Teoría de Lenguajes
Teoría de la Programación 1
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
Indique si las siguientes afirmaciones son verdaderas o falsas, justificando en cada caso.

a) Sea L_a un lenguaje Libre de Contexto no Regular. Entonces L_a^r (reverso de L_a) es Libre de Contexto no Regular.

Verdadero. Supongo que L_a^r es Regular. Como los lenguajes regulares son cerrados bajo reverso, entonces L_a sería regular. Pero L_a es NO Regular, entonces L_a^r es NO Regular. Es Libre de Contexto porque los Libres de Contexto son cerrados bajo reverso. Entonces L_a^r es LC NO Regular.

b) Sea L_b un lenguaje Libre de Contexto no Regular. Entonces existen L_1 y L_2 Regulares (distintos de vacío) tales que $L_1 \subset L_b \subset L_2$.

Verdadero. Todo lenguaje no vacío, Libre de Contexto no Regular tiene incluido un lenguaje formado por ejemplo por una de sus tiras ($w \in L_b$, $\{w\} \subseteq L_b$) y a su vez $L_b \subseteq \Sigma^*$, siendo Σ el alfabeto sobre el cual están definidas sus tiras.

c) Se cumple que $h^{-1}(h(L_c)) = L_c$ para cualquier homomorfismo definido sobre el alfabeto de L_c.

Falso. Considere el homomorfimsmo: h(0)=0, h(1)=0 sobre el alfabeto {0,1}.
Sea el $L_c = \{0^n1^n\}$ $h(L_c)= \{0^{2n}\}$.
Se puede ver que $h^{-1}(h(L_c))= \{x$ / x es de la forma $(0|1)^{2n}\}$, que es distinto de L_c.

d) El lenguaje L_b generado por la gramática G_b=(V,T,P,S) con V={S,A,X},T={a,b},
P = { S → Sb | Ab
A → XaaA | aa | aaA
Xa → aX
Xb → bb }
es Libre de Contexto

Verdadero.
Se van a hacer derivaciones para intentar descubrir la estructura de las tiras generadas por la gramática; ya que aunque tenga reglas con terminales del lado derecho no significa que el lenguaje generado no sea libre de contexto o regular.

Con S → Sb generamos cualquier cantidad de b's (al menos 1). Luego pasamos a generar a's (con al menos 1 b con la regla S → Ab) S ⇒* Abbbbbbb

Con las reglas A → XaaA | aaA se generan cantidad par de a's y una marca 'X' o no; se generan cosas del tipo A ⇒* XaaaaXaa......XaaaaaaXaaaa
Tenemos entonces: S ⇒* XaaaaXaa......XaaaaaaXaaaabbb...bbbb

Con la regla $Xa \rightarrow aX$ se mueven las 'X' a la derecha
Con la regla $Xb \rightarrow bb$ se convierten las X's en b's

De donde el lenguaje generado es el dado par la siguiente expresión regular:
aa(aa)*bb*
con lo cual el lenguaje es Regular y por Jerarquía de Chomsky es Libre de Contexto

Ejercicio 2
Sea
$L_2 = \{ b^p a^q b^r$, con $q>p+r$; $q \geq 1$, $p,r \geq$ ▯ }

a) Clasifique L_2 según la Jerarquía de Chomsky.
b) Construya una gramática simplificada G_2 / $L_2 = L(G_2)$.
c) Construya un autómata M_2 / $L_2 = L(M_2)$. ¿Es determinista? Justifique.

a) Se va a demostrar que el lenguaje L_2 no es Regular.
Demostración por contra-recíproco del Pumping Lemma para lenguajes regulares.
Sea N la constante del PL y tomamos $z = b^N a^{N+1}$; $z \in L_2$, $|z| = 2N + 1 \geq N$
$u = b^{N-j-t}$
$v = b^j$ $j \geq 1$
$w = b^t a^{N+1}$
Entonces, sea i=2, $uv^2w = b^{N-j-t} b^{j*i} b^t a^{N+1} = b^{N+j} a^{N+1} \notin L_2$ porque cantidad de b's es mayor o igual que cantidad de a's (porque $j \geq 1$).

