Repaso Practica 1 Ejercicios

Ejercicio 2.

Construir una MT, con cualquier cantidad de cintas, que acepte de la manera más eficiente posible el lenguaje L = {$a^n b^n c$ n | n ≥ 0}. Comentario: Plantear primero la idea general.

<u>Idea General:</u>

Una MT con 2 cintas:

1. Copia las a de la cinta 1 a la cinta 2.

2. Devuelve los cabezales de la cinta 2 y 1 tal como están.

3. Compara las a de la cinta 2 (se recorre para la izquierda) con las b de la cinta 1 (se recorre para la derecha). Básicamente cuando hay una a en la cinta 2 tiene que haber una b en la cinta 1. Cuando dejan de haber a en la cinta 2 tienen que dejar de haber b en la cinta 1, si sucede esto, se pasa al siguiente punto, caso contrario se rechaza.

4. Compara las a de la cinta 2 (se recorre para la derecha) con las c de la cinta 1 (se recorre para la derecha). Si cuando dejan de haber a en la cinta 2 dejan de haber c en la cinta 1, se acepta. Caso contrario se rechaza.

<u>Construcción</u>

Definición de la MT = (Q, Σ, δ, q0, qA, qR)

Q = {q0, q1, q2}

q0: copia las a de la cinta 1 a la cinta 2

q1: compara las a con las b.

q2: compara las a con las c.

Alfabeto Σ = {a, b, c, B}

	a, a	a, b	a, c	a, B	b, b	b, a	b, c	b, B	c, c	c, a	c, b	c, B	B,B	B, a	B, b	B, b
q0				q0, a, R, a, R				q1, b, S, B, L					q1, B, S B, L			
q1						q1, b, R, a, L						q2 c, S, B, R	q2, B, S B, R			
q2										q2, c, R,			qA, B, S B, S			

										a, R						

Todas las celdas en blanco son rechazos de la MT.

(verificar después)

Ejercicio 4.

Probar:

1. La clase R es cerrada con respecto a la operación de unión. Ayuda: la prueba es similar a la desarrollada para la intersección.

2. La clase RE es cerrada con respecto a la operación de intersección. Ayuda: la prueba es similar a la desarrollada para la clase R.

1.

Idea General:

Dada dos MT M1 y M2 que respectivamente aceptan L1 y L2 y paran siempre, la idea es construir una MT M que acepte L1 U L2 (L = L1 U L2) y pare siempre.

Construcción:

- M tiene 2 cintas:
- Dada la entrada w en la cinta 1, M hace:
  1) Copia la entrada w de la cinta 1 a la cinta 2.
  2) Ejecuta la M1 sobre el w en la cinta 2. Si la MT M1 para en qA, entonces M para en qA. Si la MT M1 para en qR, entonces:
     - Borra el contenido de la cinta 2 y se vuelve a copiar el w de la cinta 1 a la cinta 2.
     - Ejecuta M2 sobre el w de la cinta 2. Si la MT M2 para en qA, entonces M para en qA. Si la MT M2 para en qR entonces M para en qR.

Verificación de correctitud:

- L = (L1 U L2)

M acepta las entradas que son aceptadas por M1 o M2, es decir, acepta los inputs que son aceptados por alguna de las dos maquinas (la unión). Si una entrada es rechazada por ambas maquinas, entonces la MT M la rechazara. Si es una entrada que es aceptada por algunas de las maquinas, entonces MT M la aceptara. Por lo tanto L = (L1 U L2)

- L pertenece a R

Como M1 y M2 siempre paran, al M ser una MT que simula la ejecución de ambas maquinas y hace un copiado y borrado finito, siempre se va a detener.

2.

Idea General:

Dada dos MT M1 y M2 que respectivamente aceptan L1 y L2, la idea es construir una MT M que acepte L1 n L2 (L = L1 n L2)

Construcción:

- M tiene 2 cintas.
- Dada la entrada w en la cinta 1, M hace:
  1) Copia la entrada de la cinta 1 a la cinta 2
  2) Ejecuta la MT M1 sobre el w en la cinta 2. Si la MT M1 para en qR, M para en qR, si M1 loopea, M loopeara. Si M1 para en qA, entonces M hace lo siguiente:
     - Borra el contenido de la cinta 2 y vuelve a copiar la entrada w de la cinta 1 a la cinta 2
     - Ejecuta la MT M2 sobre el w en la cinta 2. Si la MT M2 para en qR, M para en qR, si M2 loopea, M loopeara y si M1 para en qA entonces M para en qA.

Verificación de correctitud:

- L(M) = L = L1 n L2

La MT M reconoce las entradas que son reconocidas tanto por la MT M1 como por la MT M2, es decir, reconoce las entradas que son reconocidas por ambas maquinas (la intersección). Si alguna o ambas maquinas rechazan una entrada entonces M también la rechazara, solo si ambas maquinas aceptan la entrada M la aceptara. Por lo tanto L(M) = L1 n L2.

- L pertenece a RE

Como L = L(M), existe una maquina que acepta el lenguaje (la que se acabo de construir). Esta MT M que lo acepta al estar simulando la ejecución de dos MT (M1 y M2) que pueden loopear sobre un input, entonces M también loopeara. Por lo tanto L pertenece a RE ya que existe una maquina que lo acepta (y no necesariamente para).

