Teoría de Lenguajes

2do. Parcial – Curso 2018
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 [Evaluación individual del obligatorio]

a)
i) Verdadero
ii) Falso
iii) Falso
iv) Verdadero

b)
S → 'a' S 'a'
S → 'b'

c)
i) S
ii) S
iii) No despliega nada.
iv) La tira no pertenece al lenguaje.

Ejercicio 2

Sean los siguientes lenguajes.
L_a R.E. NO Regular
L_b Libre de Contexto NO Regular
$L_c = \{a^n, n>3\}$
L_d Regular NO Finito

Indique si las siguientes afirmaciones son verdaderas o falsas, justificando en cada caso.

a) $L_a \cap L_b$ es r.e. NO Libre de Contexto
Falso. Sea $L_a = \{a^kb^k , k >0\}$ y $L_b = \{ c^kd^k , k >0\}$ donde ambos son Libres de Contexto NO Regulares. En particular, L_a es R.E. por Jerarquía de Chomsky. A su vez, $L_a \cap L_b = \Phi$ (Vacío) y el Lenguaje Vacío es Regular y por lo tanto Libre de Contexto por Jerarquia de Chomsky.

b) $L_a – L_b$ es Regular
Falso. Si consideramos los mismos lenguajes La y Lb de la parte anterior, vemos que $L_a – L_b = L_a$, el cual es Libre de Contexto NO Regular.

c) $L_a \cup L_d^c$ es Libre de Contexto
Falso. Sea $L_a = \{a^kb^kc^k, k >0\}$ y $L_d = \sum^* - \{abc\}$, siendo $\sum=\{a,b,c\}$.
Entonces $L_d^c =\{abc\}$. Entonces, $L_a \cup L_d^c = L_a$, que es R.E. NO Libre de Contexto (demostrado en teórico).

d) $L_c^c \cap L_a$ es Regular
Falso. Sea $L_a = \{a^kb^k , k >0\}$ y $L_c^c = \{ x / x \in\{a,b\}^*$ donde x no es de la forma $a^n \} \cup \{ \varepsilon ,a, aa, aaa\}$; por lo tanto $L_c^c \cap L_a = L_a$, el cual no es Regular.

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.

Ejercicio 3

Sea

$L_3 = \{ a^n b^p c^k$, con $k>p \vee k>n$, $p,n \geq 0\}$

a) Construya una gramática simplificada G_3 / $L_3 = L(G_3)$. Justifique por qué la gramática propuesta está simplificada.

Primero construimos una gramática G_3 / $L_3 = L(G_3)$

G_3:

$S \rightarrow A\ S_1 \mid S_2$
$A \rightarrow \varepsilon \mid a\ A$
$S_1 \rightarrow b\ S_1\ c \mid C$
$C \rightarrow c\ C \mid c$
$S_2 \rightarrow a\ S_2\ c \mid B\ C$
$B \rightarrow \varepsilon \mid b\ B$

Dado que no es una gramática simplificada, la simplificamos eliminando primero las producciones épsilon:

G_3':

$S \rightarrow A\ S_1 \mid S_2 \mid S_1$
$A \rightarrow a\ A \mid a$
$S_1 \rightarrow b\ S_1\ c \mid C$
$C \rightarrow c\ C \mid c$
$S_2 \rightarrow a\ S_2\ c \mid B\ C \mid C$
$B \rightarrow b\ B \mid b$

Luego eliminamos las producciones unitarias:

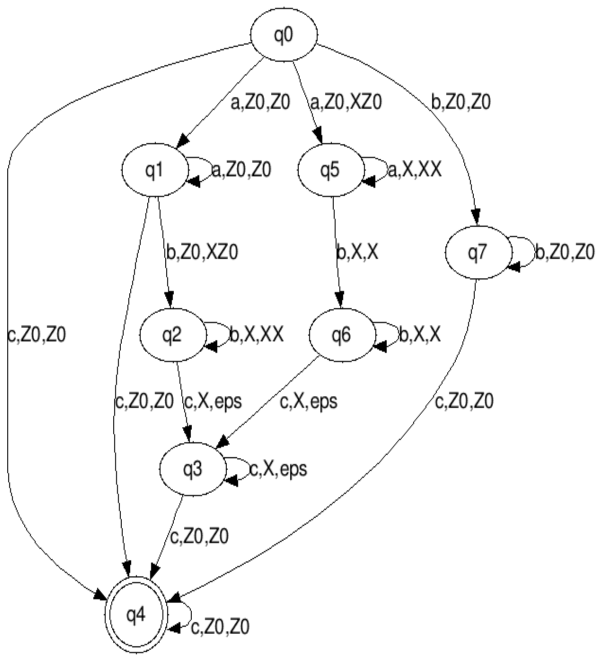
G_3'':

