Teoría de Lenguajes
Teoría de la Programación I
(Soluciones)

Consideraciones generales
i) Escriba nombre y C.I. en todas las hojas.
ii) Numere todas las hojas.
iii) En la primera hoja indique el total de hojas.
iv) Comience cada ejercicio en una hoja nueva.
v) Utilice las hojas de un solo lado.
vi) Entregue los ejercicios en orden.

Ejercicio 1

a) Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique adecuadamente cada respuesta.
Dados los siguientes lenguajes:

- L_a lenguaje libre de contexto NO regular
- L_b lenguaje recursivamente enumerable NO libre de contexto
- L_c lenguaje regular con infinitos elementos

i) Tenemos suficiente información para determinar la cantidad de clases de equivalencia de la relación R_L para L_a y L_b.

ii) $L_a \cap L_b$ es un lenguaje libre de contexto.

iii) $L_a \cup L_b$ es un lenguaje recursivamente enumerable.

iv) $L_b \cap L_c$ NO es regular.

i) **Verdadero**. L_a es libre de contexto no regular y L_b es recursivamente enumerable no libre de contexto. Por lo tanto como ni L_a ni L_b son regulares, con lo cual por el teorema de Myhill-Nerode, ambos tienen infinitas clases de equivalencia.

ii) **Falso**. No se cumple para $L_a = \{a^nb^nc^p, n>0, p>=0\}$ y $L_b = \{a^nb^nc^n, n>0\}$, ya que $L_a \cap L_b = L_b$

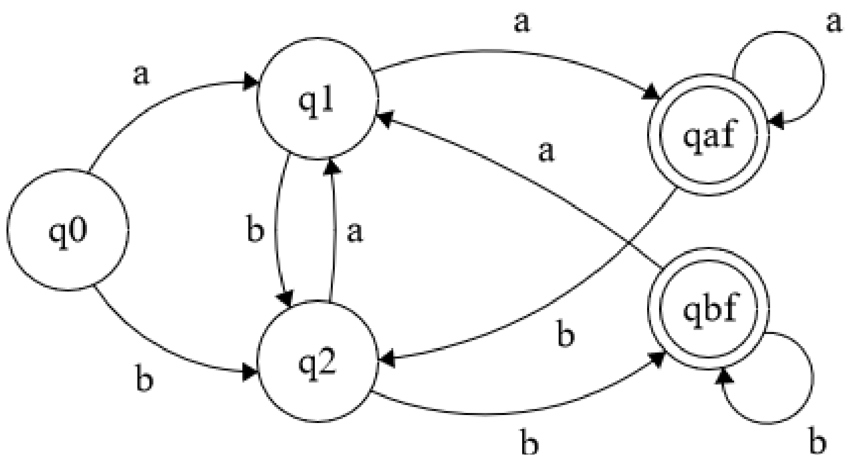
iii) **Verdadero**. Dado que tanto L_a como L_b son lenguajes recursivamente enumerables (L_a es recursivamente enumerable por ser libre de contexto y la Jerarquía de Chomnsky), existen dos gramáticas irrestrictas G_a y G_b con variables iniciales S_a y S_b / $L_a = L(G_a)$ y $L_b = L(G_b)$.
Si se construye una tercera gramática G_c con la regla $S \rightarrow S_a \mid S_b$, y con el resto de las producciones de G_a y G_b, se tiene una gramática irrestricta que genera el lenguaje, por lo que es recursivamente enumerable.

iv) **Falso**. Tomando $L_b = \{a^nb^nc^n, n > 0\}$ y $L_c = \{a^nbc, n>0\}$, tenemos que $L_b \cap L_c = \{abc\}$ es regular porque es finito.

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.

b) Sea el lenguaje L_1 = L ((a|b)*(aa|bb))

i) Construya un autómata finito M_1 / L_1 = L(M_1). ¿Es mínimo? Justifique.



Para ver si el autómata es mínimo, aplicaremos el algoritmo de minimización y veremos si el autómata resultante es el mismo:

Π_0 : [q0 q1 q2] [qa qf]

Π_1 : [q0] [q1] [q2] [qa] [qf]

Π_2 : [q0] [q1 [q2] [qa] [qf]

Como Π_2 es igual a Π_1 termina el algoritmo, y los estados resultantes son los mismos que teníamos antes por lo que el autómata original **es mínimo.**

ii) Construya una gramática simplificada G_1 / L_1 = L(G_1). Justifique porqué está simplificada.

Dado que el lenguaje es regular, se propone una gramática lineal derecha con las siguientes reglas:

S –> aQ_1 | b Q_2
Q_1 –> bQ_2 | a Q_{af} | a
Q_2 –> aQ_1 | b Q_{bf} | b
Q_{af} –> aQ_{af} | b Q_2 | a
Q_{bf} –> bQ_{bf} | a Q_1 | b

La gramática está simplificada porque no tiene producciones unitarias, ni producciones épsilon, y además todas sus variables son útiles.

iii) A partir de la definición de la relación R_L dada en el curso, muestre 2 tiras diferentes de Σ^* que estén en la misma clase de equivalencia según R_{L1} y 2 que estén en distintas. Justifique.