Ejercicio 5.

Sean L1 y L2 dos lenguajes recursivamente numerables de números naturales codificados en unario (por ejemplo, el número 5 se representa con 11111). Probar que también es recursivamente numerable el lenguaje L = {x | x es un número natural codificado en unario, y existen y, z, tales que y + z = x, con y ∈ L1, z ∈ L2}. Ayuda: la prueba es similar a la vista en clase, de la propiedad de clausura de la clase RE con respecto a la operación de concatenación.

Idea General:

L1 . L2 es el lenguaje que contiene todas las cadenas w = x1x2, en donde la subcadena x1 ∈ L1 y la subcadena x2 ∈ L2.

Dada las dos MT M1 y M2 que respectivamente aceptan los lenguajes L1 y L2, la idea es construir una MT M que acepte L1 . L2 (L = L1 . L2). Se sabe que las MT M1 y MT M2 existen puesto que L1 y L2 son recursivamente numerables y, por definición, existe una máquina que acepta al L1 otra que acepta el L2.

La idea es que MT M va a simular en "paralelo" la ejecución de M1 y M2 sobre todas las posibles particiones de la entrada (un w con n símbolos), es decir:

1) Se parte de un i = 1 que indica la cantidad a pasos a ejecutar.
2) Se ejecutan i pasos de la MT M1 sobre el símbolo 0 de la cadena, e i pasos de la MT M2 sobre los n símbolos restantes de la cadena, luego i pasos de la MT M1 sobre el símbolo 1 de la cadena e i pasos de sobre los (n – 1) símbolos restantes de la cadena... y asi siguiendo con todas las posibles particiones.
3) Se incrementa i (la cantidad de pasos) y se vuelve al punto 2.
4) Si en algún momento ambas maquinas aceptan, entonces M acepta.

Construcción:

- La MT M tiene 6 cintas
- En las cintas 2 y 3 M ejecuta M1 y M2 respectivamente.
- Dada la entrada w en la cinta 1 donde |w| = n, M hace lo siguiente:

1) Escribe el número 1 en la cinta 4. Este número será i.
2) Escribe el número 0 en la cinta 5. Este número será h
3) Escribe el numero n en la cinta 6. Este número será k.
4) Copia los primeros h símbolos de w en la cinta 2 y los k símbolos restantes de w en la cinta 3.
5) M ejecuta a lo sumo i pasos de la MT M1 sobre el contenido de la cinta 2 e i pasos de la MT M2 sobre el contenido en la cinta 3. Si en algún momento ambas maquinas paran en qA, entonces M para en qA.
6) Si h = n, entonces se incrementa i, se borra el contenido de las cintas 2, 3, 5, 6 y se vuelve al paso 2. (si no se cumple h = n este paso se ignora)
7) Se hace h = h + 1 y k = k – 1 en la cinta 5 y 6 respectivamente, se borra el contenido de las cintas 2 y 3 y se vuelve al paso 4.

Verificación de correctitud:

- L = L1 . L2

M va a reconocer las entradas que están compuestas por dos subcadenas, siendo la primera reconocida por M1 y la segunda subcadena reconocida por M2.

- Si w ∈ L entonces este w puede ser dividido en dos subcadenas en donde la primera subcadena ∈ L1 y la segunda subcadena ∈ L2, por lo tanto eventualmente tanto M1 como M2 aceptaran (M1 la primer subcadena y M2 la segunda subcadena), llevando a M a aceptar.
- Si w no pertenece a L, entonces por construcción M1 o M2 eventualmente rechazaran todas las posibles subcadenas, llevando a M a rechazar.

- L ∈ RE

Como L(M) = L, existe una MT que acepta el lenguaje (la que se construyo). Como esta MT M simula la ejecución de dos MT que pueden loopear sobre un input, entonces M loopeara. Por lo tanto L pertenece a RE ya que existe una máquina que lo acepta (y no necesariamente para).

Ejercicio 6.

Dada una MT M1 con alfabeto Σ = {0, 1}:

1. Construir una MT M2 que determine si L(M1) tiene al menos una cadena.

2. ¿Se puede construir además una MT M3 para determinar si L(M1) tiene a lo sumo una cadena? Justificar.

1. La MT M2 hace lo siguiente:

    i. Hace i=1.
    ii. Ejecuta i pasos de M1 sobre los símbolos de longitud a lo sumo i. Estos símbolos son generados por la MT M2.
    iii. Si M1 acepta, entonces M2 acepta.
    iv. Caso contrario, se incrementa i y se vuelve al paso ii.

2. No, no se puede construir un MT M3 para determinar si L(M1) tiene a lo sumo una cadena. Van a haber situaciones donde en efecto L(M1) tendrá solamente una cadena, por lo que M3 debería aceptar, pero en su lugar se quedara buscando la segunda cadena, que nunca encontrara puesto que no existe, lo que la llevara a loopear (debido a que la cantidad de strings es infinita, buscara infinitamente) y por lo tanto, rechazar.