HW
1.1

(1) Construct a Turing Machine that reverses its inputs
(Input - 0010111 ; Output - 1110100)

Turing Machine Components:-
Alphabet $S = \{\sqcup, 0, 1, *, 0', 1'\}$
External alphabet $A = \{0, 1\}$ $A \subset S$
States $Q = \{q_0, q_f, r_0, r_1, l_0, l_1, l_{01}, l_1\}$

Transition function ($\delta$) :-

a) Start from the left, from the starting position of the string
b) Replace the character with symbol $*$; capture the state
c) Proceed from left to right. Replace the Right character with the left one.
d) Proceed from right to left and replace.
e) Repeat above steps, till we reach the middle of the string.

States and their transitions

Start :-
$(q_0, 0) \to (r_0, *, +1)$
$(q_0, 1) \to (r_1, *, +1)$
$(q_0, \sqcup) \to (q_0, \sqcup, -1)$

Left to right movement:-
$(r_0, 0) \to (r_0, 0, +1)$ $(r_1, 0) \to (r_1, 1, +1)$
$(r_0, 1) \to (r_0, 1, +1)$ $(r_1, 1) \to (r_1, 1, +1)$

Right to left:-
a) Replace the Right character with left
b) Proceed and Correct the mark $*$ with Right character.

Scanned with CamScanner

a) change the direction from Right to left

$(r_0, \sqcup) \to (l_{0'}, \sqcup, -1)$
$(r_1, \sqcup) \to (l_{1'}, \sqcup, -1)$

$(l_{0'}, 0) \to (l_0, 0', -1)$ $(l_{1'}, 0) \to (l_1, 1', -1)$
$(l_{0'}, 1) \to (l_1, 0', -1)$ $(l_{1'}, 1) \to (l_1, 1', -1)$

$(r_0, 0') \to (l_{0'}, 0', -1)$ $(r_1, 0') \to (l_{1'}, 0', -1)$
$(r_0, 1') \to (l_{0'}, 1', -1)$ $(r_1, 1') \to (l_{1'}, 1', -1)$

b) Proced from Right to left
$(l_0, 0) \to (l_0, 0, -1)$ $(l_1, 0) \to (l_1, 0, -1)$
$(l_0, 1) \to (l_0, 1, -1)$ $(l_1, 1) \to (l_1, 1, -1)$

c) Replace the marks $*$ with Right character
$(l_0, *) \to (q_0, 0, +1)$
$(l_1, *) \to (q_0, 1, +1)$

d) End of the function
$(l_{0'}, *) \to (q_f, 0, +1)$
$(l_{1'}, *) \to (q_f, 1, +1)$
$(q_f, 0') \to (q_f, 0, +1)$
$(q_f, 1') \to (q_f, 1, +1)$
$(q_f, \sqcup) \to$ HALT

