

Ejercicios Autómatas Finitos

75.579 - Autómatas y gramáticas
Grado en Ingeniería Informática

Estudios de Informática, Multimedia y Telecomunicación



Ejercicio 1

Disponemos de un sistema electrónico para transmitir señales digitales binarias de longitud variable, pero se ha detectado que si se envían secuencias de más de 3 ceros seguidos, o de más de 3 unos seguidos, hay problemas de sincronización entre el emisor y el receptor, por lo que la información no llega de forma correcta.

Este problema comporta que deba utilizarse un sistema de codificación de la información, que determina un cierto lenguaje L, que evite que se generen secuencias de más de 3 bits iguales (más de 3 ceros o más de 3 unos seguidos) cuando transferimos secuencias de datos.

Este lenguaje L, pues, nos permite asegurar que no se generarán secuencias de más de 3 bits iguales. La codificación que define el lenguaje que se quiere utilizar estará formada por códigos de 5 dígitos binarios, sin utilizar las combinaciones que tengan 3 dígitos iguales al inicio, ni tampoco, aquellas combinaciones que tengan 2 dígitos iguales al final.

Para poder verificar que los códigos recibidos son correctos se quiere construir un autómata finito determinista F que permita aceptar secuencias de códigos de este lenguaje L, de 5 dígitos binarios cada uno, y que no generen secuencias de más de 3 bits iguales.

Por ejemplo, una secuencia de 3 códigos que podría recibir el autómata F sería la siguiente: 00100 00101 11101

Para construir este autómata seguiremos los siguientes pasos:

1.1)

1.1.1)

De todos estos códigos binarios de longitud 5, marcad con una x los pertenecientes al lenguaje L, es decir, aquellos códigos que no tengan 3 dígitos iguales al inicio, y que no tengan 2 dígitos iguales al final.

00000		01000		10000		11000	
00001		01001		10001		11001	
00010		01010		10010		11010	
00011		01011		10011		11011	
00100		01100		10100		11100	
00101		01101		10101		11101	
00110		01110		10110		11110	
00111		01111		10111		11111	



1.1.2)

Dad como máximo 3 códigos de cada uno de estos conjuntos definidos a partir de códigos binarios en base a los siguientes criterios:

- Indicar las subpalabras impropias de la palabra 10101:
- $\{ w \in L \mid w = w^R \}$:
- $\{ w \in L \mid |w| = 3 \}$:
- $\{ w \in L \mid |w|_1 = 2 \}$:
- $\{ w_1 \in L \mid w_2 \in L \mid w_1 \neq w_2 \wedge (w_1 \cdot w_2) \in L^C \}$ (es suficiente con dar 3 parejas de palabras de la forma siguiente: $(w_1 \cdot w_2)$) :

1.2)

A continuación se da la secuencia de estados, desde el estado inicial hasta que se hayan procesado todos los dígitos del código y se indica también si el código ha sido aceptado o no por el autómata A3 que reconoce códigos **de longitud 3** que no empiecen con 3 dígitos binarios iguales.

Autómata A3

$Q = \{ 20, 21, 22, 23, 24, 25, 26, 27 \}$

$\Sigma = \{ 0, 1 \}$

$q_0 = 20$

$F = \{ 23 \}$

Código: (estado inicial) - ... - (estados intermedios) - ... - [(estado final)/(estado pozo)] ([NO] aceptado)

- | | | |
|--------------|--------------------------|----------------|
| a) λ | : 20 | (No aceptado). |
| b) 0 | : 20 - 21 | (No aceptado). |
| c) 00 | : 20 - 21 - 22 | (No aceptado). |
| d) 000 | : 20 - 21 - 22 - 27 | (No aceptado). |
| e) 001 | : 20 - 21 - 22 - 23 | (aceptado). |
| f) 0000 | : 20 - 21 - 22 - 27 - 27 | (No aceptado). |
| g) 010 | : 20 - 21 - 26 - 23 | (aceptado). |
| h) 101 | : 20 - 24 - 26 - 23 | (aceptado). |
| i) 1101 | : 20 - 24 - 25 - 23 - 27 | (No aceptado). |
| j) 11 | : 20 - 24 - 25 | (No aceptado). |

1.2.1)

Dad el DFA del autómata A3.



