Teoría de Lenguajes
Soluciones

Ejercicio 1

Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique adecuadamente cada respuesta.

a) Si L_{a1} es recursivamente enumerable NO libre de contexto y L_{a2} es libre de contexto pero NO regular y ambos definidos sobre el mismo alfabeto Σ, entonces $L_a=L_{a1} \cap L_{a2} = \varnothing$

Falso: Sea $L_{a1} = \{a^k b^k c^k \ / \ k \geq 0, \}$ es LC No Regular y $L_{a2} = \{a^k b^k \ / \ t \geq 0 \}$ es LC
Se puede observar que $L_{a1} \cap L_{a2} = \{ \varepsilon \} \neq \varnothing$

b) Sean Σ, Δ alfabetos, $L_b \subseteq \Delta^*$, $h:\Sigma \rightarrow \Delta$ un homomorfismo y h^{-1} el homomorfismo inverso de h. Se cumple $L_b = h(h^{-1}(L_b))$ siendo L_b un lenguaje regular.

Falso. Sea el homomorfismo h definido de la siguiente forma: h(a)=b y h(b)=b, y $L_a = L(a^*)$. Entonces $h^{-1}(L_a) = \varnothing$, luego $h(h^{-1}(L_a)) = h(\varnothing) = \varnothing \neq L_a$

c) Si $L_{c1} \subseteq \Sigma^*$ es regular y $L_c = \{ x \ / \ x \in L_{c1} \ , \ \forall \ w \in \Sigma^* \ w \neq \varepsilon \ , \ xw \notin L_{c1}\}$ entonces L_c es regular.

Verdadero: Sea $M_{c1}:(Q, \Sigma, \delta, q_0, F)$ un AFD / $L_{c1} = L(M_{c1})$. Se construye un nuevo AFD Mc / Lc = L(Mc) donde se **eliminan** todas las transiciones **salientes** de los estados **finales** de M_{c1}.
Es decir $M_c:(Q', \Sigma, \delta', q_0, F)$ donde $\delta'(q_j,a) = \delta(q_j,a)$ si $q_j \notin F$ y $Q' \subseteq Q$, ya que pueden quedar estados inaccesibles en M_c.

d) $L_d = \{ x \ / \ x \in \{a,b\}^* \text{ y es de la forma } a^k \, b^p \, a^k \, b^t \text{ con } k > t \geq 0, p>0 \}$ es un lenguaje libre de contexto

Falso: Se usará el contrarecíproco del Pumping Lema para Lenguajes Libres de Contexto

Dado N constante cualquiera, elegimos $z = a^{N+1}ba^{N+1}b^N$, $z \in L_1$ y $|z|=3N+3$
Estudiamos todas las descomposiciones de z=uvwxy que cumplen $|vx|>0$ y $|vwx|\leq N$

Familias	**a^{N+1}**	**b**	**a^{N+1}**	**b^N**
1	v x			
2			v x	
3				v x
4	v x	x	x	
5	v	v	v x	
6			v x	x
7			v	v x
8	v		x	
9			v	x

Familia 1

$u=a^{N+1-p-q-r-s}$
$v=a^p$ $p+r > 0$
$w=a^q$ $p+q+r \leq N$
$x=a^r$
$y=a^sba^{N+1}b^N$

$z_i = a^{N+1-p-q-r-s} (a^p)^i a^q (a^r)^i a^sba^{N+1}b^N = a^{N+1-p+pi-r+ri-}ba^{N+1}b^N = a^{N+1-(p+q)(i-1)}ba^{N+1}b^N$
Eligiendo i=0 queda $z_0= a^{N+1-p-q}ba^{N+1}b^N$
donde la cant_a(z_0) del comienzo de la tira es $\leq$ que cant_b(z_0) las **b**'s del final.
Por lo tanto $z_0 \notin L_d$

Familia 2, razonamiento análogo pero tomando el segundo" grupo de **a**'s
Familia 3, razonamiento análogo, pero hay que tomar i=2, con lo cual ahí la cantidad de **b**'s es la que aumenta con respecto a los dos grupos de **a**'s.