## Sequence of Configurations

In each row, the state written above a tape cell is shown in brackets next to that cell.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Start | 0 ($q_0$) | 0 | 1 | 0 | 1 | 1 | 1 | |
| L → R | * | 0 ($r_0$) | 1 | 0 | 1 | 1 | 1 | |
| " | * | 0 | 1 ($r_0$) | 0 | 1 | 1 | 1 | |
| " | * | 0 | 1 | 0 ($r_0$) | 1 | 1 | 1 | |
| " | * | 0 | 1 | 0 | 1 ($r_0$) | 1 | 1 | |
| " | * | 0 | 1 | 0 | 1 | 1 ($r_0$) | 1 | |
| " | * | 0 | 1 | 0 | 1 | 1 | 1 ($r_0$) | |
| " | * | 0 | 1 | 0 | 1 | 1 | 1 | ␣ ($r_0$) |
| Interchange | * | 0 | 1 | 0 | 1 | 1 | 1 ($l_{0'}$) | |
| L ← R | * | 0 | 1 | 0 | 1 | 1 ($l_1$) | 0' | |
| " | * | 0 | 1 | 0 | 1 ($l_1$) | 1 | 0' | |
| " | * | 0 | 1 | 0 ($l_1$) | 1 | 1 | 0' | |
| " | * | 0 | 1 ($l_1$) | 0 | 1 | 1 | 0' | |
| " | * | 0 ($l_1$) | 1 | 0 | 1 | 1 | 0' | |
| " | * ($l_1$) | 0 | 1 | 0 | 1 | 1 | 0' | |
| Replace | 1 | 0 ($q_0$) | 1 | 0 | 1 | 1 | 0' | |
| L → R | 1 | * | 1 ($r_0$) | 0 | 1 | 1 | 0' | |
| " | 1 | * | 1 | 0 ($r_0$) | 1 | 1 | 0' | |
| " | 1 | * | 1 | 0 | 1 ($r_0$) | 1 | 0' | |
| " | 1 | * | 1 | 0 | 1 | 1 ($r_0$) | 0' | |
| " | 1 | * | 1 | 0 | 1 | 1 | 0' ($r_0$) | |
| Interchange | 1 | * | 1 | 0 | 1 | 1 ($l_{0'}$) | 0' | |
| L ← R | 1 | * | 1 | 0 | 1 ($l_1$) | 0' | 0' | |
| " | 1 | * | 1 | 0 ($l_1$) | 1 | 0' | 0' | |
| " | 1 | * | 1 ($l_1$) | 0 | 1 | 0' | 0' | |
| " | 1 | * ($l_1$) | 1 | 0 | 1 | 0' | 0' | |
| Replace | 1 | 1 | 1 ($q_0$) | 0 | 1 | 0' | 0' | |
| L → R | 1 | 1 | * | 0 ($r_1$) | 1 | 0' | 0' | |
| " | 1 | 1 | * | 0 | 1 ($r_1$) | 0' | 0' | |
| " | 1 | 1 | * | 0 | 1 | 0' ($r_1$) | 0' | |
| Interchange | 1 | 1 | * | 0 | 1 ($l_{1'}$) | 0' | 0' | |
| L ← R | 1 | 1 | * | 0 ($l_1$) | 1' | 0' | 0' | |
| L ← R | 1 | 1 | * ($l_1$) | 0 | 1' | 0' | 0' | |
| Replace | 1 | 1 | 1 | 0 ($q_0$) | 1' | 0' | 0' | |
| L → R | 1 | 1 | 1 | * | 1' ($r_1$) | 0' | 0' | |
| Replace | 1 | 1 | 1 | * ($l_{0'}$) | 1' | 0' | 0' | |
| END | 1 | 1 | 1 | 0 | 1' ($q_f$) | 0' | 0' | |
| END | 1 | 1 | 1 | 0 | 1 | 0' ($q_f$) | 0' | |
| END | 1 | 1 | 1 | 0 | 1 | 0 | 0' ($q_f$) | |
| | 1 | 1 | 1 | 0 | 1 | 0 | 0 | ␣ ($q_f$) |

HALT

H.W 1.5

Let $T(n)$ be the maximum number of steps performed by a Turing Machine with $\leq n$ states and $\leq n$ symbols before it terminates, starting with the empty tape. Prove that the function $T(n)$ grows faster than any Computable total function $b(n)$. $\lim_{n \to \infty} \frac{T(n)}{b(n)} = \infty$

Proof:- Let M is a TM, with Q states and S-symbols

T = function for the number of steps

when $Q = S = n$

$T = T(n)$: maximum number of steps taken by M before HALT

b is a computable total function. So, there exists a TM, say, $M'$ such that

$M'(n) = b(n)$

Say $T'$ is the number of steps taken by $M'$ using same configuration Q, S.

Since T is the maximum steps for n-symbol, n-state TM, $T' \leq T$

if $T'$ is greater that T, then such TM, $M'$ will not HALT. Because of this, we can say there exist $N \in \mathbb{N}$, such that, $n \geq N$, function $T(n)$ grows much faster than $b(n)$

Example

For $S = \{1, \sqcup\}$, $A = \{1\}$ with n = states

for a given input n,

number of maximum steps $T(n) = O[n]$ (unary representation)

Say b is a computable function to convert n to binary using same S, A. Then complexity of such TM, $M'$

$M'(n) = b(n) = O[\log n]$

As n grows higher $n \to \infty$, $O[n]$ grows faster than $O[\log n]$, therfore

$$\lim_{n \to \infty} \frac{T(n)}{b(n)} = \frac{O[n]}{O[\log n]} \to \infty$$

Scanned with CamScanner

Additional problem 1:

Write a TM that takes input as a binary number a (a string in 0 & 1) and outputs (a−1) (in binary)

Turing Machine Components:-

Alphabet $S = \{\sqcup, 0, 1\}$

External Alphabet $A = \{0, 1\}$ $(A \subset S)$

states $Q = \{q_0, r_0, r_1, l_0, l_1, l_1', q_f\}$

Transition function ($\delta$):-

1. Start from the left, traverse the input string to the right most
2. Capture the states for 0, 1 to the right end. If input string is all 0's, Halt.
3. If Input string is (0's & 1's) start traversing from left to right
4. If we have 1 as right most, change to 0 and Halt
5. If we have 0 as right most, change to 1 and proceed left till we get 1 to change to 0 and Halt. Along the way, change 0 to 1.

