

Teoría de Lenguajes
Solución 2do. Parcial – Curso 2014

Ejercicio 1 [Evaluación individual del obligatorio]

- a)
- i. Sirve
 - ii. Sirve
 - iii. No sirve, porque permite reconocer tiras con varios NOT
 - iv. Sirve
 - v. No sirve, porque desde las variables: listT y listF no se puede generar una lista de terminales (no son positivas).

- b)
- La salida i) se puede generar con las reglas a) i
- La salida ii) se puede generar con las reglas a) ii y iv

c)

Terminales	No terminales	Símbolos del meta-lenguaje
'not'	condition	::=
'or'	logic_term	?
'and'	logic_fact	(
'id'	simple_exp	)
'='	operator	*
'<'		
'>'		
'<>'		
'<='		
'>='		
'string'		
'number'		

- d) ... “de las hojas hacia la raíz”

Ejercicio 2

Dado el siguiente lenguaje:

$$L_2 = \{ x / x \in \{a,b,0,1\}^* \text{ donde } |x|_a = |x|_b > 0, |x|_0 \geq |x|_1 \geq 0 \}$$

a) Clasifique según la Jerarquía de Chomsky. Justifique.

El lenguaje L_2 es recursivamente enumerable, porque en la parte b) se construye una gramática irrestricta que lo genera o porque en la parte c) se construye una Máquina de Turing que lo reconoce.

Utilizando el contra-recíproco del Pumping Lemma, demostraremos que no es libre de contexto.

Sea N la constante del PL, consideramos la tira $z = a^N 0^N b^N 1^N$ que pertenece a L_2 y $|z| = 4N \geq N$

Consideramos todas las descomposiciones $z = uvwxy$ que cumplen $|vx| > 0$, $|vwx| \leq N$, y mostramos que en todas ellas existe algún i , para el que la tira $uv^iwx^iy \notin L_2$

Las descomposiciones son las siguientes

	aa.....aa	00.....00	bb.....bb	11.....11
Caso 1	v x			
Caso 2	v x x			
Caso 3	v v	x		
Caso 4	v	x		
Caso 5		v x		
Caso 6		v x x		
Caso 7		v v	x	
Caso 8		v	x	
Caso 9			v x	
Caso 10			v x x	
Caso 11			v v	x
Caso 12			v	x
Caso 13				v x

Caso 1:

$$u = a^j$$

$$v = a^k$$

$$w = a^l$$

$$x = a^m$$

$$y = a^{N-m-l-k-j} 0^N b^N 1^N$$

Como $k+m > 0$, entonces $z_0 = a^{N-m-k} 0^N b^N 1^N \notin L_2$, porque tiene menos a's que b's.

Caso 9:

Es análogo al caso 1, pero con menos b's que a's.

Caso 5:

Es análogo al caso 1, pero con menos 0's que 1's.

Caso 12:

Es análogo al caso 1, pero se elige $i=2$, y quedan menos 0's que 1's.

Caso 2:

$$u = a^j$$

$$v = a^k$$

$$w = a^{N-m-k-j}$$

$$x = a^m 0^l$$

$$y = 0^{N-i} b^{N1^N}$$

(Suponemos que $|x|_0 > 0$ sino estamos en el caso 1, y que $|x|_a > 0$ sino estamos en el caso 4)

$z_0 = a^{N-m-k} 0^{N-i} b^{N1^N} \notin L_2$ porque tiene menos a's que b' y/o tiene menos 0's que 1's.

Caso 3:

Es análogo al caso 2, eligiendo $i=0$, quedan menos a's que b's

Casos 6, 7, 10 y 11:

Son análogos al caso 2, pero eligiendo $i=0$, quedan menos b's que a's

Caso 4:

$$u = a^i$$

$$v = a^k$$

$$w = a^{N-j-k} 0^l$$

$$x = 0^m$$

$$y = 0^{N-l-m} b^{N1^N}$$

Como $k+m > 0$, entonces $k > 0$ y/o $m > 0$. Si $k > 0$, $z_0 = a^{N-k} 0^N b^{N1^N} \notin L_2$ porque las a's y las b's se desbalancean. Si $m > 0$, $z_0 = a^N 0^{N-m} b^{N1^N} \notin L_2$ porque la cantidad de 0's es menor a la cantidad de 1's.

Casos 8:

Es análogo al caso 4, eligiendo $i=0$ se desbalancean las a's y b's, o quedan menos 0's que 1's

Casos 12:

Es análogo al caso 4, pero se elige $i=2$, y se desbalancean las a's y b's, o quedan menos 0's que 1's

Como estas son todas las descomposiciones posibles que cumplen $|vx| > 0$, $|vwx| \leq N$, por el contra-recíproco del Pumping Lemma, demostramos que L_2 **NO** es libre de contexto

b) Construya una gramática G_2 / $L(G_2) = L_2$.

