Teoría de Lenguajes

2do. Parcial – Curso 2023
Soluciones

Ejercicio 1 [Evaluación individual del obligatorio]

a)
```
grammar = """
S -> A B | D C | E
A -> '0' A '1' | '0' '1'
B -> '2' B | '2'
C -> '1' '1' C '2' | '1' '1' '2'
D -> '0' D | '0'
E -> '0' '0' '0' E '2' | '0' '0' '0' F '2'
F -> '1' F | '1'
"""
```
Toda gramática en nltk debe tener:
i) Variables, indicadas por letras a la izquierda de la regla de producción.
ii) Flechas, que separan la parte izquierda y derecha de la regla de producción, generadas por el símbolo de menos y el símbolo de mayor.
iii) Variables y/o terminales a la derecha de la regla de producción. Los terminales deben estar entre comillas simples y si se quiere concatenar dos variables, dos terminales o una variables y un terminal, estas deben estar separadas por espacios.

b) El retorno de la función *parse* para GLC de NLTK es una lista con los árboles para la entrada y la gramática en cuestión, o la lista vacía en caso que la entrada no sea reconocida por la gramática.
Para resolver el Programa 2 se tomaba uno de los árboles generados, en caso de existir, para recorrerlo de forma recursiva y generar la salida esperada.

c) Esta respuesta depende de cada entrega, a modo de ejemplo se indican dos opciones válidas, que no son necesariamente las únicas.
Podía realizarse mediante: *word_tokenize*, que obtiene las palabras de la entrada o *RegexpTokenizer*, que usa expresiones regulares para tokenizar la entrada.

Ejercicio 2

$L_2 = \{ w\#z \ / \ w \in \Sigma^* \wedge z \in suf(w)\}$ donde *suf*(w) denota al conjunto de sufijos de la tira $w \in \Sigma^*$

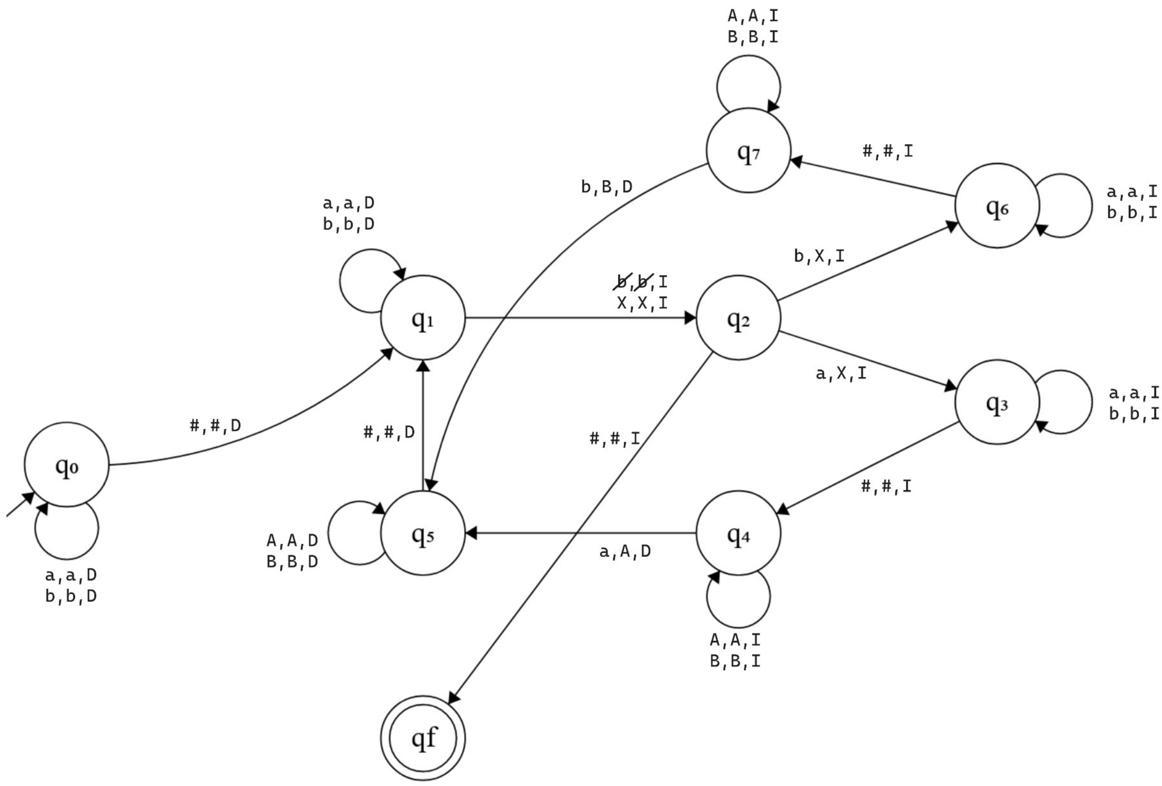
a) El sufijo z de una tira $w \in \Sigma^*$ se puede definir como $w = s.z$, con $s \in \Sigma^*$ y $z \in \Sigma^*$. Usando esa definición, el lenguaje que queremos generar es $L_2 = \{ s.z\#z \ / \ s \in \Sigma^* \wedge z \in \Sigma^*\}$

