

Modelos de computación (2015-2016)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA



Relación 2

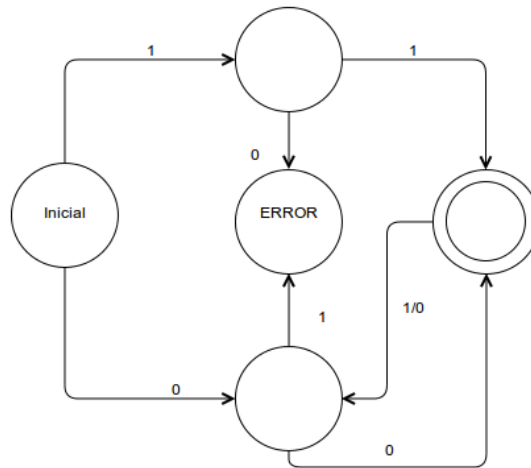
José Carlos Martínez Velázquez

26 de octubre de 2015

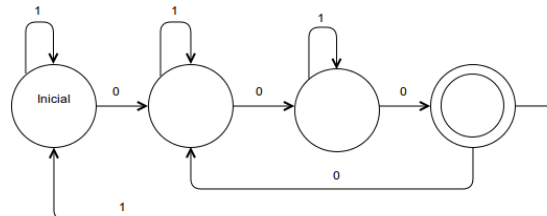
1. Dibujar los AFDs que aceptan los siguientes lenguajes con alfabeto 0,1:

Para resolver los diferentes apartados de éste ejercicio se ha utilizado una metodología que consiste en buscar el/los casos de éxito directamente y luego introducir poco a poco las restricciones del problema.

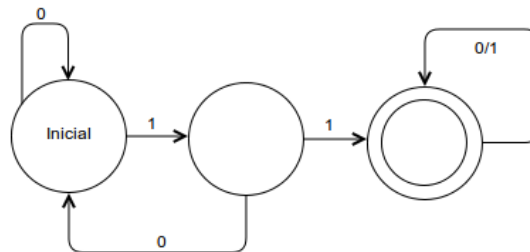
a) El lenguaje $\{11,00\}$



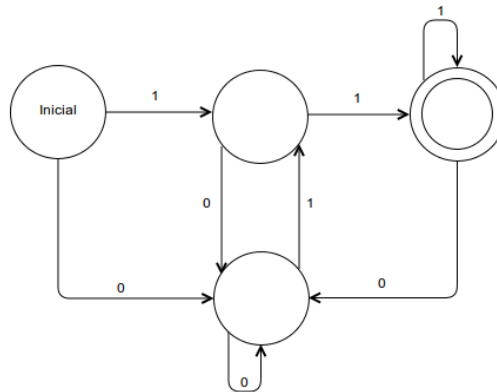
b) El lenguaje formado por las cadenas donde el número de ceros es divisible por 3.



c) El lenguaje de las palabras que contienen la subcadena 11.

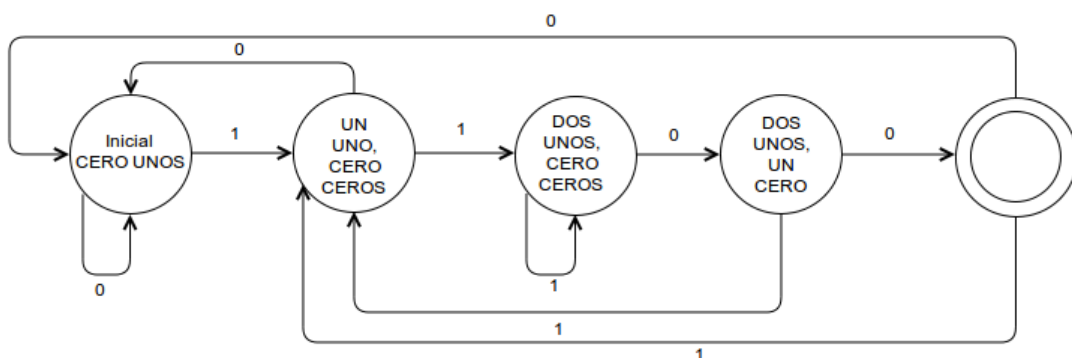


- d) El lenguaje de las palabras que empiezan o terminan (o ambas cosas) en 11.

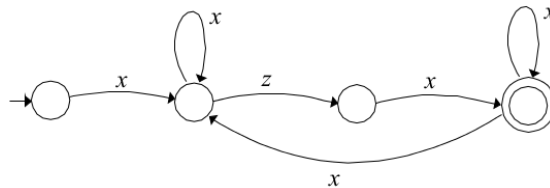


2. Diseñar un AFD que, dada una cadena binaria, detecte la palabra clave 1100, tantas veces como aparezca en la secuencia de entrada (el AFD debe llegar al estado final cada vez que detecte una ocurrencia de la palabra clave, no puede quedarse en el estado final tras encontrar la primera secuencia 1100, si sigue habiendo números detrás). Ejemplo: con la entrada 111100101100111001, el AFD debe pasar tres veces por el estado final.

Con el nombre de cada estado se ha intentado ser lo suficientemente gráfico para que se entienda que está pasando en cada iteración.



3. Dado el alfabeto x, z , queremos construir un autómata finito M tal que $L(M)$ sea el lenguaje formado por las cadenas que contienen al menos una z , y cada z está inmediatamente precedida y seguida por una x . ¿Es correcta la siguiente solución? Razone la respuesta. Si considera que la solución no es correcta, proponga un AFD alternativo.



Demostrar que éste diagrama no es correcto es más sencillo que demostrar que sí lo es. Bastaría con encontrar un caso de palabra que no sea válido o una palabra válida que no pertenezca al lenguaje se pretende definir o bien un fallo en el diagrama. Vamos a optar por la última opción. Cuando se llega a estado final, sólo se puede cambiar cuando llega un símbolo 'x', pero al ocurrir esto, se vuelve a activar el estado final y el segundo estado. Si el segundo estado se activa más de dos veces, se podrían generar dos símbolos 'z' seguidos. Para evitar ésto se propone el siguiente autómata finito determinista:

