

Teoría de Lenguajes Soluciones

Ejercicio 1

Sea $L_1 = \{ w \in \{0,1\}^*, w \text{ es de la forma } (01)^k(10)^{2p}, \text{ con } k > p \geq 0 \}$

a) Clasifique a L_1 según la Jerarquía de Chomsky.

L_1 es Libre de Contexto, NO Regular. Se demuestra que no es regular por el contra-recíproco del Pumping Lema para lenguajes regulares. Por otro lado, se justifica que es Libre de Contexto construyendo una gramática Libre de Contexto en la parte b), o por el APD de la parte c).

Dada N cte. del PL se elige $z = (01)^{N+1}(10)^{2N}$ perteneciente a L_1 y $|z| = 6N+2 \geq N$. Todas las descomposiciones de z en uvw , con $|uv| \leq N$ y $|v| \geq 1$, se pueden estudiar analizando por separado los siguientes casos:

Caso 1

$$|uv| \leq N \text{ y } |v| \text{ impar}$$

Eligiendo $i=0$, z_0 no pertenezca a L_1 , porque z_0 queda de largo impar: $6N+2 - |v|$ y todas las tiras de L_1 son de largo par: $2k+4p$.

Caso 2

$$\begin{aligned} u &= (01)^p & 2p+2q &\leq N \\ v &= (01)^q & 2q &\geq 1 \\ w &= (01)^{N+1-p-q}(10)^{2N} \end{aligned}$$

Eligiendo $i=0$, $z_0 = (01)^{p+N+1-p-q}(10)^{2N} = (01)^{N+1-q}(10)^{2N}$ siendo $k = N+1-q$ y $p = N$, resulta que z_0 no pertenece a L_1 . No se cumple que $k > p$ dado que no se cumple que $N+1-q > N$, con $2q \geq 1$.

Caso 3

$$\begin{aligned} u &= (01)^p 0 & 2p+2q+3 &\leq N \\ v &= 1(01)^q 0 & 2q &\geq 0 \\ w &= 1(10)^{N+1-p-q-2}(10)^{2N} \end{aligned}$$

Eligiendo $i=0$, $z_0 = (01)^p 0 1(01)^{N+1-p-q-2}(10)^{2N} = (01)^{p+1+N+1-p-q-2}(10)^{2N} = (01)^{N-q}(10)^{2N}$ siendo $k = N-q$ y $p = N$, resulta que z_0 no pertenece a L_1 . No se cumple que $k > p$ dado que no se cumple que $N-q > N$, con $2q \geq 0$.

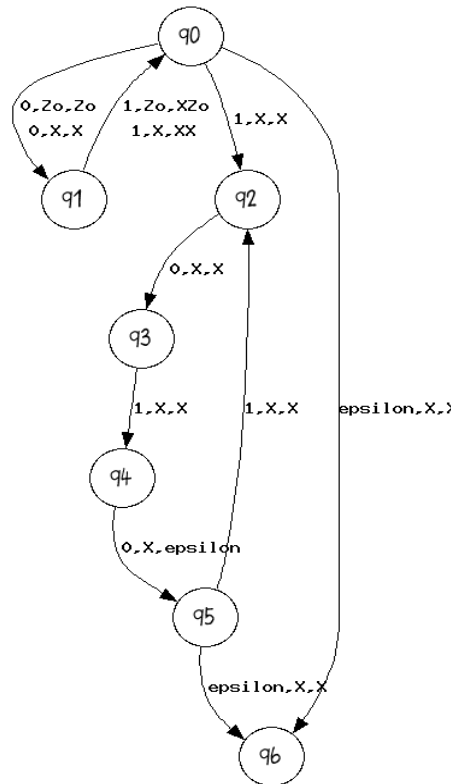
Como ya analizamos todas las descomposiciones posibles de z que cumplen $|uv| \leq N$ y $|v| \geq 1$, entonces concluimos que L_1 no es regular.

b) Construya una gramática $G_1 / L_1 = L(G_1)$. ¿Está simplificada? Justifique.

$S \rightarrow 01 S 1010 \mid 01 S \mid 01$

Está simplificada, ya que no contiene símbolos inútiles, producciones unitarias ni producciones epsilon.

c) Construya un autómata $M_1 / L_1 = L(M_1)$. ¿Es determinista? Justifique.



