_0 + 0q_4 \\ q_3 = 0q_1 + 1q_2 \\ q_4 = 1q_4 + 0q_3 = 1^*0q_3 \end{cases}$$

Empezamos aplicando el lema de Arden en q_4 y sustituimos en q_2

$q_2 = 1q_0 + 01^*0q_3$; Ahora sustituimos q_3 en q_2

$q_2 = 1q_0 + 01^*0(0q_1 + 1q_2) = 1q_0 + 01^*00q_1 + 01^*01q_2$;

Ahora aplicamos otra vez el lema de Arden.

$q_2 = (01^*01)^*1q_0 + (01^*01)^*01^*00q_1$; Sustituimos ahora q_3 en q_1.

$q_1 = 1(0q_1 + 1q_2) + 0q_2 = 10q_1 + 11q_2 + 0q_2$; Sacamos factor común y sustituimos q_2 en q_1.

$q_1 = 10q_1 + (11+0)q_2 = 10q_1 + (11+0)\left((01^*01)^*1q_0 + (01^*01)^*01^*00q_1\right) =$

$= 10q_1 + (11+0)(01^*01)^*1q_0 + (11+0)(01^*01)^*01^*00q_1$; Reducimos y Arden

$q_1 = (10 + (11+0)(01^*01)^*01^*00)q_1 + (11+0)(01^*01)^*1q_0 =$

$= (10 + (11+0)(01^*01)^*01^*00)^*(11+0)(01^*01)^*1q_0$; Sustituimos finalmente en q_0.

$q_0 = 0q_0 + 1(10 + (11+0)(01^*01)^*01^*00)^*(11+0)(01^*01)^*1q_0 + 0$;

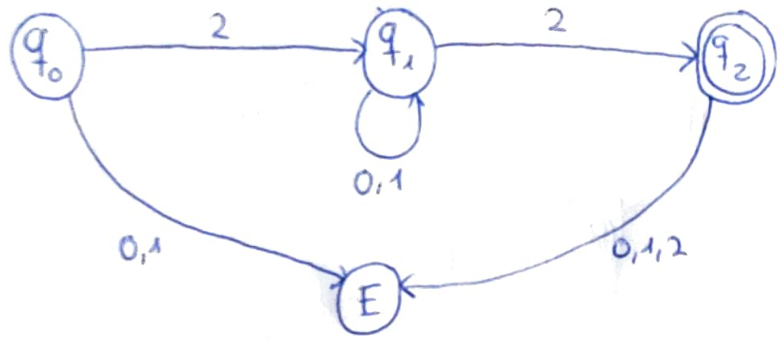
Reducimos y Arden finalmente

$q_0 = (0 + 1(10+(11+0)(01^*01)^*01^*00)^*(11+0)(01^*01)^*1)q_0 + 0$

$q_0 = \underline{(0+1(10+(11+0)(01^*01)^*01^*00)^*(11+0)(01^*01)^*1)^*0}$

→ Esta es la expresión regular que describe este lenguaje

Ejercicio 2



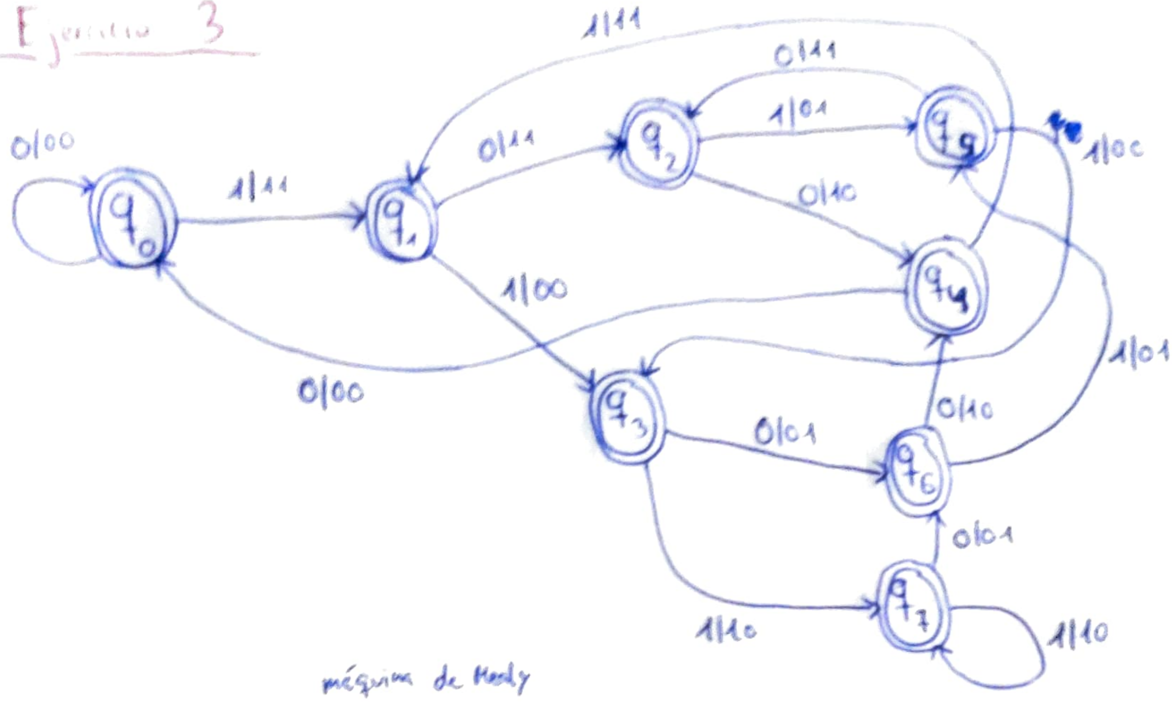
Para este autómata he usado 3 estados, q_0, q_1 y q_2.

