Teoría de Lenguajes
Soluciones

Ejercicio 1

L_1 = { $w\#w^k$ / $w \in \{0,1\}^*$ $\wedge$ $k \geq 1$ }

a) L_1 es recursivamente enumerable como vermos en la parte b) al construir una MT que lo reconoce. Demostraremos que NO es libre de contexto, utilizando el CR del PL (para Libres de Contexto).

Para eso elegimos la tira $z = 0^N1^N\#0^N1^N$, siendo N la constante del Pumping Lema (0^N1^N y k=1)

Consideramos todas las descomposiciones posibles de la tira z en uvwxy que cumplen |vx|>0 y |vwx| ≤ N, y probamos que existe un i tal que uv^iwx^iy no pertenece a L_1

Familia	0^N	1^N	#	0^N	1^N
1	v x				
2		v x			
3				v x	
4					v x
5	v	x			
6				v	x
7	v x	x			
8	v	v x			
9				v x	x
10				v	v x
11		v x	x	x	
12		v	v	v x	
13		v		x	

Familia 1
$u = 0^j$
$v = 0^k$
$w = 0^l$
$x = 0^m$
$y = 0^{N-j-k-l-m}1^N\#0^N1^N$ con k+m>0
$z_2 = 0^{N+k+m}1^N\#0^N1^N$ no pertenece al lenguaje pues al ser k+m>0, *w* tiene mas 0's a la izquierda que a la derecha del #, por tanto no se repite el mismo w

Familia 2
Es análoga, pero razonando con los 1's.

Familia 5, 7, 8
Son análogas a la Familia 1, pero razonando tanto con 0's como con 1's (crecen los 2)

Familia 3
$u = 0^N1^N\#0^l$
$v = 0^k$
$w = 0^p$
$x = 0^m$
$y = 0^{N-l-k-p-m}1^N$ con k+m > 0

$z_0 = 0^N1^N\#0^{N-k-m}1^N$ no pertenece al lenguaje pues al ser k+m>0, *w* tiene mas 0's a la izquierda que a la derecha del #, por tanto no se repite el mismo w

Familia 4
Es análoga, pero razonando con los 1's.

Familias 6, 9 y 10
Son análogas a la Familia 3, pero razonando tanto con 0's como con 1's (crecen los 2).

Familia 11
$u = 0^N1^j$
$v = 1^k$
$w = 1^{N-j-k-l}$
$x = 1^l\#0^m$
$y = 0^{N-m}1^N$
$z_0 = 0^{N-k}1^{N-l}0^{N-m}1^N$ no pertenece al lenguaje pues z_0 no tiene el #.

Familia 12
En forma análoga a la Familia 11, se puede demostrar que eligiendo i=0, z0 no pertenece al lenguaje, dado que z_0 no tiene el #.

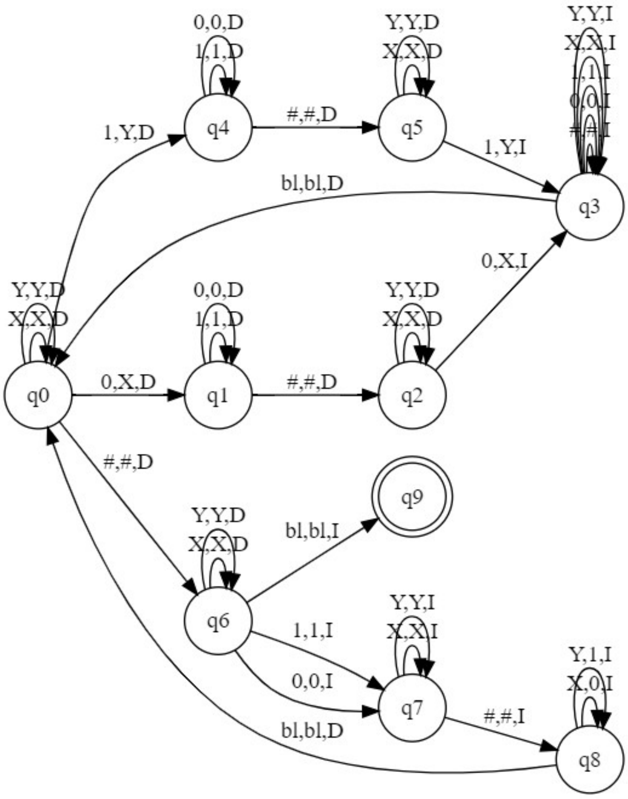
Familia 13
$u = 0^N1^j$
$v = 1^k$
$w = 1^{N-j-kl}\#0^l$
$x = 0^m$
$y = 0^{N-l-m}1^N$ con k+m > 0

Asumimos en este caso, que tanto k como m son ambos >0, sino se cae en un caso particular de alguno de los casos anteriores.