Como se sabe que no es Libre de Contexto, se da una gramática irrestricta que genera al lenguaje L_2. Tiene las siguientes reglas de producción, siendo S el símbolo inicial.

$S \rightarrow aS \mid bS \mid Z$ // genera la tira s
$Z \rightarrow aAZ \mid bBZ \mid \#$ // genera terminales y variables para el sufijo z
$Aa \rightarrow aA$ // la variable "A" avanza hacia la derecha
$Ab \rightarrow bA$
$Ba \rightarrow aB$ // la variable "B" avanza hacia la derecha
$Bb \rightarrow bB$
$A\# \rightarrow \#a$ // la variable "A" se transforma en un terminal "a" al avanzar sobre el "#"
$B\# \rightarrow \#b$ // la variable "B" se transforma en un terminal "b" al avanzar sobre el "#"

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Asimismo, todos los autómatas deben tener señalado su estado inicial y todas las gramáticas su símbolo inicial. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

b) Se construye una Máquina de Turing que reconoce L_2



Donde:

- q0 avanza sobre la tira *w* hasta llegar al #.
- q1 avanza sobre el sufijo hasta llegar al final de la cinta (blanco) o leer una previa marca X
- q2 marca el primer símbolo del sufijo que esté aún sin procesar
    - Si lo leído por q2 es una "a", va a q3:
        - q3 continúa esta tarea de rebobinado sobre los símbolos del sufijo aún sin procesar. Al encontrar el #, se mueve a la izquierda, para entrar en la zona de la cinta donde está la tira *w*
        - q4 saltea los símbolos de *w* que ya fueron marcados previamente. Al encontrar la primer "a" sin marcar, la marca y comienza a avanzar hacia la derecha.
        - q5 continúa avanzando sobre los símbolos de la tira *w* que ya fueron marcados. Al llegar al #, se mueve a la derecha y se retorna a q1, que es donde comienza un nuevo ciclo de búsqueda en el sufijo.
    - Si lo leído por q2 es una "b" va a q6:
        - q6 es análogo a q3
        - q7 es análogo a q4 con la diferencia que, lo que busca encontrar en *w*, es una "b"
        - q5 es el ya descrito
    - Si lo leído por q2 es un "#", quiere decir que ya no hay más símbolos a procesar, ya sea porque el sufijo es épsilon o porque ya se marcó todo lo que había que marcar.

Ejercicio 3

$L_3 = \{ (01)^p 0^t 1^r \;/\; t > 0 \;\wedge\;$ si $t \text{ MOD } 2 = 0$ entonces $r > p > 0$
si $t \text{ MOD } 2 = 1$ entonces $p \geq r > 0 \}$

a) Como se dice en la letra que el lenguaje L_3 no es Regular, se construye una gramática libre de contexto G_3 / $L_3 = L(G_3)$.

