

CMC Boletín 1

1.

$$L = \{x \in \{a, b, c\}^* \mid |x|_a = |x|_b = |x|_c\}$$

En	a	b	c		
Sa	✓				
Sb	✓				
Sc	✓				
M	Δ				

λ: símbolo de relleno

Para la realización de este ejercicio se empleará una MT multicinta. Cintas:

- "En": lugar donde se almacenará la palabra que se esté analizando
- "Sa, Sb, Sc": cintas que nos permitirán ver el número de apariciones de cada símbolo
- "M": cinta que marcará la primera celdilla

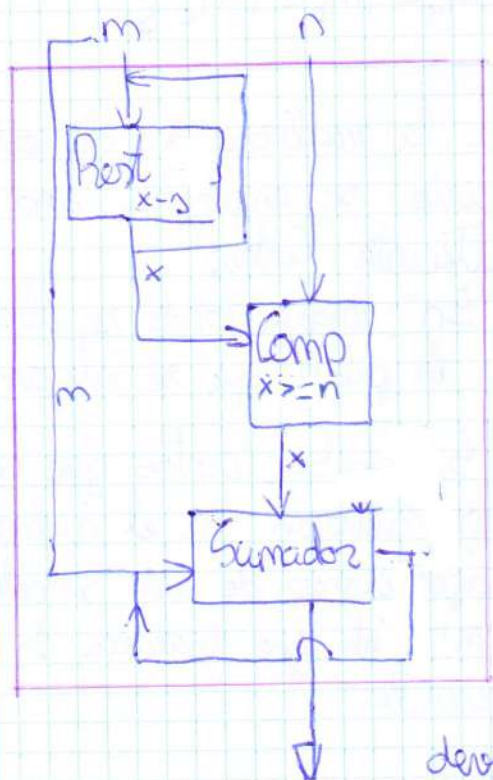
Funcionamiento:

En primer lugar se recorrerá celdilla a celdilla la cinta de entrada hasta que se encuentre λ. Para cada celdilla avanzada la cinta del signo observado se marcará y avanzará una posición. Ejemplo: en la primera celdilla de "En" hay un símbolo 'a', entonces en "Sa" se avanza el cabezal después de poner el símbolo "✓" en la cinta.

Una vez analizada toda la cinta "En" los cabezales de las cintas "Sx" se sitúan en la primera celdilla marcada con 'Δ' en la cinta M

Por último los caracteres de todas las cintas "S" avanzan hasta que una o más de una encuentren 'X'. Si en las 3 cintas se encuentra a la vez 'X' se acepta, si no se rechaza.

2.



La MT sugerida como solución consta de 3 módulos:

- Rest: recibe un número y le resta 1. Se basa en copiar a la salida todos los símbolos de la entrada menos el último.

- Comp: recibe dos números (x, n) como argumentos y

si $x \geq n$ devuelve x , si no devuelve λ . Este módulo va tachando un símbolo de x , uno de n sucesivamente

hasta que llega a λ , si ha llegado cuando tachaba símbolos de n devuelve λ si no devuelve x .

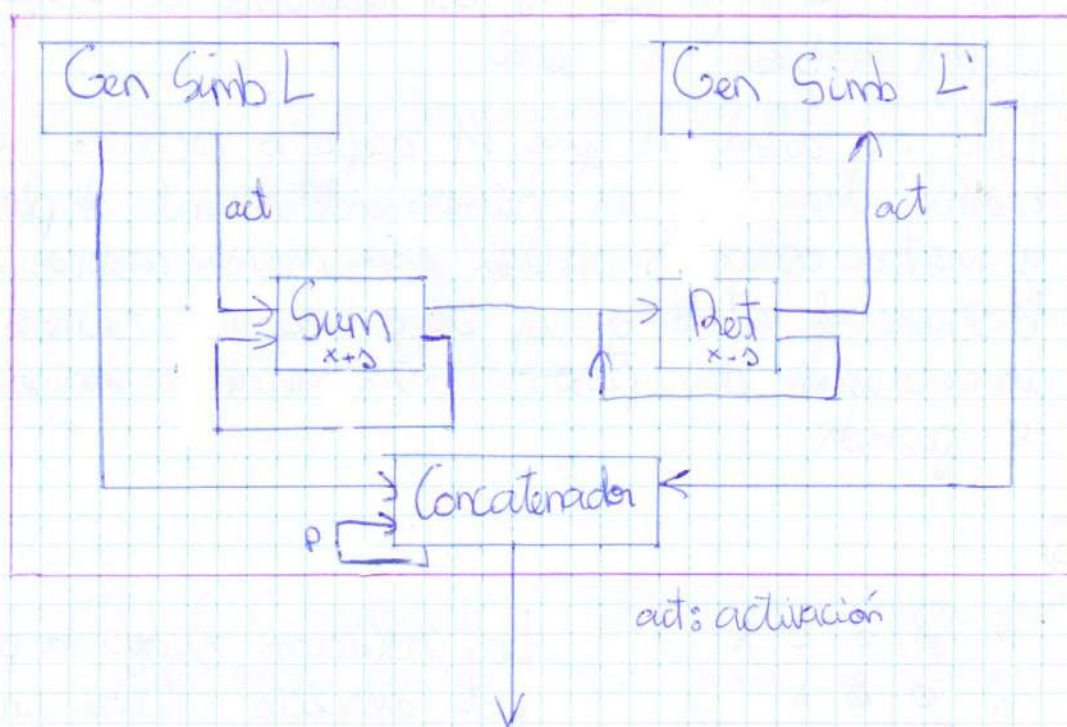
- Sumador: recibe dos parámetros y los suma (concatena los símbolos), mientras ninguno de los parámetros sea 0 realimenta su entrada con su salida.