Esta es la única descomposición de z=uvw que cumple $|uv| \leq N$ y $|v| \geq 1$.
Entonces L_2 NO es Regular.
Se verá que es Libre de Contexto al construir una Gramática Libre de Contexto G_2 / $L_2=L(G_2)$

b) Sea G_2 dada por las siguientes producciones
$S \rightarrow PQR \mid PQ \mid QR \mid Q$
$P \rightarrow bPa \mid ba$
$Q \rightarrow Qa \mid a$
$R \rightarrow aRb \mid ab$

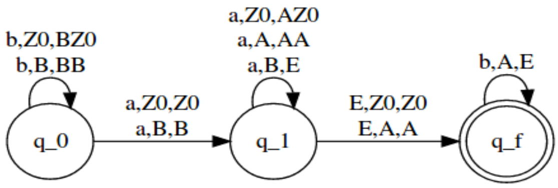
No está simplificada porque tiene una producción unitaria.
La eliminamos con el algoritmo y entonces quedan las siguientes producciones:

$S \rightarrow PQR \mid PQ \mid QR \mid Qa \mid a$
$P \rightarrow bPa \mid ba$
$Q \rightarrow Qa \mid a$
$R \rightarrow aRb \mid ab$

Esta gramática ahora si está simplificada porque no tiene producciones épsilon, ni producciones unitarias y todos sus símbolos son útiles.

c) Se construye un APD M_2 / $L_2 = L(M_2)$

Es un APD NO determinista, ya que por lo pronto existen $\delta(q\_1,\varepsilon,Zo)$ y $\delta(q\_1,a,Zo)$



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
$L_3 = \{ a^i b^j c^k$, con $k=i*j$; $i,j \geq 1\}$

a) Clasifique L_3 según la Jerarquía de Chomsky.
b) Construya una gramática G_3 / $L_3 = L(G_3)$.
c) Construya un autómata M_3 / $L_3 = L(M_3)$.

a) Clasifique L_3 según la Jerarquía de Chomsky.

L_3 es Recursivamente Enumerable y no Libre de Contexto. Se demuestra que es RE por la Gramática Irrestricta y la Máquina de Turing de las partes b) y c) respectivamente. Y que no es Libre de Contexto utilizando el PL para Libres de Contexto.

Dado N la cte. del PL, elegimos $z = a^N b^N c^{N*N}$, perteneciente a L_3 y con $|z| = N*N+2N \geq N$

Analizamos todas las descomposiciones de z = uvwxy para aquellos casos que cumplen las condiciones $|vx| > 0$ y $|vwx| \leq N$

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

Caso 1:
$u = a^p$
$v = a^q$ $\qquad q + s \geq 1$
$w = a^r$ $\qquad q+r+s \leq N$
$x = a^s$
$y = a^{N-p-q-r-s} b^N c^{N*N}$

$z_i = a^p (a^q)^i a^r (a^s)^i a^{N-p-q-r-s} b^N c^{N*N}$
$z_i = a^{N+(q+s)(i-1)+} b^N c^{N*N}$
Eligiendo i=2, $z_2 \notin L_3$, porque la multiplicación de la cantidad de a y b: (N+q+s)*N, no es igual a la cantidad de c's, que es N*N (al ser $q + s \geq 1$).

En forma análoga se demuestra:

- **Caso 5:** eligiendo i=2, $z_2 \notin L_3$, porque la multiplicación de la cantidad de a y b: N*(N+q+s), no es igual a la cantidad de c: N*N.
- **Caso 9**: eligiendo i=2, $z_2 \notin L_3$, porque la multiplicación de la cantidad de a y b: N*N, no es igual a la cantidad de c: N*N+q+s.