Familia 4

$u=a^{N+1-p-q-r}$
$v=a^p$ $p+r+s+1 > 0$ (contempla casos p y/o r y/o s = 0)
$w=a^q$ $p+q+r+s+1 \leq N$
$x=a^rba^s$
$y=a^{N+1-s}b^N$

$z_i = a^{N+1-p-q-r} (a^p)^i a^q (a^rba^s)^i a^{N+1-s}b^N = a^{N+1-p-r} (a^p)^i (a^rba^s)^i a^{N+1-s}b^N$
Eligiendo i=0 queda $z_0= a^{N+1-p-r} a^{N+1-s}b^N$
donde NO queda la menos una **b** entre los 2 substrings de **a**'s. Por lo tanto $z_0 \notin L_d$

Familia 5, razonamiento análogo y argumento análogo.

Familia 6

$u=a^{N+1}ba^{N+1-p-q-r}$
$v=a^p$ $p+r+j > 0$
$w=a^q$ $p+q+r+j \leq N$
$x=a^rb^j$
$y=b^{N-j}$

$z_i = a^{N+1}ba^{N+1-p-q-r}(a^p)^i a^q (a^rb^j)^i b^{N-j} = a^{N+1}ba^{N+1-p-r}(a^p)^i (a^rb^j)^i b^{N-j}$

- si j=0, p y/o r >0, tomamos i=0 entonces queda $z_0= a^{N+1}ba^{N+1-p-r}b^{N-j}$ con lo cual la cantidad de **a**'s del primer substring de **a**'s de z_0 es mayor que la cantidad de **a**'s del segundo substring de **a**'s. Por lo tanto $z_0 \notin L_d$
- si j >0, si r>0 tomamos i=2 y acá el problema es que en z_2, se mezclan **a**'s y **b**'s.
  si r=0 también tomando i=2, el problema es que la cantidad de **b**'s de z_2 del final de la tira es mayor o igual que la cantidad de **a**'s del primer substring de **a**'s.
Por lo tanto $z_2 \notin L_d$

Familia 7, el razonamiento es análogo.

Familia 8

$u=a^{N+1-p-q}$
$v=a^{p}$ $p+r > 0$
$w=a^{q}ba^{s}$ $p+q+r+s+1 \leq N$
$x=a^{r}$
$y=a^{N+1-s-r}b^{N}$

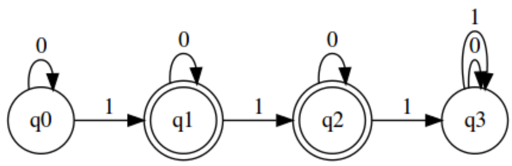
$z_i = a^{N+1-p-q} (a^p)^i a^q ba^s (a^r)^i a^{N+1-s-r} b^N = a^{N+1-p} (a^p)^i b (a^r)^i a^{N+1-r} b^N$
Eligiendo i=0 queda $z_0 = a^{N+1-p} b a^{N+1-r} b^N$
Como p+r>0, al menos uno de los 2 es >0, con lo cual la cantidad de **b**'s queda mayor o igual que el substring de **a**'s que corresponda con esa variable (p ó r)
Por lo tanto $z_0 \notin L_d$

Familia 9 es similar, sólo que el argumento será que tomanado i=2, la cantidad de **b**'s es mayor o igual que la cantidad de **a**'s del primer subtring de **a**'s.
Por lo tanto $z_2 \notin L_d$

Como estas son todas las descomposiciones posibles que cumplen |vx|>0 y |vwx|≤ N, se ha logrado demostrar por el contra-recíproco del Pumping Lema para lenguajes libres de contexto que L_d NO es un lenguaje libre de contexto, con lo cual la afirmación es Falsa.

e) Sean L_e un lenguaje regular y dos tiras x, y ∈ L_e, entonces $xR_{L_e}y$

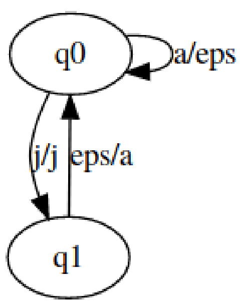
Falso: Alcanza con considerar un lenguaje regular cuyo autómata finito mínimo tenga más de un estado final. Por ejemplo, sea el lenguaje L_e de las tiras definidas sobre el alfabeto {0,1} y que contienen exactamente uno o dos 1's.



