Universidad Nacional de La Plata

Facultad de informática

Computabilidad y Complejidad

Resolución - Prácticas

Autor:

Lautaro Castro

Índice

Práctica 1

Ejercicio 1

1) Probar la siguiente ley distributiva $\boldsymbol{A \cup (B \cap C) = (A \cup B) \cap (A \cup C)}$

$x \in (A \cup (B \cap C)) \Leftrightarrow$ (definición de intersección y unión)

$x \in A \vee (x \in B \wedge x \in C) \Leftrightarrow$ (distributiva lógica proposicional)

$(x \in A \vee x \in B) \wedge (x \in A \vee x \in C) \Leftrightarrow$ (definición de intersección y unión)

$x \in ((A \cup B) \cap (A \cup C))$

Ejercicio 2

Probar la siguiente ley de Morgan: El Complemento de A unión B es igual al complemento de A intersección el complemento de B

$x \in (A \cup B)^C \Leftrightarrow$ (Definición de complemento)

$\neg(x \in A \vee x \in B) \Leftrightarrow$ (Ley de Morgan lógica proposicional)

$x \notin A \wedge x \notin B \Leftrightarrow$ (Definición de complemento y conjunción)

$A^C \wedge B^C$

Ejercicio 3

Probar que el complemento del complemento de A es igual a A

$x \in (A^C)^C \Leftrightarrow$ (Definición de complemento)

$x \notin (x \notin A) \Leftrightarrow$

$\neg(\neg(x \in A)) \Leftrightarrow$ (Doble negación lógica proposicional)

$x \in A \Leftrightarrow$

A

Ejercicio 4

Sea A el conjunto de los números naturales tales que, si son mayores que 5 o bien terminan en 5,entonces contienen algún dígito 1 ó 2.

Inciso a

Cuáles de los siguientes números pertenecen a A: 3, 5, 10, 15, 30, -10

3, 10, 15 pertenecen a A

Inciso b

Expresar el enunciado como una fórmula proposicional donde m significa mayores que 5, t es terminan en 5, u es contiene algún dígito 1 y d es contiene algún dígito 2

$(m \vee t) \rightarrow (u \vee d)$

Inciso c

Transformar la fórmula del inciso anterior de manera que no tenga una implicación y aplicar una ley de Morgan al resultado. Expresarlo en una frase.

$(m \vee t) \rightarrow (u \vee d) \Leftrightarrow$ (Implicación)

$\neg((m \vee t) \wedge \neg(u \vee d)) \Leftrightarrow$ (Morgan)

$\neg(m \vee t) \vee (u \vee d)$

A es el conjunto de los números naturales tales que no cumplen que es mayor 5 o termina en 5 o que contenga algún dígito 1 o 2.

Ejercicio 5

Sean:

$\boldsymbol{X = \{x/x \in, x \text{ es impar}\}}$

$\boldsymbol{Y = \{y/y \in, y \text{ es primo}\}}$

$\boldsymbol{Z = \{z/z \in, z \text{ es múltiplo de 3}\}}$

Describir cada uno de los siguientes conjuntos:

a) $\boldsymbol{X \cap Y}$

b) $\boldsymbol{X \cap Z}$

c) $\boldsymbol{Y \cap Z}$

d) $\boldsymbol{Z - Y}$

e) $\boldsymbol{X - (Y \cap Z)}$

f) $\boldsymbol{(Y \cap Z) - X}$

g) $\boldsymbol{X \cup Y}$

Resolución:

a) $X \cap Y = \{x : x \in: \forall a \in: n \in: (a|x) \rightarrow [(a = 0 \vee a = x) \wedge x = 2n + 1]\} = \{Y - \{2\}\}$

b) $X \cap Z = \{x : x \in: n \in: x = 2n + 1 \wedge x = 3n\}$

c) $Y \cap Z = \{3\}$

d) $Z - Y = \{z : z \in: n \in: z = 3n \wedge \neg(z = 2n + 1)\}$

e) $X - (Y \cap Z) = X - \{3\} = \{x : x \in: n \in: x = 2n + 1 \wedge x \neq 3\}$

f) $(Y \cap Z) - X = \{3\} - X = \varnothing$

g) $X \cup Y = \{x : x \in: n \in: x = 2n + 1 \vee x = 3n\}$

Ejercicio 6

Calcular los conjuntos de partes en los siguientes casos

a) $\varnothing$

b) $\boldsymbol{\{a,\ b,\ c\}}$

c) $\boldsymbol{\{\varnothing\}}$

d) $\boldsymbol{\{\varnothing,\ \{\varnothing\}}$

e) $\boldsymbol{\{a, \{b,\ c\}\}}$

Resolución:

a) $\varnothing = \varnothing$

b) $\{a,\ b,\ c\} = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$

c) $\{\varnothing\} = \{\varnothing, \{\varnothing\}\}$

d) $\{\varnothing,\ \{\varnothing\} = \{\varnothing, \{\varnothing\}, \{\{\varnothing\}\}, \{\varnothing, \{\varnothing\}\}\}$

e) $\{a, \{b,\ c\}\} = \{\varnothing, \{a\}, \{\{b,c\}\}, \{a, \{\{b,c\}\}\}$

Ejercicio 7

Presentar una lista con todos los elementos en cada uno de los conjuntos siguientes:

a) $\boldsymbol{\{x, y\} \times \{a, b, c\}}$

b) $\boldsymbol{\{a, b, c\} \times \{x, y\}}$

c) $\boldsymbol{\{x, y\} \times \{y, x\}}$

d) $\boldsymbol{\{x, y\}^2 \times \{\}}$

e) $\boldsymbol{\{\}^{10} \times \{2, 3, 4\}^{20}}$

f) $\boldsymbol{\{1\}^5 = \{(1, 1, 1, 1, 1)\}}$

g) $\boldsymbol{\{1, 2\} \times \{a\} \times \{a, b\}}$

Resolución:

a) $\{x, y\} \times \{a, b, c\} = \{(x, a), (x, b), (x, c), (y, a), (y, c), (y, c)\}$

b) $\{a, b, c\} \times \{x, y\} = \{(a, x), (a, y), (b, x), (b, y), (c, x), (c, y)\}$

c) $\{x, y\} \times \{y, x\} = \{(x, y), (x, x), (y, x), (y, y)\} = \{x, y\}^2$

d) $\{x, y\}^2 \times \{\} = \varnothing$ El conjunto vacío no contiene elementos para construir pares ordenados

e) $\{\}^{10} \times \{2, 3, 4\}^{20} = \varnothing$ El conjunto vacío no contiene elementos para construir pares ordenados

f) $\{1\}^5 = \{(1, 1, 1, 1, 1)\}$

g) $\{1, 2\} \times \{a\} \times \{a, b\} = \{(1, a, a), (1, a, b), (2, a, a), (2, a, b)\}$

Ejercicio 8

¿Cuál es el cardinal de $\boldsymbol{A \times B}$ ***si*** $\boldsymbol{|A| = n}$ ***y*** $\boldsymbol{|B| = m}$***?***

El cardinal de $|A_1 \times A_2 \times ... \times A_n| = |A_1| * |A_2| * ... * |A_3| \Rightarrow |A \times B| = n * m$

Ejercicio 9

Demostrar por inducción que si $\boldsymbol{A}$ ***es un conjunto finito*** $\boldsymbol{|A| = n \Rightarrow |p(A)| = 2^n}$

<u>Caso base</u>

Sea $A = \varnothing \Rightarrow |A| = 0$

$p(A) = \{\varnothing\} \Rightarrow |p(A)| = 1 = 2^0$

<u>Paso inductivo</u>

Sea $A = \{a_1, a_2, ..., a_k\} \Rightarrow |A| = k$

$p(A) = \{\varnothing, \{a_1\}, ...\{a_k\}, ..., \{a_1, a_2, ..., a_k\} \Rightarrow |p(A)| = 2^k$

Sea $B = \{a_1, a_2, ..., a_k, a_{k+1}\} = A \cup \{a_{k+1}\}$

Como $B = A \cup \{a_{k+1}\} \Rightarrow p(A) \subseteq p(B)$ entonces solo faltaría determinar los subconjuntos que se forman al añadir el elemento a_{k+1}.

