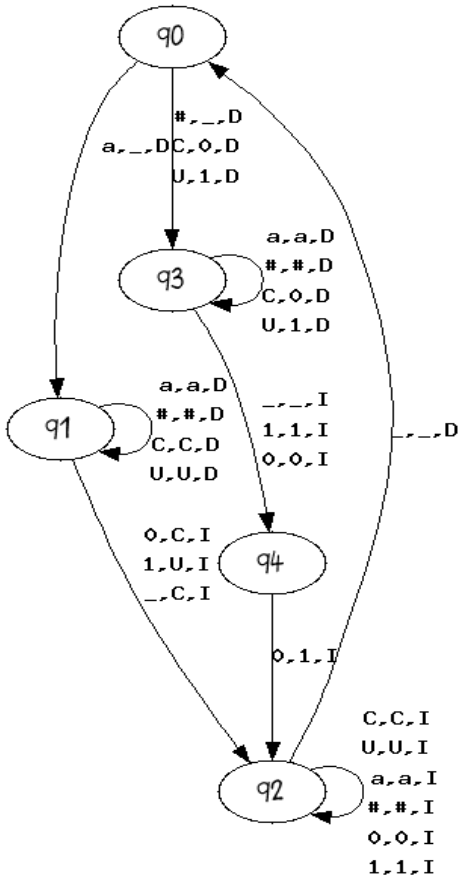
Teoría de Lenguajes – Teoría de la Programación I
Soluciones

Ejercicio 1



Ejercicio 2

Sea $L_2 = \{ a^k (ab)^p a^r / k > 0, p > r \geq 0 \}$

a) Clasifique a L_2 según la Jerarquía de Chomsky.

Sea N la cte. de PL y tomamos $z= a(ab)^{N+1} a^N$, $|z| = 1+2N+2+N = 3N+3 > N$
Se buscarán TODAS las descomposiciones de z=uvw que cumplan:

i. $|uv| <= N$
ii. $|v|>1$

Caso 1:
$u= \varepsilon$ $v=a(ab)^k$ $w=(ab)^{N+1-k} a^N$ con $k>=0$ $2k+1<=N$
$z_i= uv^iw = (a(ab)^k)^i (ab)^{N+1-k} a^N$
tomando i=0 $z_0= (ab)^{N+1-k} a^N$ NO pertenece L_2 porque no hay 2 **a**'s seguidas al principio (al menos una **a** debe haber antes del primer par de (**ab**)'s

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.**

Caso 2:
u= **ε** v=a(ab)^k a w= b(ab)$^{N-k}$ a^N con k>=0 2k+2<=N
z_i= uv^{i}w = (a(ab)^{k}a) ^{i} b(ab)$^{N-k}$ a^N
tomando i=0 z_0= b(ab)$^{N-k}$ a^N NO pertenece L_2 porque comienza en **b**

Caso 3:
u=a(ab)^j v=(ab)^k w=(ab)$^{N-j-k+1}$ a^N con k>0, j>=0, 2(j+k)+1<=N
z_i= uv^{i}w = a(ab)^j (ab)$^{N-j+(i-1)k+1}$ a^N = a(ab)$^{N+(i-1)k+1}$ a^N
tomando i=0 z_0= a(ab)$^{N-k+1}$ a^N NO pertenece a L_2 porque como k>0, la cantidad de pares de (ab) es menor o igual a la cantidad de **a**'s a continuación de esos pares

Caso 4:
u=a(ab)^j v=(ab)^k a w=b(ab)$^{N-j-k}$ a^N con k, j>=0, 2(j+k)+2<=N
z_i= uv^{i}w = a(ab)^j ((ab)^k a) ^{i} b(ab)$^{N-j-k}$ a^N =
tomando i=0 z_0= a(ab)^j b(ab)$^{N-j-k}$ a^N NO pertenece a L_2 porque o bien no hay 2 **a**'s seguidas al principio (al menos una **a** debe haber antes del primer par de (**ab**)'s que es cuando j=0; o bien aparecen dos **b**'s seguidas (j>0)