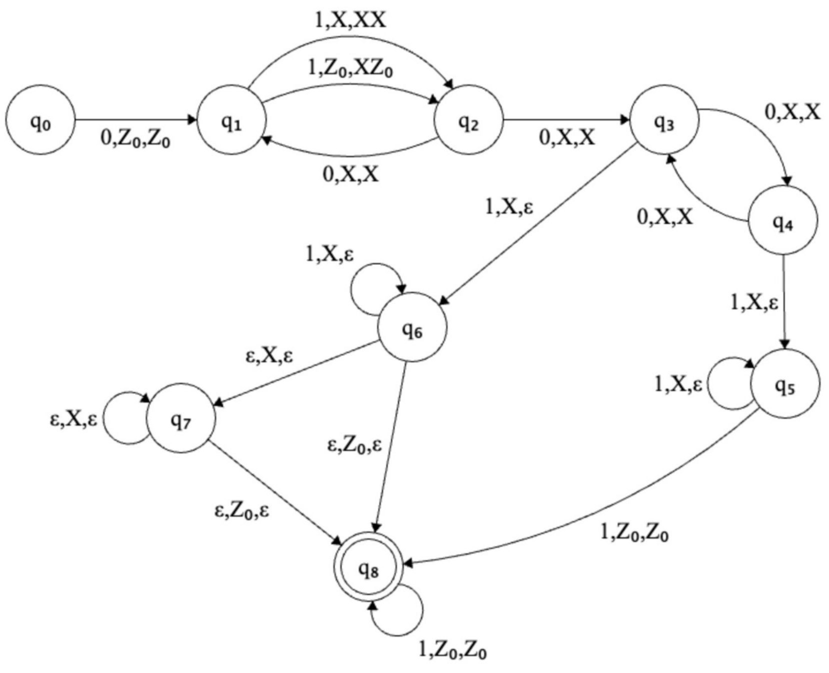
Una posible gramática con S como símbolo inicial tendría las siguientes reglas de producción:

S → 01P1 | 01 I 1 // con la P genero las tiras con cantidad par de 0's en el medio
// con la I genero las tiras con cantidad impar de 0's en el medio
P → 01P1 | P1 | A001 // Aseguro al menos un 1 al final más que 01's
A → A00 | 00 // Se generan con A una cantidad par de 0's que van en el centro
I → 01 I 1 | 01 I | A0 | 0 // acá aseguro al menos un 0 en el medio y luego genero cantidad impar de 0's entrando en las derivaciones de A

Se puede ver por las reglas, que está simplificada, ya que no contienen producciones epsilon, ni unitarias.

Además, todas las variables son útiles.

b) Se construye un APD M_3 / $L_3 = L(M_3) = L_3$



El APD dibujado – cuyo estado inicial es qo es No Determinista.

Esto puede verse por ejemplo en la transición: $\delta(q_2,0,X) = \{ (q_1,X), (q_3,X) \}$

Ejercicio 4

Indique si las siguientes afirmaciones son Verdaderas o Falsas justificando en cada caso.

a) Sean L_{a1} y L_{a2} dos lenguajes definidos sobre el mismo alfabeto Σ. Si L_{a1} es recursivamente enumerable pero no libre de contexto y L_{a2} es regular pero no finito, entonces $L_{a1} \cap L_{a2}$

i) es regular

Falso: Tomando Σ ={0,1} y siendo $L_{a1} = \{ 0^k1^k0^k, k > 0 \}$ que sabemos es recursivamente enumerable pero no libre de contexto – por teórico - y $L_{a2} = \{ 0^n1^p0^t, n,p,t > 0 \}$, que es regular, ya que existe una ER que lo define al ser los tres supraíndices de los símbolos del alfabeto mayores que 0 y en cualquier cantidad – al menos uno –, sólo respetando la secuencia se tiene 00*11*00* , y por lo tanto **$L_{a1} \cap L_{a2} = \{0^k1^k0^k, k > 0\}$**; es decir, el propio L_{a1} que se sabe recursivamente enumerable y no libre de contexto y por lo tanto no regular (por Jerarquía de Chomsky).