$S \rightarrow A\ S_1 \mid b\ S_1\ c \mid a\ S_2\ c \mid B\ C \mid c\ C \mid c$
$A \rightarrow a\ A \mid a$
$S_1 \rightarrow b\ S_1\ c \mid c\ C \mid c$
$C \rightarrow c\ C \mid c$
$S_2 \rightarrow a\ S_2\ c \mid B\ C \mid c\ C \mid c$
$B \rightarrow b\ B \mid b$

y como todas las variables son positivas y alcanzables, la última gramática obtenida (G_3'') esta simplificada, y cumple que $L_3 = L(G_3'')$

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) Construya un autómata M_3 / $L_3 = L(M_3)$. ¿Es determinista? Justifique.



El autómata es no determinista, ya que en el estado q_0 tenemos dos transiciones con la misma combinación entrada-stack; $\delta(q_0,a,Z_0) = \{ (q_1,Z_0), (q_5,XZ_0)\}$

Ejercicio 4
Sea

$L_4 = \{ a^n b^m c^p,\ \ p=\min\{n,m\},\ m,n \geq 0\} \}$

a) Clasifique L_4 según la Jerarquía de Chomsky.

L_4 es un lenguaje Recursivamente Enumrable, lo cual se demuestra con la G.I. de la parte b), y **no** es un lenguaje Libre de Contexto, lo cual se demuestra aplicando el contra-recíproco del Pumping Lema para lenguajes libres de contexto.

Dado N constante del PL, elegimos $z = a^N b^N c^N$, $z \in L_4$, $|z| = 3N \geq N$
Estudiamos todas las descomposiciones de z=uvwxy que cumplen |vx|>0 y
$|vwx| \leq N$

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

Familia 1:
$u=a^p$
$v=a^q$ $|vx| = q+s \geq 1$
$w=a^r$ $|vwq| = q+r+s \leq N$
$x=a^s$
$y=a^{N-p-q-r-s}b^Nc^N$

$z_i = a^p\,(a^q)^i\,a^r\,(a^s)^i\,a^{2N-p-q-r-s}b^Nc^N = a^{p+qi+r+si+N-p-q-r-s}b^Nc^N = a^{N+(q+s)(i-1)}b^Nc^N$

Eligiendo $i=0$ $z_0 = a^{N-(q+s)}b^Nc^N$

Entonces tenemos que $z_0 \notin L_4$ ya que como $q+s \geq 1$, la cantidad de **a**'s es menor que la cantidad de **b**'s y la cantidad de **c**'s deja de ser el min {cant_a,cant_b} al permanecer con la misma cantidad de **b**'s.

Familia 2
Es análoga a la Familia 1, pero el razonamiento es con las **b**'s

Familia 3
Esta familia merece un un razonamiento diferente, porque la cantidad que varía es la de **c**'s. En este caso se podría tomar tanto i=0, como i>1, ya que la cantidad de **c**'s va a diferir de la de **a**'s y **b**'s (que permanecen fijas), con lo cual $z_i \notin L_4$.

Familia 4
La descomposición de esta familia sería:
$u=a^{N-p-q}$
$v=a^p$ $|vx| = p+s \geq 1$
$w=a^qb^r$ $|vwq| = p+q+r+s \leq N$
$x=b^s$
$y=b^{N-r-s}c^N$

$z_i = a^{N-p-q}\,(a^p)^i\,a^qb^r(b^s)^i\,b^{N-r-s}c^N = a^{N+(i-1)p}b^{N+(i-1)s}c^N$

Tomando i=0, $z_0=a^{N-p}b^{N-s}c^N$
Si p>0 y s>0, la cantidad de **c**'s permanece *fija* y mayor que la cantidad de **a**'s y la cantidad de **b**'s.
Si alguno de los 2 (p ó s) es igual a 0 (ambas no pueden serlo por contradecir |vx|>0) – supongamos s=0 - la cantidad de **a**'s es menor que la cantidad de **b**'s y la cantidad de **c**'s deja de ser el min {cant_a,cant_b} al permanecer con la misma cantidad de **b**'s. Entonces $z_0 \notin L_4$.

