Teoría de Lenguajes

2do. Parcial – Curso 2021
Soluciones

Ejercicio 1 [Evaluación individual del obligatorio]

a)

```
grammar = """
   S -> 'a' S 'b' | 'c' A
   A -> 'c' A 'd' | 'c' 'd'
   """
```

b)

Ejercicio 2

Sea el siguiente lenguaje $L_2 = \{x\#y \;/\; x,y \in L((0|1)^*),\ |x|_1 = 2|y|_0,\ \ |y|_1 = 2|x|_0\}$

Ejemplos de tiras de L_2:

#
110#110
00#1111
10111#0110

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

a) El lenguaje L propuesto es recursivamente enumerable, lo que se demuestra en la parte b) dando una gramática irrestricta que lo genera o en la parte c) mediante una máquina de Turing que lo reconoce. No es un lenguaje libre de contexto, lo cual se prueba aplicando el contra-recíproco del Pumping Lema para lenguajes libres de contexto.
Dado N, elegimos para el pumping la tira $Z= 1^{2N}0^N\#0^N1^{2N}$ (otra posibilidad es $Z= 0^N1^{2N}\#1^{2N}0^N$).
Tenemos que $|Z| = 6N+1 > N$, y además se cumple que $Z \in L$ ya que los unos antes del numeral son el doble de los ceros después del mismo y viceversa.
Analizamos las descomposiciones de z = uvwxy y consideramos aquellas familias que cumplen las condiciones $|vx| \geq 1$ y $|vwx| \leq N$. Toda descomposición pertenece a una de trece familias posibles:

familia	1^{2N}	0^N	#	0^N	1^{2N}
1	v x				
2	v x	x			
3	v	x			
4	v	v x			
5		v x			
6		v x	x	x	
7		v		x	

8		v	v	v x	
9				v x	
10				v x	x
11				v	x
12				v	v x
13					v x

Familia 1:
$u = 1^p$ $p >= 0$
$v = 1^q$ $q+s >= 1$
$w = 1^r$ $q+r+s <= N$
$x = 1^s$
$y = 1^{2N-p-q-r-s}0^N\#0^N1^{2N}$

$z_i = 1^{2N+(q+s)(i-1)}0^N\#0^N1^{2N}$
$z_0 = 1^{2N-(q+s)}0^N\#0^N1^{2N}$
Como $q+s >= 1$, el largo de la primera secuencia de 1's antes de # es menor estricto que el doble del largo de la secuencia de 0's después de # ($2N-(q+s) < 2N$), y por lo tanto $z_0 \notin L$.

Familias 5 ,9,13:
Se demuestran con el mismo argumento que la familia 1, tomando $i = 0$. Para la familia 5 los 1's de y son más del doble que los 0's de x, para la familia 9 los 1's de x son más del doble que los 0's de y, y para la familia 13 los 1's de y son menos del doble que los 0's de x.

Familia 2:
$u = 1^{2N-p-q-r}$
$v = 1^p$ $p+r+s>= 1$
$w = 1^q$ $p+q+r+s<=N$
$x = 1^r0^s$ (se asume $r>0$ y $s>0$, sino estaríamos en los caso de las familias 1 y 3)
$y = 0^{N-s}\#0^N1^{2N}$

$z_i = 1^{2N+p(i-1)-r}(1^r0^s)^i0^{N-s}\#0^N1^{2N}$
$z_0 = 1^{2N-p-r}0^{N-s}\#0^N1^{2N}$
Como $s >0 \rightarrow N-s < N$ y en z_0 los 0's de x son menos de la mitad que los 1's de y $\rightarrow z_2 \notin L$.

Familias 4,10 y 12:
Se demuestran con el mismo argumento que la familia 2, tomando $i = 0$. En z_0, para la familia 4 los 1's de x son menos del doble que los 0's de y, para la familia 10 los 1's de y son menos del doble que los 0's de x, y para la familia 12 los 0's de y son menos de la mitad que los 1's de x.

Familia 3:
$u = 1^{2N-p-q}$
$v = 1^p$ $p+s>= 1$
$w = 1^q0^r$ $p+q+r+s<=N$
$x = 0^s$
$y = 0^{N-s-r}\#0^N1^{2N}$

$z_i = 1^{2N+p(i-1)}0^{N+s(i-1)-}\#0^N1^{2N}$
$z_0 = 1^{2N-p}0^{N-s}\#0^N1^{2N}$
Como $p+s>0$, consideramos 2 casos:
1) $p>0 \rightarrow 2N-p < 2N$ y en z_0 los 1's de x son menos del doble que los 0's de y $\rightarrow z_0 \notin L$.

2) $s>0 \rightarrow N-s < N$ y en z_0 los 0's de x son menos de la mitad que los 1's de y $\rightarrow z_0 \notin L$.