Caso 5:
u=a(ab)^{j}a v=b(ab)^k w=(ab)$^{N-j-k}$ a^N con k, j>=0, 2(j+k)+3<=N
z_i= uv^{i}w = a(ab)^j a (b(ab)^k) ^{i} (ab)$^{N-j-k}$ a^N = a a (b(ab)^k) ^{i} (ab)$^{N-k}$ a^N
tomando i=0 z_0= aa(ab)$^{N-k}$ a^N NO pertenece a L_2 porque como k>=0, la cantidad de pares de (ab) es menor o igual a la cantidad de **a**'s a continuación de esos pares

Caso 6:
u=a(ab)^{j}a v=b(ab)^{k}a w=b(ab)$^{N-j-k-1}$ a^N con k, j>=0, 2(j+k)+4<=N

z_i= uv^{i}w = a(ab)^j a (b(ab)^k a) ^{i} b(ab)$^{N-j-k-1}$ a^N
tomando i=0 z_0= aab(ab)$^{N-k-1}$ a^N = a(ab)$^{N-k}$ a^N NO pertenece a L_2 porque como k>=0, la cantidad de pares de (ab) es menor o igual a la cantidad de **a**'s a continuación de esos pares

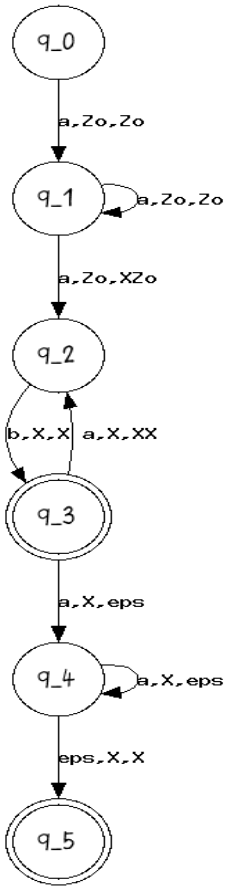
Estas son todas las descomposiciones que satisfacen |uv| <= N y 2) |v|>1
Por lo tanto, el lenguaje L_2 **NO es Regular**.
Para ver que es libre de contexto, se construye una GLC en parte b)

b) Construya una gramática simplificada G_2 / L_2 = L(G_2).

S → aS | aX
X → abXa | abX | ab

Es una gramática libre de contexto (por la estructura de sus reglas) y que además está simplificada al no tener símbolos inútiles, producciones epsilon y no tener producciones unitarias

c) Construya un autómata M_2 / $L_2 = L(M_2)$.¿Es determinista? Justifique. En caso de construir un autómata no determinista, muestre aquellos estados donde se presenta el no determinismo.



Se trata de un APD NO Determinista.
El no determinismo se observa en los siguientes estados y transiciones:
$\delta(q_1,a,Zo) = \{(q_1,Zo), (q_2,XZo)\}$
$\delta(q_3,a,X) = \{(q_2,XX), (q_4,eps)\}$
$\delta(q_4,a,X) = \{(q_4,eps)\}$ y existe $\delta(q_4,eps,X) = \{(q_5,X)\}$

Ejercicio 3
Decir si las siguientes afirmaciones son Verdaderas o Falsas. Justifique adecuadamente cada respuesta.

a) Sean Σ, Δ alfabetos, $L_a \subseteq \Sigma^*$ y $h:\Sigma \rightarrow \Delta$ un homomorfismo.
Si $h(L_a)$ es regular, entonces L_a es regular.

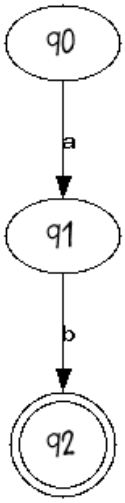
Falso.
Se puede demostrar con el siguiente contra-ejemplo. Sea $L_a = \{0^n1^n / n \geq 0\}$ y $h: \{0,1\} \rightarrow \{2\}^* / h(0)=2$ y $h(1)=2$.
Entonces $h(L_a) = \{2^n2^n / n \geq 0\} = L((22)^*)$ es regular, pero L_a no es regular.

