eg Construct TM for addition fun^n for the unary number system.

Sol^n The unary no. is made up of only one character.

eg no. 5 can be written in unary no. system as 11111.

In this TM we are going to perform addition of 2 unary nos.

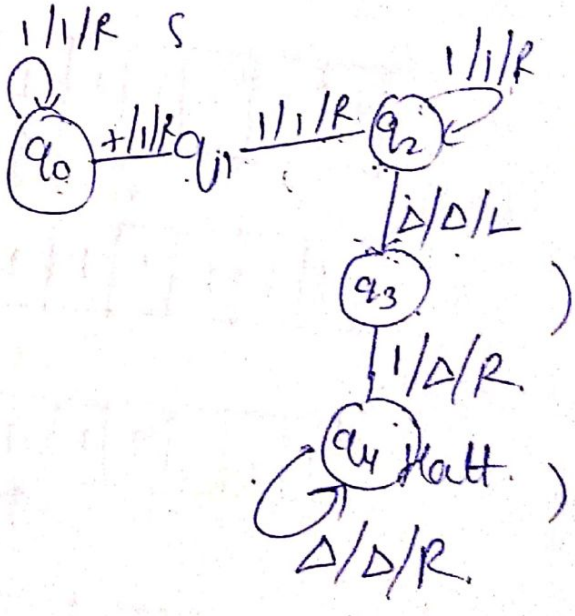
for eg 3 + 2 = 5

ie 111 + 11 = 11111

we simply replace + by 1 & move ahead right for searching end of the string we will convert last 1 to Δ

~~111+11Δ Move right~~

Instantaneous Description.



① $\delta(q_0, 1) = (q_1, 1, R)$

② $\delta(q_1, +) = (q_2, 1, R)$

③ $\delta(q_2, 1) = (q_2, 1, R)$

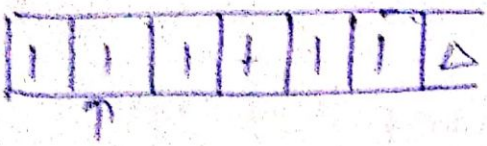
④ $\delta(q_2, \Delta) = (q_3, \Delta, L)$

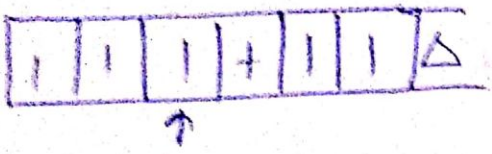
⑤ $\delta(q_3, 1) = (q_4, \Delta, R)$

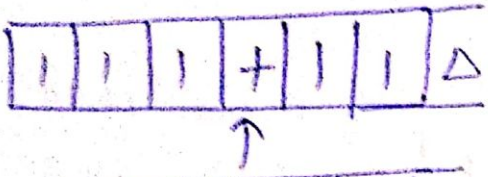
⑥ $\delta(q_4, \Delta) = (q_4$ (Halt state, $\Delta, R)$

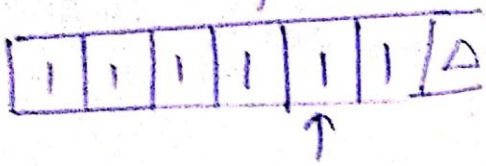
Suppose we have 111+11 = 11111

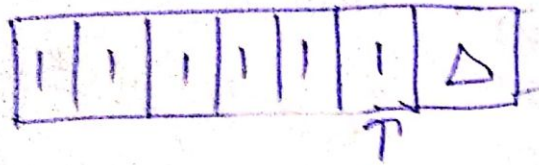

move right

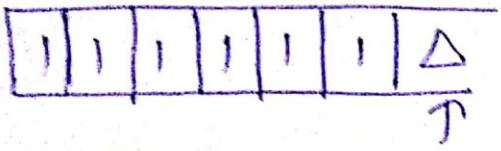

"

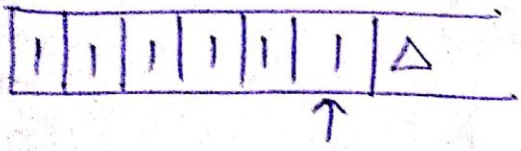

"

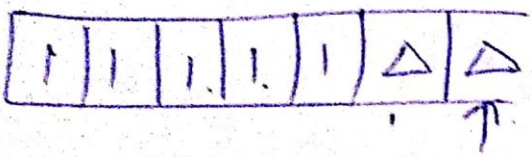

Convert + to 1 & move right.


move right


move right


move left.

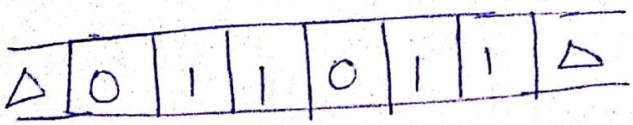

Convert 1 to △ & move right


Halt.

— X —

Two Way Infinite Tape.

→



The tape head as usual can move in forward & backward direction.

→ 2 way infinite tape turing m/c can also be denoted by $M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, F)$

↓

In 2 way infinite tape it placed at both the ends of the tape. ~~On both~~ and string lies b/w these Δ.