ii) no regular

Falso: Este caso, también es Falso. Tomando Σ ={0,1} y siendo $L_{a1} = \{ 0^k1^k0^k, k >0\}$ que es recursivamente enumerable pero no libre de contexto – por teórico – y $L_{a2} = \{ 0^t10, t > 0 \}$ que es regular, ya que existe una ER que lo define al ser t>0 comienza con cualquier cantidad de 0's – al menos uno – y termina en 01 – 00*10 - , se tiene que **$L_{a1} \cap L_{a2}$ = { 010 }** que es finito y por lo tanto regular.

b) Sean L_{b1} y L_{b2} dos lenguajes definidos sobre el mismo alfabeto Σ. Si L_{b1} es recursivamente enumerable y L_{b2} es regular, entonces siempre se cumple que $L_{b2} \subset L_{b1}$.

Falso. Se plantea el siguiente contraejemplo. Sea nuevamente ∑ ={0,1} , $L_{b1}=\{ 0^k1^k0^k, k\geq 0 \}$ y L_{b2} =dado por la expresión regular 0*1*0*. Acá se cumple que $L_{b1} \subset L_{b2}$

c) Para el lenguaje $L_c = \{ a^k (ab)^r b^s$, con $k, r, s > 0 \wedge r \text{ MOD } 2 = 1 \}$ no es posible construir una gramática lineal izquierda simplificada G_c / $L_c = L(G_c)$. Si su respuesta es Verdadero, debe justificar adecuadamente. Si su respuesta es Falso, construya una gramática con esas características.

Falso: El lenguaje L_c es un lenguaje regular, ya que no hay restricciones en cuanto a las cantidades de los símbolos, sólo en el orden en que aparecen en las tiras: secuencias de una o más **a**'s, seguidas de secuencias de largo impar de pares - al menos uno - de **ab**'s y seguidos de secuencias de una o más **b**'s.

Una ER podría ser: **aa*ab(abab)*bb***

Por ser un lenguaje regular, siempre será posible construir una gramática lineal izquierda (y derecha también).
A continuación se escribe una gramática lineal izquierda con las siguientes reglas de producción y símbolo inicial S:

S → Sb | Xb
X → Xabab | Aab
A → Aa | a

Se puede ver que está simplificada porque no tienen producciones epsilon, no tiene producciones unitarias y todas sus variables son útiles.

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Asimismo, todos los autómatas deben tener señalado su estado inicial y todas las gramáticas su símbolo inicial. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

d) El lenguaje $L_d = \{ 0^{m*k}1^{2k}0^{m+k+j}$, con $m \in \{0,1\} \wedge k\geq 0 \wedge j > 0 \}$ es libre de contexto.

Falso: Se usará el contrarecíproco del Pumping Lema para Lenguajes Libres de Contexto para probar que no es Libre de Contexto.

Dado N constante cualquiera, elegimos $z = 0^N 1^{2N} 0^{N+2}$, $z \in L_d$ y $|z| = 4N+1$
Se consideró: m=1 , k=N , j=1
Notar que cant_0(z) del final siempre es mayor que cant_0(z) del comienzo en la menos **2** 0's.

Se estudian todas las descomposiciones de z=uvwxy que cumplen $|vx|>0$ y $|vwx|\leq N$

Familias	0^N	1^{2N}	0^{N+2}
1	v x		
2		v x	
3			v x
4	v x	x	
5	v	v x	
6		v x	x
7		v	v x
8	v	x	
9		v	x

Familia 1

u=$0^{N-p-q-r-s}$
v=0^p p+r > 0
w=0^q p+q+r ≤ N
x=0^r
y=$0^s1^{2N}0^{N+2}$

$z_i = 0^{N-p-q-r-s} (0^p)^i 0^q (0^r)^i 0^s 1^{2N}0^{N+2} = 0^{N-p+pi-r+ri-}1^{2N}0^{N+2} = 0^{N+(p+r)(i-1)}1^{2N}0^{N+2}$
Eligiendo i=2 queda $z_2= 0^{N+p+r}1^{2N}0^{N+2}$
donde la 2*cant_0(z_2) del comienzo de la tira es > cant_1(z_2) que le siguen ya que p+r>0
Por lo tanto $z_2 \notin L_d$

