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	X	10001	X	11001	X
00010		01010	X	10010	X	11010	X
00011		01011		10011		11011	
00100		01100		10100		11100	
00101	X	01101	X	10101	X	11101	
00110	X	01110	x	10110	X	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:

- Indicad las subpalabras impropias de la palabra 10101: { λ,10101}
- { w ϵ L | w = w^R } : { 01110, 10001, 10101 }
- { w ϵ L | |w| = 3 } : { Ø }
- { w ϵ L | $|w|_1$ = 2 } : { 01010, 00110, 00101}
- { w1 ϵ L i w2 ϵ L | w1 ≠ w2 i (w1·w2) ϵ L^C } (es suficiente con dar 3 parejas de palabras de la forma siguiente: (w1·w2)) : {01010 00110, 00101 01110, 10001 10101}

1.2)

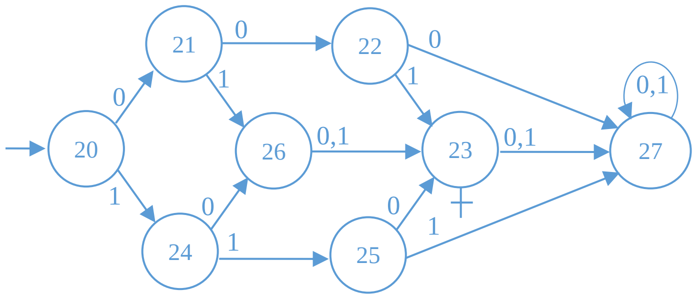
A continuación se da la secuencia de estados, desde el estado inicial hasta que se hayan procesado todos los dígitos del código y se indica también si el código ha sido aceptado o no por el autómata A3 que reconoce códigos **de longitud 3** que no empiecen con 3 dígitos binarios iguales.

Autómata A3
Q = { 20, 21, 22, 23, 24, 25, 26, 27 }
∑ = { 0, 1 }
q0 = 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:

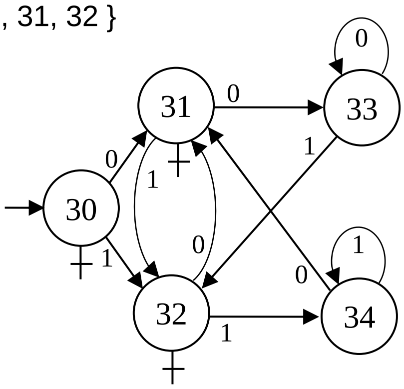
Código: (estado inicial) - ... - (estados intermedios) - ... - [(estado final)/(estado pozo)] ([NO] aceptado)

k) 010 : 20 - 21 - 26 - 23 (aceptado).
l) 110 : 20 - 24 - 25 - 23 (aceptado).
m) 001 : 20 - 21 - 22 - 23 (aceptado).
n) 111 : 20 - 24 - 25 - 27 (No aceptado).
o) 1 : 20 - 24 (No aceptado).

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 }
∑ = { 0, 1 }
q0 = 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:

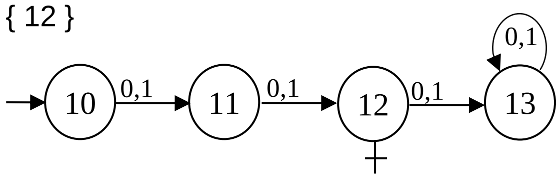
Código: (estado inicial) - ... - (estados intermedios) - ... - [(estado final)/(estado poz0)] ([NO] aceptado)

a) λ	: 30	(aceptado).
b) 0	: 30 - 31	(aceptado).
c) 00	: 30 - 31 - 33	(No aceptado).
d) 000	: 30 - 31 - 33 - 33	(No aceptado).
e) 010	: 30 - 31 - 32 - 31	(aceptado).
f) 111	: 30 - 32 - 34 - 34	(No aceptado).
g) 001	: 30 - 31 - 33 - 32	(aceptado).
h) 110	: 30 - 32 - 34 - 31	(aceptado).

1.4

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

Autómata C2
Q = { 10, 11, 12, 13 }
∑ = { 0, 1 }
q0 = 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.

∩	0	1
→1030	1131	1132
1131	1233	1232
1132	1231	1234
+1231	1333	1332
+1232	1331	1334
1233	1333	1332
1234	1331	1334
1331	1333	1332
1332	1331	1334
1333	1333	1332
1334	1331	1334

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.