Es no determinista, porque tiene una transición épsilon con X el tope del stack que sale de q_0 , y otra transición con X en el tope del stack que sale de q_0 .

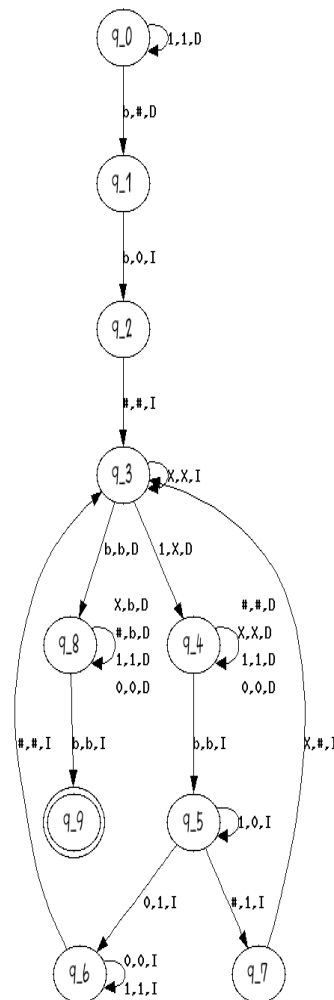
Ejercicio 2

Considere la función $f: \{1\}^* \rightarrow \{0,1\}^*$, que, dada una tira $w=1^n$, con $n>0$, devuelve una tira $w'=b_n$, siendo b_n la representación binaria de n . Por ejemplo:

$$\begin{aligned} f(1) &= 1 \\ f(11) &= 10 \\ f(111) &= 11 \\ f(11111111) &= 1000 \end{aligned}$$

Construya una Máquina de Turing que compute la función f .

Idea: la máquina comienza con el número en unario en la cinta. Se desplaza hasta el final de la tira y agrega '#0' a la derecha. Luego, por cada 1 que tiene a la izquierda del numeral, lo marca con X, y se mueve hasta la derecha de la tira, y empieza a moverse a la izquierda. Si encuentra un 0, lo transforma en 1 y terminó de sumar. Si encuentra un 1, lo transforma en 0, y sigue hacia la izquierda llevando el carry. Si llega al '#' con el carry, entonces cambia el '#' por un 1, y la X que siempre tiene a su izquierda, por un nuevo '#'. Y va a buscar el siguiente 1. Si ya no hay más, "limpia" las marcas temporales X y #, va a la derecha, y finaliza.



Ejercicio 3

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

- Sean L_1, L_2, L_3 lenguajes tales que L_1, L_2, L_3 es regular, y L_1 y L_3 son regulares. Entonces L_2 es regular.
- Si L_4 es un lenguaje regular no vacío, no existe un lenguaje L_5 no regular, tal que $L_4 \subseteq L_5$.
- Si L_6 y L_7 son lenguajes tales que $L_6 \cap L_7$ es finito, y L_7 es libre de contexto no regular, entonces L_6 es regular.
- La unión de un lenguaje recursivamente enumerable y un lenguaje libre de contexto nunca es regular.
- El lenguaje $\text{pref}(\{a^k b^k c^k / k \geq 0\})$ el Libre de Contexto NO Regular.

a) **Falso.**

Si consideramos $L_1 = L(1^*)$, $L_3 = L(0^*)$ y $L_2 = \{1^k 0^k / k \geq 0\}$ entonces L_1, L_2, L_3 quedaría $\{1^*\} \cdot \{1^k 0^k\} \cdot \{0^*\} = L(1^* 0^*)$ que es regular y sin embargo L_2 no es regular

b) **Falso**

Si consideramos $L_4 = L(10)$ y $L_5 = \{1^k 0^k / k \geq 0\}$, L_4 está incluido en L_5 y L_5 no es regular

c) **Falso.**

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.

Si consideramos $L_6 = \{1^N 0^N / N \geq 0\}$ y $L_7 = \{0^N 1^N / N \geq 0\}$ entonces,
 $L_6 \cap L_7 = \{\epsilon\}$ que es un conjunto finito y L_6 no es regular

d) **Falso.**

Por la Jerarquía de Chomsky, todo lenguaje regular es un lenguaje libre de contexto y también un lenguaje recursivamente enumerable.