Si se analiza, los subconjuntos que añade este elemento extra, son la combinación con cada uno de los subconjuntos formado por el resto de elementos, es decir $x \in p(A) \Rightarrow (x \cup \{a_{k+1}\}) \subseteq p(B)$

$$\underbrace{\varnothing \cup a_{k+1} \ ... \ \{a_k\} \cup a_{k+1} \ ... \ \{a_1, a_2, ..., a_k\} \cup a_{k+1}}_{|p(A)| = 2^k \text{, entonces el elemento } a_{k+1} \text{ añade } 2^k \text{ subconjuntos}}$$

$\therefore |p(B)| = 2^k + 2^k = 2 * 2^k = 2^{k+1}$

Ejercicio 10

Mostrar que $|N \times N| = |N^+|$

1) $|N^+| \leq |N \times N| \Rightarrow f_1 : N^+ \to N \times N, f_1(n) = (n, n)$
2) $|N \times N| \leq |N^+| \Rightarrow f_2 : N \times N \to N, f_2(i, j) = \frac{(i+j)(i+j+1)}{2} + i + 1$

$\therefore |N^+| = |N \times N|$

Ejercicio 11

Mostrar que $|Q^+| = |N|$, siendo Q^+ el conjunto de los racionales positivos

1) $|N| \leq |Q^+| \Rightarrow f_1 : N \to Q^+, f_1(n) = \frac{n}{1}$
2) $|Q^+| \leq |N| \Rightarrow f_2 : Q \to N \times N, f_2(i, j) = (i, j)$ ($N \leq N \times N \leq Q$ transitividad)

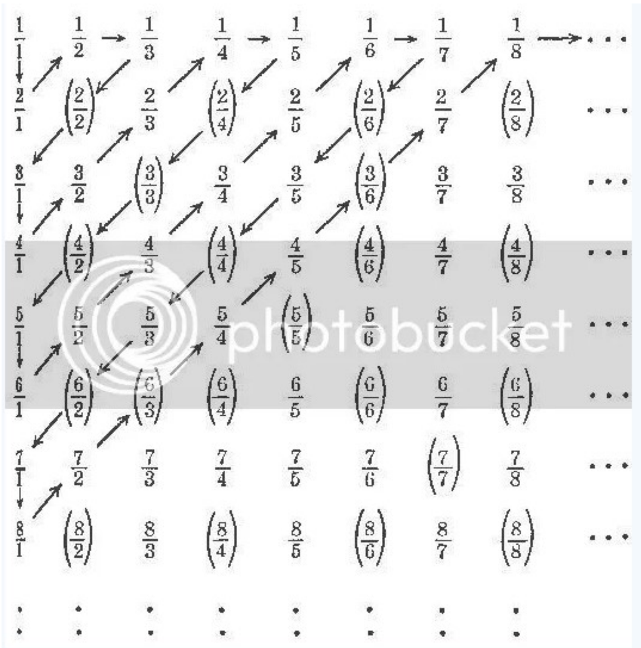

Figura 0.1

Como se puede observar en la imagen Figura 0.1, si colocamos de forma ordenada los números racionales en un arreglo bidimensional, y asignamos un número natural a partir de seguir un orden diagonal (eliminando las fracciones redundantes, que están entre paréntesis), existe una función inyectiva $f_2 : Q^+ \to N$

$\therefore |N| = |Q^+|$

Explicación más detallada

Ejercicio 13

Dar un ejemplo de 2 conjuntos disjuntos no vacíos, A y B tales que:

a) $|\boldsymbol{A}| < |\boldsymbol{B}| < |\boldsymbol{A} \cup \boldsymbol{B}|$

b) $|\boldsymbol{A}| < |\boldsymbol{B}| = |\boldsymbol{A} \cup \boldsymbol{B}|$

c) $|\boldsymbol{A}| = |\boldsymbol{B}| = |\boldsymbol{A} \cup \boldsymbol{B}|$

Resolución:

a) $|A| < |B| < |A \cup B|$; $A = \{a\}$ $B = \{b, c\}$

b) $|A| < |B| = |A \cup B|$; $A = \{a\}$ $B = \{a, b\}$

c) $|A| = |B| = |A \cup B|$; $A = \{a\}$ $B = \{a\}$

Ejercicio 14

Mostrar que $|\boldsymbol{N} - \{\boldsymbol{7, 9, 15, 34, 21, 344, 990}\}| = |\boldsymbol{N}|$

Sea $|N - \{7, 9, 15, 34, 21, 344, 990\}| = N^*$

1) $|N^*| \leq |N| \Rightarrow f_1 : N^* \to N, f_1(n) = n$

2) $|N| \leq |N^*| \Rightarrow f_2 : N \to N^*, f_2(n) = n + 991$

$\therefore |N| = |N^*|$

Práctica 2

Ejercicio 1

construir las siguientes MT:

Inciso a

Construir una máquina de Turing que haga un corrimiento a derecha de la cadena binaria en la cinta, marcando con un símbolo especial '#' la celda que corresponde al primer símbolo desplazado. $\boldsymbol{\Gamma = \{B, \#, 0, 1\}}$

MT de cómputo $< Q, \Sigma, \Gamma, \delta, q_0 >$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B, H\}$

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$

$\delta(q_0, 0) = (q_2, \#, D)$; $\delta(q_0, 1) = (q_1, \#, D)$

$\delta(q_1, 1) = (q_1, 1, D)$; $\delta(q_1, 0) = (q_2, 1, D)$; $\delta(q_1, B) = (q_3, 1, D)$

$\delta(q_2, 0) = (q_2, 0, D)$; $\delta(q_2, 1) = (q_1, 0, D)$; $\delta(q_2, B) = (q_3, 0, D)$

Inciso b

Y otra que haga un corrimiento a izquierda.

MT de cómputo $< Q, \Sigma, \Gamma, \delta, q_0 >$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B, H\}$

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$

$\delta(q_0, 0) = (q_0, 0, D)$; $\delta(q_0, 1) = (q_0, 1, D)$; $\delta(q_0, B) = (q_1, B, I)$

$\delta(q_1, 0) = (q_2, \#, I)$; $\delta(q_1, 1) = (q_3, \#, I)$

$\delta(q_2, 0) = (q_2, 0, I)$; $\delta(q_2, 1) = (q_3, 0, I)$; $\delta(q_2, B) = (q_4, 0, I)$

$\delta(q_3, 0) = (q_2, 1, I)$; $\delta(q_3, 1) = (q_3, 1, I)$; $\delta(q_3, B) = (q_4, 1, I)$

Ejercicio 2

construir las siguientes MT:

Inciso a

Construir una máquina de Turing M tal que L(M) $\boldsymbol{= 0^n 1^n / n \geq 1}$ ***y mostrar la traza de computación de M para las entradas*** $\boldsymbol{w_1 = 0011}$ ***y*** $\boldsymbol{w_2 = 011}$***.***

MT: $< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B\}$

$\delta : Q \times \Gamma \rightarrow (Q \cup \{q_A, q_R\}) \times \Gamma \times \{I, D\}$

δ	0	1	B
q_0	(q_1, B, D)	$(q_R, 1, D)$	(q_R, B, D)
q_1	$(q_1, 0, D)$	$(q_1, 1, D)$	(q_2, B, I)
q_2	$(q_R, 0, D)$	(q_3, B, I)	(q_R, B, D)
q_3	$(q_3, 0, I)$	$(q_3, 1, I)$	(q_4, B, D)
q_4	(q_1, B, D)	$(q_R, 1, D)$	(q_A, B, D)

Entrada $w_1 = 0011$:

$Bq_0 0011B \vdash_M \; BBq_1 011B \vdash_M \; BB0q_1 11B \vdash_M \; BB01q_1 1B \vdash_M$

$BB011q_1 B \vdash_M \; BB01q_2 1B \vdash_M \; BB0q_3 1BB \vdash_M \; BBq_3 01BB \vdash_M$

$Bq_3 B01BB \vdash_M \; BBq_4 01BB \vdash_M \; BBBq_1 1BB \vdash_M \; BBB1q_1 BB \vdash_M$

$BBBq_2 1BB \vdash_M \; BBq_3 BBBB \vdash_M \; BBBq_4 BBB \vdash_M \; BBBBq_A BB$

Entrada $w_2 = 011$:

$q_0 011B \vdash_M \; Bq_1 11B \vdash_M \; B1q_1 1B \vdash_M \; B11q_1 B \vdash_M$

$B1q_2 1B \vdash_M \; Bq_3 1BB \vdash_M \; q_3 B1BB \vdash_M \; Bq_4 1BB \vdash_M$

$B1q_R BB$

Inciso b

Construir una máquina de Turing que busque en la cinta el patrón abab y se detenga si y sólo si encuentra ese patrón. $\boldsymbol{\Gamma = \{a, b, c, B\}}$

MT: $< Q, \Sigma, \Gamma, \delta, q_0, q_d >$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, B\}$

$\delta : Q \times \Gamma \rightarrow (Q \cup \{q_d\}) \times \Gamma \times \{I, D\}$

Tabla 1: Alternativa 1

δ	a	b	c	B
q_0	(q_1, a, D)	(q_0, b, D)	(q_0, c, D)	(q_0, B, D)
q_1	(q_4, a, D)	(q_2, b, D)	(q_4, c, D)	(q_4, B, D)
q_2	(q_3, a, D)	(q_4, b, D)	(q_4, c, D)	(q_4, B, D)
q_3	(q_4, a, D)	(q_d, b, D)	(q_4, c, D)	(q_4, B, D)
q_4	(q_0, a, I)	(q_0, b, I)	(q_0, c, I)	(q_0, B, D)

Tabla 2: alternativa 2

δ	a	b	c	B
q_0	(q_1, a, D)	(q_0, b, D)	(q_0, c, D)	(q_0, B, D)
q_1	(q_1, a, D)	(q_2, b, D)	(q_0, c, D)	(q_0, B, D)
q_2	(q_3, a, D)	(q_0, b, D)	(q_0, c, D)	(q_0, B, D)
q_3	(q_1, a, D)	(q_d, b, D)	(q_0, c, D)	(q_0, B, D)

Ejercicio 3

Construir máquinas de Turing para computar las siguientes funciones:

Inciso a

Suma unaria. $\boldsymbol{\Sigma = \{+, 1\}}$.