$\equiv_0$
C1 = {1030,1131,1132,1233,1234,1331,1332,1333,1334}
C2 = {1231,1232}

$\equiv_1$
C11={1030,1233,1234,
1331,1332,1333,1334}
C12 = {1131}
C13 = {1132}
C2 = {1231,1232}

	0		1	
→1030	1131	C1	1132	C1
1131	1233	C1	1232	C2
1132	1231	C2	1234	C1
+1231	1333	C1	1332	C1
+1232	1331	C1	1334	C1
1233	1333	C1	1332	C1
1234	1331	C1	1334	C1
1331	1333	C1	1332	C1
1332	1331	C1	1334	C1
1333	1333	C1	1332	C1
1334	1331	C1	1334	C1

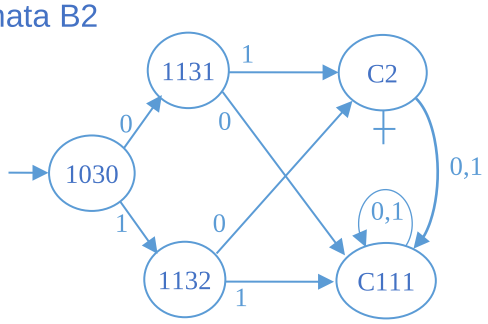
$\equiv_2$
C111={1233,1234,
1331,1332,1333,1334}
C112={1030}
C12 = {1131}
C13 = {1132}
C2 = {1231,1232}

	0		1	
→1030	1131	C12	1132	C13
1131	1233	C1	1232	C2
1132	1231	C2	1234	C1
+1231	1333	C1	1332	C1
+1232	1331	C1	1334	C1
1233	1333	C1	1332	C1
1234	1331	C1	1334	C1
1331	1333	C1	1332	C1
1332	1331	C1	1334	C1
1333	1333	C1	1332	C1
1334	1331	C1	1334	C1

Ahora ya no hay más divisiones en la tabla de transiciones y por lo tanto este será el autómata mínimo.

El DFA mínimo obtenido es el siguiente:

Autómata B2



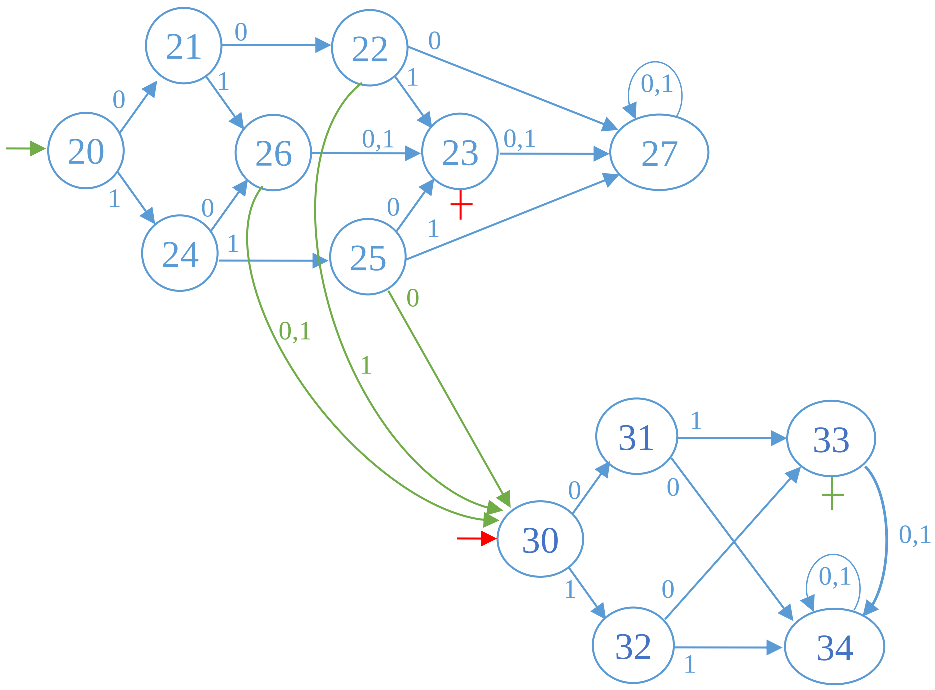
1.5

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

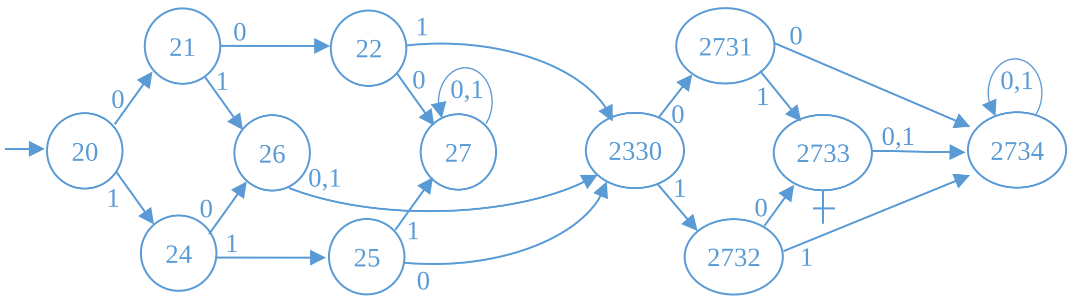
Autómata D



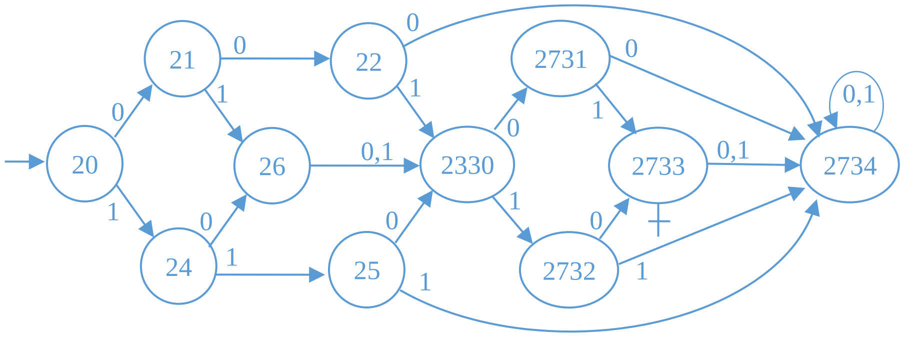
1.5.2
Obtened el DFA D equivalente del autómata D obtenido en el apartado anterior.

Determ.	0	1
→20	21	24
21	22	26
24	26	25
22	27	2330
25	2330	27
26	2330	2330
27	27	27
2330	2731	2732
2731	2734	2733
2732	2733	2733
+2733	2734	2734
2734	2734	2734

Autómata D determinista



Autómata D determinista mínimo (se unifica el estado 27 con el estado 2734).
Esto no se solicitaba



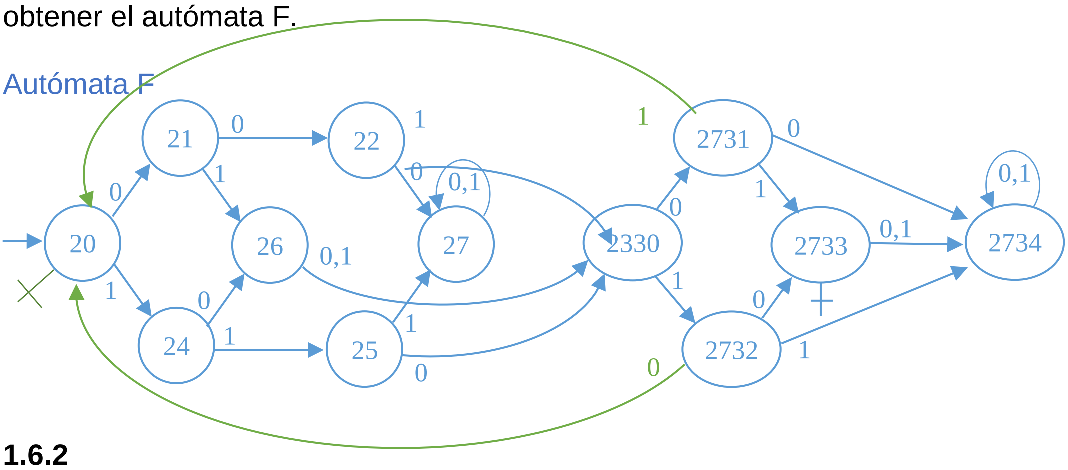
1.6

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.

