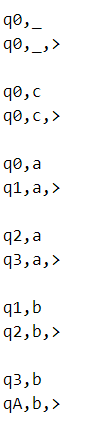
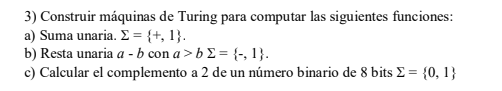


a) Con dos cintas es fácil

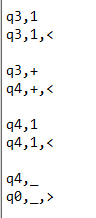
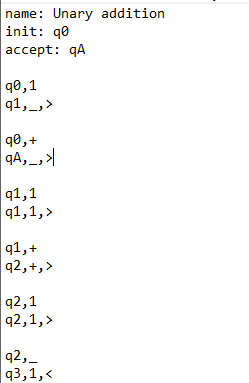
//con la coma separamos las cintas, en la primera esta el 0011 pedido y después en la cinta 2 vamos creando la misma cantidad de 0 que en la primera y cuando encuentra 1 empezamos a borrarlos hacia la izquierda en la cinta 2, y si ambos coinciden al quedar en blanco se acepta.

q­00011, q­i├ M 0q011, 0q­0├ M 00q­011, 00q­0 ├ M 001q­11, 0q­1 ├ M 001q­11, q­1 ├ M 0101q1B, Bq1 ├ 0101BqA, qAB

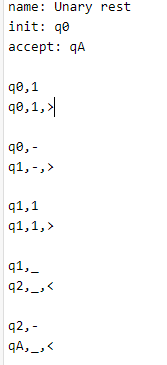
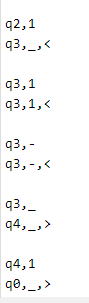
b) 



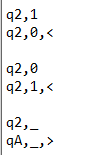
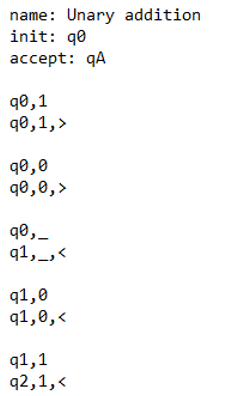
a)

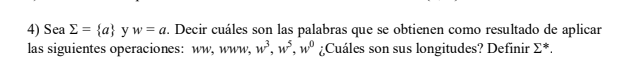


b)

c)





ww es aa, www es aaa, w^3 es aaa, w^5 es aaaaa, w^0 es lambda, la longitud es literal 2 para el primero, 3 para el segundo, 3 para el tercero imagino? y todo así…

 = lambda, a, a.a, a.a.a … y asi



ww es abaaba, www es abaabaaba, w^3 es abaabaaba, w^5 es abaabaabaabaaba, w^0 es aba^0 o 1?

 = lambda, a, b, a.a, a.b, b.a, a.a.a, a.a.b … y así



lambda

a

b

c

aa

ab

ac

ba

bb

bc

ca

cb

cc 

L = {0w / w ∈ Σ\*} Estás diciendo que tu palabra empieza con 0 y al lado tiene un carácter cualquiera

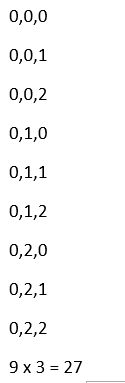
L = {0w^n / w ∈ Σ\* y n >= 0} Acá estás aceptando todas las cadenas que empiecen con 0 (0, 00, 01, 001, 011, 0111111)

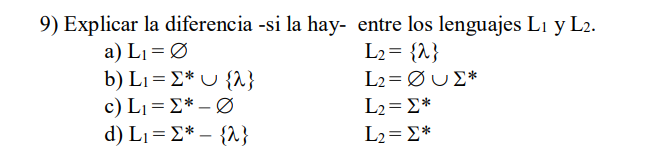
{w/ w es un lenguaje valido cuando la cadena es 1 y cuando es 0} Solo acepta cero o uno.

