<u>Laboratorio 8</u>

EJERCICIOS PROPUESTOS: Para cada ejercicio diseñar la Máquina de Turing. Además definir sus estados y transiciones. Finalmente explicar cómo aplica la MT para resolver el problema.

1) Diseñar una Máquina de Turing que calcule el complemento a 1 de un número binario. (Es decir, que sustituya los 0's por 1's y los 1's por 0's).

M = (S;Σ;Γ;Δ;q1;δ)

Donde:

Γ : es el alfabeto de símbolos de la cinta

Σ : es el alfabeto de símbolos de entrada

Δ : es el símbolo en blanco

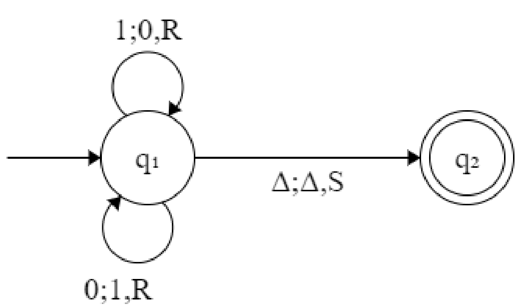
S: es un conjunto finito de estados

q1 : es el estado inicial

δ : es una funcion de transicion parcial

En el estado q1 lo que hacemos es hacer que calcule el complemento de 1 y que el cabezal tenga un movimiento a la derecha

Cuando cumpla la tarea se parará la máquina



δ(q1 , 1;0,R) ->q1 δ(q1 , Δ;Δ,L) -> q2 δ(q1 , 0;1,R) -> q1

2) Diseñar una Máquina de Turing que obtenga el sucesor de un número en codificación unaria. Considerar en la codificación unaria que el 0 se representa por la cadena vacía, el 1 por 1, el 2 por 11, etc.

M = (S;Σ;Γ;Δ;q1;δ)

Donde:

Γ : es el alfabeto de símbolos de la cinta
Σ : es el alfabeto de símbolos de entrada
Δ : es el símbolo en blanco
S: es un conjunto finito de estados
q1 : es el estado inicial
δ : es una funcion de transicion parcial

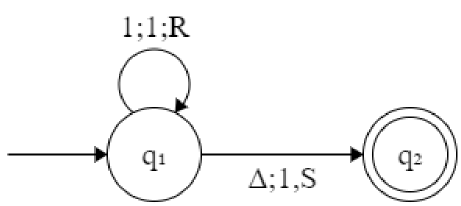
0 = lambda ,

1 -> 1 ,

2 -> 11 ;

3 -> 111

En el mismo bucle podemos generar todos los 1's que queramos para poder cumplir la condición que mencionamos anteriormente, luego la maquina se para con S



δ(q1 , 1;1,R) -> q1 δ(q1 , Δ;1,S)->q2

3) Diseñar una Máquina de Turing que obtenga el predecesor de un número en codificación unaria. Considerar la codificación unaria del 0 igual que en el ejercicio 2.

M = (S;Σ;Γ;Δ;q1;δ)

Donde:

Γ : es el alfabeto de símbolos de la cinta
Σ : es el alfabeto de símbolos de entrada
Δ : es el símbolo en blanco

S: es un conjunto finito de estados
q1 : es el estado inicial
δ : es una funcion de transicion parcial

0 = lambda ,

1 -> 1 ,

2 -> 11 ;

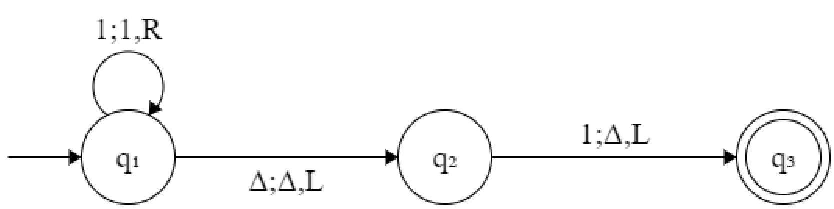
3 -> 111

Lo que queremos hacer que escriba lea digamos un 3 en unario entonces por consiguiente queremos generar el 2 en unario ya que es el su predecesor

111 -> generar 11

Como en el primer ejercicio generamos todos los números unarios que queramos y al final nos quedamos en una cadena delta por lo que retrocedemos para llegar a un 1 EJM 11***1***

Con el cual con la transición 1;Δ,L leemos ese 1 mencionado y nos movemos a la izquierda y nos quedamos con 11 por lo tanto la máquina acepta dicha cadena



δ(q1 , 1;1,R)-> q1   δ(q2 , 1;Δ,L)-> q3   δ(q1 , Δ;Δ,L)->q2

4) Diseñar una Máquina de Turing que calcule la paridad de un número binario. Es decir, si el número de 1's de la cadena es par, se añade un 0 al final, y si es impar, se añade un 1.

M = (S;Σ;Γ;Δ;q1;δ)

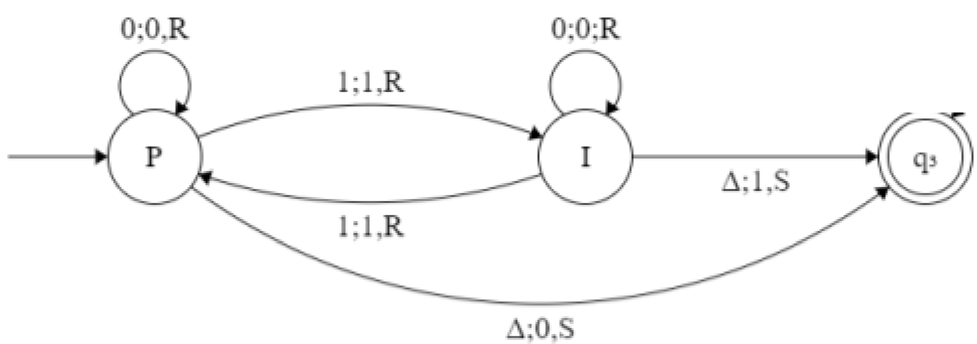
Donde:
Γ : es el alfabeto de símbolos de la cinta
Σ : es el alfabeto de símbolos de entrada
Δ : es el símbolo en blanco

S: es un conjunto finito de estados
q1 : es el estado inicial
δ : es una funcion de transicion parcial

P representa el número de 1’s de la caden par

I representa el número de 1’s impar

- En el bucle del estado P lo que queremos decir es que si lee un 0 entonces se mantiene es ese mismo estado pero si lee un 1 se mueve a la derecha para cambiar de estado
- Para los casos en los que se lee un Delta nos trasladamos al estado de aceptación y escribimos un 0 para mantener la cadena PAR y en caso que lea otro 1 para conseguir la cadena IMPAR



δ(P , 0;0,R)-> P
δ(P , 1;1,R)-> I
δ(I , 1;1,R)->P
δ(I , 0;0,R)->I
δ(I , Δ;1,S)->q3
δ(P, Δ;0,S)->q3