Practica 4

1) Construir una máquina de Turing que escriba en la primera cinta las palabras de {0,1}* en orden canónico separadas por un símbolo ";". Obviamente esta máquina nunca se detiene.

   q0,B,B
   q1,;,D,0,S

   q1,B,0
   q1,0,D,0,D

   q1,B,1
   q1,1,D,1,D

   q1,B,B
   q3,;,S,B,I

   q3,;,0
   q4,;,S,1,I

   q3,;,1
   q5,;,S,0,I

   q5,;,B
   q1,;,D,0,S

   q5,;,0
   q4,;,S,1,I

   q5,;,1
   q5,;,S,0,I

   q4,;,0
   q4,;,S,0,I

   q4,;,1

q4,;,S,1,I

q4,;,B
q1,;,D,B,D

2) Sean Σ ={a,b} y $\mathcal{L}$ el conjunto de todos los lenguajes definidos sobre Σ. Diga si las siguientes afirmaciones son verdaderas o falsas:

a) $\mathcal{L} - R = \emptyset$
   Falso
   $L_D \in \mathcal{L}$ pero no a R, por lo que si se hace $\mathcal{L} - R$ por lo menos L_D seguirá estando en $\mathcal{L}$

b) $\Sigma^* \in R$
   Verdadera
   Existe una MT tal que acepta el lenguaje $\Sigma *$ ($q0w \vdash qA$)

c) $ab \in \Sigma *$
   Verdadera
   $\Sigma * = (\lambda,a,b,aa,ab,ba,bb,\ldots)$

d) $RE - R \neq \emptyset$
   Verdadero
   Por absurdo: $RE - R = \emptyset$
   $Lu \in (RE - R)$
   $Lu \in \emptyset$ (ABSURDO)
   $\therefore RE - R \neq \emptyset$

e) $\emptyset \in RE$
   Verdadero
   Existe una MT que acepta el lenguaje $\emptyset$ tal que ($q0w \vdash qR$) $\therefore \emptyset \in R$ y por definición $\emptyset \in RE$

f) $CO\text{-}R \subset CO\text{-}RE$
   Verdadero
   $L \in CO\text{-}R \Rightarrow$ (def. Co-R)

L̶ ∈ R ⇒ (def. R y RE)

L̶ ∈ RE ⇒ (def. Co-Re)

L ∈ CO-RE

g) {λ} ∈ (ℒ – CO-RE)

Falso

Existe una MT tal que acepte el lenguaje {λ} y sea recursivo (q0w ⊢ qA, w =λ) o (q0w ⊢ qR))

∴ {λ} ∈ R ⇒ {λ} ∈ CO-RE (Teorema 3)

∴ {λ} ∉ (ℒ – CO-RE)

h) CO-RE = RE

Falso

Por absurdo: CO-RE - RE = ∅

L_D ∈ (CO-RE - RE)

L_D ∈ ∅ (ABSURDO)

∴ CO-RE ≠ RE

i) a ∈ R

Falso

R = {∅, {λ}, {a}, {b}, {aa}, {ab}, …) *a ∉ R.*

*"a" no es un lenguaje. El conjunto R se compone de lenguajes (conjuntos de cadenas), no de cadenas individuales (*a es una cadena y {a} es el lenguaje que contiene la cadena a)

La diferencia entre "a ∉ R" y "{a} ∈ R" se debe a la forma en que se definen los lenguajes y las cadenas en la teoría de la computabilidad.

<u>a ∉ R (a no pertenece a R):</u>

En el contexto de la teoría de la computabilidad, el conjunto de lenguajes recursivos (R) está formado por lenguajes que son decidibles por una máquina de Turing, es decir, hay una máquina de Turing que puede aceptar o rechazar cada cadena de entrada en un tiempo finito. Un lenguaje se considera recursivo si hay una MT que siempre se detiene y decide si una cadena dada está en el lenguaje o no.

La notación "a ∉ R" significa que la cadena "a" no pertenece a ningún lenguaje recursivo. Esto implica que no existe una máquina de Turing que pueda tomar la cadena "a" como entrada y siempre detenerse en un tiempo finito para decidir si "a" está en el lenguaje o no. En otras palabras, "a" es una cadena que no es decidible por ninguna máquina de Turing.

<u>{a} ∈ R ({a} pertenece a R):</u>

Por otro lado, cuando tienes un conjunto que contiene una sola cadena, como "{a}", esta notación se refiere al lenguaje que contiene solo esa cadena. En este caso, el lenguaje es *{a}. El hecho de que "{a} $\in$ R" significa que el lenguaje que contiene solo la cadena "a" es un lenguaje recursivo. Esto implica que existe una máquina de Turing que puede decidir en tiempo finito si una cadena dada es igual a "a" o no.*

j) RE $\cup$ R = $\mathcal{L}$

Falso

RE $\cup$ R = RE

$L_D \in \mathcal{L}$ pero $L_D \notin$ RE

$\therefore$ RE $\cup$ R $\neq \mathcal{L}$

k) ($\mathcal{L}$ – RE) = CO-RE

Falso

Existe L = {1w / w $\in$ LD } $\cup$ {0w / w $\notin$ LD } que $\notin$ (RE $\cup$ CO-RE)

Existe un lenguaje en $\mathcal{L}$ - RE que no está en CO-RE

$\therefore$ ($\mathcal{L}$ – RE) $\neq$ CO-RE, ($\mathcal{L}$ – RE) = L $\cup$ CO-RE

l) {a} $\in$ RE

Verdadero

{a} $\in$ R $\Rightarrow$ (def. R y RE)

{a} $\in$ RE

3)