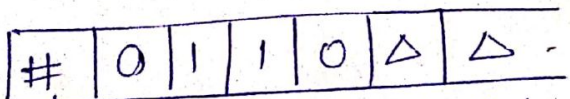
Theorem

L is recognized by a TM with a 2 way infinite tape if & only if it is recognized by a TM with one way infinite tape.

Let M_1 be a TM with one way infinite tape & can be denoted as $M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_1, B, F_1)$

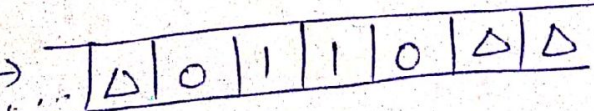
Similarly, M_2 be a TM with 2 way infinite tape. & can be denoted as $M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_2, B, F_2)$

M_1 →



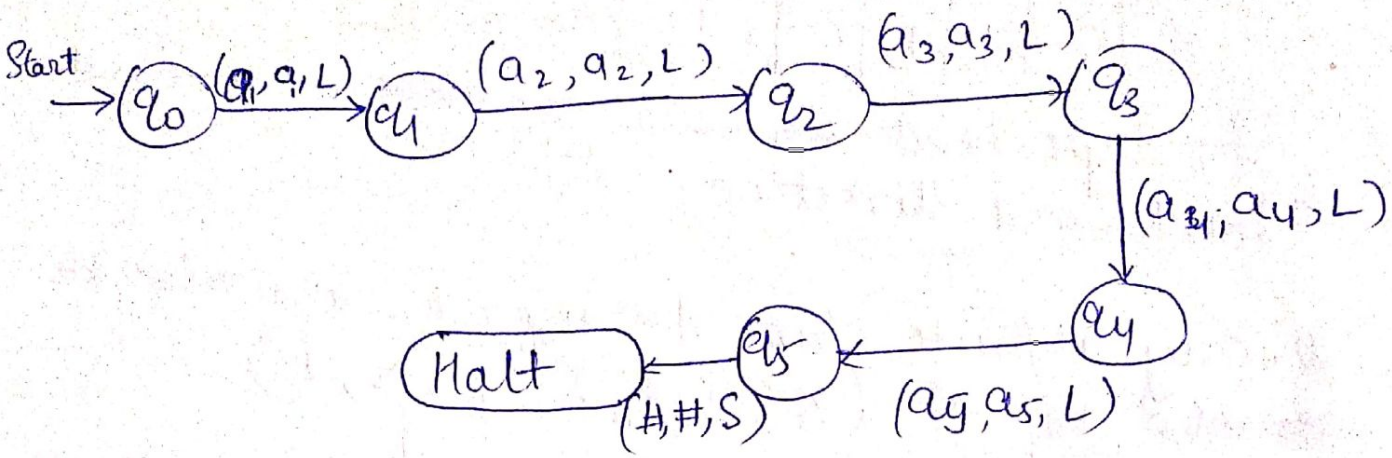
↳ It indicates the left side ~~ter~~ mination.

M_2 →



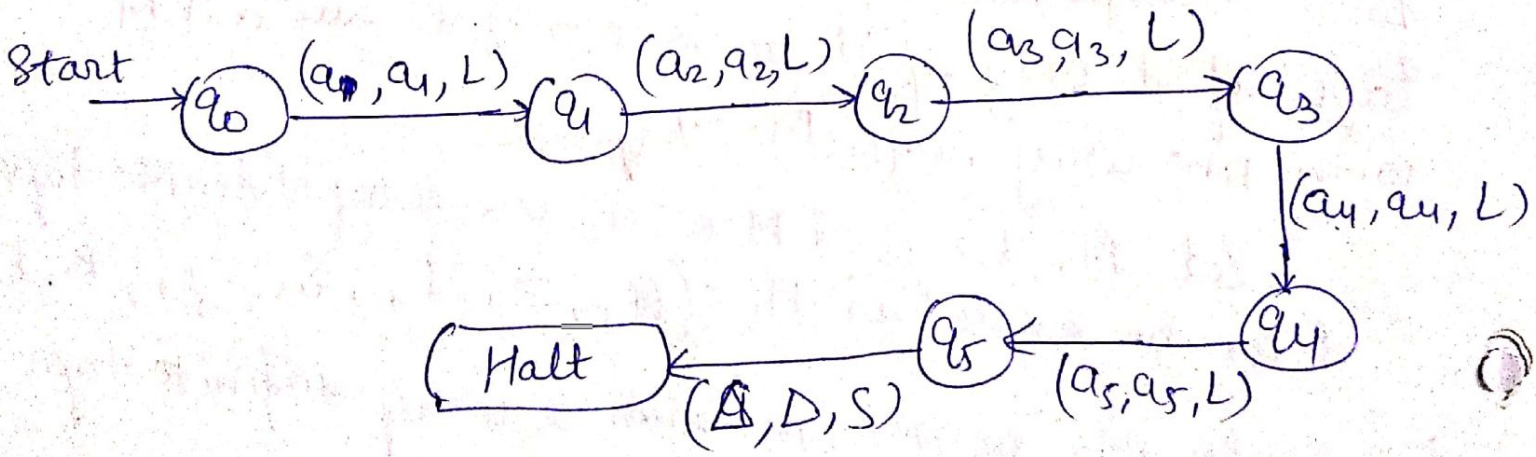
The TM M_1 can be

#	a_5	a_4	a_3	a_2	a_1	Δ	Δ	...



The TM M_2 can be

...	Δ	a_5	a_4	a_3	a_2	a_1	Δ	Δ	...



