Teoría de Lenguajes
Soluciones

Ejercicio 1

a) Falso.

Dos cadenas w1 y w2 están en la misma clase de equivalencia bajo RL si, para cualquier cadena z, se cumple:

w1z∈L ⇔ w2z∈L

Vamos a verificarlo con un contraejemplo.

Tomemos el lenguaje regular L={w∈{a,b}* | w tiene la forma $a^m b^n$, con m,n≥0}

Este lenguaje está formado por cadenas que consisten en un bloque de aa's seguido de un bloque de bb's (en ese orden).

Tomemos las siguientes tiras:

- w1=a
- w2=b
- z = a

La tira w1.z es la tira "aa" que pertenece a L, mientras que la tira w2.z es la tira "ba" que no pertenece a L.
Ambas cadenas, w1 y w2, están en L, pero no pertenecen a la misma clase de equivalencia bajo la relación RL. Por lo que la afirmación es **falsa.**

b) **Falso.**
Tomemos los lenguajes $L_i = 0^i 1^i$ cada uno de estos lenguajes es regular, ya que consiste únicamente de una tira. Si definimos L como: L = $\cup_{i=1}^{\infty}$ L_i = {01,0011,000111,...} = {x : $0^k 1^k$}, obtenemos un lenguaje que es libre de contexto pero no regular.

Entonces podemos concluir que es **falso** decir que la unión infinita de lenguajes regulares es regular.

c) **Falso.**

Tomemos los lenguajes L_i = {a^i} siendo el alfabeto Σ = {a}. Cada uno de esos lenguajes es regular (formados por la tira de "i" a's).
Entonces si tomamos L = $\cup_{i=1}^{\infty}$ L_i = {a,aa,aaa,...} que puede representarse por la expresión regular aa*. Con lo cual tampoco es posible afirmar que la unión infinita de lenguajes regulares sea no regular.

d) **Falso.**
Si el complemento de la intersección de L_1 y L_2 es el conjunto vacío, significa que:

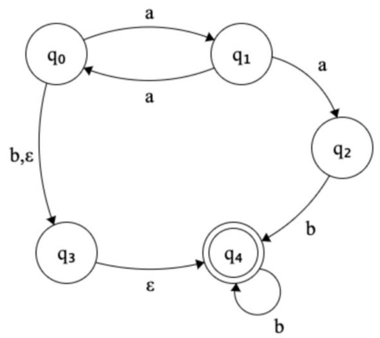
$$(L_1 \cap L_2)^C = \phi \Leftrightarrow (L_1 \cap L_2) = \Sigma^¿$$

Si la intersección de L_1 y L_2 contiene todas las cadenas posibles sobre el alfabeto Σ entonces tanto L_1 como L_2 tiene que contener todas las cadenas posibles sobre el alfabeto.

Si definimos L_1 y L_2 como el mismo lenguaje L={w ∈ {a}* | tal que w tiene la forma a^m con m≥0 }, entonces su intersección es L, que coincide con $\Sigma^¿$, por lo tanto su complemente es vacío. Sin embargo L_1 es libre de contexto y regular, por lo tanto la afirmación es **falsa.**

Ejercicio 2

a) AFND-ε



<u>Pasaje AFND-ε → AFND</u>

Cálculo de las ε-clausura de cada estado:

$\varepsilon_{cl}(q_0) = \{q_0,q_3,q_4\}$
$\varepsilon_{cl}(q_1) = \{q_1\}$
$\varepsilon_{cl}(q_2) = \{q_2\}$
$\varepsilon_{cl}(q_3) = \{q_3,q_4\}$
$\varepsilon_{cl}(q_4) = \{q_4\}$

