Turing Machine As a Computer of Integer Functions ①

It is also used as a computer of functions from integers to integer. The traditional approach is to represent integers in unary form. The integer $i \geq 0$ respresented by the string 0^i.

Ex:

Construct a TM that Perform addition operation.

Sol:

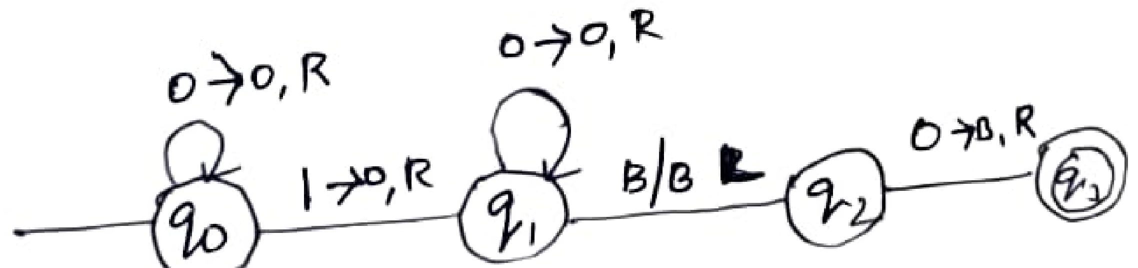
The function is defined as $f(x+y) = x+y$

x is given by 0^x

y is given by 0^y

logic

The input is placed on the tape $0^x 1 0^y$. The sum of these two values are Performed by replacing the intermediate 1 by 0 and replacing the last 0 by blank Symbol.



Transition table

	0	1	B
→ q0	(q0, 0, R)	(q1, 0, R)	-
q1	(q1, 0, R)	-	(q2, B, L)
q2	(q3, B, R)	-	-
* q3	-	-	-

Ex:

$x = 3,\ y = 2,\ f(x+y)$

$q_0 0^3 1 0^2 \Rightarrow q_0 000100 \vdash 0q_0 00100$

$\vdash 00q_0 0100 \vdash 000q_0 100 \vdash$

$0000q_1 00 \vdash 00000q_1 0 \vdash$

$000000q_1 B \vdash 00000q_2 0B \vdash$

$00000Bq_3$

Ex: 2

Construct TM to compute the function $f : N \to N$ such that $f(x) = x+1$

Sol :-

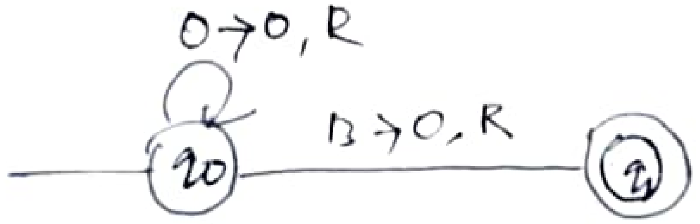
x is represented as 0^x

$f(x) = x+1 = 0^{x+1}$

logic

The o/p contains one more 0 than the i/p.

Initially TM is at q_0. At q_0 if it reads a blank symbol skipping 0's, replace with 0 and enter the final state



Transition table

	0	B
$\rightarrow q_0$	$(q_0, 0, R)$	$(q_1, 0, R)$
$* q_1$	—	—

Ex: 3

Design a TM to compute $f(x) = x+2$

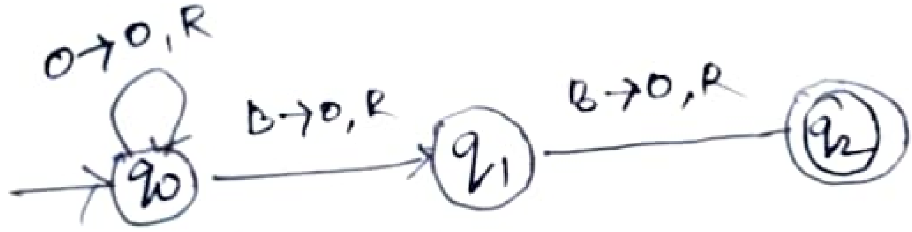
Sol :

x can be represented as 0^x

to find 0^{x+2}

Logic

Initially the TM is at state q_0. At q_0, if it finds a blank symbol, replace it with 0 and enters state q_1. At q_1 it finds a bland symbol. Replace it with 0 and enter the final state q_2



Transition diagram

	0	B
→ q_0	$(q_0, 0, R)$	$(q_1, 0, R)$
q_1	–	$(q_2, 0, R)$
* q_2	–	–

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

$\Sigma = \{0\}$

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

q_0 = initial state

$\{q_2\}$ = final state

Subtraction

Ex: Construct a TM that Perform subtraction

$$f(m, n) = \begin{cases} m > n, & m-n \\ m \leq n, & 0 \end{cases}$$

Sol:-

$m = 3 \quad n = 2$

111011

Step 1

●	1	1	0	1	1	B	B	B	...