1.2.2)

Indicad la secuencia de estados y si los códigos siguientes son aceptados o no por el autómata antes definido:

Código: (estado inicial) - ... - (estados intermedios) - ... - [(estado final)/(estado pozo)] ([NO] aceptado)

- k) 010 :
- l) 110 :
- m) 001 :
- n) 111 :
- o) 1 :

1.3)

A continuación se da el autómata B, que reconoce códigos **de cualquier tamaño** que no terminen con 2 dígitos binarios iguales.

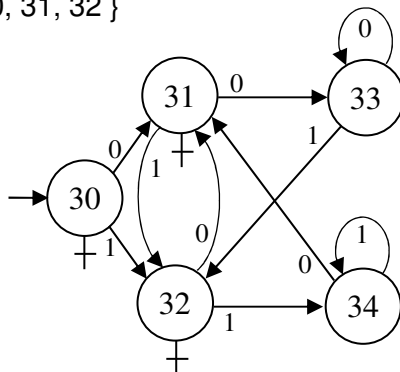
Autómata B

$Q = \{ 30, 31, 32, 33, 34 \}$

$\Sigma = \{ 0, 1 \}$

$q_0 = 30$

$F = \{ 30, 31, 32 \}$



1.3.1)

Indicad la secuencia de estados y si los códigos siguientes son aceptados o no por el autómata antes definido:

Código: (estado inicial) - ... - (estados intermedios) - ... - [(estado final)/(estado pozo)] ([NO] aceptado)

- a) λ :
- b) 0 :
- c) 00 :
- d) 000 :
- e) 010 :
- f) 111 :
- g) 001 :
- h) 110 :



1.4

A continuación se da el autómata C2 que reconoce códigos de longitud 2.

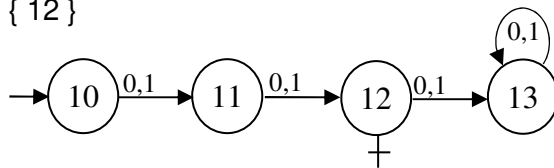
Autómata C2

$Q = \{ 10, 11, 12, 13 \}$

$\Sigma = \{ 0, 1 \}$

$q_0 = 10$

$F = \{ 12 \}$



1.4.1

Dad la tabla de transiciones que resulta de hacer la intersección por producto cartesiano del autómata B y del autómata C2 para obtener un nuevo autómata que acepte códigos de longitud 2 que no tengan 2 dígitos iguales.

1.4.2

Dad el DFA (Autómata Finito Determinista) **mínimo**, que denominaremos B2, equivalente al autómata finito determinista correspondiente a la tabla de transiciones anterior. Se deben mostrar todos los pasos seguidos para obtener el autómata mínimo y justificar el porqué es mínimo.

1.5

A partir del autómata A3 obtenido en el punto 1.2 y el autómata B2 obtenido en el punto 1.4, queremos obtener el autómata D que acepte Códigos de 5 dígitos binarios, sin utilizar las combinaciones que tengan 3 dígitos iguales al inicio, ni tampoco las combinaciones que tengan 2 dígitos iguales al final. Para obtenerlo hay que concatenar estos dos autómatas.

1.5.1

Realizad la concatenación del autómata A3 y B2 para obtener el autómata D

1.5.2

Obtened el DFA D equivalente del autómata D obtenido en el apartado anterior.



1.6

Finalmente, para obtener el autómata finito determinista F que permita aceptar secuencias de códigos de este lenguaje L de 5 dígitos binarios cada uno, que no tengan **secuencias** de más de 3 bits iguales, hay que hacer el cierre positivo de Kleene.

1.6.1

Haced el cierre positivo de Kleene del DFA D obtenido en el punto anterior para obtener el autómata F .

1.6.2

Obtened la tabla de transiciones del DFA F equivalente del autómata F obtenido en el apartado anterior.