States and their transitions:-

start:-

$(q_0, 0) \to (r_0, 0, +1)$

$(q_0, 1) \to (r_1, 1, +1)$

traverse right:-

$(r_0, 0) \to (r_0, 0, +1)$

$(r_0, 1) \to (r_1, 1, +1)$

$(r_1, 0) \to (r_1, 0, +1)$

$(r_1, 1) \to (r_1, 1, +1)$

$(r_1, \sqcup) \to (l_0, \sqcup, -1)$

traverse left:-

$(l_0, 0) \to (l_1', 1, -1)$

$(l_0, 1) \to (l_1, 0, -1)$

$(l_1, 0) \to (q_f, 0, -1)$

$(l_1, 1) \to (q_f, 0, -1)$

$(l_1', 0) \to (l_1', 1, -1)$

$(l_1', 1) \to (l_1, 0, -1)$

HALT:-

$(r_0, \sqcup) \to (q_f, 0, -1)$

a) If All zeros

b) If we reach second from left ~~If right most is 1~~

~~c) If we reach 1 from left If right most is 0~~

## Sequence of Configurations:-

Ex-1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Start | $\overset{q_0}{1}$ | 1 | 0 | 0 | 1 | | (25) |
| L → R | 1 | $\overset{r_1}{1}$ | 0 | 0 | 1 | | |
| " | 1 | 1 | $\overset{r_1}{0}$ | 0 | 1 | | |
| " | 1 | 1 | 0 | $\overset{r_1}{0}$ | 1 | | |
| " | 1 | 1 | 0 | 0 | $\overset{r_1}{1}$ | | |
| " | 1 | 1 | 0 | 0 | 1 | $\overset{r_1}{\sqcup}$ | |
| L ← R | 1 | 1 | 0 | 0 | $\overset{l_0}{1}$ | | |
| " | 1 | 1 | 0 | $\overset{l_1}{0}$ | 0 | | |
| Stop | 1 | 1 | $\overset{q_f}{0}$ | 0 | 0 | | (24) |

Ex-2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Start | $\overset{q_0}{1}$ | 0 | 0 | 0 | 0 | | (16) |
| L → R | 1 | $\overset{r_1}{0}$ | 0 | 0 | 0 | | |
| " | 1 | 0 | $\overset{r_1}{0}$ | 0 | 0 | | |
| " | 1 | 0 | 0 | $\overset{r_1}{0}$ | 0 | | |
| " | 1 | 0 | 0 | 0 | $\overset{r_1}{0}$ | | |
| " | 1 | 0 | 0 | 0 | 0 | $\overset{r_1}{\sqcup}$ | |
| L ← R | 1 | 0 | 0 | 0 | $\overset{l_0}{0}$ | | |
| " | 1 | 0 | 0 | $\overset{l_1}{0}$ | 1 | | |
| " | 1 | 0 | $\overset{l_1}{0}$ | 1 | 1 | | |
| " | 1 | $\overset{l_1}{0}$ | 1 | 1 | 1 | | |
| " | $\overset{l_1}{1}$ | 1 | 1 | 1 | 1 | | |
| " | $\overset{l_1}{0}$ | 1 | 1 | 1 | 1 | | |
| Stop | $\overset{q_f}{0}$ | 1 | 1 | 1 | 1 | | (15) |

# Additional problem 2

Let $T = \{M \mid M$ is a Turing machine$\}$
Find a function $f: T \to N$ such that if $N$ and $M$ are Turing machines and $f(N) = f(M)$ then $N = M$ (that is, $N$ and $M$ have the same alphabet, same transition function etc)

For a given TM, M, Transition function is the set of Configurations formed by states, symbols and shift

States $Q = \{q_0, q_1 \dots q_{n_1}\}$
external Alphabet $A = \{a_1, a_2 \dots a_{n_2}\}$
Alphabet $S = \{s_1, s_2 \dots s_{n_3}\}$
blank symbol $\sqcup$

Transition function $\delta$ is a set of values over
$$\delta(q, s_p) = (q', s', \Delta p) \quad \Delta p \text{ - shift in position.}$$

In this case our function $f: T \to N$ where
$N$ is the set of integers
we need to assign integer to each value for Q, A, S
Q - $n_1$ states
A - $n_2$ external alphabets
S - $n_3$ symbols
By using the property :- Any number can be represented by factors of prime numbers, say 2, 3, 5
we can represent each TM as
$$f(M) = 2^{n_1} \cdot 3^{n_2} \cdot 5^{n_3}$$
Using same notation, we can say
$$f(N) = 2^{n_1} \cdot 3^{n_2} \cdot 5^{n_3}$$

Since $f(M) = f(N)$
we can say Turing machines $N = M$.