Familia 11:
Se demuestra en forma análoga a la familia 3, tomando i = 0.

Familia 6:
$u = 1^{2N} 0^{2N-p-q-r}$
$v = 0^p$ $p+r+s+1>= 1$
$w = 0^q$ $p+q+r+s+1<=N$
$x = 0^r\#0^s$
$y = 0^{N-s}\#0^N1^{2N}$

$z_i = 1^{2N}0^{N+p(i-1)-r}(0^r\#0^s)^i0^{N-s}1^{2N}$
$z_2 = 1^{2N}0^{N+p}\#0^{s+r}\#0^N1^{2N}$
Como z_2 contiene dos # $\rightarrow z_2 \notin L$.

Familia 8:
Se demuestra en forma análoga a la famila 6, tomando i = 2.

Familia 7:
$u = 1^{2N}0^{N-p-q}$
$v = 0^p$ $p+s >= 1$
$w = 0^q\#0^r$ $p+q+r+s+1 <=N$
$x = 0^s$
$y = 0^{N-r-s}1^{2N}$

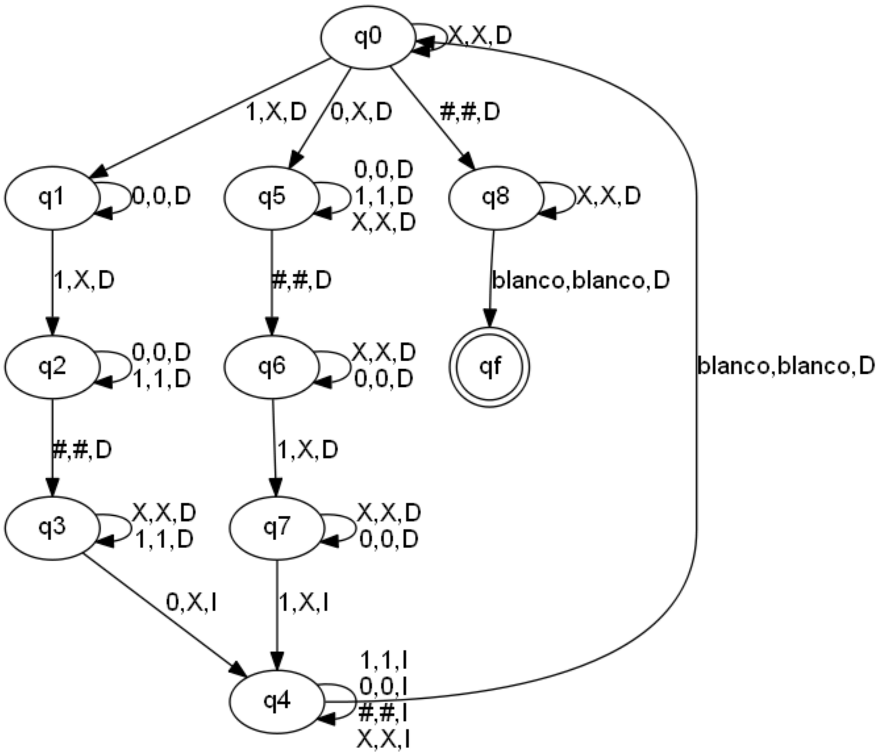
$z_i = 1^{2N} 0^{N+p(i-1)} \#0^{N+s(i-1)}1^{2N}$
$z_0 = 1^{2N} 0^{N-p}\#0^{N-s}1^{2N}$
Como $p+s > 0$, debe verificarse al menos uno de los siguientes casos:
1) $p>0$: $N-p < N$, lo que implica que la cantidad de 0´s en la subsecuencia x antes de # es menor que la mitad de 1´s en la subsecuencia y después de # $\rightarrow z_0 \notin L$.
2) $s>0$: $N-s < N$, lo que implica que la cantidad de 0´s en la subsecuencia y después de # es menor que la mitad de 1´s en la subsecuencia x antes de # $\rightarrow z_0 \notin L$.

Como estas son todas las descomposiciones que cumplen las condiciones $|vx|\geq1$ y $|vwx|\leq N$ y para cada una encontramos un i / $z_i \notin L$, entonces podemos afirmar que **L no es Libre de Contexto.**

b) Se construirá una gramática irrestricta G / L = L(G).

S -> 11S0 | 0S11 | #
01 -> 10
10 -> 01

c) Se construirá un autómata M / L = L(M).



La MT presentada es determinista, ya no hay más de una transición para todo par (estado, símbolo de cinta).