Esta MT recibe m y n , n alimenta en todo momento al módulo Comp y m se utiliza como entrada de Rest y Sumador la primera vez.

El módulo Rest genera números cada vez menores, para cada número generado en temp se comprueba que sea superior al límite inferior ese número se pasa a Sumador que lleva la suma parcial acumulada que suma con dicho número

3.

$L \diamond L'$ es recursivamente enumerable ya que puede ser generado por la siguiente MT:



Esta MT consta de 5 módulos:

- Gen Simb L / Gen Simb L': generan símbolos del lenguaje correspondiente
- Sumador: cuando recibe una señal de activación suma s a su valor. Funcionamiento: al recibir un número concatena ese número a '0'.
- Restador: resta s a su valor. Su funcionamiento es equivalente al módulo Rest del ejercicio 2

- Concatenador: concatena el valor de su entrada p con el símbolo que le lleve

A grandes rasgos el funcionamiento de esta MT se basa en generar la palabra x y conocer su longitud gracias al Sumador para a continuación concatenarle $|x|$ símbolos del lenguaje L .

4

Si el lenguaje $L(M)$ es recursivo el lenguaje $L_q(M)$ también lo será.

Esto es debido a que M acepta o rechaza la cadena dada en un número indeterminado de pasos y nuestra operación analiza esa misma cadena x buscando el estado q y luego hace un número indeterminado de pasos hasta alcanzar un estado de parada.

5.

m	\emptyset	\emptyset	0	λ	
n	\emptyset	\emptyset	λ		
$copm$					
res	0	0	λ		

La solución sugerida para este ejercicio utiliza una MT formada por las siguientes cintas:

- m : almacena el segundo parámetro
- n : almacena el primer parámetro
- $copm$: almacena una copia del valor m
- res : almacena el valor a devolver

Funcionamiento:

En primer lugar se copia el valor n en las cintas n y res , y el valor m en las cintas m y $copm$.

A continuación se tachan símbolos de m y n a la vez hasta que se encuentre un símbolo de relleno. Si se encuentra el primer símbolo en n se acepta el valor en res , si no se copia $copm$ en res y se acepta el valor de res .

6.

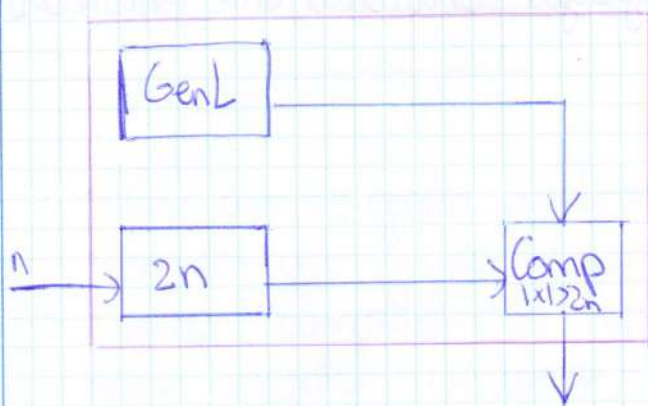
$No, \mu(L, L')$ no es también recursivamente enumerable.

Esto es debido a que la operación μ utiliza el complemento de L' y la complementación es una operación no cerrada para los lenguajes recursivamente enumerables.

7.

a y b

$P_n(L)$ es recursivo y recursivamente enumerable ya que existe una MT que genere el lenguaje en orden canónico



La MT propuesta consta de los siguientes tres módulos:

- GenL: genera palabras del lenguaje L en orden canónico
- $2n$: Recibe un número y duplica su valor. Su funcionamiento básico es concatenar n consigo mismo.

- Comp: recibe dos palabras y si $|x| > |n|$ acepta x .
 Su funcionamiento se basa en tachar símbolos de las dos palabras hasta que encuentre un símbolo blanco, si lo encuentra antes en x acepta la palabra, si no la rechaza.