{w/ w es un lenguaje valido cuando la cadena es 101 y cuando es 010}



Siempre va a ser la cantidad de elementos Elevado a la n (longitud), es decir para 1 de longitud serian 3 elementos, para 2 serian 9, para 3 serian 27 y asi…



1. El primero es un lenguaje vació y el segundo es un lenguaje con una cadena vacía
2. Es un lenguaje con todas las cadenas y una cadena vacía, el segundo son todos los lenguajes con un lenguaje vació por lo que serían lo mismo.
3. Es un lenguaje con todas las cadenas sin vació y el otro es un lenguaje con todas las cadenas, por lo tanto son iguales
4. Es un lenguaje con todas las cadenas sin una cadena vacía y el otro son todas las cadenas, por lo tanto no son iguales



Si los naturales son infinito contable esto también, imagino que se justifica con el {1,2,3…}

a) Para probar el enunciado hay que demostrar que |Sigma| <= |N|

Por tanto hay que encontrar una funcion inyectiva f tal que:

f:Sigma -> N

Sigma = 0,1,2,3,4,5,6,7,8,9

Sigma\* = 0,1,2,3,4,5,6,7,8,9, 10, 11, 12,..., 21, ...9321

Por definición sabemos que Sigma es finito.

si sigma es finito entonces las palabras que se pueden generar

con el mismo son ordenables. Lo importante es conocer la |Sigma|

pues esto nos dará el criterio de conversión para la función de

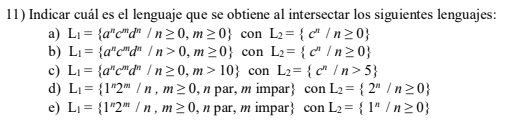
mapeo. Con la cantidad de elementos de sigma bien se puede definir

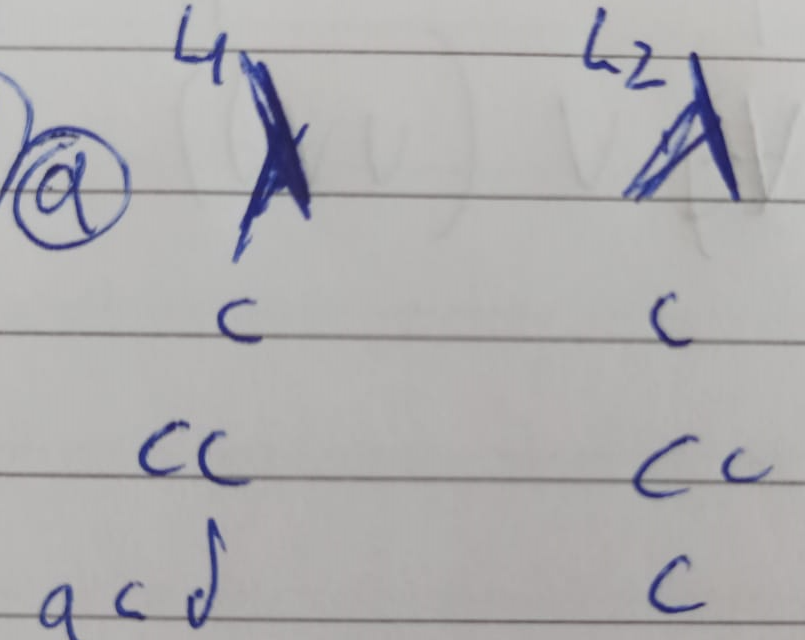
un estilo de ordenamiento con base |Sigma| y la definición de la

función sería tan simple como una función que convierte de una base

a otra la cual claramente existe. Por ejemplo, convertir de base 2 a

decimal (si fuese que |Sigma| es 2)



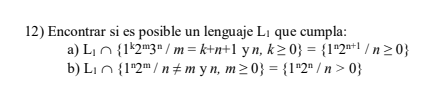
a) Se obtiene L2 

b) Nunca se encuentran porque siempre existe un a o d.

