UNIT-IV TURING MACHINE

Introduction:

⇒ A Turing machine is an 'automatic machine' that manipulates the input strings according to the transition rules.

⇒ It was invented by Alan Turing in 1936. Turing machine became the most powerful general model of computation.

Description of a TM / Informal definition:

Turing machine is composed of

1. Input tape with tape head
2. Finite set of transition rules.
3. Output tape with tape head.

⇒ The input tape contains the input symbols. The tape is divided into a number of cells. each cell storing one symbol.

⇒ The input tape can also serve as output tape, that contains the resultant value of the operation done by TM.

⇒ The tape head is used to read a symbol from the input tape or write a symbol on the output tape.

⇒ The tape is capable of reading/writing one symbol at a time.

⇒ Several head can be used to perform parallel processes/operations on the head

⇒ The transition rules are responsible for performing the required operations using its rules and based on the input read.

Nature of TM:

The Turing machines are computing machines that are more powerful than the FA and PDA. It is given as,

Finite Automata $\subseteq$ Deterministic PDA $\subseteq$ NPDA $\subseteq$ TM.

FA, DPDA, NPDA are less powerful than TM since these machines have no control over the i/p and cannot modify their own inputs.

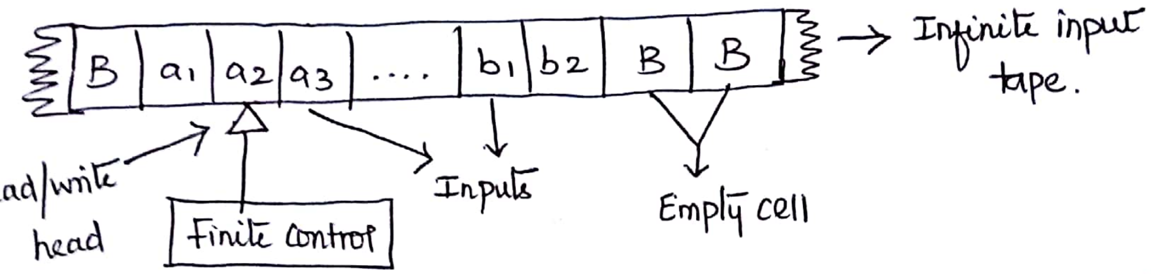
Model of Turing Machine

⇒ TM shall be thought of as a finite state machine with a read/write head.

⇒ The input is stored on an input tape that is divided into a number of individual cells.

⇒ The cell contains blank symbol 'B' when there is no input in it.

⇒ Each cell can store one symbol in it.



⇒ The read/write head of the system reads from and writes data to the i/p tapes.

⇒ The head performs its operation on one cell at a time.

⇒ The movement of the head can be,

- Left (moving backward)
- Right (moving forward)
- None (no movement)

⇒ The head is capable of

- Reading a symbol
- Modifying a symbol.
- Move previous (L) | Move next (R)
- Halt.

Formal Definition of TM.

A Turing machine, M is a 7-Tuple given by,

$$\boxed{M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)}$$

Where

$Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Finite set of input symbols on input tape

$\Gamma \rightarrow$ Finite set of tape symbols $[\Sigma \cup B]$ [$B \rightarrow$ blank symbol]

$\delta \rightarrow$ Transition function given by,

$(q, a) = (q', b, M)$ $M \rightarrow$ Movement (left, Right, No movement)

$q_0 \rightarrow$ Initial state $[q_0 \in Q]$

$B \rightarrow$ Blank symbol representing empty cell $[B \in \Gamma]$

$F \rightarrow$ set of final states $[F \subseteq Q]$

Instantaneous description for TM.

Instantaneous description for TM is the snapshot of how the input string is processed by the Turing machine.

It describes,

→ The input string

→ position of the head

→ State of the machine

$(q, a) = (q', b, M) \rightarrow$ Movement $(L/R/N)$.

q → Current State; a → I/p; q' → Next State; b → replaced symbol

Problems:

1) Design a Turing machine that accepts all strings of the form $a^n b^n$ for $n \geq 1$ and rejects all other strings.