Autómata F



1.6.2

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

Determ.	0	1
→20	21	24
21	22	26
24	26	25
22	27	2330
25	2330	27
26	2330	2330
27	27	27
2330	2731	2732
2731	2734	273320
2732	273320	2734
+273320	273421	273424
2734	2734	2734
273421	273422	273426
273424	273426	273425
273422	273427	27342330
273425	27342330	273427
273427	273427	273427
27342330	27342731	27342732
27342731	27342734	2734273320
27342732	2734273320	2734
+2734273320	273421	273424

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.

No.
No podríamos concatenar A3 y B2, deberíamos hacer la intersección de A, B y C4.
Donde A sería un autómata que reconocería códigos de cualquier tamaño que no empiece con 3 dígitos binarios iguales.
Donde C4 sería un autómata que reconoce códigos de longitud 4.

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).

No.
No podríamos hacer la concatenación de A3 y B2, deberíamos hacer la intersección de A, B y C6 o la intersección de B y C3 concatenado después con A3, pero también deberíamos tener en cuenta que un código de 6 dígitos, que no empieza por 3 dígitos iguales o acaba con 2 dígitos iguales, puede tener secuencias de más de 3 dígitos iguales (011.110 y 100001), lo que no ocurría con los códigos de longitud 5.
Donde A sería un autómata que reconoce códigos de cualquier tamaño que no empiecen con 3 dígitos binarios iguales.
Donde C6 sería un autómata que reconoce códigos de longitud 6

.

Ejercicio 2

Dado el autómata finito determinista A = (Q, Σ , q0,F, δ), donde Q = {1, 2, 3, 4, 5, 6, 7 , 8 , 9, 10}, Σ = { 0, 1 }, q0=1, F={ 9 } i δ definida per la tabla siguiente:

δ	**0**	**1**
→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.

L1 = 0L2 + 1L3
L2 = 0L4 + 1L6
L3 = 0L6 + 1L5
L4 = 0L10 + 1L8 = 1L8
L5 = 0L7 + 1L10 = 0L7
L6 = 0L7 + 1L8
L7 = 0L10 + 1L9 = 1L9
L8 = 0L9 + 1L10 = 0L9
L9 = 0L2 + 1L3 + λ = L1 + λ
L10 = 0L10 + 1L10 = (0+1)L10 = Ø

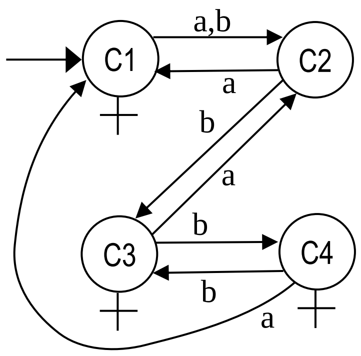
L1 = 0L2 + 1L3 = 00L4 + 01L6 + 10L6 + 11L5
= 00L4 + 11L5 + (01+10)L6
L1 = 00(1L8) + 11(0L7) + (01+10)(0L7 + 1L8)
L1 = (110+010+100)L7 + (001+011+101)L8
L1 = (110+010+100)(1L9) + (001+011+101) (0L9)
L1 = (1101+0101+1001)L9 + (0010+0110+1010)L9
L1 = (1101+0101+1001+0010+0110+1010)L9

L1 = (1101+0101+1001+0010+0110+1010)(L1 + λ)
L1 = (1101+0101+1001+0010+0110+1010)L1
+ (1101+0101+1001+0010+0110+1010) λ
L1 = (1101+0101+1001+0010+0110+1010)*
+ (1101+0101+1001+0010+0110+1010)
L1 = (1101+0101+1001+0010+0110+1010)⁺

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.

.



LC1 = aLC2 + bLC2 + λ = (a+b)LC2 + λ = (a+b) (ba+bb)*(aLC1 + bb + b) + λ

=((a+b)(ba+bb)*a)* ((a+b) (ba+bb)*(bb+b)+λ

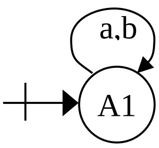
LC2 = aLC1 + bLC3 = aLC1 + b(a+b)LC2 + bb + b = (ba+bb)*(aLC1 + bb + b)

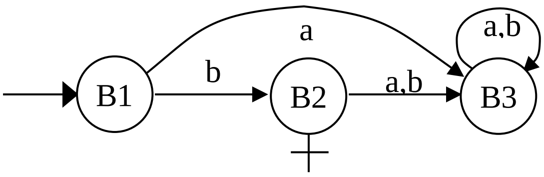
LC3 = aLC2 + bLC4 + λ = (a+b)LC2 + b + λ