Familia 5
La descomposición de esta familia sería:
$u=a^Nb^{N-p-q}$
$v=b^p$ $|vx| = p+s \geq 1$
$w=b^qc^r$ $|vwq| = p+q+r+s \leq N$
$x=c^s$
$y=c^{N-r-s}$

$z_i = a^N\,b^{N-p-q}(b^p)^i\,b^qc^r(c^s)^ic^{N-r-s} = a^Nb^{N+(i-1)p}c^{N+(i-1)s}$

Acá la discusión es la siguiente:
- si p=0, entonces s>0, y tomando i=2, $z_2=a^Nb^Nc^{N+s}$ y la cantidad de **c**'s es mayor que la de **a**'s y **b**'s (que quedan fijas)
- si s = 0, entonces p>0, y tomando i=0, $z_0=a^Nb^{N-p}c^N$ y la cantidad de **c**'s es mayor que la de **b**'s (que a su vez es menor que la de **a**'s)

- si p>0 y s>0, tomando i=2, $z_2=a^Nb^{N+p}c^{N+s}$ y por lo pronto la cantidad de **c**'s es mayor que la de **a**'s (que queda fija)
Entonces según sea el caso, $z_i \notin L_4$

Familia 6
La descomposición de esta familia sería:
$u=a^{N-p-q-r}$
$v=a^p$ $|vx| = p+r+s \geq 1$
$w=a^q$ $|vwq| = p+q+r+s \leq N$
$x=a^rb^s$
$y=b^{N-s}c^N$

$z_i = a^{N-p-q-r}(a^p)^i a^q(a^rb^s)^i b^{N-s}c^N$
Asumimos que r >0 y s> (sino caen en los casos anteriores).
Tomando i=2, $z_2 = a^{N+p-r}a^rb^sa^rb^Nc^N$ donde se mezclan **a**'s y **b**'s; y $z_2 \notin L_4$.

Familias 7,8 y 9
La 7 es análoga a la Familia 6.
La 8 y 9 son también análogas, tomando i=2 se mezclan **b**'s y **c**'s.

Como hemos estudiado todas las descomposiciones posibles que cumplen, $|vx|>0$ y $|vwx| \leq N$ se concluye que L_4 **no** es un lenguaje libre de contexto.

Como se dijo al comienzo, en la parte b) se construirá una GI y con eso quedará demostrado que L_4 es recursivamente enumerable.

b) Construya una gramática G_4 / $L_4 = L(G_4)$.

Se dará una gramática irrestricta G_4 que genera L_4.

La idea de la GI propuesta es generar primero tiras de la forma $(aD)^n$ $(bE)^m$ con n,m≥ 0 y luego por cada pareja (D,E) generar una c al final de la tira. Si quedan D's o E's sobrantes sin emparejar se eliminan sin generar c's. Se emplean delimitadores de inicio y fin de tira que se eliminan entre sí al final.

G_4 = ({S,I,A,B,C,D,E,F}, {a,b,c},P,S}

P tiene las siguientes producciones:

S → IABF // producción inicial con delimitadores
A → aDA | ε // genera $(aD)^n$ con n >=0
B → bEB | ε // genera $(bE)^m$ con m >=0. Obtenemos una tira IaDaD…aDbEbE…bEF.
Da → aD // D se desplaza a la derecha ignorando las a's
Db → bD // D se desplaza a la derecha ignorando las b's
Dc → cD // D se desplaza a la derecha ignorando las c's
aE → Ea // E se desplaza a la izquierda ignorando las a's
bE → Eb // E se desplaza a la izquierda ignorando las b's
cE → Ec // E se desplaza a la izquierda ignorando las c's
DE → c // por cada par (a,b) se genera una c
DF → F // si hay mas a's que b's se eliminan las D's sobrantes
IE → I // si hay mas b's que a's se eliminan las E's sobrantes
Ia → aI // el delimitador izquierdo se desplaza a la derecha ignorando a's
Ib → bI // el delimitador izquierdo se desplaza a la derecha ignorando b's
cF → Fc // el delimitador derecho se desplaza a la izquierda ignorando c's
ca → ac // se ordenan correctamente las c's a la derecha de las a's
cb → bc // se ordenan correctamente las c's a la derecha de las b's
IF → ε // se eliminan los delimitadores generando una tira válida del lenguaje

Observar que el desplazamiento a la derecha del delimitador I se ve frenado cuando encuentra una c. A su vez el desplazamiento a la izquierda del delimitador F es frenado al encontrar a's o b's. Esto implica que si hay c's a la izquierda de a's o b's, la única posibilidad de que se encuentren I y F y se derive una tira de terminales puros es que se corrija primero el orden de todos los pares invertidos ca y cb, por lo que se garantiza que no se derivan tiras que no pertenecen al lenguaje.