// Genero a y b controlando la cantidad

$S \rightarrow aSb \mid aRb \mid ab$

// Genero 0 y 1 (no obligatorio), permitiendo que cantidad de 0 sea mayor a cantidad de 1

$R \rightarrow 0R1 \mid 0R \mid \text{eps}$

// "Desorden" de terminales

$ab \rightarrow ba$

$a0 \rightarrow 0a$

$a1 \rightarrow 1a$

$ba \rightarrow ab$

$b0 \rightarrow 0b$

$b1 \rightarrow 1b$

$0a \rightarrow a0$

$0b \rightarrow b0$

$01 \rightarrow 10$

$1a \rightarrow a1$

1b -> b1
10 -> 10

c) Construya un autómata que reconozca a L_2 .

Avanzo de izquierda a derecha

Si viene a, marco con A y voy a buscar b que marco con X

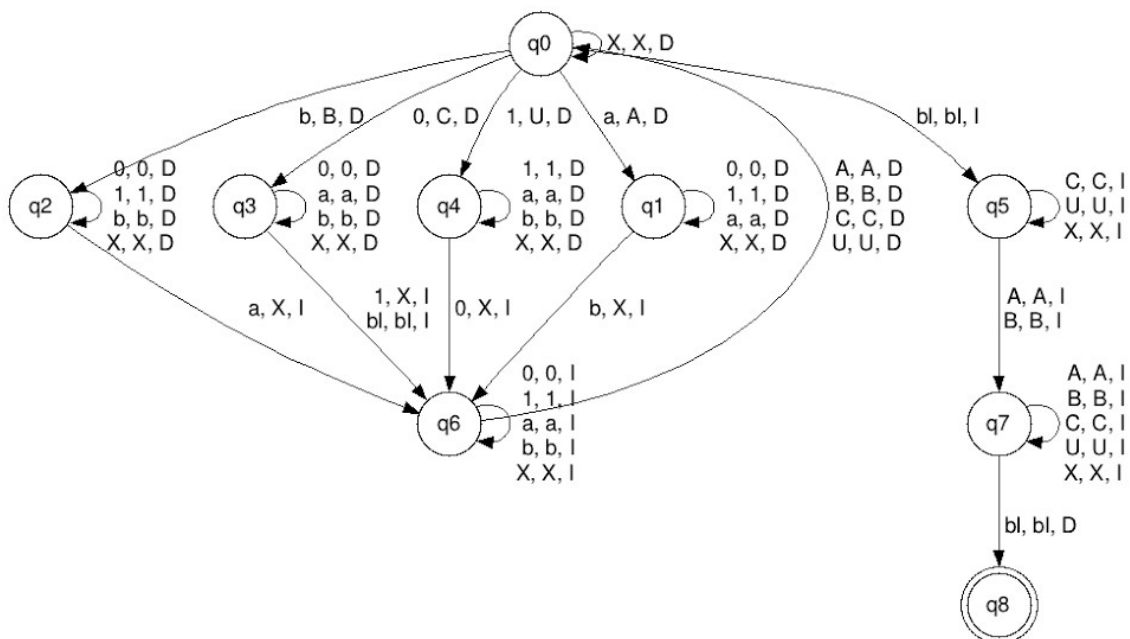
Si viene b, idem a pero marcando B

Si viene 0, marco con C y voy a buscar 1. Si no encuentro vuelvo sin problema (encuentro blanco)

Si viene 1, marco con U y busco 0, que marco con X (ídem a y b)

Si viene X, salteo X hasta encontrar a, b, 0, 1. Si encuentro blanco, vuelvo saltando A, B, C, U, X, y si llego a blanco llego a estado final

Vuelvo hasta primer A, B, C o U saltando a, b, 0, 1 y X



Ejercicio 3

Dado el siguiente lenguaje:

$$L_3 = \{ x / x \in \{a,b\}^* \text{ es de la forma } a^{2k+t}b^{k+t}a^p ; k,t,p > 0 \}$$

- a) Construya una gramática simplificada $G_3 / L(G_3) = L_3$. Explique porqué la gramática construida está simplificada.

$S \rightarrow Sa \mid Ra$
 $R \rightarrow aaRb \mid aaXb$
 $X \rightarrow aXb \mid ab$

Está simplificada porque no contienen producciones epsilon, ni producciones unitarias. Además, todos los símbolos son útiles

- b) Construya un autómata que reconozca a L_3 . ¿Es determinista? Justifique.