Caso 3:
$u = a^{N-p-q}$

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.

$v = a^p$ $p + s \geq 1$
$w = a^q b^r$ $p + q + r + s \leq N$
$x = b^s$
$y = b^{N-r-s} c^{N*N}$

$z_i = a^{N-p-q}\ (a^b)^i\ a^q b^r\ (b^s)^i\ b^{N-r-s} c^{N*N}$
$z_i = a^{N+p(i-1)}\ b^{N+s(i-1)} c^{N*N}$
Eligiendo i=2, $z_2 \notin L_3$, porque la multiplicación de la cantidad de a y b: (N+p)*(N+s), no es igual a la cantidad de c's que es N*N.

Caso 7:
$u = a^N b^{N-p-q}$
$v = b^p$ $p + s \geq 1$
$w = b^q c^r$ $p+q+r+s \leq N$
$x = c^s$
$y = c^{N*N-r-s}$

$z_i = a^N b^{N-p-q}\ (b^p)^i\ b^q c^r\ (c^s)^i\ c^{N*N-r-s}$
$z_i = a^N\ b^{N+-p(i-1)}\ c^{N*N+s(i-1)}$
Eligiendo i=2, $z_2 = a^N\ b^{N+-p}\ c^{N*N+s}$

Si p=0, entonces s>0, y la cantidad de c's (N*N+s) no es igual a la multiplicación de la cantidad de a y b (N*N), por lo tanto $z_2 \notin L_3$.

Si p>0, entonces la cantidad de b es N+p (mayor o igual a N+1). Para que z_2 pertenezca a L_3, la cantidad de c debería ser N*(N+p), que es mayor igual a $N*(N+1) = N^2+N$.
Sin embargo como $p+q+r+s \leq N$, se cumple que s<N, entonces la cantidad de c es: $N^2+s < N^2+N = N*(N+1) \leq N*(N+p)$.
Por lo tanto $z_2 \notin L_3$, porque la cantidad de c's N^2+s es menor que (y por lo tanto distinto) a la multiplicación de la cantidad de a y b: N*(N+p).

Caso 2:
$u = a^{N-p-q-r}$
$v = a^p$ $p + r + s \geq 1$
$w = a^q$ $p + q + r + s \leq N$
$x = a^r b^s$
$y = b^{N-s} c^{N*N}$

además consideramos:

- r>0 sino caemos en el caso 3, y
- s>0 sino caemos en el caso 1.

$z_i = a^{N-p-q-r}\ (a^p)^i\ a^q\ (a^r b^s)^i\ b^{N-s} c^{N*N}$
Eligiendo i=2, $z_2 \notin L_3$, porque se mezclan a con b.

En forma análoga se demuestran los casos: 4, 6 y 8. Eligiendo i=2, $z_2 \notin L_3$, porque se mezclan a's con b's, o b's con c's.

Estas son todas las descomposiciones que cumplen las condiciones $|vx| \geq 1$ y $|vwx| \leq N$, y para cada una encontramos un i tal que $z_i \notin L_3$ entonces **podemos afirmar que L_3 no es libre de contexto**.