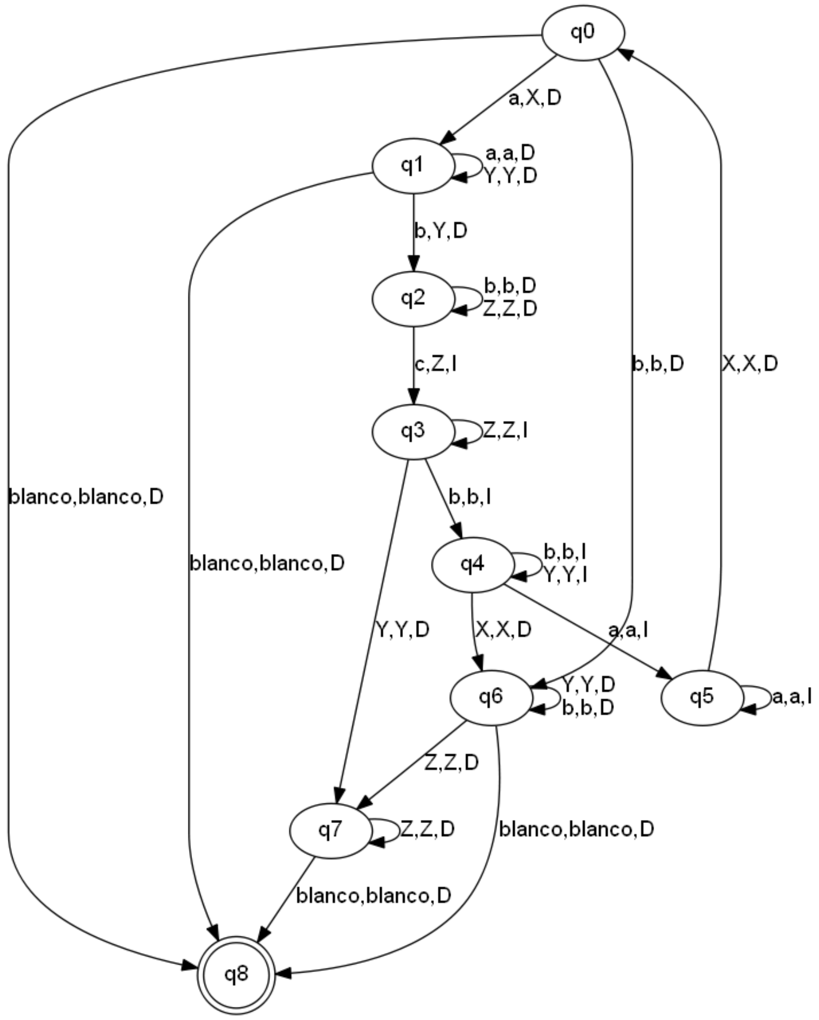
Otra variante a la gramática propuesta:

S → M F
M → a M b C | A | B // M genera el mínimo de letras, luego A o B decantan para un // lado o el otro.
A → a A | ε
B → b B | ε
C b → b C // genero las C's al final.
C F → Fc
F → ε