Si tomamos la tira x= 010 , y= 0101, ambas ∈ L_e pero no se cumple $xR_{L_e}y$ ya que cuando z = 1, **x1 ∈ L$_e$** (porque tiene dos 1's) sin embargo **y1 ∉ L$_e$** (ya que queda con tres 1's)

Ejercicio 2

a)



b) Dado el siguiente autómata finito M_2:({q0, q1, q2, q3, q4}, {0,1}, δ, q0, {q3}) siendo d dado por:

	0	**1**	**ε**
q0	{q1 , q3}	{q1}	{q4}
q1	Φ	{q0 , q2}	Φ
q2	Φ	{q3}	Φ
q3	Φ	Φ	Φ
q4	Φ	{q0 , q2}	{q3}

i) Construya el autómata finito mínimo M_2' / $L(M_2)=L(M_2')$

Pasaje AFND-ε → AFND, se aplica δ'(q,a) = ε-clausura(δ(ε-clausura(q),a))

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

	0	**1**
q0	{q1, q3}	{q0, q1, q2, q3, q4}
q1	Φ	{q0, q2, q3, q4}
q2	Φ	{q3}
q3	Φ	Φ
q4	Φ	{q0, q2, q3, q4}

F' = {q0, q3}

Pasaje AFND → AFD

	δ''	**0**	**1**
p0	[q0]	[q1, q3]	[q0, q1, q2, q3, q4]
p1	[q1, q3]	-	[q0, q2, q3, q4]
p2	[q0, q1, q2, q3, q4]	[q1, q3]	[q0, q1, q2, q3, q4]
p3	[q0, q2, q3, q4]	[q1, q3]	[q0, q1, q2, q3, q4]

F'' = {p0, p1, p2, p3}

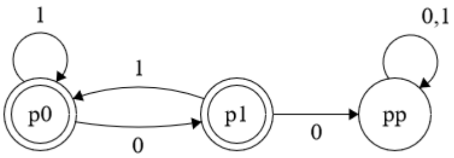
<u>Minimización</u>
π0 [p0, p1, p2, p3]
π1 [p1] [p0, p2, p3]
π2 [p1] [p0, p2, p3]

AFD Mínimo

δ	**0**	**1**
p0	p1	p0
p1	-	p0

Estados finales: p0 y p1

Si agregamos el estado pozo para que el autómata sea completo, lo cual nos servirá para la siguiente parte, el resultado es el siguiente:



ii) Defina la relación R_L. Dé las clases de R_L mediante expresiones regulares.

Dados x,y ∈ Σ*: x R_L y si ∀ z ∈ Σ*, xz ∈ L ^ yz ∈ L
v xz ∉ L ^ yz ∉ L

Como el AFD encontrado es mínimo y completo, se pueden plantear las ecuaciones características asociadas a cada estado del autómata, cuyas incógnitas son los conjuntos de palabras (tiras) que permiten pasar desde el estado inicial a cada estado del AFD. En la resolución se utiliza el Lema de Arden: la única solución a la ecuación
X = rX | s es: X = r*s si ε ∉ L(r).

$X_0 = X_0.1 \mid X_1.1 \mid \varepsilon$
$X_1 = X_0.0$
$X_p = X_p.0 \mid X_p.1 \mid X_1.0$

Sustituyendo, aplicando el Lema de Arden y factorizando tenemos:

$X_0 = X_0.1 \mid X_0.01 \mid \varepsilon = X_0 (1 \mid 01) \mid \varepsilon$
$X_0 = \varepsilon.(1 \mid 01)^* =$ **(1 | 01)***

$X_1 =$ **(1 | 01)*.0**

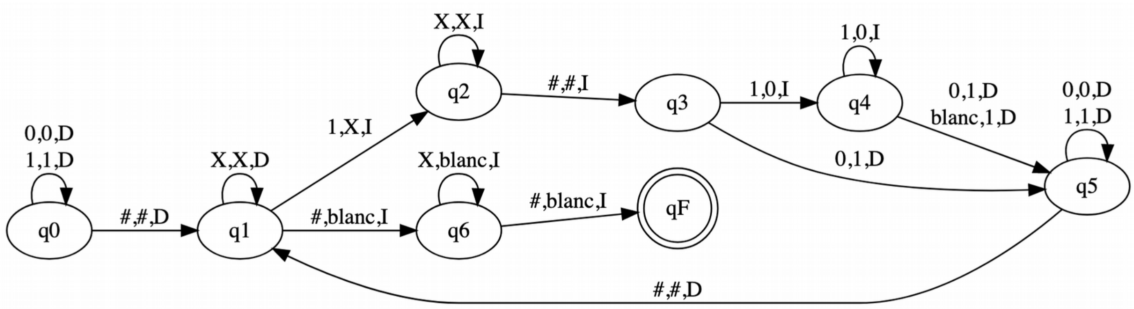
$X_p = X_p (0 \mid 1) \mid (1 \mid 01)^* 00$
$X_p =$ **(1 | 01)* . 00 . (0 | 1)***

iii) Dé una expresión regular que defina el lenguaje L(M). Justifique.