$\delta'(q_0,a) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_0),a)) = \varepsilon_{cl}(\tilde{\delta}(\{q_0,q_3,q_4\},a)) = \varepsilon_{cl}(\{q_1\}) = \{q_1\}$
$\delta'(q_0,b) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_0),b)) = \varepsilon_{cl}(\tilde{\delta}(\{q_0,q_3,q_4\},b)) = \varepsilon_{cl}(\{q_3,q_4\}) = \{q_3,q_4\}$
$\delta'(q_1,a) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_1),a)) = \varepsilon_{cl}(\tilde{\delta}(\{q_1\},a)) = \varepsilon_{cl}(\{q_0,q_2\}) = \{q_0,q_2,q_3,q_4\}$
$\delta'(q_1,b) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_1),b)) = \varepsilon_{cl}(\tilde{\delta}(\{q_1\},b)) = \varepsilon_{cl}(\{\}) = \{\}$
$\delta'(q_2,a) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_2),a)) = \varepsilon_{cl}(\tilde{\delta}(\{q_2\},a)) = \varepsilon_{cl}(\{\}) = \{\}$
$\delta'(q_2,b) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_2),b)) = \varepsilon_{cl}(\tilde{\delta}(\{q_2\},b)) = \varepsilon_{cl}(\{q_4\}) = \{q_4\}$
$\delta'(q_3,a) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_3),a)) = \varepsilon_{cl}(\tilde{\delta}(\{q_3, q_4\},a)) = \varepsilon_{cl}(\{\}) = \{\}$
$\delta'(q_3,b) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_3),b)) = \varepsilon_{cl}(\tilde{\delta}(\{q_3, q_4\},b)) = \varepsilon_{cl}(\{q_4\}) = \{q_4\}$
$\delta'(q_4,a) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_4),a)) = \varepsilon_{cl}(\tilde{\delta}(\{q_4\},a)) = \varepsilon_{cl}(\{\}) = \{\}$
$\delta'(q_4,b) = \varepsilon_{cl}(\tilde{\delta}(\varepsilon_{cl}(q_4),b)) = \varepsilon_{cl}(\tilde{\delta}(\{q_4\},b)) = \varepsilon_{cl}(\{q_4\}) = \{q_4\}$

y $F'=\{q_0,q_4\}$

<u>Pasaje AFND → AFD</u>

δ″	a	b
$[q_0]$	$[q_1]$	$[q_3q_4]$
$[q_1]$	$[q_0q_2q_3q_4]$	[]
$[q_3q_4]$	[]	$[q_4]$
$[q_0q_2q_3q_4]$	$[q_1]$	$[q_3q_4]$
$[q_4]$	[]	$[q_4]$

y $F''=\{q_0,q_{34},q_{0234},q_4\}$

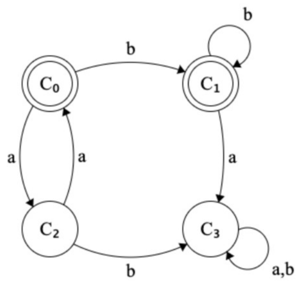
Agregamos el estado pozo para la minimización

δ″	a	b
$[q_0]$	$[q_1]$	$[q_3q_4]$
$[q_1]$	$[q_0q_2q_3q_4]$	$[q_p]$
$[q_3q_4]$	$[q_p]$	$[q_4]$
$[q_0q_2q_3q_4]$	$[q_1]$	$[q_3q_4]$
$[q_4]$	$[q_p]$	$[q_4]$
$[q_p]$	$[q_p]$	$[q_p]$

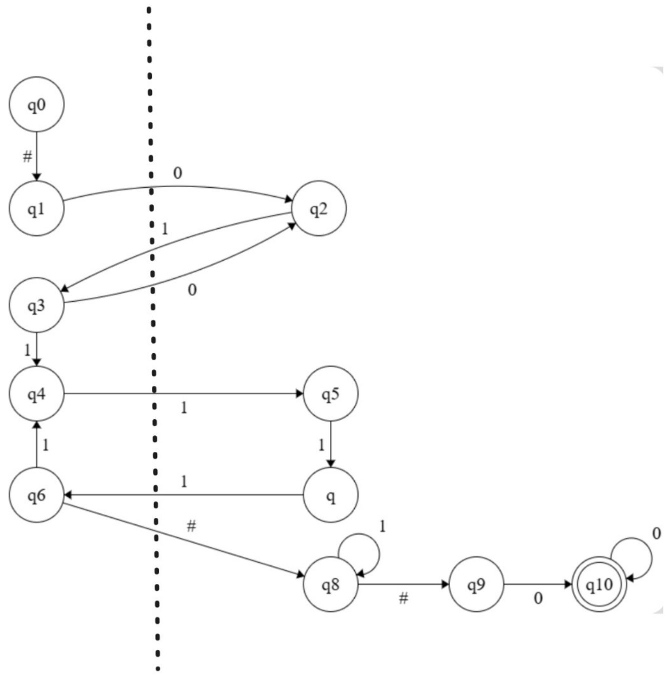
<u>Pasaje AFD → AFD- Mínimo</u>

$\Pi_0 = \{q_0,q_{34},q_{0234},q_4\}\ \{q_1,q_p\}$
$\Pi_1 = \{q_0,q_{34},q_{0234},q_4\}\ \{q_1\}\ \{q_p\}$
$\Pi_2 = \{q_0,q_{0234}\}\ \{q_{34},q_4\}\ \{q_1\}\ \{q_p\}$
$\Pi_3 = \{q_0,q_{0234}\}\ \{q_{34},q_4\}\ \{q_1\}\ \{q_p\}$
$C_0\sim\{q_0,q_{0234}\}$, $C_1\sim\{q_{34},q_4\}$, $C_2\sim\{q_1\}$, $C_3\sim\{q_p\}$