A grandes rasgos la MT va generando palabras en orden canónico y aceptando aquellas que cumplan la condición

c y d

En ambos casos se plantea la diferencia entre dos lenguajes. Dicha operación se puede escribir de la siguiente manera:

$$P_m(L) - P_n(L) = P_m(L) \cap \overline{P_n(L)}$$

Una vez visto este planteamiento se puede concluir:

- La operación es recursiva ya que tanto la intersección como el complemento son operaciones cerradas para los lenguajes recursivos.
- La operación no es recursivamente enumerable ya que la complementación no es una operación cerrada para los lenguajes recursivamente enumerables.

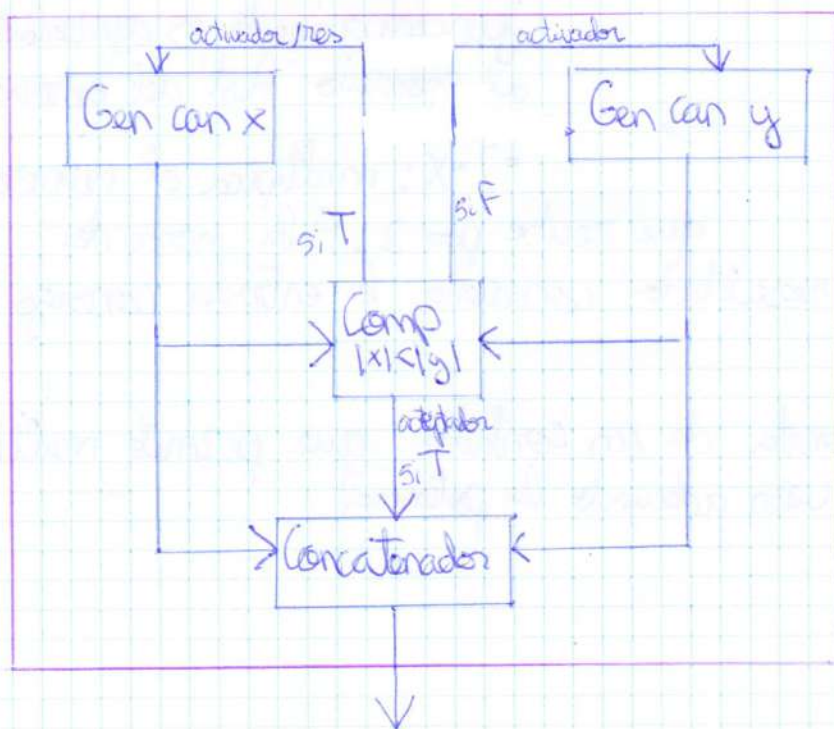
8

La operación P es cerrada tanto para los lenguajes recursivos como los recursivamente enumerables.

Esto es debido a que la operación P consiste en concatenación y potencia. Ambas operaciones son cerradas para los lenguajes recursivos y recursivamente enumerables de manera que la operación P también es cerrada.

9

Ya que existe una MT que puede generar $P(L)$ en orden canónico $P(L)$ es cerrada para los lenguajes recursivos y los recursivamente enumerables. La MT en cuestión es:

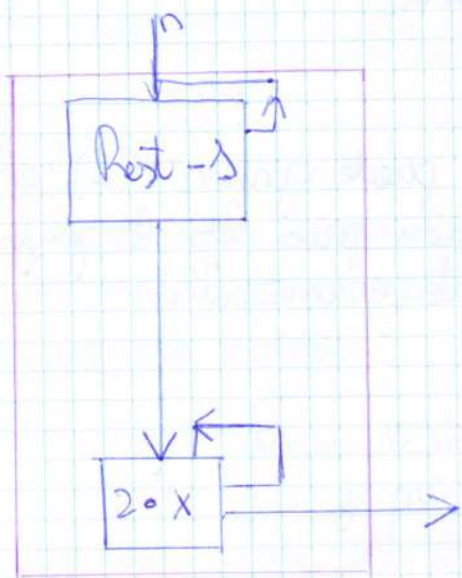


- Gen can x: genera palabras de L_1 en orden canónico
- Gen can y: genera palabras de L_2 en orden canónico
- Comp: comprueba si $|x| < |y|$. Tacha símbolos de x e y hasta encontrar un blanco, si lo encuentra antes en x envía una señal de activación a "Gen can y", sino se la envía a "Gen can x" y a "Concatenador".

- Concatenador: al recibir la señal de activación concatena la palabra x con la y . Su funcionamiento es similar al de "Concatenador" del ejercicio 3.

Esta MT genera palabras de L_1 y L_2 y las concatena si $|x| < |y|$

10



La MT propuesta consta de dos módulos

- Rest-1: Resta 1 al número que recibe. Su funcionamiento es equivalente al módulo "Rest" del ejercicio 3.

• $2 \cdot X$: multiplica el número que recibe por 2. A la hora de calcular el resultado concatena la entrada consigo misma.

Esta MT consta de un contador que permite multiplicar el número de veces adecuado la potencia