Por lo tanto basta con tomar 2 lenguajes regulares, y al realizar la unión, nos queda un lenguaje regular ya que existe la gramática que lo genera, para probar esto basta con hacer la unión de las gramáticas de los respectivos lenguajes, teniendo la precaución de que no existen símbolos variables en común a ambas gramáticas

e) **Falso.**

Se demostrará que $\text{pref}(a^k b^k c^k)$ no es un lenguaje libre de contexto aplicando el contrarrecíproco del PL. Sea N la cte. del PL, elijo $z = a^N b^N c^N$ y considero todas las descomposiciones que satisfacen: $|vx| > 0$ $|vwx| \leq N$

	a...a	b...b	c...c
1	vx		
2	$v x \dots$	$\dots x$	
3	v	x	
4	$v \dots$	$\dots v x$	
5		vx	
6		$v x \dots$	$\dots x$
7		v	x
8		$v \dots$	$\dots v x$
9			$v x$

Caso 1

$$\begin{aligned} u &= a^j \\ v &= a^k \\ w &= a^l \\ x &= a^p \\ y &= a^{N-j-k-l-p} b^N c^N \end{aligned} \quad \begin{aligned} k+l+p &\leq N \\ k+p &> 0 \end{aligned}$$

$z_i = a^{N+(i-1)(k+p)} b^N c^N$, con $i=0$, $z_0 = a^{N-k-p} b^N c^N$, como $k+p > 0$, la cantidad de a's es menor que las b's y por lo tanto no puede ser prefijo de $a^N b^N c^N$ (ya que todos los prefijos tienen mayor o igual cantidad de a's que de b's).

El caso 5 es análogo (quedan menos b's que c's).

Caso 9

$$\begin{aligned} u &= a^N b^N c^j \\ v &= c^k \\ w &= c^l \\ x &= c^p \\ y &= c^{N-p-l-k-j} \end{aligned} \quad \begin{aligned} k+l+p &\leq N \\ k+p &> 0 \end{aligned}$$

$z_i = a^N b^N c^{N+i(k+p)}$, con $i=2$, $z_0 = a^N b^N c^{N+k+p}$, como $k+p > 0$, la cantidad de c's es mayor que las b's y por lo tanto no puede ser prefijo de $a^N b^N c^N$ (ya que todos los prefijos tienen mayor o igual cantidad de b's que de c's).

Caso 2

$$\begin{aligned} u &= a^{N-k-l-p} \\ v &= a^k \\ w &= a^l \\ x &= a^p b^q \end{aligned} \quad k+l+p+q < N$$

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.

$$y = b^{N-q}c^N$$

$z_i = a^{N+(i-1)k-p}(a^p b^q)^i b^{N-q} c^N$, con $i=2$, z_2 no pertenece a L por mezclar a's y b's (los prefijos siempre mantienen el orden de primero a's, luego b's, luego c's).

Los casos 4, 6 y 8 son análogos.

Caso 3

$$u = a^{N-k-l}$$

$$v = a^k$$

$$w = a^l b^p$$

$$x = b^q$$

$$y = b^{N-p-q} c^N$$

$$k+l+p+q < N$$

$$k+q > 0$$

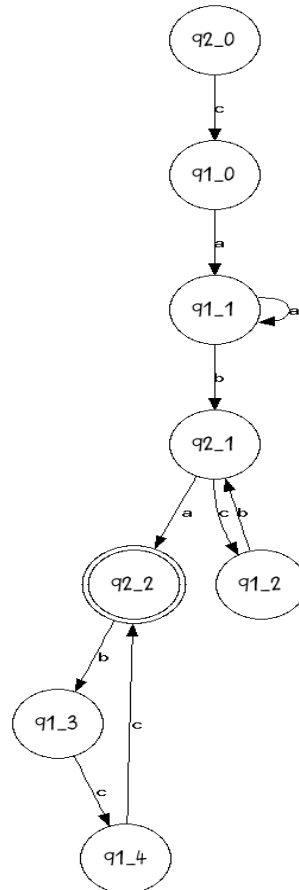
$z_i = a^{N+(i-1)k} b^{N+(i-1)q} c^N$, con $i=0$, $z_0 = a^{N-k-l} b^{N-q} c^N$. Como $k+q > 0$, al menos uno de los índices debe ser mayor que cero. Por ende, hay menos a's o b's que c's, lo cual no es posible para los prefijos de $a^n b^n c^n$.