MT: $< Q, \Sigma, \Gamma, \delta, q_0, q_d >$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{+, 1\}$

$\Gamma = \{+, 1B\}$

$\delta : Q \times \Gamma \to (Q \cup \{q_d\}) \times \Gamma \times \{I, D\}$

δ	1	+	B
q_0	$(q_0, 1, D)$	$(q_1, 1, D)$	(q_d, B, D)
q_1	$(q_1, 1, D)$	$(q_d, +, D)$	(q_2, B, I)
q_2	(q_d, B, I)	$(q_d, +, I)$	(q_d, B, I)

Inciso b

Resta unaria $\boldsymbol{a - b}$ con $\boldsymbol{a > b}$ $\boldsymbol{\Sigma = \{-, 1\}}$.

$MT :< Q, \Sigma, \Gamma, \delta, q_0, q_d$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Gamma = \{-, 1, B\}$

$\delta : Q \times \Gamma \to (Q \cup \{q_d\}) \times \Gamma \times \{I, D\}$

δ	1	-	B
q_0	$(q_0, 1, D)$	$(q_1, -, D)$	q_d, B, I
q_1	$(q_2, -, I)$	$(q_1, -, D)$	(q_3, B, I)
q_2	$(q_1, -, D)$	$(q_2, -, I)$	(q_2, B, D)
q_3	$(q_d, 1, I)$	(q_3, B, I)	(q_d, B, I)

Inciso c

Calcular el complemento a 2 de un número binario de 8 bits con $\boldsymbol{\Sigma = \{0, 1\}}$

Una forma de hallar el opuesto de un número binario positivo en complemento a dos es comenzar por la derecha (el dígito menos significativo), copiando el número original (de derecha a izquierda) hasta encontrar el primer 1, después de haber copiado el 1, se niegan (complementan) los dígitos restantes (es decir, copia un 0 si aparece un 1, o un 1 si aparece un 0). Este método es mucho más rápido para las personas, pues no utiliza el complemento a uno en su conversión

$MT :< Q, \Sigma, \Gamma, \delta, q_0, q_d >$

$Q = \{q_0, q_1, q_2\}$

$\Gamma = \{0, 1, B\}$

$\delta : Q \times \Gamma \rightarrow (Q \cup \{q_d\}) \times \Gamma \times \{I, D\}$

δ	0	1	B
q_0	$(q_0, 0, D)$	$(q_0, 1, D)$	(q_1, B, I)
q_1	$(q_1, 0, I)$	$(q_2, 1, I)$	(qd, B, I)
q_2	$(q_2, 1, I)$	$(q_2, 0, I)$	(qd, B, D)

Ejercicio 4

Sea $\Sigma = \{a\}$ y $w = a$. Decir cuáles son las palabras que se obtienen como resultado de aplicar las siguientes operaciones: ww, www, w^3, w^5, w^0 ¿Cuáles son sus longitudes? Definir Σ^*

$ww = aa;\ |ww| = 2$

$www = aaa;\ |www| = 3$

$w^3 = aaa;\ |www| = 3$

$w^5 = aaaaa;\ |w^5| = 5$

$w^0 = \lambda;\ |w^0| = 0$

$\Sigma^* = \{\lambda, a, aa, aaa, ...\} = \{a^n, n \geq 0\}$

Ejercicio 5

Idem al ejercicio anterior, pero con $\Sigma = \{a, b\}$ y $w = aba$.

$ww = abaaba;\ |ww| = 6$

$www = abaabaaba;\ |www| = 9$

$w^3 = abaabaaba;\ |w^3| = 9$

$w^5 = abaabaabaabaaba;\ |w^5| = 15$

$w^0 = \lambda;\ |w^0| = 0$

$\Sigma^* = \{a^n b^m, n \geq 0 \wedge m \geq 0\}$

Ejercicio 6

Sea $\Sigma = \{a, b, c\}$, escriba las 13 cadenas más cortas de Σ^*.

1. $|\lambda| = 0$
2. $|a| = 1$
3. $|b| = 1$
4. $|aa| = 2$
5. $|ab| = 2$

6. $|ba| = 2$
7. $|bb| = 2$
8. $|aaa| = 3$
9. $|aba| = 3$
10. $|abb| = 3$
11. $|aab| = 3$
12. $|bbb| = 3$
13. $|bba| = 3$

Ejercicio 7

Dar tres ejemplos de lenguajes basados en el alfabeto $\{0, 1\}$

1. $\{w \in \Sigma^* : 1^n, n \geq 0\}$
2. $\{w \in \Sigma^* : 0^n \, n \geq 0\}$
3. $\varnothing$

Ejercicio 8

¿Cuántas cadenas de longitud 3 hay en $\{0, 1, 2\}*$, y cuántas de longitud n?

Dado que las distintas cadenas se forman generando variaciones de los distintos símbolos **con repetición**, entonces si se dispone de n símbolos, la cantidad de cadenas con longitud m es: n^m

$\therefore$

cantidad de cadenas de longitud 3 $= 3^3 = 27$

cantidad e cadenas de longitud n $= 3^m$

Ejercicio 9

Explicar la diferencia -si la hay- entre los lenguajes L1 y L2.

a) $L_1 = \varnothing \quad L_2 = \{\lambda\}$

   L_1 no contiene ningún elemento mientras que L_2 contiene a λ

b) $L_1 = \Sigma^* \cup \{\lambda\} \quad L_2 = \varnothing \cup \Sigma^*$

   Son iguales:

   $\lambda \subseteq \Sigma^* \rightarrow \Sigma^* \cup \{\lambda\} = \Sigma^*$

   $\varnothing$ es neutro en la unión $\rightarrow \varnothing \cup \Sigma^* = \Sigma^*$

c) $L_1 = \Sigma^* - \varnothing \quad L_2 = \Sigma^*$

   Son iguales:

   $\varnothing$ es neutro en la diferencia $\rightarrow \Sigma^* - \varnothing = \Sigma^*$

d) $L_1 = \Sigma^* - \{\lambda\} \quad L_2 = \Sigma^*$ Son diferentes ya que $\Sigma^* - \lambda = \Sigma \therefore L_1 \neq L_2$

Ejercicio 10

Mostrar que Σ^* es infinito contable.

Simplemente Magia...

La idea general para encontrar una función inyectiva tal que $|\Sigma^*| \leq |\mathbb{N}|$ se defina un sistema posicional a partir de todos los símbolos de Σ^*. De modo que la función pueda expresarse a través del calculo del teorema fundamental de la numeración.

Sea $|\Sigma| = n$, y $\Sigma = \{s_1, ..., s_n\}$ entonces se definirá una función que mapea cada símbolo de Σ a un número.

Se define función $g(x)$:

$$
\begin{aligned}
g(s_1) = &\quad 0 \\
g(s_2) = &\quad 1 \\
... & \\
g(s_n) = &\quad \text{n}
\end{aligned}
$$

Sea $c(w, n)$ (denominada función caracter), una función que mapea el símbolo n-esimo de la palabra 'w', donde $n > 0$. Es decir si $w = argentina$, $c(w, 3) = g$. Entonces, se puede definir la siguiente función inyectiva para $\Sigma^* \to \mathbb{N}$:

$f(x) = \sum_{i=1}^{|x|} g(c(x,i)) * |\Sigma^*|^{i-1}$

$\therefore |\Sigma^*| \leq |\mathbb{N}|$

Ejercicio 11

Indicar cuál es el lenguaje que se obtiene al intersectar los siguientes lenguajes:

1. $L_1 = \{a^n c^m d^n / n \geq 0, m \geq 0\}$ con $L_2 = \{c^n / n \geq 0\}$

   $L_1 \cap L_2 = \{c^n, n \geq 0\}$
2. $L_1 = \{a^n c^m d^n / n > 0, m \geq 0\}$ con $L_2 = \{c^n / n \geq 0\}$

   $L_1 \cap L_2 = \{\lambda\}$
3. $L_1 = \{a^n c^m d^n / n \geq 0, m > 10\}$ con $L_2 = \{c^n / n > 5\}$

   $L_1 \cap L_2 = \{c^n, n > 10\}$
4. $L_1 = \{1^n 2^m / n, m \geq 0, \text{ n par, m impar}\}$ con $L_2 = \{2^n / n \geq 0\}$

   $L_1 \cap L_2 = \{2^n, n \geq 0 \wedge \text{ n impar}\}$
5. $L_1 = \{1^n 2^m / n, m \geq 0, \text{n par, m impar}\}$ con $L_2 = \{1^n / n \geq 0\}$

   $L_1 \cap L_2 = \varnothing$

Ejercicio 12

Encontrar si es posible un lenguaje L1 que cumpla:

a) $L_1 \cap \{1^k 2^m 3^n / m = k + n + 1 \ \wedge n, k \geq 0\} = \{1^n 2^{n+1} / n \geq 0\}$

   $L_1 = \{1^n 2^{n+1} / n \geq 0\}$

b) $L_1 \cap \{1^n 2^m / n \neq m \wedge n, m \geq 0\} = \{1^n 2^n / n > 0\}$

   No es posible encontrar un L_1 que cumpla esa igualdad

Ejercicio 13

Conteste las siguientes preguntas sobre Máquinas de Turing

Inciso a

¿Puede el alfabeto de la cinta ($\boldsymbol{\Gamma}$) ser el mismo que el alfabeto de entrada ($\boldsymbol{\Sigma}$)

No, no pueden ser el mismo. El alfabeto de entrada representa la cadena de símbolos con la que se forma la cadena de entrada, y esta no puede tener símbolos B en medio de la cadena ya que este representa la ausencia de símbolos. Por lo tanto: $\Sigma \subseteq \Gamma \wedge B \in (\Gamma - \Sigma)$

Inciso b

¿Puede una máquina de Turing tener un único estado?