Las tiras *aa* y *aaa* pertenecen a la misma clase de equivalencia porque al ser procesadas en el autómata que reconoce a L_1 (el de la parte i), terminan en el mismo estado, por lo que pertenecen a la misma clase de R_M y por lo tanto a la misma clase de R_L (al ser el autómata completo y mínimo).

Por otro lado, se pueden tomar las tiras *b* y *a*, que no pertenecen a la misma clase de equivalencia ya que por ejemplo existe z = *b* tal que *b*z pertenece a L_1 pero *a*z no pertenece a L_1.

Ejercicio 2 **[Teoría de Lenguajes]**

Sea x una tira que pertenece a $\{a,b,c\}^*$. Se dice que hay una **inversión** entre dos símbolos distintos cualesquiera de la tira x (no necesariamente consecutivos) si el primero de esos dos símbolos que aparece en la tira figura después que el segundo en el alfabeto (***ba*** es una inversión, y ***ab*** no).
A modo de ejemplo, la tira ***cbba*** tiene 5 inversiones: c con la primera b, c con la segunda b, c con a, la primera b con a y la segunda b con a.

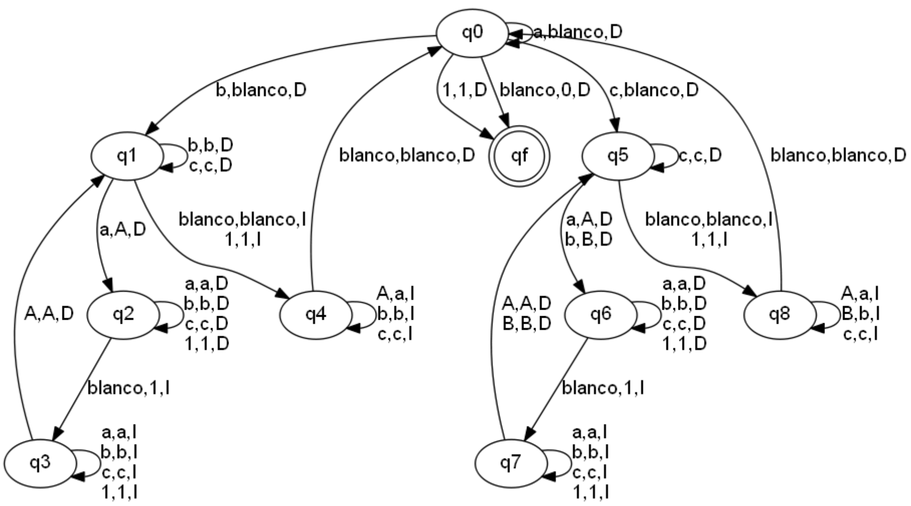
Construya una Máquina de Turing que compute una función que recibe una tira x de $\{a,b,c\}^*$
y devuelve el número de inversiones en x, en notación unaria (tira de unos consecutivos de largo igual al número de inversiones en x).
Si en la tira de entrada no hay inversiones, la máquina devuelve 0.

Ejemplos:

Entrada	Salida
bab	1
acba	111
aabc	0
ccacb	11111
ε	0

Nota: en la configuración final de la MT, en la cinta sólo debe quedar el resultado.

La idea de la MT es procesar la tira de izquierda a derecha buscando para el símbolo actual todas las inversiones que hay con los símbolos que le siguen en la tira. Si el símbolo actual es una a no hay inversiones y se sustituye por blanco de inmediato; si el símbolo actual es una b se computan las inversiones con todas las a's que le siguen y si el símbolo actual es una c se computan inversiones con todas las a's y b's que le siguen.



Ejercicio 3

Sea el lenguaje $L_3 = \{ x_1x_2...x_k\#1^k \ / \ x_i \in \{01, 010\}, \ k \geq 0 \}$

a) Clasifique L_3 según la Jerarquía de Chomsky
b) Construya una gramática G_3 / $L_3 = L(G_3)$ de acuerdo con la categoría del lenguaje.
c) Construya un autómata M_3 / $L_3 = L(M_3)$ de acuerdo con la categoría del lenguaje. ¿Es determinista? Justifique.

a) L_3 es libre de contexto no regular. Que es libre de contexto se demuestra con la gramática de la parte b) o el APD de la parte c), y que no es regular aplicando el CR del PL para lenguajes regulares.

Dado N la constante del Pumping Lema, se elige $z = (01)^N\#1^N$
Las descomposiciones z = uvw a estudiar que cumplen $|uv| \leq N$ y $|v| \geq 1$ son:

Caso 1.-
$u = (01)^p$ $\quad p \geq 0$
$v = (01)^q$ $\quad q \geq 1 \quad 2p+2q \leq N$
$w = (01)^{N-p-q}\#1^N$

$z_i = (01)^p\,(01)^{qi}\,(01)^{N-p-q}\#1^N = (01)^{N+(i-1)q}\#1^N$
Eligiendo i = 0, queda $z_0 = (01)^{N-q}\#1^N$, quedando cantidad de pares 01 anterior al # menor a la cantidad de 1s que hay después del #, lo cual no puede pasar. Por otro lado, puede verse que antes del #, viene una cantidad de símbolos que pueden ser entre 2 y 3 veces la cantidad de 1s después del # que son los posibles largos de cada x_j. Por eso z_0 no pertenece a L_3.