En cualquier caso, al elegir i=0, z_0 no pertenece al lenguaje, porque la cantidad de 0's de la tira a la derecha del # siempre queda menor que la de 0's de la izquierda, o el razonamiento igual con los 1's de la izquierda y la derecha.

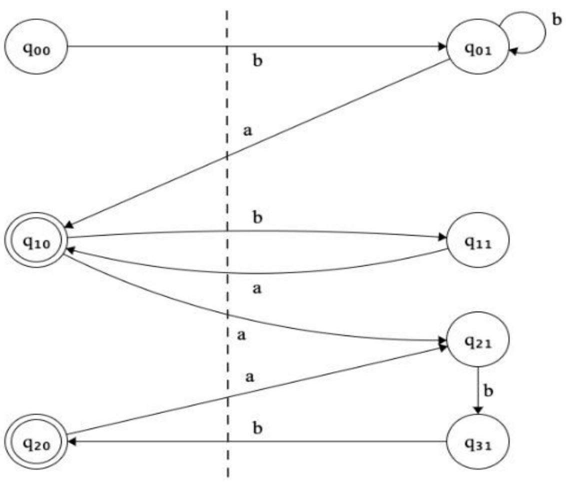
Como estas son todas las descomposiciones posibles de z en uvwxy que cumplen |vx|>0 y |vwx| ≤N, por el CR del PL el lenguaje **L_1 NO es libre de contexto**.

b) Se construye una MT determinista M_1 de acuerdo al modelo presentado en el curso / $L_1 = L(M_1)$

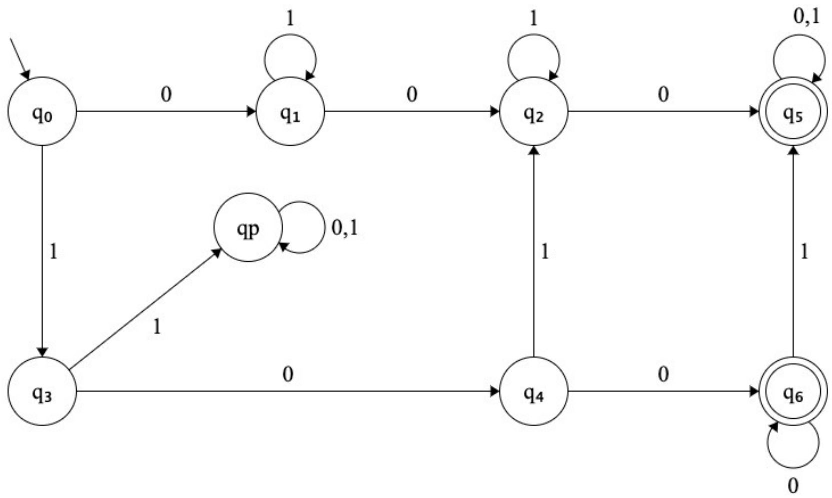


Ejercicio 2

a) $L_{2a} = \{ < b^j a^k , b^n a^j b^m > : \text{con } m=2k , m,k,n \geq 0 , j>0\}$

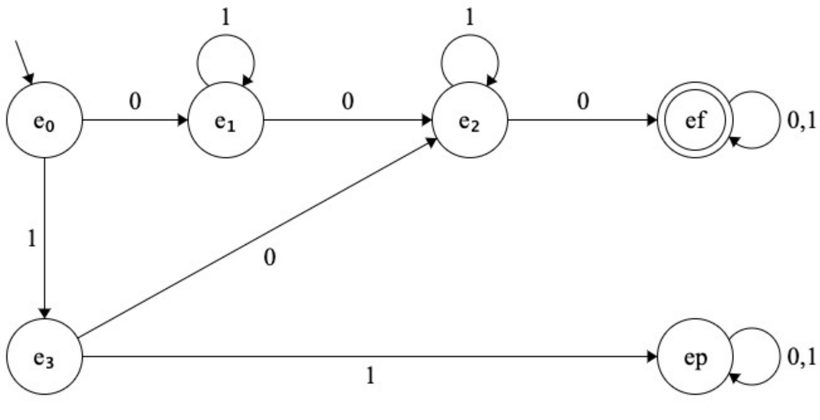


b) Para hallar las clases de R_L calcularemos las clases de R_M. Para eso, completamos el autómata (agregando el pozo) y lo minimizamos:



π_0	[q0 q1 q2 q3 q4 qp]			[q5 q6]		
π_1	[q0 q1 q3 qp]			[q2 q4]	[q5 q6]	
π_2	[q0]	[q1]	[q3]	[qp]	[q2 q4]	[q5 q6]
π_3	[q0]	[q1]	[q3]	[qp]	[q2 q4]	[q5 q6]
Renombre	e0	e1	e3	ep	e2	ef