Sí, es posible. Suponga la siguiente maquina de turing que limpia los símbolos de un alfabeto de la cinta:

$MT :< Q, \Sigma, \Gamma, \delta, q_0 >$

$Q = \{q_0\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B\}$

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{I, D\}$

$\delta(q_0, 0) = (q_2, B, D)$

$\delta(q_0, 1) = (q_2, B, D)$

Inciso c

¿Cuántos lenguajes existen definidos sobre el alfabeto $\boldsymbol{\Sigma = \{0, 1\}}$? ¿y sobre $\boldsymbol{\Sigma = \{1\}}$?

Infinitos lenguajes. A partir de cada alfabeto pueden formarse infinitas combinaciones de cadenas, en consecuencia sobre esa infinitas cadenas pueden definirse infinitos lenguajes

Inciso d

¿Cuáles de los siguientes conjuntos son lenguajes definidos sobre $\boldsymbol{\Sigma}$? $\varnothing, \boldsymbol{\Sigma}, \boldsymbol{\Sigma^*}, \{\boldsymbol{\lambda}\}, \{\boldsymbol{\lambda}\} \cup \boldsymbol{\Sigma}, \{\varnothing\}$

1) $\varnothing \in \Sigma^*$ (El vacío está contenido en todo conjunto)
2) $\Sigma \in \Sigma^*$ (Σ^* se forma a partir de los elementos Σ)
3) $\Sigma^* \in \Sigma^*$ (Identidad)
4) $\{\lambda\} \in \Sigma^*$ (La cadena vacía forma parte de las palabras que pueden formarse a partir del alfabeto)
5) $(\{\lambda\} \cup \Sigma) \in \Sigma^*$ (Por 2 y 4)
6) $\{\varnothing \notin \Sigma^*\}$

Ejercicio 13

Sea la siguiente máquina de Turing:

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, B\}$

$\delta(q, s) = (q', s', m) tq$

$q \in Q \quad q' \in Q \cup \{q_R\} \quad s, s' \in \Gamma \quad m \in \{D, I\}$

¿Reconoce el lenguaje $\{\lambda\}$? Si no es así indique cuál es el lenguaje que reconoce

No, no reconoce el lenguaje $\{\lambda\}$, dado que todas las transiciones definidas o llevan a un $q \in Q$ o sino a q_R, de esta manera no se define ningún estado que lleve a q_A, por lo cual el lenguaje reconocido es $\varnothing$.

Ejercicio 14

Sea $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R \rangle$, donde en cada caso se asume que las transiciones $\delta(\cdot)$ no especificadas llevan a la detención en q_R. Determinar $L(M)$.

a) $Q = \{q_0, q_1\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B\}$

$\delta(q_0, 0) = (q_0, 0, I)$
$\delta(q_0, B) = (q_0, B, D)$
$\delta(q_0, 1) = (q_1, 1, D)$

b) $Q = \{q_0, q_1\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B\}$

$\delta(q_0, 0) = (q_1, B, D)$
$\delta(q_1, B) = (q_A, B, D)$
$\delta(q_1, 0) = (q_A, 0, D)$
$\delta(q_1, 1) = (q_A, 1, D)$

c) $Q = \{q_0, q_1\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B\}$

$\delta(q_0, 0) = (q_0, 0, I)$
$\delta(q_0, B) = (q_0, B, D)$
$\delta(q_0, 1) = (q_1, 1, D)$
$\delta(q_1, 0) = (q_0, B, I)$
$\delta(q_1, B) = (q_0, B, D)$

d) $Q = \{q_0\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B\}$

$\delta(q_0, 1) = (q_0, B, I)$
$\delta(q_0, 0) = (q_A, B, I)$
$\delta(q_0, B) = (q_0, B, D)$

e) $Q = \{q_0, q_1\}$; $\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B\}$

$\delta(q_0, 0) = (q_1, B, D)$
$\delta(q_1, 0) = (q_1, 1, D)$
$\delta(q_1, 1) = (q_1, 0, D)$
$\delta(q_1, B) = (q_A, 1, D)$

Algún día lo realizaré...

Práctica 3

Ejercicio 1

Construir máquinas de Turing que acepten los siguientes lenguajes

a) $L1 = \Sigma^*$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0\}$

$\Sigma = \{s_1, s_2, ..., s_n\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times \Gamma \to Q \cup \{q_A, q_R\} \times \Gamma \times \{I, D, S\}$

$\delta(q_0, s') = (q_A, s_i, S) \quad (\forall s')(s' \in \Gamma)$

b) $L2 = \{\lambda\}$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0\}$

$\Sigma = \{s_1, s_2, ..., s_n\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times \Gamma \to Q \cup \{q_A, q_R\} \times \Gamma \times \{I, D, S\}$

$\delta(q_0, B) = (q_A, B, S)$

$\delta(q_0, s') = (q_R, s', S) \quad (\forall s')(s' \in \Gamma)$

c) $L3 = \varnothing$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0\}$

$\Sigma = \{s_1, s_2, ..., s_n\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times \Gamma \to Q \cup \{q_A, q_R\} \times \Gamma \times \{I, D, S\}$

$\delta(q_0, s') = (q_R, s', S) \quad (\forall s')(s' \in \Gamma)$

d) $L4 = \{0^n 1^{2n} / n \geq 0\}$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >; \quad k = 3$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times (\Gamma)^k =\to Q \cup \{q_A, q_R\} \times (\Gamma \times \{I, D, S\})^k$

$\delta(q_0, (B, B, B)) = (q_A, (B, B, B), (S, S, S))$

$\delta(q_0, (0, B, B)) = (q_1, (B, B, B), (S, S, S))$

$\delta(q_1, (0, B, B)) = (q_1, (0, 0, 0), (D, D, D))$

$\delta(q_1, (1, B, B)) = (q_2, (1, B, B), (S, I, I))$

$\delta(q_2, (1, 0, 0)) = (q_A, (1, B, 0), (D, I, S))$

$\delta(q_2, (1, B, 0)) = (q_3, (1, B, 0), (S, S, S))$

$\delta(q_3, (1, B, 0)) = (q_3, (1, B, B), (D, S, I))$

$\delta(q_3, (B, B, B)) = (q_A, (B, B, B), (S, S, S))$

$\delta(q', s') = (q_R, s', (S, S, S))$ $(\forall(q', s'))((q', s') :$ pertenece al conjunto de las transiciones faltantes)

En lenguaje coloquial, las transiciones faltantes van todas a q_R

e) $L5 = \{a^n b^n c^n / n \geq 0\}$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >; \quad k = 3$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times (\Gamma)^k =\rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{I, D, S\})^k$

$\delta(q_0, (B, B, B)) = (q_A, (B, B, B), (S, S, S))$

$\delta(q_0, (a, B, B)) = (q_1, (a, B, B), (S, S, S))$

$\delta(q_1, (a, B, B)) = (q_1, (a, b, c), (>, >, >))$

$\delta(q_1, (b, B, B)) = (q_2, (b, B, B), (S, I, I))$

$\delta(q_2, (b, b, c)) = (q_2, (b, B, c), (>, <, -))$

$\delta(q_2, (c, B, c)) = (q_3, (c, B, c), (S, S, S))$

$\delta(q_3, (c, B, c)) = (q_3, (c, B, B), (D, S, I))$

$\delta(q_3, (B, B, B)) = (q_A, (B, B, B), (S, S, S))$

$\delta(q', s') = (q_R, s', (S, S, S))$ $(\forall(q', s'))((q', s') :$ pertenece al conjunto de las transiciones faltantes)

f) $L6 = \{a^n b^m c^k / k = n + m, n, m \geq 1\}$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >; \quad k = 2$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times (\Gamma)^k =\rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{I, D, S\})^k$

$\delta(q_0, a, B) = (q_0, (a, c), (D, D))$

$\delta(q_0, b, B) = (q_1, (b, c), (D, D))$

$\delta(q_1, b, B) = (q_1, (b, c), (D, D))$

$\delta(q_1, c, B) = (q_2, (c, B), (S, I))$

$\delta(q_2, c, c) = (q_2, (c, B), (D, I))$

$\delta(q_2, B, B) = (q_A, (B, B), (S, S))$

$\delta(q', s') = (q_R, s', (S, S, S))$ $(\forall(q', s'))((q', s') :$ pertenece al conjunto de las transiciones faltantes)

g) $L7 = \{ww^r/w \in \{0,1\}*\}$, donde w^res el reverso de w

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >; \quad k = 2$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times (\Gamma)^k =\rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{I, D, S\})^k$