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

Falso.
Se puede demostrar con el siguiente contra-ejemplo. Sea $L_b = \{0, 1\}$ regular y
h: {0,1} → {2}* / h(0)=2 / h(1)=22, entonces $h(L_b) = \{2,22\}$,
pero $h^{-1}(h(L_b)) = \{x / h(x) \in h(L_b) \} = \{0,00,1\}$ distinto a L_b.

c) Sea k un natural, se define $L_k = \{ a^k b^k \}$ (k es "fijo")
Se cumple que R_L (para $L=L_k$) tiene 2k + 1 clases de equivalencia.

Falso.
Se puede demostrar con el siguiente contra-ejemplo. Consideremos L_1, para el cual existe el siguiente AFD mínimo M que lo reconoce:



Donde las clases de equivalencias de R_M son 4 contando el estado pozo. Y por el teorema de Myhill–Nerode sabemos que las clases de equivalencias de R_M coinciden con las de R_L, por lo tanto no se cumple que R_L (para $L=L_1$) tenga 2k+1 = 3.

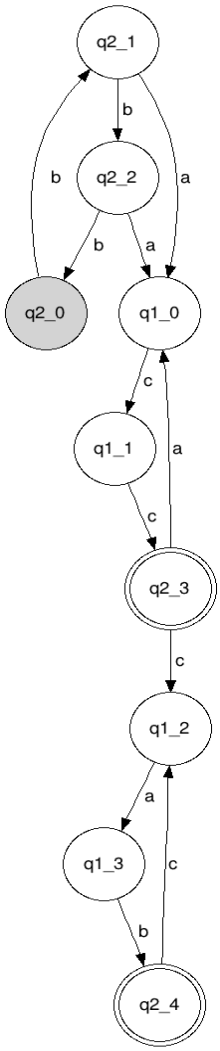
d) Si L_d es un lenguaje libre de contexto, entonces L_d es un lenguaje infinito.

Falso.
Se puede demostrar con el siguiente contra-ejemplo: Sea $L_d = \{a\}$ es libre de contexto, que podemos representar con la siguiente gramática S → a, pero L_d es finito.

Ejercicio 4

Construya un Autómata Finito Determinista de dos cintas que acepte la relación:

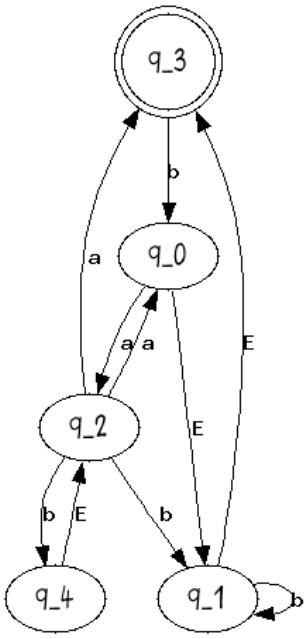
$\{<c^{2k}(ab)^t, b^n a^k c^t> / t \geq 0, k > 0, n > 0, n \bmod 3 \neq 0 \}$



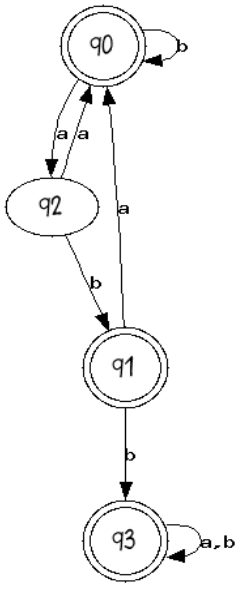
Ejercicio 5 **[Teoría de Lenguajes]**

a) Sean (x,y) ∈ Σ*, xRLy ⇔ ∀ z ∈ Σ* se cumple que (xz ∈ L ∧ yz ∈ L) o bien (xz ∉ L ∧ yz ∉ L), es decir que las concatenaciones xz y yz o bien ambas pertenecen al lenguaje o ambas no pertenecen al lenguaje.