Solution:

Procedure:

(i) change the leftmost a to x

(ii) Move forward right side and change the rightmost b to y.

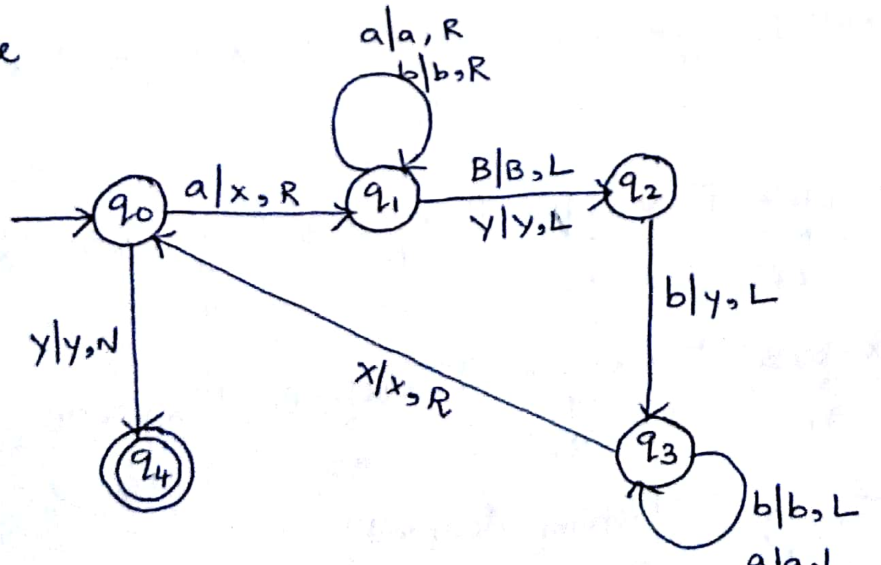
(iii) Move left and change the leftmost a, which is at position immediate right of x, to x.

(iv) Move right and change the rightmost b, which is at immediate left of y, to y

(v) Repeat step (iii) and (iv) until there are no more b's equal to a's.

(vi) Halt the TM.

Turing machine



Description of the TM.

The turing Machine, M is given by,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Where, $Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, B, x, y\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_4\}$

δ is given by,

Q \ $\Sigma \cup \Gamma$	a	b	x	y	B
$\rightarrow q_0$	(q_1, x, R)	–	–	(q_4, y, N)	–
q_1	(q_1, a, R)	(q_1, b, R)	–	(q_2, y, L)	(q_2, B, L)
q_2	-	(q_3, y, L)	–	–	–
q_3	(q_3, a, L)	(q_3, b, L)	(q_0, x, R)	–	–
$* q_4$	ϕ	ϕ	ϕ	ϕ	ϕ

Instantaneous description for the string w = "aabb"

$\underset{\uparrow q_0}{a}abbB \vdash x\underset{\uparrow q_1}{a}bbB \vdash xa\underset{\uparrow q_1}{b}bB \vdash xab\underset{\uparrow q_1}{b}B \vdash xabb\underset{\uparrow q_1}{B}$

$\vdash xab\underset{\uparrow q_2}{b}B \vdash xa\underset{\uparrow q_3}{b}yB \vdash x\underset{\uparrow q_3}{a}byB \vdash \underset{\uparrow q_3}{x}abyB \vdash x\underset{\uparrow q_0}{a}byB$

$\vdash xx\underset{\uparrow q_1}{b}yB \vdash xxb\underset{\uparrow q_1}{y}B \vdash xx\underset{\uparrow q_2}{b}yB \vdash x\underset{\uparrow q_3}{x}yyB \vdash xx\underset{\uparrow q_0}{y}yB$

$\vdash xx\underset{\uparrow q_4}{y}yB$ [String Accepted]

2) Design a TM that perform right shift over $\Sigma = \{0, 1\}$.

Solution:

To perform: shift the input string right by one place.

Procedure:

(i) Move right side to the last character from the initial character.