El autómata mínimo queda definido de la siguiente manera:



Planteamos ahora el sistema de ecuaciones para hacer el cálculo de las clases de R_M

X0 = ε
X1 = X0.0 | X1.1
X2 = X1.0 | X2.1 | X3.0
X3 = X0.1
XF = XF.0 | XF.1 | X2.0
XP = XP.0 | XP.1 | X3.1

Usando el lema de Arden en su versión X = X r | s = s.r*, operamos para resolver el sistema:

X1 = ε.0 | X1.1 = X1. 1 | 0 ⇒ X1 = 0.1*
X3 = ε.1 ⇒ X3 = 1
XP = XP(0|1) | 1.1 ⇒ XP = 11(0|1)*

X2 = (0.1*).0 | X2.1 | (1).0 = X2.1 | (01*0 | 10) ⇒ (01*0 | 10).1* = 01*01* | 101*

XF = XF.0 | XF.1 | X2.0 = XF (0|1) | (01*01* | 101*).0 = XF (0|1) | (01*01*0 | 101*0)
⇒ (01*01*0 | 101*0).(0|1)*

Como el autómata era mínimo, entonces las clases calculadas de RM coinciden con las de RL.

c) Se aplican los algoritmos vistos en el curso para eliminar transiciones epsilon y luego obtener el AFD solicitado

ε-cl(q0) = {q0,q1,q4}
ε-cl(q1) = {q1,q4}
ε-cl(q2) = {q2}
ε-cl(q3) = {q3}
ε-cl(q4) = {q4}

δ'(q0,a) = {q4}	δ'(q0,b) = {q1,q2,q3,q4}
δ'(q1,a) = {q4}	δ'(q1,b) = {q1,q3,q4}
δ'(q2,a) = {q1,q2,q4}	δ'(q2,b) = {q3}
δ'(q3,a) = {}	δ'(q3,b) = {q3,q4}
δ'(q4,a) = {}	δ'(q4,b) = {}

Los estados finales serán: F' = {q0, q4}

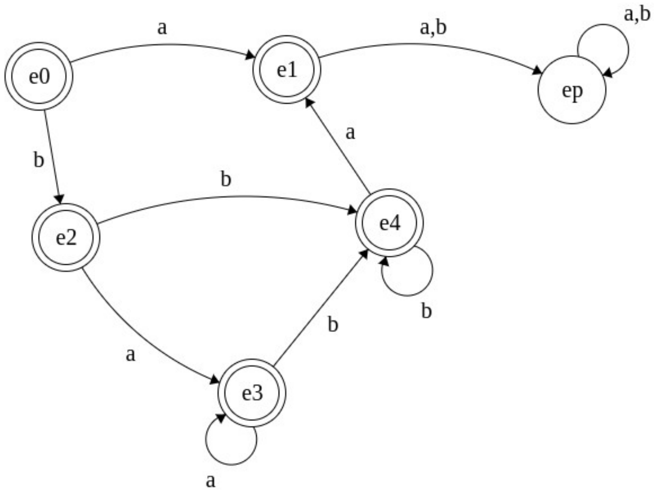
Ahora realizamos el pasaje AFND-AFD para obtener un posible autómata D.

δ''	a	b
e0: [q0] ∈ F	[q4]	[q1 q2 q3 q4]
e1: [q4] ∈ F	[ ]	[ ]
e2: [q1 q2 q3 q4] ∈ F	[q1 q2 q4]	[q1 q3 q4]
e3: [q1 q2 q4] ∈ F	[q1 q2 q4]	[q1 q3 q4]
e4: [q1 q3 q4] ∈ F	[q4]	[q1 q3 q4]

Renombramos los estados, agregamos el pozo, y marcamos los estados finales:

δ''	**a**	**b**
e0 ϵ F	e1	e2
e1 ϵ F	ep	ep
e2 ϵ F	e3	e4
e3 ϵ F	e3	e4
e4 ϵ F	e1	e4
ep	ep	ep

El autómata resultante queda diagramado de la siguiente manera:



Ejercicio 3

L_3 = { x / x ∈ Σ * y es de la forma $a^p b^q c^r$, con |p-r|=|q-r|, r>0, p≥0, q≥0 }

a) El lenguaje L NO es regular. Se demostrará utilizando el contrarecíproco del Pumping Lema para Lenguajes Regulares (PL1).

Sea N la cte. del PL , elegimos $z = a^N b^N c$, se verifica que z ϵ L (|p-r|=|q-r|=N-1), con |z| = 2N+1 > N.