$\delta(q0,(0,B)) = (q1,(B,0),(D,S))$

$\delta(q0,(1,B)) = (q1,(B,1),(D,S))$

$\delta(q0,(B,B)) = (qA,(B,B),(S,S))$

$\delta(q1,(0,0)) = (q1,(0,0),(D,S))$

$\delta(q1,(0,1)) = (q1,(0,1),(D,S))$

$\delta(q1,(1,1)) = (q1,(1,1),(D,S))$

$\delta(q1,(1,0)) = (q1,(1,0),(D,S))$

$\delta(q1,(B,0)) = (q2,(B,0),(I,S))$

$\delta(q1,(B,1)) = (q2,(B,1),(I,S))$

$\delta(q2,(0,0)) = (q3,(0,B),(I,S))$

$\delta(q2,(1,1)) = (q3,(1,B),(I,S))$

$\delta(q3,(0,B)) = (q3,(0,B),(I,S))$

$\delta(q3,(1,B)) = (q3,(1,B),(I,S))$

$\delta(q3,(B,B)) = (q0,(B,B),(D,S))$

$\delta(q', s') = (q_R, s', (S, S, S)) \quad (\forall(q', s'))((q', s') :$ pertenece al conjunto de las transiciones faltantes)

h) $L8 = L7 \cup \{w^0 w^R/w \in \{0,1\}*\} \cup \{w^1 w^R/w \in \{0,1\}*\}$

Idem al inciso g); ya que $L7 \cup \{w^0 w^R/w \in \{0,1\}*\} \cup \{w^1 w^R/w \in \{0,1\}*\} = L7$ porque expresan los mismos patrones, ya que $w^0 = \lambda$ y $w^1 = w$, y dentro de $\{0,1\}*$ una de las posibles cadenas a formar es la cadena vacía, es decir λ

Ejercicio 2

Construya una Máquina de Turing de 2 cintas que implemente un contador binario en la segunda cinta para contabilizar la cantidad de letras 'a' que aparecen en el input de la primera cinta. Con $\boldsymbol{\Sigma = \{a, b\}}$; $\boldsymbol{\Gamma = \{a, b, 0, 1, B\}}$

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_d >; \quad k = 2$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, 0, 1, B\}$

$\delta = Q \times (\Gamma)^k =\rightarrow Q \cup \{q_A, q_R\} \times (\Gamma \times \{I, D, S\})^k$

$\delta(q0,(a,B)) = (q0,(a,1),(D,S))$

$\delta(q0,(a,0)) = (q0,(a,1),(D,S))$

$\delta(q0,(a,1)) = (q1,(a,1),(S,S))$

$\delta(q0,(b,B)) = (q0,(b,B),(D,S))$

$\delta(q0,(b,0)) = (q0,(b,0),(D,S))$

$\delta(q0,(b,1)) = (q0,(b,1),(D,S))$

$\delta(q0,(B,B)) = (qd,(B,0),(S,S))$

$\delta(q0,(B,0)) = (qd,(B,0),(S,S))$

$\delta(q0,(B,1)) = (qd,(B,1),(S,S))$

$\delta(q1,(a,B)) = (q2,(a,1),(S,S))$

$\delta(q1,(a,0)) = (q2,(a,1),(S,S))$

$\delta(q1,(a,1)) = (q1,(a,0),(S,D))$

$\delta(q2,(a,0)) = (q2,(a,0),(S,I))$

$\delta(q2,(a,1)) = (q2,(a,1),(S,I))$

$\delta(q2,(a,B)) = (q0,(a,B),(D,D))$

$\delta(q',s') = (q_d,s',(S,S,S)) \quad (\forall(q',s'))((q',s'):$ pertenece al conjunto de las transiciones faltantes$)$

El resto de ejercicios aún deben pasarse en limpio...

Práctica 4

Ejercicio 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.

La ejecución de esta MT puede probarse en https://turingmachinesimulator.com: pegue el siguiente código fuente y **compile** (el input de la cinta debe ser vacío o blanco, se podría separar por ; pero es complejizar en vano, esta máquina simplemente genera las cadenas sobre-escribiendolas):

```
name: L = 0,1*
init: qI
accept: qA

qI,_
qR,0,-

qR,0
qR,0,>

qR,1
qR,1,>

qR,_
qW,_,<

qW,0
qR,1,-

qW,1
qW,0,<

qW,_
qR,1,-
```

Aclaraciones:

- La sintaxis de una transición es:
  - 'estado leido', 'símbolo leido'
    'siguiente estado', 'símbolo a escribir', 'movimiento de la cinta'
  - Ejemplos:
    qI,_
    qD,0,-

    Significa $\delta(q_I, B) = (q_D, 0, Static)$
- los movimientos de la cinta pueden ser '<' (izquierda), '>' (derecha), '-' (permanecer sin movimiento). El símbolo '_' (gruión bajo) representa celda vacía o blanco.
- Estados:
  - qI (qInitial): es el estado inicial quien se encarga de setear la cinta en 0
  - qR (qRight): estado que arrastra la cinta al final del input hasta encontrar Blanco

- qW (qWrite): estado que se encarga de actualizar el contador haciendo el carry.

Ejercicio 2

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

- $\mathcal{L} - R = \varnothing$ (Falso)

  Existen lenguajes como L_U, L_D, etc, que no pertenecen a R. Entonces no es posible que aquella resta de conjuntos de $\varnothing$

- $RE - R \neq \varnothing$ (Verdadero)

  Existen lenguajes como como L_D que pertenecen a RE y no a R. Entonces no es posible que aquella resta de conjuntos de $\varnothing$

- $\{\lambda\} \in (\mathcal{L} - CO\text{-}RE)$ (Verdadero)

  Es posible la definición del lenguaje L = $\{\lambda\}$ a través de una MT M que solamente acepte la cadena vacía, y rechase en caso contrario, en consecuencia $L \in R$. Y como es sabido $R = Co$R $\Rightarrow$ $L \in (\mathcal{L} - Co\text{-}RE)$

- $RE \cup R = \mathcal{L}$ (Falso)

  Existen lenguajes como L_D que no pertenecen a RE ni a R. Por lo cual esta afirmación no es posible.

- $\Sigma* \in R$ (Verdadero)

  La definición de una MT M tal que $L(M) = \Sigma^*$ es muy simple, solamente se necesita de un estado inicial q_0 la cual para cualquier símbolo pasa a q_A. De esta manera para todo input, M se estaría deteniendo, y en consecuencia $\Sigma^* \in R$

- $\varnothing \in RE$ (Verdadero)

  Es posible la definición de una MT M tal que $L(M) = \varnothing$, esta consitiría simplemente en rechazar cualquier entrada de Σ^*

- $CO\text{-}RE = RE$ (Falso)

  Existen lenguajes como L_D que no pertencen a RE, sin embargo su complemento $\overline{L_D}$ si pertenece a RE. Por lo cual la afirmación no es posible.

- $(\mathcal{L} - RE) = CO\text{-}RE$ (Falso)

  Existen lenguajes como $L = \{1w/w \in L_D\} \cup \{0w/w \notin L_D\}$ que no pertenecen a RE ni Co-RE

- $ab \in \Sigma*$ (Verdadero)

  $\Sigma*$ representa todas las cadenas formables a partir de Σ y $a, b \in \Sigma$

- $CO\text{-}R \subset CO\text{-}RE$ (Verdadero)

  Por definición un lenguaje que es R, necesariamente debe ser RE. Por lo cual la afirmación es verdadera.

- $a \in R$ (Falso)

  a es un elemento, mientras que R es un conjunto de conjuntos. Por lo cual la afirmación no es posible

- $\{a\} \in RE$ (Verdadero)

  Es posible la definición de una MT M tal que acepte 'a' y se detenga en caso contrario.

Ejercicio 3

Si $L \in (RE - R)$

a) ¿Existirá alguna máquina de Turing que rechace parando en qR si su entrada está en L y rechace loopeando si su entrada no está en L?

Sí, de acuerdo a las suposiciones existe una MT M tal que L(M) = L. Entonces de esta manera se podría construir una MT M' que simula M sobre cualquier entrada w. Entonces M' responde de la siguiente manera ante w:

- Si M acepta w $\Rightarrow$ M' rechaza w.
- Si M rechaza w $\Rightarrow$ M' loopea.
- Si M loopea sobre w $\Rightarrow$ M' loopea de igual manera.

b) ¿Existirá alguna máquina de Turing que rechace loopeando si su entrada está en L y rechace parando en q_R si su entrada no está en L?

No, no podría existir. Ya que la suposición $L \in (RE - R)$ implica que L al no pertenecer a R supone que no hay una MT M que se detenga para cualquier entrada de Σ^*

c) De existir, que lenguaje reconocería esta máquina de Turing.

Esta máquina reconocería al conjunto vacío. Es decir que L(M) = $\varnothing$, ya que rechazaría loopeando si su entrada está en L, y cuando no está en L también rechazaría parando en qR, en síntesis, rechaza todo.