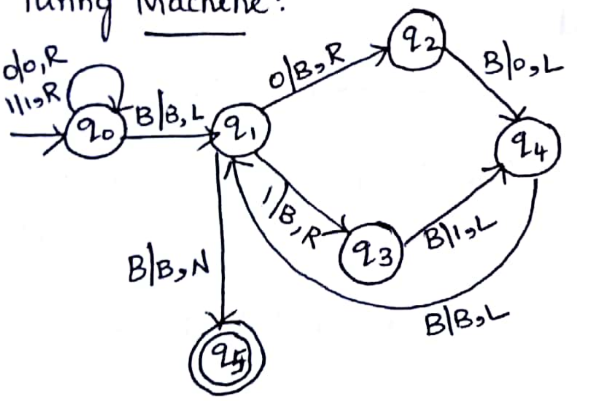
(ii) If the character is '0' replace it with 'B' and move one step right to replace the immediate 'B' to '0'.

(iii) If the character is '1' replace it with 'B' and move one step right to replace the immediate 'B' to 1

(iv) After processing as above, move left one step to perform step (ii) or (iii) on the next right-most unprocessed character.

(v) perform step-(iv) until the all characters are processed.

Turing Machine:



Description of Turing Machine

The Turing Machine is given by,

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Where,

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$ $\quad q_0 = \{q_0\}$

$\Sigma = \{0, 1\}$ $\quad B = \{B\}$

$\Gamma = \{0, 1, B\}$ $\quad F = \{q_5\}$.

Transition function δ is given by.

Q \ $\Sigma \cup \Gamma$	0	1	B
$\to q_0$	$(q_1, 0, R)$	$(q_0, 1, R)$	(q_1, B, L)
q_1	(q_2, B, R)	(q_3, B, R)	(q_5, B, N)
q_2	-	-	$(q_4, 0, L)$
q_3	-	-	$(q_4, 1, L)$
q_4	-	-	(q_1, B, L)
$*q_5$	ϕ	ϕ	ϕ

Instantaneous description for w = "0101"

$$\underset{q_0}{\underset{\uparrow}{0}}101B \vdash 0\underset{q_0}{\underset{\uparrow}{1}}01B \vdash 01\underset{q_0}{\underset{\uparrow}{0}}1B \vdash 010\underset{q_0}{\underset{\uparrow}{1}}B$$

$$\vdash 0101\underset{q_0}{\underset{\uparrow}{B}} \vdash 010\underset{q_1}{\underset{\uparrow}{1}}B \vdash 010B\underset{q_3}{\underset{\uparrow}{B}} \vdash 010\underset{q_4}{\underset{\uparrow}{B}}1$$

$$\vdash 01\underset{q_1}{\underset{\uparrow}{0}}B1 \vdash 01B\underset{q_2}{\underset{\uparrow}{B}}1 \vdash 01\underset{q_4}{\underset{\uparrow}{B}}01 \vdash 0\underset{q_1}{\underset{\uparrow}{1}}B01$$

$$\vdash 0B\underset{q_3}{\underset{\uparrow}{B}}01 \vdash 0\underset{q_4}{\underset{\uparrow}{B}}101 \vdash \underset{q_1}{\underset{\uparrow}{0}}B101 \vdash B\underset{q_2}{\underset{\uparrow}{B}}101$$

$$\vdash \underset{q_4}{\underset{\uparrow}{B}}0101 \vdash \underset{q_1}{\underset{\uparrow}{B}}B0101 \vdash \underset{q_5}{\underset{\uparrow}{B}}0101 \quad \text{Accept}$$

3) Construct a Turing machine to make a copy of a string over $\Sigma = \{0, 1\}$

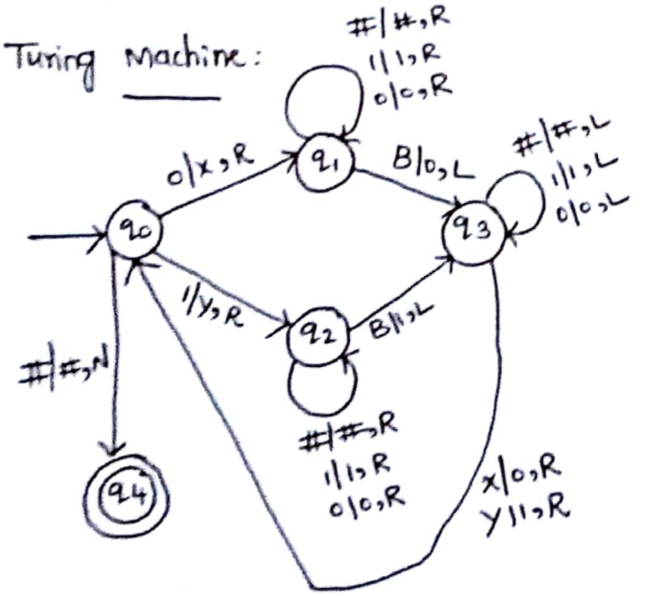
Solution:

Assume: A sequence of input symbols followed by '#' at the end.

Procedure:

1. If the symbol is '0', move right until 'B' is reached. Mark '0' as 'x' to indicate that it is processed.
2. Replace the symbol 'B' by '0'. Move left until 'x' is reached. Mark 'x' as '0' and move right.
3. If the symbol is '1', move right until 'B' is reached. Mark '1' as 'y' to indicate that it is processed.
4. Replace 'B' by 1 and move left until y is reached. Mark 'y' as 1 and move right.
5. Process step 1 and 2 if '0' is the input symbol; else perform step 4 and 5 to process y.
6. Stop and reach state if # is reached.

Turing Machine:



Description of Turing machine

The Turing machine, M is given by

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Where,

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, x, y, \#, B\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_4\}$

Transition function δ is given by

δ \ $\Sigma \cup \Gamma$	0	1	#	B	x	y
$\rightarrow q_0$	(q_1, x, R)	(q_2, y, R)	$(q_4, \#, N)$	–	–	–
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_1, \#, R)$	$(q_3, 0, L)$	–	–
q_2	$(q_2, 0, R)$	$(q_2, 1, R)$	$(q_2, \#, R)$	$(q_3, 1, L)$	–	–
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_3, \#, L)$	–	$(q_0, 0, R)$	$(q_0, 1, R)$
$* q_4$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

Instantaneous description for $w = 101$

$\underset{\underset{q_0}{\uparrow}}{1}01\#B \vdash y\underset{\underset{q_2}{\uparrow}}{0}1\#B \vdash y0\underset{\underset{q_2}{\uparrow}}{1}\#B \vdash y01\underset{\underset{q_2}{\uparrow}}{\#}B \vdash y01\#\underset{\underset{q_2}{\uparrow}}{B} \vdash y01\underset{\underset{q_3}{\uparrow}}{\#}1B$

$\vdash y0\underset{\underset{q_3}{\uparrow}}{1}\#1B \vdash y\underset{\underset{q_3}{\uparrow}}{0}1\#1B \vdash \underset{\underset{q_3}{\uparrow}}{y}01\#1B \vdash 1\underset{\underset{q_0}{\uparrow}}{0}1\#1B \vdash 1x\underset{\underset{q_1}{\uparrow}}{1}\#1B \vdash 1x1\underset{\underset{q_1}{\uparrow}}{\#}1B$

$\vdash 1x1\#\underset{\underset{q_1}{\uparrow}}{1}B \vdash 1x1\#1\underset{\underset{q_1}{\uparrow}}{B} \vdash 1x1\#\underset{\underset{q_3}{\uparrow}}{1}0B \vdash 1x1\underset{\underset{q_3}{\uparrow}}{\#}10B \vdash 1x\underset{\underset{q_3}{\uparrow}}{1}\#10B \vdash 1\underset{\underset{q_3}{\uparrow}}{x}1\#10B$

$\vdash 10\underset{\underset{q_0}{\uparrow}}{1}\#10B \vdash 10y\underset{\underset{q_2}{\uparrow}}{\#}10B \vdash 10y\#\underset{\underset{q_2}{\uparrow}}{1}0B \vdash 10y\#1\underset{\underset{q_2}{\uparrow}}{0}B \vdash 10y\#10\underset{\underset{q_2}{\uparrow}}{B}$

$\vdash 10y\#1\underset{\underset{q_3}{\uparrow}}{0}1B \vdash 10y\#\underset{\underset{q_3}{\uparrow}}{1}01B \vdash 10y\underset{\underset{q_3}{\uparrow}}{\#}101B \vdash 10\underset{\underset{q_3}{\uparrow}}{y}\#101B \vdash 101\underset{\underset{q_0}{\uparrow}}{\#}101B$.