LC4 = aLC1 + bLC3 + λ = LC2+λ

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:

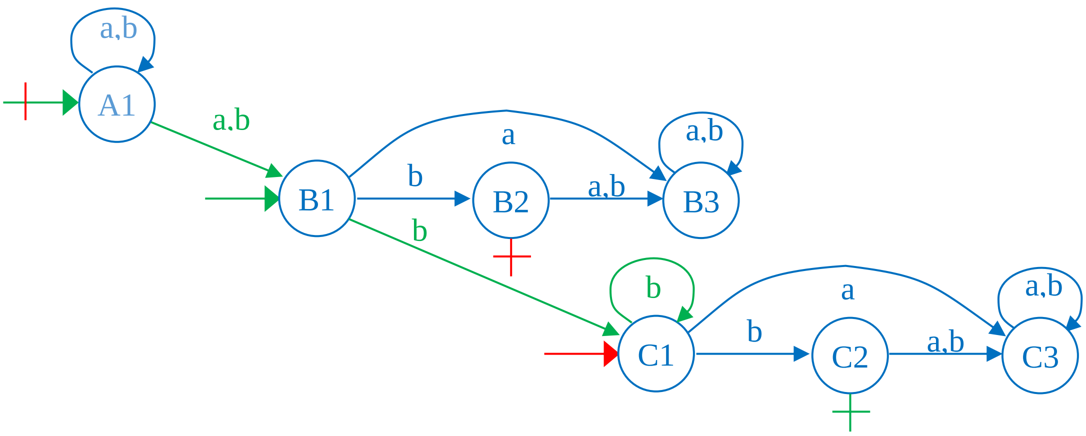
a) Indicad qué lenguaje reconoce cada autómata (dad la expresión regular):

Autómata A: (a+b)*

Autómata B: (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

Hay que concatenar (a+b)*·b·b⁺



¿Las palabras b, bb, bbb, bbb son aceptadas por este autómata?
b no és aceptada, bb, bbb y bbbb son aceptadas.

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

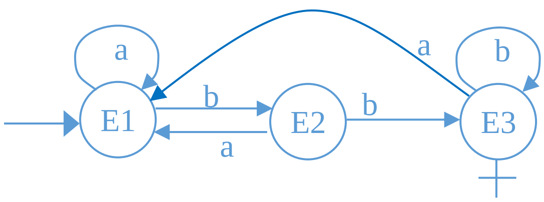
Δ		a		b	
→A1B1	C1	A1B1B3	C3	A1B1B2C1	C2
A1B1B2C1	C2	A1B1B3C3	C4	A1B1B2C1B3C2	C5
A1B1B3	C3	A1B1B3	C3	A1B1B2C1B3	C6
A1B1B3C3	C4	A1B1B3C3	C4	A1B1B2C1B3	C7
+A1B1B2C1B3C2	C5	A1B1B3C3	C4	A1B1B2C1B3C2C3	C8
A1B1B2C1B3	C6	A1B1B3C3	C4	A1B1B2C1B3C2	C5
A1B1B2C1B3C3	C7	A1B1B3C3	C4	A1B1B2C1B3C2C3	C8
+A1B1B2C1B3C2C3	C8	A1B1B3C3	C4	A1B1B2C1B3C2C3	C8

d) Dad el DFA **mínimo** que permita reconocer el lenguaje representado por la expresión regular [(ab + b) (ab + b) * (a + λ)], 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

E1	C1, C2, C3, C4, C6, C7
E3	C5, C8

E1	C1, C3, C4,
E2	C2, C6, C7

Ya no hay más divisiones en la tabla de transiciones, y por tanto, este es el DFA mínimo.



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 +].

LE1 = aLE1 + bLE2 = aLE1 + bb*aLE1 + bb*b = (a+bb*a)*bb*b=(a+b$^+$a)*bb$^+$
LE2 = aLE1 + bLE3 = aLE1 + bL2 + b = b*(aLE1 + b)
LE3 = aLE1 + bLE3 + λ = L2 + λ

La forma de la expresión regular no es idéntica, pero sí muy parecida y equivalente.
La b de (a + b) * queda integrada en bb +
(a + b + a) es para tener en cuenta las a s antes de las b 's del final.