Teoría de Lenguajes
Parcial Integrador – Curso 2020

Consideraciones generales
- i) Escriba nombre, C.I. y **número de parcial** 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 [Evaluación individual del obligatorio]
a) Escriba una expresión regular con la sintaxis de Python que reconozca las tiras que cumplan lo siguiente:

"Una o más letras en minúscula, seguidas por ':' o ';' y luego cero o más dígitos".

Ejemplos: "hola:" "a;32145" "ejemplo:1000"

Una posible expresión regular es: '[a-z]+(:|;) \d*'

b) Explique, en alto nivel, cómo resolvió el *programa1* del segundo laboratorio, que trata sobre el juego en el que se ingresan fichas por diferentes entradas. ¿Sus variables se correspondían con estados del juego? En caso de que sí, explique brevemente cómo.

Para esta parte no había una única solución, sino que dependía de cómo lo habían modelado en el laboratorio.

Ejercicio 2 [30 puntos]
Sea el lenguaje L_2 / L_2 = L(r), siendo r = (0|1)*010*(0|1)

a) Defina la relación R_L para un lenguaje L cualquiera

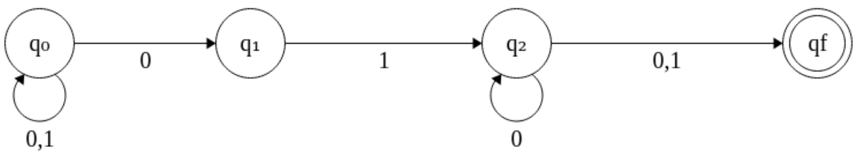
Sean dos tiras x,y $\in \Sigma^*$ y L un lenguaje cualquiera. Decimos que

$$x \; R_L \; y \leftrightarrow (\forall\, z \in \Sigma^* \; x.z \text{ sii } y.z)$$

b) ¿Cuántas clases define la relación R_{L2} ? Justifique formalmente su respuesta.

La estrategia será hallar un AFD mínimo para el lenguaje L_2 , para luego hacer uso del Teorema de Myhill–Nerode y dar la cantidad de clases de equivalencia en función del índice de $R_{M.}$

Lo primero que haremos es dar un AFND que reconozca L(r). Su construcción es sencilla dado que se deduce observando la expresión regular:



Ahora lo minizaremos:

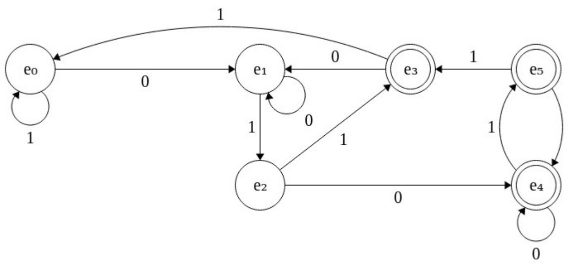
	0	1
$[q_0]$	$[q_0\,q_1]$	$[q_0]$
$[q_0\,q_1]$	$[q_0\,q_1]$	$[q_0\,q_2]$
$[q_0\,q_2]$	$[q_0\,q_1 q_2 q_f]$	$[q_0\,q_f]$
$[q_0\,q_f]$	$[q_0\,q_1]$	$[q_0]$
$[q_0\,q_1 q_2 q_f]$	$[q_0\,q_1 q_2 q_f]$	$[q_0\,q_2 q_f]$
$[q_0\,q_2 q_f]$	$[q_0\,q_1 q_2 q_f]$	$[q_0\,q_f]$

Donde $F=\{[q_0 q_f], [q_0\, q_1 q_2 q_f], [q_0 q_2 q_f]\}$

Reescribimos la tabla, renombrando los estados para mayor practicidad:

		0	1
$[q_0]$	e_0	e_1	e_0
$[q_0\,q_1]$	e_1	e_1	e_2
$[q_0\,q_2]$	e_2	e_4	e_3
$[q_0\,q_f]$	e_3	e_1	e_0
$[q_0\,q_1 q_2 q_f]$	e_4	e_4	e_5
$[q_0\,q_2 q_f]$	e_5	e_4	e_3

Donde $F=\{e_3, e_4, e_5\}$. El diagrama resulta:



Entonces ahora resta minimizar.

π_0 $[e_0 e_1 e_2]$ $[e_3 e_4 e_5]$
π_1 $[e_0 e_1][e_2]$ $[e_3][e_4 e_5]$
π_2 $[e_0][e_1][e_2]$ $[e_3][e_4][e_5]$
π_3 $[e_0][e_1][e_2]$ $[e_3][e_4][e_5]$

El algoritmo finaliza, obteniendo el mismo autómata con exactamente la misma cantidad de estados, las mismas transiciones y mismos estados finales.

Como el AFD es mínimo, por el Teorema de Myhill–Nerode el índice de R_L es el mismo que el de R_M. Por tanto, como el índice de R_M es 6, el de R_L también lo será. Entonces la relación R_{L2} define **6 clases de equivalencia**.

c) Escriba una gramática lineal izquierda simplificada G_2 / $L_2 = L(G_2)$.