Ejercicio 4

Sea* $\mathbf{L} = \{\mathbf{w}|$ *Existe alguna Máquina de Turing M que acepta w*$\}$ ¿$\mathbf{L} \in \mathbf{R}$*? Justifique.

Para cualquier w_i es posible definir una MT M_i que acepte w_i, simplemente definiendo una secuencia de transciones para cada cada simbolo s_j perteneciente a w_i, y rechazando en q_R en caso contrario. Entonces el lenguaje $L = \Sigma^*$ y $L \in R$ ya que la máquina que reconoce Σ^* es una MT que acepta todo en un solo paso.

Ejercicio 5

Conteste y justifique:

a) ¿$\mathcal{L}$ es un conjunto infinito contable?

Ya se ha demostrado los siguientes puntos:

- $|\Sigma^*| = |N|$
- $|A| < |p(A)|$

$\therefore \mathcal{L}$ no es infinito contable

b) ¿RE es un conjunto infinito contable?

Sí, RE es un ocnjunto infinito contable. Por definición para cada $L \in RE$, existe una MT M tal que $L(M) = L$, por lo tanto cada M podría codificarse en binario permitiendo mapear cada L de acuerdo a un número natural igual al valor decimal de la codificación binario de la MT M que la acepta.

c) ¿$\mathcal{L} - RE$ es un conjunto infinito contable?

De forma general se puede intuir que si a un conjunto no numerable, se le extrae un subconjunto que si es numerable, aún permanecerán los elementos que hacen que el conjunto siga siendo no numerale.

dem. Suponiendo que $\mathcal{L} - RE$ es numerable, se puede llegar a una contradicción:

$(\mathcal{L} - RE) \cup RE$ (es numerable, por unión de conjuntos numerables)

$\Rightarrow \mathcal{L}$ (es numerable)

Lo cual lleva a una contradicción, ya que está demostrado que $\mathcal{L}$ no es numerable.

$\therefore \mathcal{L} - RE$ es un conjunto no numerable

d) Existe algún lenguaje $L \in \mathcal{L}$, tal que L sea infinito no contable

Desde un punto de vista informal, esto no puede ser posible, ya que se ha demostrado que Σ^* es contable, en consecuencia todo subconjunto de este también será contable.

dem. Suponiendo que $(\exists L)(L : L \in \mathcal{L} \wedge L$ es incontable) entonces se puede llegar al siguiente contradicción:

$(L \cup L^C = \Sigma^*)$ es incontable (por unión de conjunto incontable con cualquier otro)

Lo cual lleva a una contradicción porque Σ^* es contable

$\therefore (\nexists L)(L : L \in \mathcal{L} \wedge L$ es incontable)

Ejercicio 6

Sea L un lenguaje definido sobre $\boldsymbol{\Sigma}$. Demostrar que:

a) $\overline{\mathbf{L}} \notin \mathbf{R} \Rightarrow \mathbf{L} \notin \mathbf{R}$

dem. Suponiendo que si $\overline{L} \notin R \Rightarrow L \in R$ se generan contradicciones:

$L \in R$

$\Rightarrow \overline{L} \in Co\text{-}R$ (Teorema Co-R = R)

Lo cual lleva a una contradicción porque se asumió que $\overline{L} \notin R$

b) $(\mathbf{L_1} \in \mathbf{RE})\mathbf{AND}(\mathbf{L_2} \in \mathbf{RE}) \Rightarrow \mathbf{L_1} \cap \mathbf{L_2} \in \mathbf{RE}$

dem. Dado que tanto L_1 como L_2 pertenecen a RE, implica que existen una MT M_1 que reconoce L_1 y una M_2 que reconoce L_2

Entonces es posible construir una MT M de la siguiente manera:

1) Dada un entrada w, M simula M_1 sobre w. Entonces:

- Si M_1 rechaza w $\Rightarrow$ M rechaza w
- Si M_1 loopea sobre w $\Rightarrow$ M loopea sobre w
- Si M_1 acepta w $\Rightarrow$ M pasa al paso (2)

2) M simula M_2 sobre w. Entonces:

- Si M_2 rechaza w $\Rightarrow$ M rechaza w
- Si M_2 loopea sobre w $\Rightarrow$ M loopea sobre w
- Si M_2 acepta w $\Rightarrow$ M acepta w

Así de esta manera se tiene una MT M que acepta $L_1 \cap L_2$, ya que emulando ambas máquinas M solo reconocerá entradas que son reconocidas tanto por M_1 como por M_2.

c) $(\mathbf{L_1} \in \mathbf{RE})\mathbf{AND}(\mathbf{L_2} \in \mathbf{RE}) \Rightarrow \mathbf{L_1} \cup \mathbf{L_2} \in \mathbf{RE}$

dem. Dado que L_1 y L_2 pertenecen a RE, existen las correspondientes MT M_1 y M_2 que reconocen a cada una.

Entonces es posible construir una MT M que reconozca su unión, ejecutando paralelamente M_1 y M_2. Entonces de esta manera si la entrada w se encuentra dentro de $L_1 \cup L_2$ M podrá reconocerla, caso contrario o ambas loopearan o se necesitará de que ambas paren en q_R para que M rechace w o loopea indefinidamente.

d) La unión de un número finito de lenguajes recursivamente enumerables es un lenguaje recursivamente enumerable.

dem. Aplicar la misma estrategia que (c) solo que aplicado a N lenguajes. Es decir que se simulan paralelamente los N lenguajes L_n tal que o alguno acepta la entrada, o todos la rechazan o loopean infinitamente.

Ejercicio 7

Para los casos a), b) y c) del punto anterior ¿valen las recíprocas? Justifique.

a) $L \notin R \Rightarrow \overline{L} \notin R$

dem. Suponiendo que $L \notin R \Rightarrow \overline{L} \in R$ se puede llegar a una contradicción:

$L \notin R \Rightarrow \overline{L} \in R$

$\Rightarrow \overline{\overline{L}} \in R$ (porque R = Co-R)

$\Rightarrow L \in R$ (doble complemento)

Lo cual implica una contradicción

$\therefore L \notin R \Rightarrow \overline{L} \notin R$, entonces la recíproca es válida

b) $\mathbf{L_1 \cap L_2 \in RE \Rightarrow (L_1 \in RE) AND (L_2 \in RE)}$

dem. Esto no es cierto, puede darse el siguiente contraejemplo:

$\overline{L_U} \notin RE$

$\overline{L_U} \cap_U = \varnothing$

$\varnothing \in RE$

$\therefore$ la recíproca es válida

c) $\mathbf{L_1 \cup L_2 \in RE \Rightarrow (L_1 \in RE) AND (L_2 \in RE)}$

dem. Esto no es cierto, puede darse el siguiente contraejemplo:

$\overline{L_U} \notin Re$

$\overline{L_U} \cup L_U = \Sigma^*$

$\Sigma^* \in RE$

$\therefore$ la recíproca es inválida

Ejercicio 8

Si L es un subconjunto de un lenguaje recursivamente enumerable, ¿Puede afirmarse entonces que L es recursivamente enumerable? Justifique.

No puede afirmarse, ya que existen lenguajes como $L_D, \overline{L_U}$ que no pertenecen a RE. Sin embargo son todos subconjuntos de Σ^* que representa un conjunto enumarable.

Ejercicio 9

9) Dado L1, un lenguaje recursivo cualquiera

$\boldsymbol{L_2 = \{<M>|L(M) = L_1\}}$

$\boldsymbol{L_3 = \{<M>|L(M) = L_1 \text{ y } M \text{ siempre se detiene}\}}$

Determine si $\boldsymbol{(L_2 - L_3) = \varnothing}$***.Justifique su respuesta***

Dado que L_1 es R entonces existe una MT M tal que siempre se detiene para cualquier entrada.

Teniendo esto en cuenta, $M \in L_2 \wedge M \in L_3$, sin embargo se puede construir una MT M' tal que simula M, pero cuando M rechaza una entrada M' loopea indefinidamente.

$\therefore M' \in L_2 \wedge M' \notin L_3 \Rightarrow L_2 \nsubseteq L_3 \Rightarrow (L_2 - L_3) \neq \varnothing$

Ejercicio 10

Sean los lenguajes $\boldsymbol{L = \{<M>|M \text{ siempre se detiene}\}}$ ***y*** $\boldsymbol{L_R = \{<M>|L(M) \in R\}}$***. Cuál es la afirmación correcta: VERIFICAR***

a) $\boldsymbol{L \subset L_R}$

b) $\boldsymbol{L \supset L_R}$

c) $\boldsymbol{L = L_R}$

La respuesta correcta es (a), esto porque L_R reconoce MTs cuyo lenguaje pertenece a R, lo cual implica que puede contener más de una MT para determina lenguaje, que serían la MT M que siempre se detiene, y otras MTs que si bien no son R podrían ser RE y pertenecen a L_R ya que por la máquina M se considera que el lenguaje que representan es R. En cambio L solo incluye las MTs que siempre se detienen M.

Entonces Supongamos una MT <M>que siempre se detiene, esta pertenecerá tanto a L como L_R. Sin embargo, se puede construir una MT <M'>tal que que actúa igual que <M>a diferencia que en vez de pasar a q_R loopea indefinidamente.