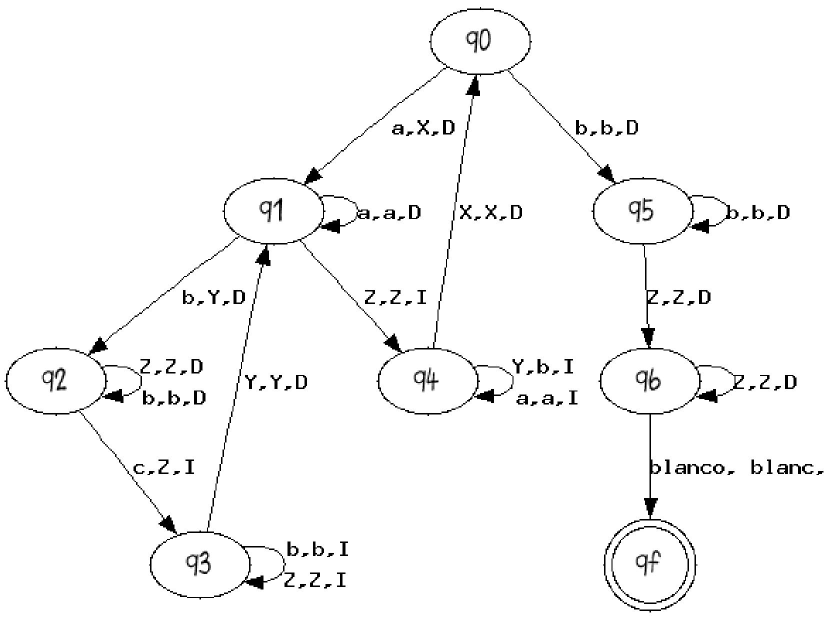
b) Se dará una gramática irrestricta G_3 que genera L_3.

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.

G_3 = ({S,I,A,B,F,M,C}, {a,b,c},P,S} con P formado por las siguientes producciones:
{
S →IABF
A →aA|a // genera a^i con i>0
B →bB|b // genera b^j con j>0
Ia →aIM // por cada a se genera un multiplicador M
Ma→aM // M se desplaza a la derecha ignorando las a's
Mb→bCM // por cada b que encuentra M se genera una C, que luego se convertirá en c
Mc →c // cuando se termina la tira de b's, se elimina M
MF →F // contempla el caso del M que se topa con el delimitador cuando no hay c's
Cb→bC // C se desplaza a la derecha ignorando b's
Cc→cc // cuando C encuentra un terminal c adiciona otra c
CF →cF // contempla el caso del C que se topa con el delimitador cuando no hay c's
Ib →bI // el delimitador izquierdo se desplaza a la derecha ignorando b's
Ic→cI // e ignorando c's
IF→ ε } // se eliminan los delimitadores generando una tira válida del lenguaje

La idea de la GI propuesta es generar primero las tiras de la forma a^ib^j con i,j ≥ 1 y luego por cada símbolo a de la tira agregar j símbolos c's al final de la tira.

c)

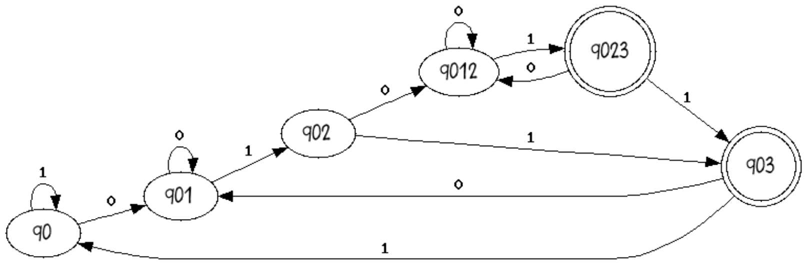


Ejercicio 4 - Teoría de Lengujaes
a) i) Defina R_M para un AFD M cualquiera.
ii) Sea L_4 el lenguaje dado por la expresión regular (0|1)* 010*1. ¿Puede afirmar que se cumple 10011 R_{M4} 0101, siendo ambas tiras que pertenecen a L_4 y M_4 un AFD / L_4 = L(M_4) ? Justifique.
iii) ¿Y si M_4 fuera el AFD Mínimo? Justifique.

a) i) Dados un AFD M / L=L(M) y dos tiras x, z; se dice que
$$x R_M z \Leftrightarrow \delta\hat{}(q_o,x)=\delta\hat{}(q_o,z)$$

ii) Lo que se expresa en la definición de la parte (i) es que dos tiras están en la relación R_M si llegan al mismo estado partiendo del estado inicial.
La afirmación sería verdadera si se construye un AFD y se comprueba que llegar al mismo estado (sea final o no).