AFD- Mínimo



b) L_{2b}= { <#0^k1^j#, 1^t#0^r> con k,j,r>0; t ≥ k+j; j MOD 2 = 0 }



Ejercicio 3

a)

L_{31}={ $x = c^p w c^k$ / $w \in \{a,b\}^*$ $\wedge$ $|w|_a > |w|_b > 0 \wedge k > p > 0$ }

Este lenguaje es libre de contexto, no regular.

Primero se probará que no es regular utilizando el contra-recíproco del PL para lenguajes regulares. Luego se verá en la parte b) porque es libre de conbtexto.

Sea $N \in \mathbb{N}$ y se elige $z = ca^{N+1}b^Ncc \in L_{31}$ y $|z| = 2N+4 \geq N$ (p=1, k=2)

Ahora se estudian todas las descomposiciones de z = uvw talque $|uv| \leq n$ y $|v| \geq 1$ tratando de encontrar un $i \geq 0$ para c/descomposición tal que $uv^iw \notin L_{31}$

Familia 1

$u = \varepsilon$
$v = ca^q$ con $1+q \leq N$ y $q \geq 0$
$w = a^{N+1-q}\,b^Ncc$

$z_i = uv^iw = (ca^q)^i\, a^{N+1-q}\, b^Ncc$

Tomando i=0 se ve que $uv^0w \notin L_{31}$, porque la tira z_0 no comienza con **c**

Familia 2
$u = ca^q$
$v = a^p$ con $1+q+p \leq N$ y $p \geq 1$
$w = a^{N+1-q-p} b^N cc$

$z_i = uv^iw = ca^q(a^p)^i a^{N+1-q-p} b^N cc = ca^q(a^p)^i a^{N+1-q-p} b^N cc = ca^q a^{pi} a^{N+1-q-p} b^N cc =$
$ca^{N+1+(i-1)p} b^N cc$

Tomando nuevamente i=0 se ve que $uv^0w = ca^{N+1-p} b^N cc \notin L_{31}$, porque como $p \geq 1$, la tira z_0 contiene menos o igual cantidad de **a**'s que de **b**'s

Por lo tanto, como estas son todas las descomposiciones posibles de z en uvw que cumplen $|uv| \leq n$ y $|v| \geq 1$, por el CR del PL L_{31} **no** es regular.

Nota: También podría haberse tomado la tira $z = c^N aabc^{N+1}$ que controla la relación de las **c**' y pone un caso particular entre **a**'s y **b**'s
En ese caso tendría sólo una Familia para discutir.

L_{32}={ w / w $\in$ {a,b,c}*, comienzan con **c** $\wedge$ $|w|_a > 1 \wedge |w|_b \geq 0 \wedge |w|_c \geq 1$ }

Este lenguaje es regular, ya que lo podemos expresar a través de la siguiente expresión regular: **c(a|b|c)*a(a|b|c)*a(a|b|c)***

Justificación:
- comienza en una c
- luego hay que asegurar que vengan al menos dos **a**'s por la condición >1
- antes, entre medio y después de esas **a**'s, pueden venir cualquier cantidad de símbolos {a, b, c} y en cualquier orden

b) i. Construcción de gramáticas

Para L_{31} se va a construir una gramática libre de contexto
Teniendo en cuenta que la desigualdad k > p, se puede expresar como $\exists$ j>0 $\in$ N /
k = p + j y por tanto el lenguaje se puede reescribir como:

L_{31}= {x = $c^p.w.c^{p+j}$ / w $\in$ {a,b}* $\wedge$ $|w|_a > |w|_b > 0$ $\wedge$ j>0 $\wedge$ p>0}
= {x = $c^p.w.c^pc^j$ / w $\in$ {a,b}* $\wedge$ $|w|_a > |w|_b > 0$ $\wedge$ j>0 $\wedge$ p>0}

Por lo tanto, la gramática libre de contexto propuesta a continuación se basa en generar las c^p de los extremos y con otra variable generar las c's a la derecha para que sean mayores a p. Por otro lado, generar la tira w en el centro.
Para generar la tira w, damos todas las posibilidades en que se pueden disponer pares de "a" y "b", obligando a generar al menos una "a" extra.