Entonces sucede lo siguiente:

- <M'>$\notin L$
- <M'>$\in L_R$, porque <M'>reconoce el mismo lenguaje que <M>y justamente este lenguaje pertenece a R porque existe la MT <M>que se detiene para toda entrada.

$\therefore L \subset L_R$

Ejercicio 11

11) Encuentre una justificación para cada una de las siguientes afirmaciones

a) $\varnothing \in$ **RE**

Sí, la construcción de la respectiva MT M, consistiría en un estado q_0 que rechaza toda entrada posible. De esta manera $M \in R \Rightarrow M \in RE$

b) Si L es un lenguaje formado por una sola palabra, entonces L $\in$ **R** VERIFICAR

Sí, es posible la construcción de una MT M tal que $L(M) = L$ y siempre se detiene.

Supongamos que L solo acepta una única palabra w, así podemos decir que w es la concatenación de una serie de símbolos s_i, es decir $w = s_0.s_1.s_2. \; ... \; .s_n$. Entonces se puede construir una MT M donde cada estado q_i necesita del símbolo s_i para pasar a la siguiente transición hasta llegar a s_n que transiciona a q_A, y donde cualquier símbolo s_j tal que $j \neq i$, implica una transición hacia q_R en el estado q_i

$M =< Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R >$

$Q = \{q_0, q_1, ..., q_n\, q_A, q_R\}$

$\Sigma = \{s_0, s_1, ..., s_n\}$

$\Gamma = \Sigma \cup \{B\}$

$\delta = Q \times \Sigma \rightarrow Q \times \Sigma \times \{D, I, S\}$

$(q_i, s_i) = (q_{i+1}, s_i, D)$ $(\forall i)(i : i >= 0 \wedge i < n)$

$(q_n, s_n) = (q_A, s_n, D)$

$(q_i, s_j) = (q_R, s_j, D)$ $(\forall i)(i : 0 >= i <= n \wedge i \neq j)$

$\therefore L \in R$

c) Si L es un lenguaje finito, entonces L $\in$ R VERIFICAR

Se puede demostrar que la afirmación es verdadera, a través de los siguientes razonamientos:

Dado que L es un lenguaje finito, este se compondrá de una cantidad n finita de palabras. Entonces si tomamos estas palabras en orden canónico, por (b) para cada w_i existe una MT M_i tal que $L(M_i) = \{w_i\}$ y M_i siempre se detiene. Entonces se podría crear una MT M_F de la siguiente manera:

1) Para cada w M_F simula cada M_i en orden canónico. Entonces puede suceder lo siguiente:

- Si algún M_i acepta w $\Rightarrow$ M_F acepta w
- Si todos los M_i rechazan w $\Rightarrow$ M_F rechaza w

Ejercicio 12

Demuestre que si el Halting Problem (HP) es un lenguaje recursivo entonces podría construirse una máquina de Turing que acepte el lenguaje universal L_U, y que se detenga para todo $w \in \Sigma^*$ ¿Qué puede decir entonces sobre la recursividad de HP? VERIFICAR

$L_U = \{(<M>, w)/M \textit{ acepta } w\}$

$HP = \{(<M>, w)/M \textit{ se detiene con input } w\}$

Si $HP \in R$, entonces existe una MT M_{HP} que siempre se detiene.

Entonces se puede construir una MT M_U tal que $L(M_U) = L_U$ y siempre se detiene. Esta consistiría en:

1) Se analizar el par (<M>, w), si es inválido se rechaza

2) Se emula HP sobre el par (<M>, w). Entonces:

- Si HP rechaza (<M>, w) $\Rightarrow$ que (<M>, w) loopea indefinidamente, entonces M_U rechaza (<M>, w)
- Caso contrario se emula M sobre w

3) Cuando se emula M sobre w:

- Si M acepta w $\Rightarrow$ M_U acepta w
- Si M rechaza w $\Rightarrow$ M_U rechaza w

De esta manera M_U se detiene para toda entrada, y $L_U \in R$, pero está demostrado que $L_U \in (RE - R)$ por lo cual es un absurdo.

$\therefore HP \notin R$

Ejercicio 13

Demuestre que $L_{NV} \in RE$

$\boldsymbol{L_{NV} = \{(<M>)/L(M) \neq \varnothing\}}$

El lenguaje L_{NV} se comprende como todas las codificaciones M tal que aceptan al menos una palabra w.

Se puede construir una MT M_{NV} tal que para cada <M>, lo ejecuta de **a pasos** sobre todo $w \in \Sigma^*$ en orden canónico, así de esta manera:

1. Si <M>acepta algún w $\Rightarrow L_{NV}$ acepta <M>. Eventualmente si se ejecuta <M>de a pasos sobre los diferentes $w \in \Sigma^*$ se terminará encontrando algún w que reconoce. No se quedará en loop ya que al ejecutarse de a pasos en algún momento se probará con los demás w (inputs).
2. Si <M>no reconoce ningun input $\Rightarrow L_{NV}$ loopeara indefinidamente.

De esta manera $L_{NV} \in RE$

Práctica 5

Nota: Los casos de codificación inválida son triviales, por lo cual no es necesario tenerlos en cuenta para realizar demostraciones. Ya que se sobre-entienden esos casos especiales.

Ejercicio 1

Sean $\mathbf{L_1}$ y $\mathbf{L_2}$, dos lenguajes definidos sobre $\mathbf{\{0,1\}*}$

$\mathbf{L_1 = \{0^n1 | n \geq 0\}}$

$\mathbf{L_2 = \{1^n0 | n \geq 0\}}$

Inciso a

Demuestre que existe una reducción ($\mathbf{L_1}$ α $\mathbf{L_2}$)

Para demostrar la reducción $L_1 \; \alpha \; L_2$, es necesario demostrar dos aspectos:

1) $(\exists M_f)(M_f | M_f$ siempre se detiene)
2) $w \in L_1 \Leftrightarrow M_f(w) \in L_2$

1) Suponga la siguiente MT M_f:

$\delta(q_0, 0) = (q_0, 1, D)$

$\delta(q_0, 1) = (q_0, 0, D)$

$\delta(q_0, B) = (q_d, B, S)$

Así de esta manera, por construcción de M_f este se detiene eventualmente para cualquier $w \in \{0,1\}*$

2) $w \in L_1$

$\Leftrightarrow w = 0^n1, n \geq 0$ (Por construcción de L_1)

$\Leftrightarrow M_f(w) = 1^n0, n \geq 0$ (Por construcción de M_f)

$\Leftrightarrow M_f(w) \in L_2$ (Por definición de L_2)

$\therefore w \in L_1 \Leftrightarrow M_f(w) \in L_2$

Inciso b

Idem para $\mathbf{L_2 = \{\lambda\}}$

Para demostrar la reducción $L_1 \; \alpha \; L_2$, es necesario demostrar dos aspectos:

1) $(\exists M_f)(M_f | M_f$ siempre se detiene)
2) $w \in L_1 \Leftrightarrow M_f(w) \in L_2$

1) Se puede construir una MT M_f de la siguiente manera:

- Dado w, se hace un chequeo sintáctico de que $w = 0^n1, n \geq 0$
  - Si w es inválido, entonces se deja w1 en la cinta.
  - Si w es válido, se borra la cinta y se deja λ en ella.

Claramente M_f se detiene, ya que la entrada es finita

2) $w \in L_1 \Leftrightarrow M_f(w) \in L_2$

La demostración se hará en dos pasos:

2.1) $w \in L_1$

$\Rightarrow w = 0^n 1, n \geq 0$ (por definición de L_1)

$\Rightarrow M_f(w) = \lambda$ (por construcción de M_f)

$\Rightarrow M_f(w) \in L_2$ (por definición de L_2)

2.2) $w \notin L_1$

$\Rightarrow w \neq 0^n 1, n \geq 0$ (por definición de L_1)

$\Rightarrow M_f(w) = w1$ (por construcción de M_f)

$\Rightarrow M_f(w) \notin L_2$

$\therefore w \in L_1 \Leftrightarrow M_f(w) \in L_2$

Inciso c

Idem para $\mathbf{L_2 = \{1^n 0 | n > 0\}}$

Para demostrar la reducción $L_1 \ \alpha \ L_2$, es necesario demostrar dos aspectos:

1) $(\exists M_f)(M_f | M_f$ siempre se detiene$)$
2) $w \in L_1 \Leftrightarrow M_f(w) \in L_2$

1) Suponga la siguiente MT M_f:

$\delta(q_0, s) = (q_1, s, I)\ (\forall s)(s \mid s \in \{0, 1, B\})$

$\delta(q_1, s) = (q_2, 1, D)\ (\forall s)(s \mid s \in \{0, 1, B\})$

$\delta(q_2, 0) = (q_2, 1, D)$

$\delta(q_2, 1) = (q_2, 0, D)$

$\delta(q_2, B) = (q_d, B, S)$

Así de esta manera, por construcción de M_f este se detiene eventualmente para cualquier $w \in \{0, 1\}*$. A diferencia del inciso (a), esta máquina además de complementar los bits, convierte la salida de (a) de w a 1w.