1.7

1.7.1

Si quisiéramos hacer un autómata que permita aceptar secuencias de códigos de este lenguaje L , pero de 4 dígitos binarios cada uno que no tengan secuencias de más de 3 bits iguales, ¿podríamos hacerlo de la misma manera que lo hemos hecho para códigos de 5 dígitos? Explica brevemente por qué (no es necesario obtener ningún autómata, ni hacer ningún cálculo, sólo indicar la estrategia y qué la puede condicionar).

1.7.2

Si quisiéramos hacer un autómata que permita aceptar secuencias de códigos de este lenguaje L , pero de 6 o más dígitos binarios cada uno que no tengan secuencias de más de 3 bits iguales, ¿podríamos hacerlo de la misma manera que lo hemos hecho para códigos de 5 dígitos? Explica brevemente por qué (no es necesario obtener ningún autómata, ni hacer ningún cálculo, sólo indicar la estrategia y qué la puede condicionar).



Ejercicio 2

Dado el autómata finito determinista $A = (Q, \Sigma, q_0, F, \delta)$, donde $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $\Sigma = \{0, 1\}$, $q_0=1$, $F=\{9\}$ i δ definida per la tabla siguiente:

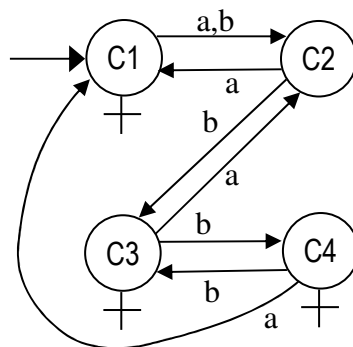
δ	0	1
$\rightarrow 1$	2	3
2	4	6
3	6	5
4	10	8
5	7	10
6	7	8
7	10	9
8	9	10
+9	2	3
10	10	10

Determinad una expresión regular que represente el lenguaje L reconocido por este autómata definiendo los lenguajes asociados a cada estado del autómata.

Ejercicio 3

Dad el sistema de ecuaciones que describe el autómata dado. Aplicando el Lema de Arden encuentra la expresión regular que describe el lenguaje aceptado por el autómata.

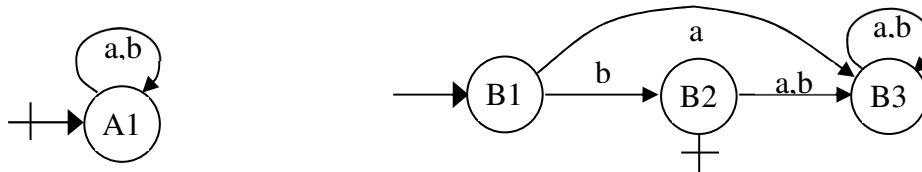
.





Ejercicio 4

Se quiere construir un **autómata finito determinista (DFA) y mínimo** que acepte todas las palabras sobre el alfabeto $\{a, b\}$ del lenguaje representado por la expresión regular $[(a + b)^* bb +]$ a partir de los autómatas siguientes:



Contestad los apartados siguientes:

a) Indicad qué lenguaje reconoce cada autómata (dad la expresión regular):

Autómata A:

Autómata B:

b) ¿Qué operaciones hay que hacer sobre los autómatas dados para obtener el autómata que acepte el lenguaje representado por la expresión regular $[(a + b)^* bb +]$?

Dibujad el autómata obtenido

¿Las palabras b , bb , bbb , bbb son aceptadas por este autómata?

c) Dad la tabla de transiciones del autómata finito **determinista** (DFA) que reconoce el lenguaje representado por la expresión regular $[(a + b)^* bb +]$

d) Dad el DFA **mínimo** que permita reconocer el lenguaje representado por la expresión regular $[(ab + b)(ab + b)^*(a + \lambda)]$, a partir de la tabla de transiciones del DFA obtenido en el apartado anterior. Para simplificar la minimización podéis re-etiquetar los Estados de la tabla de transiciones

e) Dad el sistema de ecuaciones que describe el DFA **mínimo** obtenido. Aplicando el Lema de Arden encuentrad la expresión regular que describe el lenguaje aceptado por el autómata mínimo. Indicad si la expresión regular obtenida es equivalente a la expresión regular dada $[(a + b)^* bb +]$.