b)
i. Se construye AFND con transiciones Epsilon; el cual se muestra en el siguiente diagrama:



Luego se obtiene el AFD correspondiente y el AFD Mínimo; el cual se muestra en el siguiente diagrama:



ii. Luego, por el Teorema de Myhill-Nerode, las clases de equivalencia de RM (con M el autómata mínimo equivalente) equivalen con las clases de RL. Por lo cual tenemos 4 clases. Planteamos el sistema de ecuaciones correspondiente:

Y0 = ε| Y0b | Y2a | Y1a
Y1 = Y2b
Y2 = Y0a
Y3 = Y1b | Y3(a|b)

Resolviendo el sistema, tenemos una ER para cada clase de RL
Y0 = (b|aa|aba)*
Y1 = (b|aa|aba)*ab
Y2 = (b|aa|aba)*a
Y3 = (b|aa|aba)*abb(a|b)*

iii. Queremos hallar una expresión regular que defina L(M). Dado que las clases de RM asociadas a un estado final definen el conjunto de tiras de Σ* que partiendo del estado inicial llegan al estado final en cuestión, L(M') = L(Y0 | Y1 | Y3), ya que q0,q1 y q3 son los estados finales de M'. Por lo tanto, (b|aa|aba)* | (b|aa|aba)*ab | (b|aa|aba)*abb(a|b)* es una E.R. que define L(M').

Ejercicio 5 [Teoría de la Programación 1]

a) i. FALSO.

Supongamos que **f** fuera computable: existe una macro **MF** que la computa.
Sea **MK** la macro que dado un índice **i**, devuelve del siguiente programa:

```
PROGRAM (X0)
    X1:=EVAL_PROG(i, i)
RESULT(X1)
```

Claramente, este programa no depende de su entrada: converge siempre cuando el programa de índice **i** con entrada **i** lo hace; y diverge siempre en caso contrario.
Sea el siguiente programa de índice **r**:

```
PROGRAM (X0)
    X1:=MK(X0);
    X2:=MF(X1,X1,X0)
RESULT(X2)
```

Este programa converge siempre ya que MK y MF siempre lo hacen. Además:

$$<l_x(r),i>\Rightarrow 1 \text{ sii } \exists\, b \text{ tq. } <l_x(MK(i)), i>\downarrow \text{ y } <l_x(MK(i)), b>\downarrow \text{ sii } <l_x(i), i>\downarrow$$

Pero entonces, el programa de índice **r** computa **q.** Absurdo.

ii. **FALSO**.

El siguiente programa computa **g**:

```
PROGRAM (X0)
    X1:=FST(X0);
    X2:=SND(X0);
    X3:=SND(X2);
    X2:=FST(X2);      X1:=p; X2:=q; X3:=a
    X5:=1;                  pasos
    X10:=0;                       flag y resultado
    WHILE (X10=0) DO
        X4:=0;                    posible b
        WHILE (X4 <> X5) DO
                X6:=EVAL_PROG_STEP(X1,X3,X5);        < lx(p),a >
                X7:=EVAL_PROG_STEP(X2,X4,X5);        < lx(q),b >
                X8:=EVAL_PROG_STEP(X1,X4,X5);        < lx(p),b >
                X9:=EVAL_PROG_STEP(X2,X3,X5);        < lx(q),a >
                IF (FST(X6)=1 &  FST(X7)=1)  THEN
                        X10:=1
                ELSE
                    IF (FST(X8)=1 &  FST(X9)=1) THEN
                            X10:=1
                    FI;
                FI;
                X4:=SUC(X4)
            END;
            X5:=SUC(X5)
    END;
RESULT(X10)
```

b) **VERDADERO.**

Si puedo transformar un problema NP-completo en tiempo polinomial a uno P, puedo solucionar todos los problemas de NP en tiempo polinomial (transformo el problema al NP-completo y éste al P, todo en tiempo polinomial).

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.**