S → cSc | cXcJ
X → Xab | aXb | abX | Xba | bXa | baX | aX | Xa
    | Yab | aYb | abY | Yba | bYa | baY
Y → aY | a
J → Jc | c

Puede verificarse que esta gramática está simplificada, ya que no tiene símbolos inútiles, ni producciones epsilon ni producciones unitarias.

Para L_{32} se va a construir una gramática regular con las siguientes reglas:

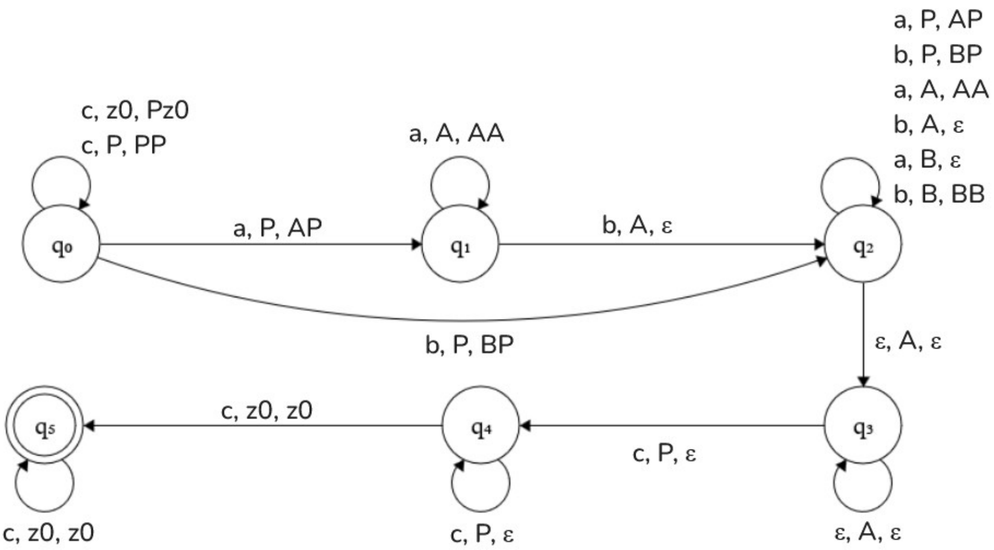
S → cA
A → aB | aA | bA | cA
B → aC | aB | bB | cB | a
C → aC | bC | cC | a | b | c

También pued verificarse que esta gramática está simplificada, ya que no tiene símbolos inútiles, ni producciones epsilon ni producciones unitarias.

ii. Construcción de autómatas

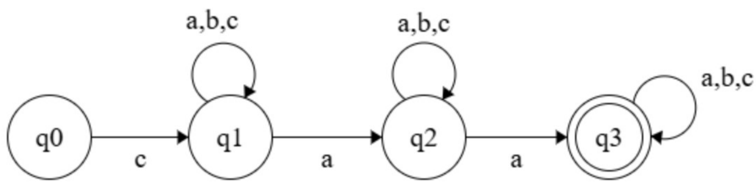
A continuación se presenta un APD que reconoce al lenguaje L_{31}.
La semántica de los estados es la siguiente:

- q0: aún se están leyendo los primeros p símbolos "c"
- q1: se leyó una "a" perteneciente a la subtira w. Aún resta por leer una "b" para que se cumpla la condición $|w|_b>0$
- q2: ya se leyó la "b" obligatoria. Ahora, se leerá el resto de la subtira w
- q3: luego de procesar todos los símbolos de w, el tope de stack con marca "A" confirma que $|w|_a>|w|_b$
- q4: se leen los últimos p símbolos "c"
- q5: se leen los j símbolos "c"



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.**

Puede verificarse que este APD es <u>NO Determinista</u>, ya que existe por ejemplo δ(q2,a,A) y también existe δ(q2,ε,A); es decir, para un mismo estado y mismo tope del stack, una transición que consume entrada y otra que no (con ε).



El autómata es NO Determinista ya que, por ejemplo, en q1 cuando se lee una 'a' o se puede ir a q2 – que sería una de las 2 obligatorias – o quedarse en q1.

Ejercicio 4