En este caso, se ve haciendo un AFD que ambas tiras llegan a estados diferentes.

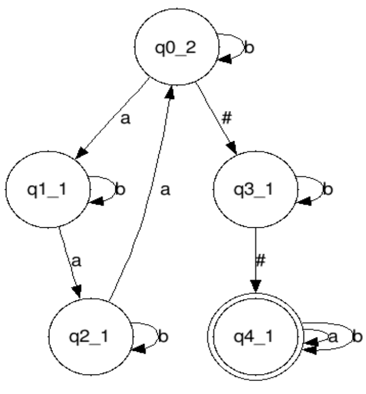


$\delta\hat{}(q_o,10011)=q_{23}$
$\delta\hat{}(q_o,0101)=q_{023}$

iii) El hecho de que sea mínimo no altera en nada la respuesta anterior, ya que la definición no menciona nada de que el AFD sea o no mínimo.

Se hace el AFD Mínimo de todas formas para verificarlo.
Qué si se aplica el algoritmo, da el mismo autómata.

b) Construya un autómata de dos cintas que reconozca el lenguaje formado por pares de tiras <x#y,v#>, con x,y,v ∈ {a,b}* / $|x|_a = 2|v|_a$



Ejercicio 4 – Teoría de la Programación 1

a) Defina la relación de transformación polinómica entre problemas ($\propto$).
b) Defina las clases P y NP
c) Qué es un problema NP-completo?
d) Sean los siguientes problemas:
Problema A (K-Clique). Sea $G_A=(V_A,E_A)$ un grafo no dirigido y K un natural positivo. ¿Existe un subgrafo de G_A completo de K o más vértices?

Problema B (Máximo subgrafo común). Sea $G_B=(V_B,E_B)$ y $G_C=(V_C,E_C)$ dos grafos no dirigidos y H un natural positivo. ¿Existen subconjuntos $E_B' \subseteq E_B$ con conjunto de vértices $V_B' \subseteq V_B$ y $E_C' \subseteq E_C$ con conjunto de vértices $V_C' \subseteq V_C$ tales que: $|E_B'|=|E_C'| \geq H$ y $G_B'=(V_B',E_B')$ es isomorfo a $G_C'=(V_C',E_C')$?

Sabiendo que A es NP-Completo, demuestre que también B es completo para la clase NP.

a) Una transformación o reducción polinomial de un problema de decisión Π_1 a uno Π_2 es una función polinomial que transforma una instancia I_1 de Π_1 en una instancia I_2 de Π_2 tal que I_1 tiene respuesta SI para Π_1 si y sólo si I_2 tiene respuesta SI para Π_2.

Asimismo, un problema de decisiónon Π_1 se reduce polinomialmente a otro problema de decisión Π_2 , $\Pi_1 \propto_p \Pi_2$, si existe una transformación polinomial de Π_1 a Π_2.

b) Un problema de decisión pertenece a la clase P (polinomial) si existe un algoritmo polinomial para resolverlo.

Un problema de decisión pertenece a la clase NP si dada una instancia de SI y evidencia de la misma, puede ser verificada en tiempo polinomial.

c) Un problema de decisión es NP-completo si:
- $\Pi \in NP$
- $\forall\ \Pi' \in NP,\ \Pi' \propto_p \Pi$

d) Se probará que A $\propto$ B.

Sea un grafo $G_A=(V_A,E_A)$ y K, parámetros del problema del K-clique; se construye una instancia del problema B del siguiente modo:

- $G_B = G_A$
- G_C = grafo completo de K vértices (K*(K-1) / 2 aristas)
- H = K*(K-1) / 2

G_B y G_C tienen subgrafos isomorfos de H o más aristas $\Leftrightarrow$ $G_B = G_A$ tiene un k-Clique

Notar que:
- Como $H=|E_C|$, cualquier subgrafo de G_C con al menos H aristas debe ser igual al propio G_C.
- Se puede ver además que la transformación del problema es polinomial.