Ejercicio 3

Sea el siguiente lenguaje $L_3 = \{a^p b^{s+r} \# c^s: p>r; p,r>0, s\geq 0\}$

a) Construya una gramática G_3 tal que $L(G_3)=L_3$.
b) ¿Cuál es la tira más corta de L_3? Realice una derivación para esa tira, según la gramática escrita en la parte a).
c) Dé un autómata apropiado M_3 tal que $L(M_3) = L_3$. ¿Es determinista?
d) Considere la siguiente modificación a L_3: $L' = \{a^p b^{s+r} \# c^q: p>r; p,r>0, s,q \geq 0\}$. Construya una gramática G' simplificada tal que L(G') = L'.

Justifique por qué está simplificada.

a) Se reescribe la expresión del lenguaje $L = \{a^p b^r b^s \# c^s: p>r; p,r>0, s\geq 0\}$

S -> AB
A -> aAb | aA | aab
B -> bBc | #

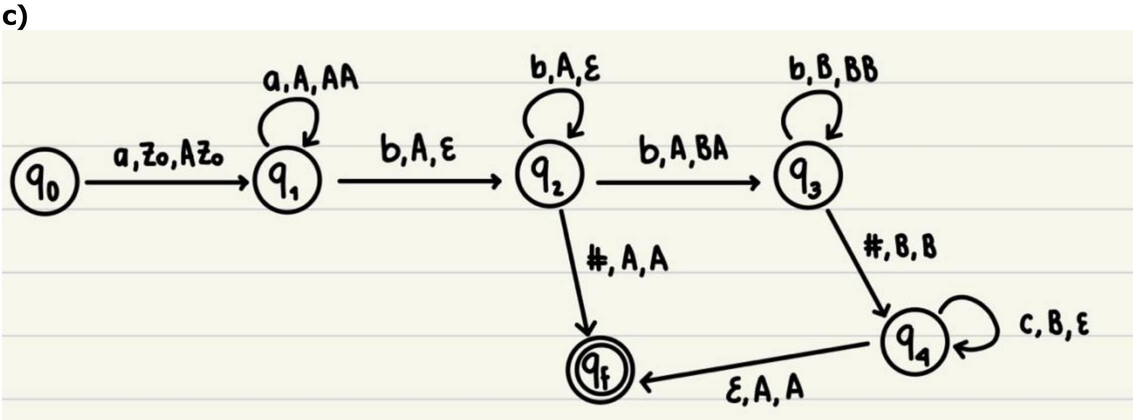
Una solución alternativa es:

S -> ABC
A -> aA | a
B -> aBb | ab
C -> bCc | #

b) La tira más corta es "aab#". La derivación: S => AB => aabB=> aab#

Para la gramática alternativa una posible derivación es: S => ABC => aBC => aabC => aab#

c)



El APD es no determinista, ya que se cumple que al estar en el estado q_2 y teniendo a A en el tope del stack, se tienen dos transiciones posibles asociadas al símbolo b: una al estado q_2 y otra al estado q_3

d) El lenguaje $L' = \{a^p b^{s+r} \# c^q: p>r; p,r>0, s,q \geq 0\}$ es regular, dado que L' = L(aaa*bb*#c*). Por tanto, haremos una gramática lineal derecha que lo genere:

S -> aaA
A -> aA | bB
B -> bB | #C | #
C -> cC | c

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.

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

No era pedido en el ejercicio, pero la justificación de que el lenguaje L′ es regular puede ser probada mediante la igualdad con el lenguaje generado por una E.R. Consideremos:

- $L' = \{a^p b^{s+r} \# c^q : p>r;\ p,r>0,\ s,q \geq 0\}$
- $L'' = L(aaa^*bb^*\#c^*)$

La prueba de que $L' \subseteq L''$ es trivial, pues surge de cómo generar las tiras de L′ usando la expresión regular aaa*bb*#c*.

Para probar $L'' \subseteq L'$ tomemos x, una tira de L″ y veamos cómo se corresponde a L′.

$x \in L'' \Rightarrow x = a.a.a^k\ b.b^j \# c^m \quad k,j,m \geq 0 \Rightarrow x = a^{k+2}\ b^{j+1} \# c^m \quad k,j,m \geq 0$

Primero veamos cómo se corresponden los índices de L′ a k,j y m.

- $p = k+2$
- $s = j$
- $r = 1$
- $q = m$

Podemos comprobar, entonces, que estos índices cumplen las condiciones de L′:

- $p>r$ se cumple porque $k+2 > 1$
- $p>0$ se cumple porque $k+2 > 0$
- $r>0$ se cumple porque $1 > 0$
- $s\geq0$ se cumple porque $j\geq0$
- $q\geq0$ se cumple porque $m\geq0$

Entonces $L'' \subseteq L$ y como también se cumple $L' \subseteq L'' \Rightarrow L' = L'' \Rightarrow$ L′ es generado por una expresión regular $\Rightarrow$ L′ es regular