La gramática lineal izquierda surge de razonar la producción de las tiras de L(r) desde la derecha hacia la izquierda. Entonces:

S → A0 | A1
A → A0 | B01 | 01
B → B0 | B1 | 0 | 1

Por último, la gramática G está simplificada dado que:

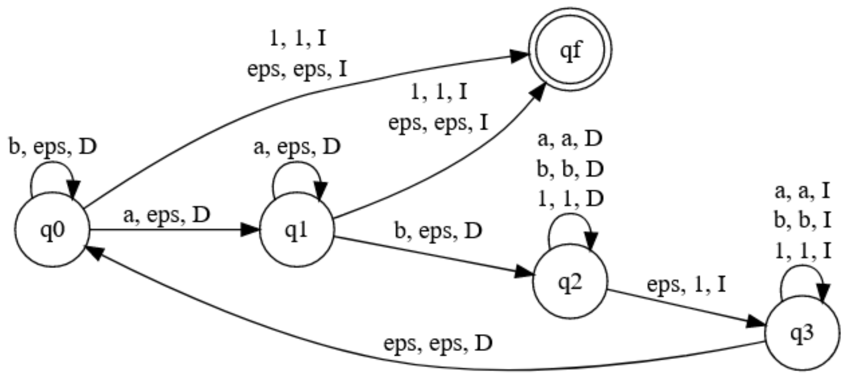
- No tiene ninguna producción épsilon
- No tiene ninguna producción unitarias
- Todas sus variables son alcanzables y positivas

Ejercicio 3 [20 puntos]
Sea la función ***f*** :{a,b}* → 1* / ***f*** (w)=1^k, donde w pertenece a {a,b}* y k es la cantidad de ocurrencias del substring ‘ab’ en w.

Ejemplos:

w	f(w)
bbabbaa	1
babaabb	11
bbbba	ε
aaababbab	111

Construya una Máquina de Turing que compute la función ***f***.



Notar que:

- al computar la función, la MT debe devolver la cinta conteniendo solamente la salida. Por tanto, se debe escribir símbolos blancos alrededor de ella
- en el caso f(bbbba)=ε la salida válida es la cinta solamente con símbolos blancos

Ejercicio 4 [40 puntos]
Sea $L_4 = \{ 10^m (01)^n / m > n \geq 0$ y $m \bmod 2 = 0\}$

a) Clasifique L_4 según la Jerarquía de Chomsky. Justifique.

El lenguaje es libre de contexto no-regular. Para probar que no es regular se utilizará el contrarrecíproco del pumping-lemma; para probar que es libre de contexto se dará una gramática que lo genere (o un autómata que lo reconozca).

Sea N la cte del PL.
Se elige $z= 10^{2N+2} (01)^{2N+1}$ / $|z| = 1+2N+2+2*(2N+1) = 6N+5 \geq N$.
Se consideran todas las descomposiciones de z = uvw que cumplan (i) $|uv| \leq N$ y (ii) $|v| \geq 1$
Se tiene entonces dos familias a estudiar y encontrar $i \geq 0$ / $uv^i w \notin L_4$

Caso 1:
$u = \varepsilon$
$v = 10^q$ con $q \geq 0$
$w= 0^{(2N+2)-q}(01)^{2N+1}$

Por tanto, $z_i = (10^q)^i\, 0^{(2N+2)-q}\, (01)^{2N+1}$
Tomando i=0, $z_0=0^{2N+2-q} (01)^{2N+1}$
Donde se puede observar que z_0 NO comienza con 1, por lo que $z_0 \notin L_4$

Caso 2:
$u=10^p$
$v = 0^q$ con $q \geq 1$
$w= 0^{(2N+2)-p-q} (01)^{2N+1}$

Por tanto, $z_i = 10^p 0^{iq} 0^{(2N+2)-p-q} (01)^{2N+1} = 10^{2N+2+q(i-1)} (01)^{2N+1}$
Tomando i=0, $z_0 = 10^{2N+2-q} (01)^{2N+1}$
Donde se puede observar que para respetar la relación entre los superíndices m y n de L_4 se debe cumplir:

$$2N+2-q > 2N+1$$
$$2-q>1$$
$$1>q$$

pero se tenia que $q \geq 1$, por lo que $z_0 \notin L_4$

Como se probó para todas las descomposiciones posibles que cumplan (i) y (ii), se encontró un i de forma que $uv^i w \notin L_4$, entonces por el contrarrecíproco del pumping-lemma podemos afirmar que L_4 NO es un lenguaje regular.

En la siguiente parte daremos una gramática libre de contexto, probando que es un lenguaje libre de contexto, concluyendo que L_4 es libre de contexto no-regular.

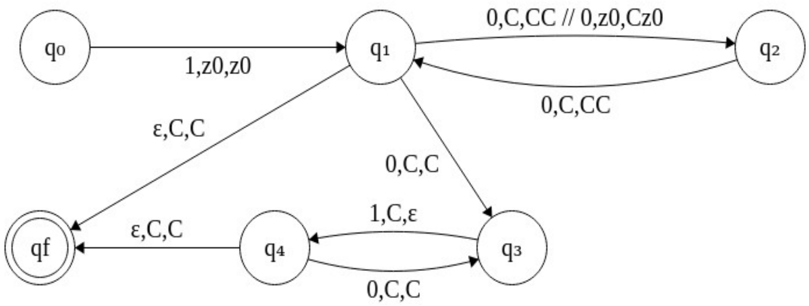
b) Construya una gramática G_4 / $L_4=L(G_4)$.

S -> 1A
A -> 00A0101| 00A | 00 | 0001

Con la primer producción de A se produce n=m=2k
Con la segunda, se permite incrementar el índice m, controlando paridad.
Con la tercera, se fuerza a que el terminal **con índice n par** respete la desigualdad m>n
Con la cuarta, se fuerza a que el terminal **con índice n impar** respete la desigualdad m>n

Nota: Se valora positivamente la simplicidad de las soluciones propuestas, así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

c) Construya un autómata M_4 / L_4=L(M_4). ¿Es determinista? Justifique.



Este autómata push-down reconoce por estado final al lenguaje L_4.

El autómata **no es determinista** dado que estando en el estado q_1 , al tener C en el tope del stack, se tiene una transición épsilon hacia el estado final q_f o una transición al estado q_3 al leer un 0.