La expresión regular del lenguaje es la unión de las clases de equivalencia asociadas a los estados finales.

L(M) = $X_0 \mid X_1$ = (1 | 01)* | (1 | 01)*.0
L(M) = **(1 | 01)* . (0 | ε)**

Ejercicio 3



Ejercicio 4

Sea L_4 = { x / x es de la forma $a^m b^r c^t$ con m,t impar, r=(m+t)/2 }

a) El lenguaje es libre de contexto, lo cual se demuestra por el A.P.D. de la parte b) y/o la GLC de la parte c).

A continuación demostraremos que no es un lenguaje regular aplicando el contra-recíproco del Pumping Lema para lenguajes regulares. 1

Dado n, elegimos z = $a^{2n+1} b^{2n+1} c^{2n+1}$, |z| = 6n+3 >= n, y z pertenece a L dado que r = 2n+1 = (2n+1+2n+1)/2 = (m+t/)2.

Estudiamos todas las descomposiciones z en uvw tal que: |uv| <= n y |v| >= 1

Caso	a^{2n+1}	b^{2n+1}	c^{2n+1}
1	u v w	w	w

Caso 1

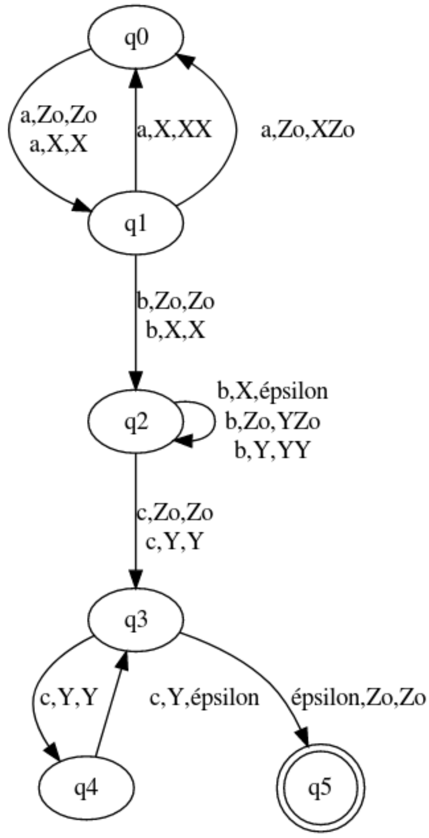
u = a^p p+q <= n
v = a^q q >= 1
w = $a^{2n+1-p-q} b^{2n+1} c^{2n+1}$

$z_i = a^p (a^q)^i a^{2n+1-p-q} b^{2n+1} c^{2n+1} = a^{2n+1+q(i-1)} b^{2n+1} c^{2n+1}$

Eligiendo i = 2 queda $z_i = a^{2n+1+q} b^{2n+1} c^{2n+1}$ que no pertenece a L_4 dado que la suma de la cantidad de a_s y c_s dividido 2 no es igual a la cantidad de b_s.

Como este es el único caso a estudiar, concluimos por el contra-recíproco del P.L. 1 que L_4 no es regular.

b)



Es un APD determinista, porque dado un estado, un tope del stack y una símbolo de la tira a consumir existe una solo transición posible.

c) Como L_4 es un lenguaje libre de contexto no regular, se dará una gramática independiente de contexto simplificada G_4 que genera las tiras de L_4.

Observar que las tiras de L_4 pueden expresarse como $x = a^m b^{(m-1)/2} b b^{(t-1)/2} c^t$ con m,t impares.

Esto sugiere la gramática G_4 = ({S,A,B}, {a,b,c},P,S}, con P dado por:

S → AbB
A → aaAb | a
B → bBcc | c

G_4 está simplificada porque:

1) No tiene producciones-ε
2) No tiene producciones unitarias
3) Todas las variables son alcanzables desde S (A y B a través de S→ AbB)
4) Todas las variables son positivas.
   A y B directamente a través de las producciones A→ a y B → c.
   S → AbB con A y B positivas.