Q Construct TM for obtaining 2's complement of a given binary no.

Sol The logic for computing 2's complement is,

(i) We read binary string from LSB to MSB. From LSB, we keep track all 0's as it is & move left till we do not get 1.

(ii) After getting 1 from LSB we move left & ∴ we convert 0 to 1 & 1 to 0 & go on moving towards left.

The process continues upto leftmost △.

eg : 0 1 1 0

1's complement for 1 0 0 1

+ 1

1 0 1 0 → 2's complement.

o/p 0 1 1 0

o/p 0 1 1 0

2's com 1 0 1 0

C C NC

q_0 → Searching △

q_1 → R to L change

q_2 → first occurance of 1 from LSB

ID

① $\delta(q_0, 0) = (q_0, 0, R)$

② $\delta(q_0, 1) = (q_0, 1, R)$

③ $\delta(q_0, \triangle) = (q_1, \triangle, L)$

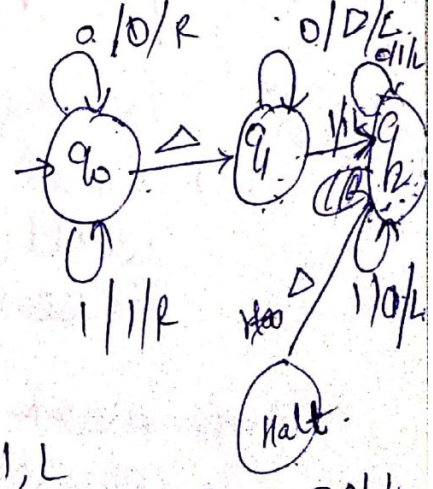
④ $\delta(q_1, 0) = (q_1, 0, L)$

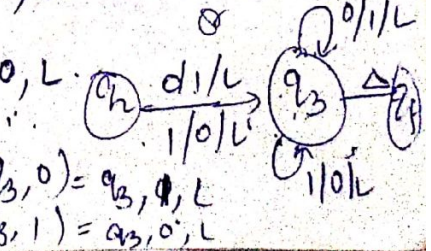
⑤ $\delta(q_1, 1) = (q_2, 1, L)$

⑥ $\delta(q_2, 0) = (q_2, \phi, L) / q_3, 1, L$

⑦ $\delta(q_2, 1) = (q_2, 0, L) / q_3, 0, L$

⑧ $\delta(q_2, \triangle) = (q_{Halt}, \triangle, S)$

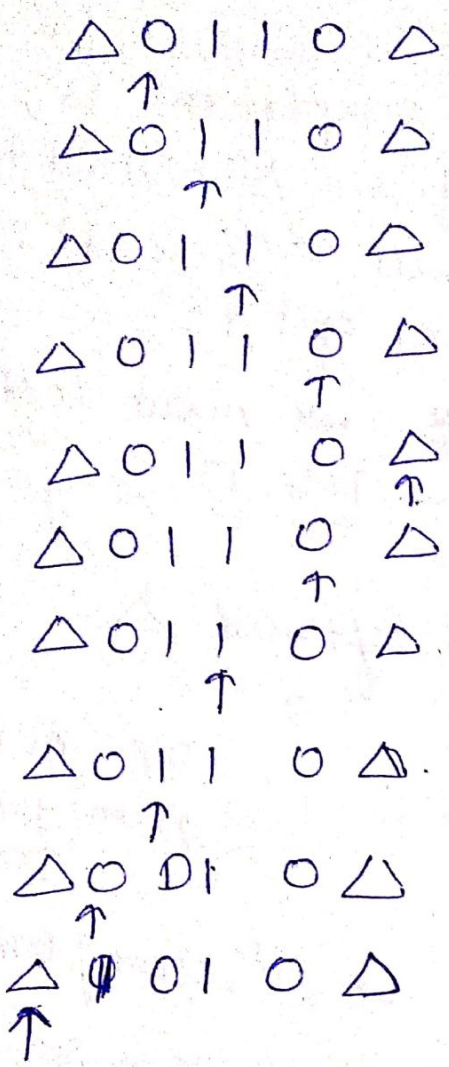




$(q_3, 0) = q_3, 1, L$

$(q_3, 1) = q_3, 0, L$

Consider the string 0 1 1 0



Move Right

"

"

"

Move left

It is 0 so keep it as it is & move left

It is 1st 1 from LSB so keep as it is & move left.

Convert 1 to 0 & move left

Convert 0 to 1 & move left

Since △ is reached, stop.