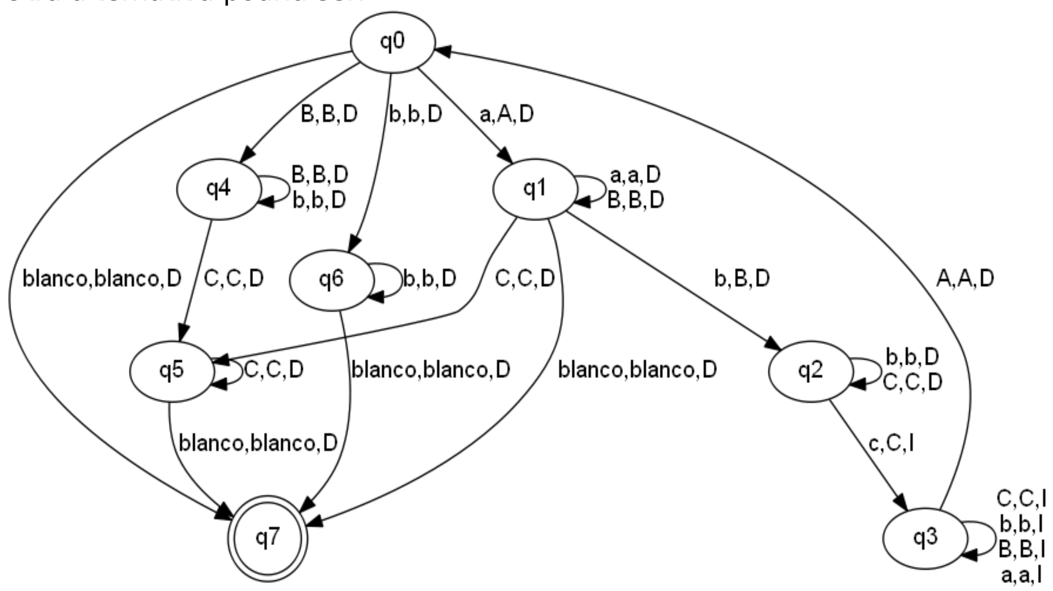
c) Construya un autómata M_4 / $L_4 = L(M_4)$.

Idea de la MT M_4 propuesta:

Si el símbolo inicial leído es un blanco (tira de entrada vacía), se va directo al estado final de aceptación.
Si el símbolo inicial leído es una b, se recorre la cinta verificando que no hayan c's, en cuyo caso se va al estado final de aceptación.
Si el símbolo inicial leído es una a, se marca con X y se recorre la cinta buscando una b o un blanco. En el caso de encontrar un blanco (tira compuesta solo por a's), se va al estado final de aceptación. Si se encuentra una b, se marca con Y dicha celda. Esto significa que en una tira válida debe existir al menos una c a la derecha de la b encontrada. Se recorre la cinta hacia la derecha buscando una c. Si se encuentra una c, se marca con Z y el cabezal se mueve hacia la izquierda, para repetir el proceso buscando la primer a no marcada de otro par (a,b). Si en este recorrido a la izquierda se encuentra una Y antes que una b, esto quiere decir que no hay más b's en la tira de entrada, por lo que ser recorre la cinta hacia la derecha buscando un blanco que indique que no hayan más c's sin marcar (ya que en este caso p= m = min{n,m}), si se lo encuentra se está ante una tira válida y se va al estado final de aceptación. Si en el recorrido a la izquierda se encuentra una X antes que una a, esto significa que no hay más a's en la tira de entrada, por lo que ser recorre la cinta hacia la derecha buscando un blanco que indique que no hayan más c's sin marcar (ya que en este caso p =n = min{n,m}), si se lo encuentra se está ante una tira válida y se va al estado final de aceptación.



Otra alternativa podría ser:



Ejercicio 5

Sea

$L_5 = \{ 0^n 1^p 2^k$, con $k,p > 0$, $n \geq 0\}$

Construya una gramática lineal izquierda simplificada G_5 / $L_5 = L(G_5)$. Justifique por qué la gramática propuesta está simplificada.

G_5:

$S \rightarrow S_2$
$S_2 \rightarrow S_1\ 2 \mid S_2\ 2$
$S_1 \rightarrow S_0\ 1 \mid S_1\ 1$
$S_0 \rightarrow \varepsilon \mid S_0\ 0$

Dado que esta gramática no está simplificada, procedemos en primer lugar a eliminar las producciones épsilon:

G_5':

$S \rightarrow S_2$
$S_2 \rightarrow S_1\ 2 \mid S_2\ 2$
$S_1 \rightarrow S_0\ 1 \mid S_1\ 1 \mid 1$
$S_0 \rightarrow S_0\ 0 \mid 0$

Luego eliminamos las producciones unitarias:

G_5'':

$S \rightarrow S_1\ 2 \mid S_2\ 2$
$S_2 \rightarrow S_1\ 2 \mid S_2\ 2$
$S_1 \rightarrow S_0\ 1 \mid S_1\ 1 \mid 1$
$S_0 \rightarrow S_0\ 0 \mid 0$

y como todas las variables son positivas y alcanzables, la última gramática obtenida (G_5'') esta simplificada, y cumple que $L_5 = L(G_5'')$