↑ q_0

(2)

B	1	1	0	1	1	B	B	B	...

↑ q_1

(3)

B	1	1	0	1	1	B	B	B	...

← q_2

(4)

B	1	1	0	1	B	B	B	B	...

← q_3

(5)

B	1	1	0	1	B	B	B	B	...

←

(6)

B	B	1	0	1	B	B	B	B	...

→

(7)

B	B	1	0	1	B	B	B	...

→

(8)

B	B	1	0	B	B	B	B	...

←

(9)

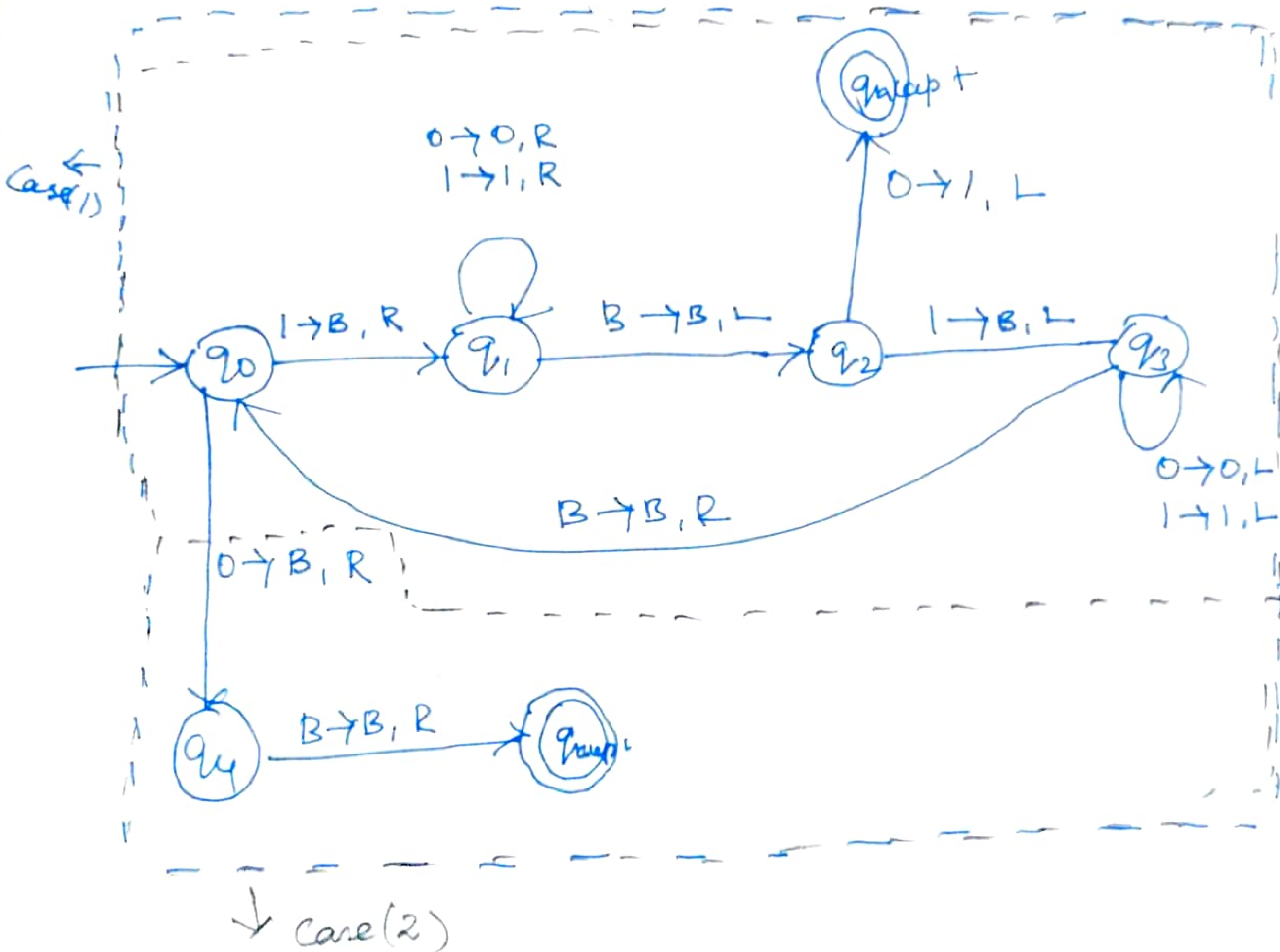
B	B	B	0	B	B	B	...	

→ →

(10)

B	B	B	$\boxed{1}$	B	B	B	B

←


Case(1)
q0
q1
q2
q3
q4
qaccept
qaccept
0→0, R
1→1, R
1→B, R
B→B, L
0→1, L
1→B, L
0→0, L
1→1, L
B→B, R
0→B, R
B→B, R
Case(2)