Para el caso 7, elegimos $i=2$ y quedan más c's que a's, lo cual no es posible para las tiras del lenguaje.

Como estas son todas las descomposiciones que cumplen $|vwx| \leq N$ y $|vx| > 0$ y, para cada uno de ellos, existe un i tal que z_i no está en el lenguaje], L no es libre de contexto.

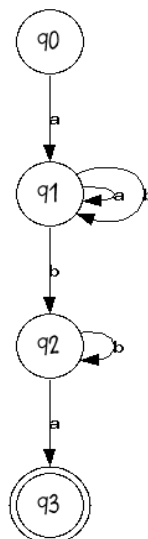
Ejercicio 4

a) Construya un Autómata Finito Determinista de dos cintas que acepte la relación:
 $\{ \langle a^k b^n c^{2t}, c^n a b^t \rangle \mid n, k > 0, t \geq 0 \}$



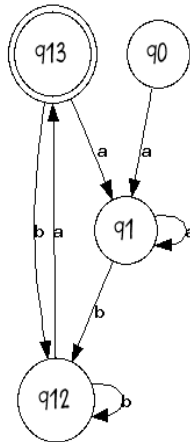
b) Dé las clases de equivalencia, según la relación RL, para el lenguaje
 $L = L(a|b)^*bb^*a$

Se construye un AFND a partir de la ER



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.

Luego se obtiene el AFD correspondiente y el AFDMínimo; el cual se muestra en el siguiente diagrama:



Luego, por el Teorema de Myhill-Nerode, las clases de equivalencia de RM (con M el autómata mínimo equivalente) equivalen con las clases de RL. Por lo cual tenemos 5 clases (la quinta correspondiente al estado pozo).

Planteamos el sistema de ecuaciones correspondiente:

$$\begin{aligned}
 X_0 &= \varepsilon \\
 X_1 &= X_0a \mid X_1a \mid X_{13}a \\
 X_{12} &= X_1b \mid X_{12}b \mid X_{13}b \\
 X_{13} &= X_{12}a \\
 X_p &= b(a|b)^*
 \end{aligned}$$

Resolviendo el sistema, tenemos una ER para cada clase de RL

$$\begin{aligned}
 X_0 &= \varepsilon \\
 X_1 &= a(a \mid b(b|ab)^*aa)^* \\
 X_{12} &= a(a \mid b(b|ab)^*aa)^*b(b|ab)^* \\
 X_{13} &= a(a \mid b(b|ab)^*aa)^*b(b|ab)^*a \\
 X_p &= b(a|b)^*
 \end{aligned}$$

c) Escriba una gramática simplificada $G_b / L_b = L(G_b)$, siendo L_b el lenguaje de la parte (b)

Como es un lenguaje regular se construye una gramática regular; en este caso será una lineal derecha.

$$\begin{aligned}
 S &\rightarrow aA \\
 A &\rightarrow aA \mid bA \mid bB \\
 B &\rightarrow a \mid bB
 \end{aligned}$$

Está simplificada porque no tiene símbolos inútiles, ni producciones épsilon ni producciones unitarias

d) Construya un autómata con salida $M: (Q, \Sigma, \Lambda, \delta, \lambda, q_0)$, con $\Sigma = \{a, b\}$; $\Lambda = \{c, \#\}$;
 $\lambda: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow (\Lambda \cup \{\epsilon\})$ que reciba una tira de símbolos definidos sobre Σ , y, leyendo de izquierda a derecha, transforme cada secuencia 'aba' en 'c', insertando un símbolo '#' cuando se lee la primera secuencia 'aba' y además para separar secuencias de 'aba' no solapadas. Ejemplos:

Entrada	Salida
ababaaba	#cc#c
abbababb	#c
bbbbbaabbb	ϵ
ababababaaaaba	#cccc#c
aba	#c
abaaabababa	#c#ccc