Se estudian todas las descomposiciones de z=uvw que cumplen: |v|>0 y |uv|≤ N

Caso 1:

$u=a^s$ $s \geq 0$

$v=a^t$ $t>0$ $s+t \leq N$

$w= a^{N-s-t} b^N c$

$z_i = uv^iw = a^{N+t(i-1)} b^Nc$

Se elige i=0 → $z_0 = a^{N-t} b^Nc$ no pertenece a L_3 ya que como t >0, se tiene para z_0 que $|p-r| = N-t-1 < N-1 = |q-r|$.

Como no hay más descomposiciones que cumplen $|v|>0$ y $|uv| \leq N$, por el CR del PL1 concluimos que L_3 NO es Regular.

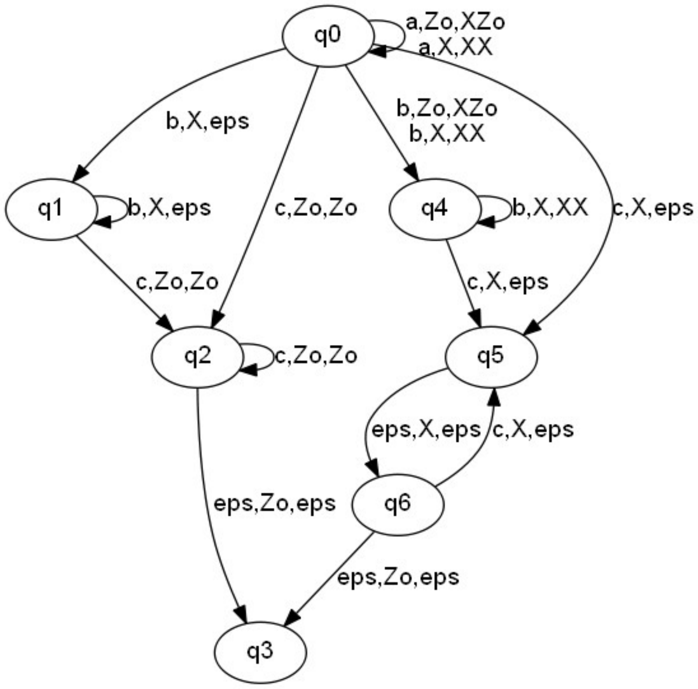
En la parte b) se construye una APD que reconoce L_3, con lo cual clasificamos a L_3 como Lenguaje Libre de Contexto NO Regular.

b) Se construye un autómata APD M_3/ $L(M_3)=L_3$.

Observamos que la condición |p-r|=|q-r| es equivalente a a que se cumpla al menos una de las siguientes condiciones:

1) p = q
2) p + q = 2r

Se construirá un APD que reconoce el lenguaje por stack vacío. La idea es mantener en el stack una cantidad de símbolos igual a los valores hasta el momento de |p-q| y de |p+q-2r|, y aceptar solamente en caso de que al terminarse la tira de entrada el stack esté vacío, controlando que r > 0. En el diagrama de la figura, la rama que sale por q_1 corresponde a reconocer la condición 1) ; acumula un símbolo en el stack por cada a y lo saca por cada b de la entrada. La rama que sale por q_4 implementa la condición 2); acumula un símbolo en el stack por cada a y b de la entrada y saca dos simbolos por cada c.



M_3 es un APD no determinista (APDND), ya que desde el estado inicial q0 y con valor X en el stack, para el símbolo de entrada b hay transiciones a dos estados distintos:

δ(q0,b,X)={(q1,eps),(q4,XX)}.

Nota: Las gramáticas y los autómatas **deben corresponderse** con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Se valora positivamente la simplicidad de las soluciones propuestas, así como una breve explicación de éstas. **Todas las respuestas deben estar debidamente justificadas.**

c) Se construye una gramática G_3 / $L(G_3)=L$

$G_3 = (\{S, A, B, C, D\}, \{a,b,c\},P,S\}$

con P formado por las siguientes producciones:

S → AB | C
A → aAb | ε
B → cB | c
C → aaCc | abDc | bbDc | aac
D → bbDc | ε

La idea es definir una subgramática que genere las tiras con p=q (S → AB) y otra que genere las tiras con p+q = 2r (S → C).

Falta ahora ver si está simplificada.

Se ve que no, que tiene al menos producciones unitarias y con ε

Por consiguiente, aplicando en cascada esos algoritmos, la gramática anterior G_3 queda con las siguientes producciones:

S → AB | cB | c | aaCc | abDc | bbDc | aac | abc | bbc
A → aAb | ab
B → cB | c
C → aaCc | abDc | bbDc | aac | abc | bbc
D → bbDc | bbc

Se observa que todas las variables involucradas son útiles (positivas y alcanzables).