Caso 2.-
$u = (01)^p0$ $\quad p \geq 0$
$v = 1(01)^q$ $\quad q \geq 0 \quad 2p+2q+2 \leq N$
$w = (01)^{N-p-q-1}\#1^N$

$z_i = (01)^p\,0\,(1(01)^q)^i\,(01)^{N-p-q-1}\#1^N = (01)^p\,0\,(1(01)^q)^i\,(01)^{N-p-q-1}\#1^N$

Eligiendo i = 2, queda $z_2 = (01)^p\,01(01)^q\,1(01)^q\,(01)^{N-p-q-1}\#1^N$
Acá el problema es que la tira z_2 tiene dos 1's consecutivos, lo cual no puede ser por la estructura del lenguaje L_3. Por eso z_2 no pertenece a L_3.

Caso 3.-
$u = (01)^p$ $\quad p \geq 0$
$v = (01)^q0$ $\quad q \geq 0 \quad 2p+2q+1 \leq N$
$w = 1(01)^{N-p-q-1}\#1^N$

$z_i = (01)^p\,((01)^q0)^i\,1(01)^{N-p-q-1}\#1^N$

Eligiendo i = 0, queda $z_0 = (01)^p\,1(01)^{N-p-q-1}\#1^N$
- si p=0 $\quad 1(01)^{N-q-1}\#1^N$ la tira z_0 comienza en 1
- si p>0 $\quad (01)^p\,1(01)^{N-p-q-1}\#1^N$ la tira z_0 tiene dos 1's consecutivos

En cualquier caso, z_2 no pertenece a L_3.

Caso 4.-
$u = (01)^p0$ $p > 0$
$v = (10)^q$ con $q \geq 1$ $2p+2q+1 \leq N$
$w = 1(01)^{N-p-q-1}\#1^N$

$z_i = (01)^p\, 0\, ((10)^q)^i\, 1(01)^{N-p-q-1}\#1^N$

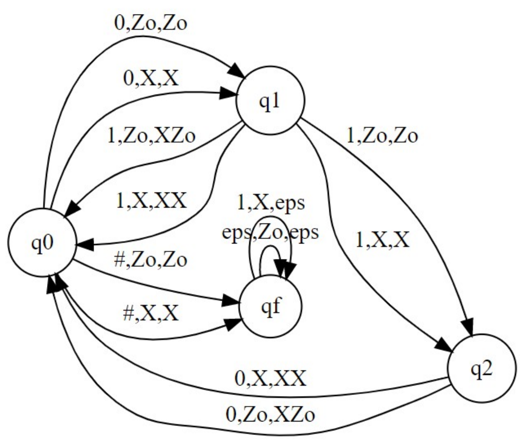
Eligiendo i = 0, queda $z_0 = (01)^p\, 01\, (01)^{N-p-q-1}\#1^N = (01)^{N-q}\#1^N$
Cómo $q \geq 1$, la cantidad de pares (01) es menor que la cantidad de 1's después del # con lo cual no es válido. Por eso z_2 no pertenece a L_3.

Estas son todas las descomposiciones de z = uvw que cumplen $|uv| \leq N$ y $|v| \geq 1$ con lo cual L_3 NO es regular.
Como se dijo al comienzo, en la parte b) se muestra que es libre de contexto construyendo una gramática.

b) Una posible gramática libre de contexto que genera L_3 puede ser la que tenga las siguientes reglas:

S → 01 S 1 | 010 S 1 | #

c) Se construye un autómata con stack



Es un APD No determinista, ya que se tiene por ejemplo $\delta(q1,1,X) = \{(q0,XX), (q2,X)\}$

Ejercicio 2 **[Teoría de la Programación I]**

i) Explique la diferencia entre un problema P y uno NP
ii) Sean los conjuntos de naturales A y B tales que A es decidible y B es r.e. pero no decidible. Puede afirmar que B-A necesariamente es r.e.?

i) P es una clase de complejidad que representa el conjunto de todos los problemas de decisión que se pueden resolver en tiempo polinómico.
Es decir, dada una instancia del problema, la respuesta "sí" o "no" se puede decidir en tiempo polinómico.

Por otro lado, NP es una clase de complejidad que representa el conjunto de todos los problemas de decisión para los cuales las instancias donde la respuesta es "si" tienen pruebas que se pueden verificar en tiempo polinómico.

ii) **Si.** Observar que $B-A = B \cap A^c$. Como A es un conjunto decidible, su complemento también lo es (teorema 17) y por lo tanto también es r.e. (teorema 18). Como los conjuntos r.e. son cerrados bajo la intersección (teorema 13), $B \cap A^c$ es r.e.

Nota: los números de teoremas hacen referencia a los apuntes de Computabilidad que están en la página del curso.