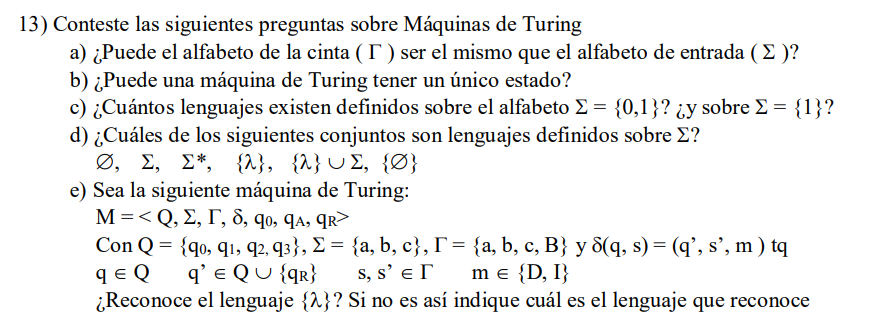
c) Solo se encuentran cuando m> 10 y el n de Ln1 sea 0, y el de L2 sea mayor a 10

d) Se encuentran cuando n del L1 sea 0 y m>=0 y el n de L2 sea impar y mayor a 0

e) Se encuentran cuando n del L1 sea 0 y m>=0 y el n de L2 sea par y mayor a 0

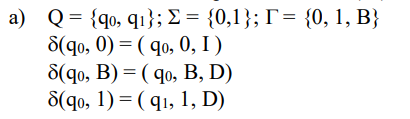


1. Este creo que queda como el 2do y el b) creo que no se puede xq dan distintos

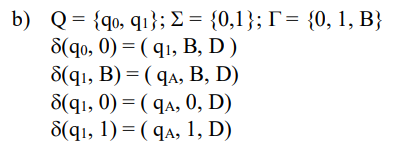


1. Si puede, aunque en general el alfabeto de entrada tiende a no tener cadena vacía.
2. No puede, mínimo se necesita q0 y un estado para aceptar.
3. Infinitos porque podes hacer 0000… y lo mismo con 1 o combinándolos. Por eso también con 1 son infinitos. Al ser infinitos conjuntos de palabras seria equivalente a generar infinitos subconjuntos de palabras, es decir tendríamos infinitos lenguajes.
4. Σ, {λ}, {λ} ∪ Σ y **Ø** son los que pueden ser definidos sobre Σ
5. No, porque q’ es Q U qR entonces siempre termina porque no tiene qA el q’, nunca llegaremos a qA

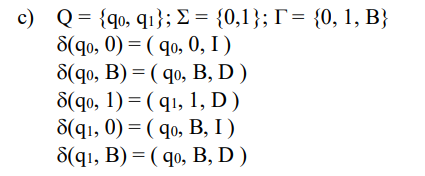




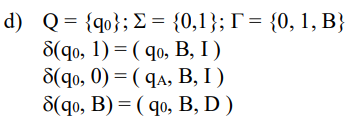
Acepta lenguaje vacío porque no tiene ninguna transición que pare en qA



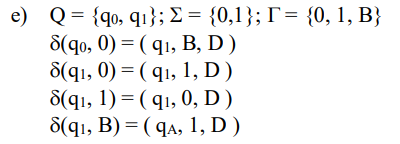
Acepta el lenguaje L = {0w^n / w ∈ Σ\* y n >= 0} o sea, todas las cadenas que empiezan con 0



Acepta lenguaje vacio porque no tiene ninguna transición que pare en qA



Acepta los lenguajes con cadenas que tengan un 0 al inicio sino se queda yendo infinitamente a la izquierda a menos que haya blanco, ahí va a la derecha



Acepta los lenguajes con cadenas que empiecen con cero y luego cambia los ceros por unos y los unos por ceros, al blanco del final le pone un 1. L = {0w^n / w ∈ Σ\* y n >= 0}