Familia 2, razonamiento análogo con el mismo i=0, y lo que falla es que la cant_1(z_0) < 2*cant_0(z_0). Por tanto $z_0 \notin L_d$

Familia 3, razonamiento análogo, pero hay que tomar i=0, con lo cual ahí cant_0(z_2) del final es ≤ cant_0(z_2) del comienzo, ya que quedaría N+2-2(p+r) ≤ N al ser p+r>0. Por tanto $z_0 \notin L_d$

Familia 4

u=$0^{N-p-q-r}$
v=0^p p+r+s > 0 (s>0 sino estamos en Familia 1
w=0^q p+q+r+s ≤ N r>0 sino estamos en Familia 8)
x=0^r1^s
y=$1^{2N-s} 0^{N+2}$

$z_i = 0^{N-p-q-r} (0^p)^i 0^q (0^r1^s)^i 1^{2N-s}0^{N+2} = 0^{N-p-r} (0^p)^i (0^r1^s)^i 1^{2N-s}0^{N+2}$
Eligiendo i=2 queda $z_2= 0^{N+r}1^s 0^r 1^{2N} 0^{N+2}$

donde z_2 no queda de la forma de las tiras del lenguaje al aparecer **0**'s entre los **1**'s (porque r>0). Por lo tanto $z_2 \notin L_d$

Familia 5, razonamiento análogo y mismo argumento.

Familia 6 y 7 el razonamiento análogo, mismas restricciones pero se mezclan los **1**'s con los **0**'s del final.

Familia 8

$u=0^{N-p-q}$
$v=0^p$ $\quad p+r > 0$
$w=0^q1^s$ $\quad p+q+r+s \leq N$
$x=1^r$
$y=1^{2N-s-r}0^{N+2}$

$z_i = 0^{N-p-q}(0^p)^i 0^q 1^s (1^r)^i 1^{2N-s-r} 0^{N+2} = 0^{N-p}(0^p)^i (1^r)^i 1^{2N-r} 0^{N+2}$
Eligiendo i=3 queda $z_3 = 0^{N+2p}1^{2N+r}0^{N+2}$
Como p+r>0, al menos uno de los 2 es >0, con lo cual:
- si p>0, cant_0(z_3) del comienzo es $\geq$ cant_0(z_3) del final de la tira; $z_3 \notin L_d$
- si p=0, entonces r>0, por lo tanto cant_1(z_3) > 2*cant_0(z_3) del comienzo; $z_3 \notin L_d$

Familia 9

$u=0^N1^{2N-p-s}$
$v=1^p$ $\quad p+r > 0$
$w=1^s0^q$ $\quad p+q+r+s \leq N$
$x=0^r$
$y=0^{N+2-q-r}$

$z_i = 0^N1^{2N-p-s}(1^p)^i 1^s 0^q (0^r)^i 0^{N+2-q-r} = 0^N1^{2N-p}(1^p)^i (0^r)^i 0^{N+2-r} = 0^N1^{2N+(i-1)p} 0^{N+2+(i-1)r}$
Eligiendo i=0 queda $z_0 = 0^N1^{2N-p}0^{N+2-r}$
Como p+r>0, al menos uno de los 2 es >0, con lo cual:
- si p>0, cant_0(z_0) del comienzo es > cant_1(z_0); $z_0 \notin L_d$
- si p=0, entonces r>0, por lo tanto no se cumple que cant_0(z_0) del final tenga al menos 2 **0**'s más que cant_0(z_0) del comienzo; $z_0 \notin L_d$

Como estas son todas las descomposiciones posibles que cumplen |vx|>0 y |vwx|≤ N, se ha logrado demostrar por el contra-recíproco del Pumping Lema que L_d NO un lenguaje Libre de Contexto, con lo cual la afirmación es **Falsa**.

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Asimismo, todos los autómatas deben tener señalado su estado inicial y todas las gramáticas su símbolo inicial. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.