q_0 es el estado inicial, por el cual solo puede avanzar al estado q_1, si se añade un 2, debido a la restricción del enunciado de empezar la palabra por 2. En caso de poner un 0, o un 1, se iría al estado de error puesto que la palabra no empezaría por 2.

En q_1, es el estado ~~de~~ intermedio donde podemos añadir todos los ~~0s~~ 0's y 1's que queramos, y en caso de añadir un 2, pasaríamos al estado q_3.

Finalmente en q_3, que es ~~el~~ un estado terminal del autómata, ya tendríamos la palabra que ha empezado por 2 y ha añadido un 2 al final sin tener ningún 2 entre medias, cumpliendo así las restricciones, y es por ello, que añadir 0, 1 o 2, llevaría al estado de error puesto que tendríamos un 2 entre medias, incumpliendo así la restricción.

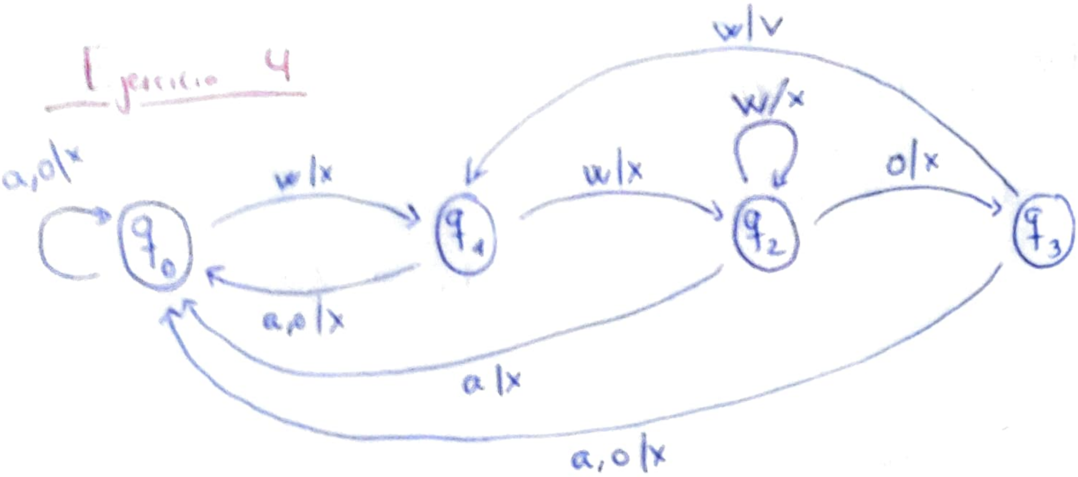
Ejercicio 3



Para realizar esta ~~[illegible]~~ máquina de Mealy, he necesitado 8 estados que van de q_0 a q_7, y son todos terminales. Cada uno de los estados representa una de las posibles combinaciones que se pueden hacer con los 3 bits: actual, anterior y segundo anterior. En la siguiente tabla podemos ver a que combinacion corresponde cada estado, y que codificación le corresponde.

SEGUNDO BIT ANTERIOR	BIT ANTERIOR	BIT ACTUAL	ESTADO	CODIFICACIÓN
0	0	0	q_0	00
0	0	1	q_1	11
0	1	0	q_2	11
0	1	1	q_3	00
1	0	0	q_4	10
1	0	1	q_5	01
1	1	0	q_6	01
1	1	1	q_7	10

Ejercicio 4



Para realizar esta máquina de Mealy, utilizaremos 4 estados, de q_0 a q_3. q_0 es el estado inicial donde podemos añadir todas las 'a' y 'o' que se quieran, y en el momento que añadimos una w pasaremos a q_1. En q_1 tendríamos la primera 'w' de la subcadena 'wwow', por lo que si añadimos una 'a' ou una 'o', volveríamos al estado inicial q_0. Si añadiéramos una 'w' pasaríamos a q_2. En q_2 tenemos las dos primeras 'w' de la subcadena, si añadiéramos 'w', seguiríamos en q_2 ~~porque~~ porque tendríamos las 2 primeras 'w', añadir una 'a' devolvería al estado inicial q_0 y añadir una 'o', pasaría a q_3. En q_3, tendríamos 'wwo', faltando una 'w' por introducir, y en ese caso iría a q_1, esto es debido a la restricción de que se pueden leer cadenas solapadas como puede ser 'wwowwow', siendo la 'w' que va de q_3 a q_1 la primera de la segunda subcadena solapada. ~~En caso de que~~ Pues al añadir la última 'w' se ~~para~~ enciende el led verde, y al añadir 'a' o 'o', devolvería a q_0.