2) $w \in L_1$

$\Leftrightarrow w = 0^n 1, n \geq 0$ (Por construcción de L_1)

$\Leftrightarrow M_f(w) = 11^n 0, n \geq 0$ (Por construcción de M_f)

$\Leftrightarrow M_f(w) \in L_2$ (Por definición de L_2)

$\therefore w \in L_1 \Leftrightarrow M_f(w) \in L_2$

Ejercicio 2

Sean $\mathbf{L_1}$ y $\mathbf{L_2}$, dos lenguajes tales que existe una reducción ($\mathbf{L_1 \ \alpha \ L_2}$)

Inciso a

Qué se puede afirmar de $\mathbf{L_1}$ si se sabe que $\mathbf{L_2 \in R}$

Se puede afirmar que $L_1 \in R$. Ya que está demostrado (en la teoría) que es posible la construcción de una MT si se dan la condiciones especificadas, de forma tal que si $L_2 \in R \Rightarrow L_1 in R$

Inciso b

***Qué se puede afirmar de* $\mathbf{L_1}$ *si se sabe que* $\mathbf{L_2} \in$ (CO-RE − RE)**

Se puede afirmar que $L_1 \notin RE$:

$L_2 \in (Co\text{-}RE - RE)$

$\Rightarrow \overline{L_2} \in RE$ (definición de Co-RE)

$\Rightarrow \overline{L_1} \in RE$ (Teorema $L_2 \in RE \Rightarrow L_1 \in RE$)

$\Rightarrow L_1 \in Co\text{-}RE$ (Por definición de Co-RE)

Inciso c

***Qué se puede afirmar de* $\mathbf{L_2}$ *si se sabe que* $\mathbf{L_1} \in \mathbf{R}$**

No es posible realizar alguna afirmación.

Inciso d

***Qué se puede afirmar de* $\mathbf{L_2}$ *si se sabe que* L1 $\in$ (CO-RE − RE)**

$L_1 \in (Co\text{-}RE - RE)$

$\Rightarrow L_1 \notin RE$

$\Rightarrow L_2 \notin RE$ (Teorema. $L_1 \notin R \Rightarrow L_2 \notin R \Leftrightarrow L_1 \; \alpha \; L_2$)

$\Rightarrow L_2 \notin R$

Ejercicio 3

Sean HP y* $\mathbf{L_u}$ *los lenguajes Halting Problem y Lenguaje Universal respectivamente.

HP = {(<*M*>, w)/M *se detiene con input w*}

$\mathbf{L_u}$ = {(<*M*>, w)/M *acepta w*}

***Demuestre que existe una reducción* HP α $\mathbf{L_u}$**

Para demostrar que $HP \; \alpha \; L_U$, se deben demostrar dos aspectos:

1. $(\exists M_f)(M_f \mid M_f$ siempre se detiene)
2. $w \in HP \Leftrightarrow M_f(w) \in L_u$

Se puede construir una MT M_f tal que $M_f((\text{<M>}, w)) = (\text{<M'>}, w)$, donde <M'> trabaja de la siguiente forma:

- Si (<M>, w) es un par inválido, entonces M_f borra la cinta y deja λ como salida
- Si (<M>, w) es un par válido, pero <M>es inválido, entonces M_f codifica <M'>, borrando <M>y definiendo las quíntuplas $(q_0, s, q_A, s, S)(\forall s)(s \mid s \in \Gamma)$, es decir L(M') $= \Sigma^*$
- caso contrario construye <M'>, buscando todas los estados q_R y los reemplaza con q_A.

1) M_f realiza una serie de pasos finitos, entonces por construcción siempre se detiene.

2) La demostración se hará en dos pasos:

2.1) $(\text{<M>}, w) \in HP$

a) $\Rightarrow$ M se detiene rechazando o aceptando w

$\Rightarrow$ M' acepta w (por construcción de M', M_f intercambia q_R por q_A)

$\Rightarrow$ (<M'>, w) $\in L_U$

b) $\Rightarrow$ <M> es un codificación inválida

$\Rightarrow$ M' acepta w (por construcción de M', L(M') = Σ^*)

$\Rightarrow$ (<M'>,w) $\in L_U$

2.2) (<M>, w) $\notin HP$

a) $\Rightarrow$ M loopea indefinidamente sobre w

$\Rightarrow$ M' loopea indefinidamente sobre w (por construcción de M', M_f solo intercambia q_R por q_A, si M loopeaba, M' va a seguir loopeando igualmente)

$\Rightarrow$ (<M'>, w) $\notin L_U$

b) $\Rightarrow$ (<M>, w) es un pár inválido

$\Rightarrow$ (<M'>, w)= λ

$\Rightarrow$ (<M'>, w) $\notin L_U$

Ejercicio 4

Sea* $\mathbf{HP_\lambda}$ *el problema de detención a partir de la cinta en blanco

$\mathbf{HP_\lambda = \{<}$***M*****>/M *se detiene con input* $\boldsymbol{\lambda}\}$**

***Demuestre que existe una reducción* HP $\boldsymbol{\alpha}$ $\mathbf{HP_\lambda}$**

Para demostrar la reducción $HP \ \alpha \ HP_\lambda$, es necesario demostrar dos aspectos:

1. $(\exists M_f)(M_f \mid M_f$ siempre se detiene)
2. $w \in HP \Leftrightarrow M_f(w) \in HP_\lambda$

Se puede construir una MT M_f, donde dado un par (<M>, w), M_f(<M>, w) = <M'>; donde <M'>está construido de la siguiente manera:

- Se construye <M'>escribiendo las quíntuplas necesarias para que M' borre la entrada y escriba w en la cinta, posicione el cabezal y simule M sobre w. Así M' para en q_A o q_R para cualquier input Σ^*

1) Claramente M_f se detiene luego de realizar una cantidad finita de pasos.

2) $w \in HP \Leftrightarrow M_f(w) \in HP_\lambda$

La demostración se hará en dos pasos:

2.1) (<M>$, w) \in HP$

a) $\Rightarrow$ M para en q_A o q_R con w (por definición de HP)

$\Rightarrow$ M' para en q_A o q_R con cualquier entrada (por construcción de M')

$\Rightarrow$<M'>$\in HP_\lambda$

2.2) Si (<M>,w)$\notin HP \Rightarrow$

a) Si M loopea sobre w

$\Rightarrow$ <M'>también loopea sobre w

$\Rightarrow$ <M'>$\notin HP_\lambda$

$\therefore HP \; \alpha \; HP_\lambda$

Ejercicio 5

Demuestre que $\mathbf{L_V \notin RE}$

$\mathbf{L_V = \{(<M>)/L(M) = \varnothing\}}$.

Considere que si <M>es un código inválido de máquina de Turing también pertenece a $\mathbf{L_V}$ ***(ya que no reconoce ningún string). Así*** $\mathbf{L_V}$ ***es el complemento del lenguaje*** $\mathbf{L_{NV} = \{(<M>)/L(M) \neq \varnothing\}}$ ***(Ayuda: puede utilizar el complemento de*** $\mathbf{L_u}$ ***para encontrar una reducción)***

Para demostrar que $L_V \notin RE$, se llevará a la reducción de $\overline{L_U} \; \alpha \; L_V$, ya que hay un teorema que si $L_1 \; \alpha \; L_2$ entonces $L_1 \notin RE \Rightarrow L_2 \notin RE$

Para demostrar la reducción se debe cumplir que:

1. Existe una M_f tal que siempre se detiene
2. $w \in \overline{L_U} \Leftrightarrow M_f(w) \in L_V$

Se puede construir una MT M_f tal que (<M>, w) = <M'> , donde M_f realiza primero un chequeo sintáctico, si el par es inválido entonces M_f borra la cinta y codifica <M'> de forma que rechace toda entrada. Por ejemplo, definiendo las quíntuplas $(q_0, s, q_R, s, S)(\forall s)(s \mid s \in \Sigma \cup \{B\})$

Caso contrario se codifica <M'> agregando las quíntuplas necesarias para borrar la entrada, hardcodear w, posicionar el cabezal al inicio de la cinta y ejecutar M sobre w.

1) M_f se detiene, ya que realiza una serie de pasos finitos.

2) La demostración se hará en dos pasos:

2.1) (<M>, w) $\in \overline{L_U} \Rightarrow$

a) $\Rightarrow$ M rechaza o loopea sobre w

$\Rightarrow$ M' rechaza o loopea sobre w

$\Rightarrow$ M' rechaza o loopea cualquier entrada (porque esta hardcodea w para cualquier entrada)

$\Rightarrow L(M') = \varnothing$

$\Rightarrow$ <M'>$\in L_V$

2.2) (<M>, w) $\notin \overline{L_U} \Rightarrow$

$\Rightarrow$ M acepta w

$\Rightarrow$ M' acepta w

$\Rightarrow$ M' acepta toda entrada (porque esta hardcodea w para cualquier entrada)

$\Rightarrow L(M') \neq \varnothing$

$\Rightarrow$ <M'>$\notin L_V$

$\therefore \overline{L_U} \; \alpha \; L_V \Rightarrow L_V \notin RE$