$\vdash 101\underset{\underset{q_4}{\uparrow}}{\#}101B \Rightarrow$ Accept.

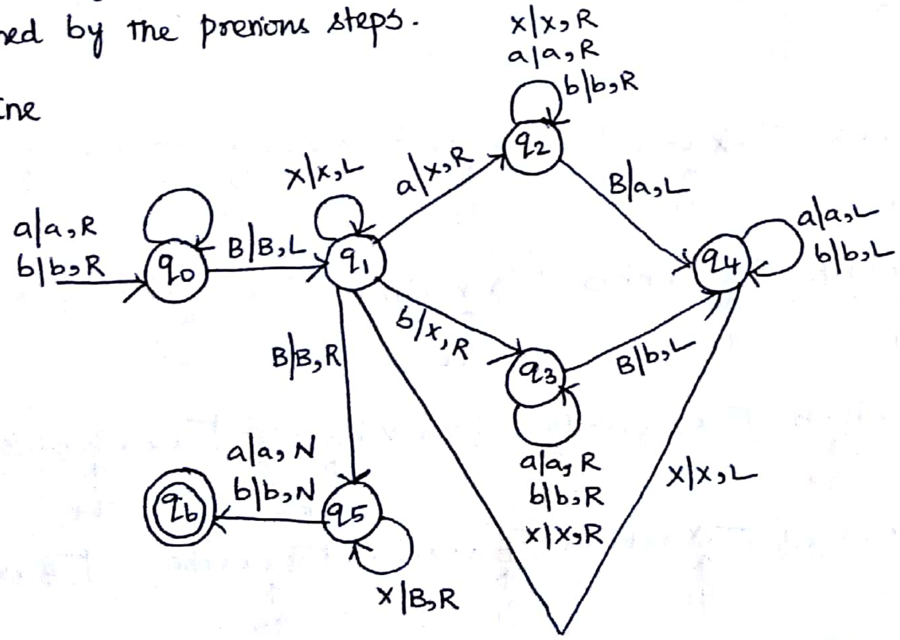
4) Design a TM to reverse a string over $\Sigma = \{a, b\}$

Solution:

Procedure:

1. Move to the last symbol, replace x for a or x for b and move right to convert the corresponding B to 'a' or 'b' accordingly.
2. Move left until the symbol left to x is reached.
3. Perform step 1 and 2 until B is reached while traversing left.
4. Replace every x to B to make the cells empty since the reverse of the string is performed by the previous steps.

Turing Machine



Description of TM:

TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

where,

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, x, B\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_6\}$

Transition function δ is given by.

Q \ $\Sigma \cup \Gamma$	a	b	X	B
q_0	(q_0, a, R)	(q_0, b, R)	–	(q_1, B, L)
q_1	(q_2, X, R)	(q_3, X, R)	(q_1, X, L)	(q_5, B, R)
q_2	(q_2, a, R)	(q_2, b, R)	(q_2, X, R)	(q_4, a, L)
q_3	(q_3, a, R)	(q_3, b, R)	(q_3, X, R)	(q_4, b, L)
q_4	(q_4, a, L)	(q_4, b, L)	(q_1, X, L)	–
q_5	(q_6, a, N)	(q_6, b, N)	(q_5, B, R)	–
q_6	ϕ	ϕ	ϕ	ϕ

Instantaneous description for $w = abb$.

$\underset{\uparrow q_0}{a}bbB \vdash a\underset{\uparrow q_0}{b}bB \vdash ab\underset{\uparrow q_0}{b}B \vdash abb\underset{\uparrow q_0}{B} \vdash ab\underset{\uparrow q_1}{b}B \vdash abX\underset{\uparrow q_3}{B} \vdash ab\underset{\uparrow q_4}{X}bB$

$\vdash a\underset{\uparrow q_1}{b}XbB \vdash aX\underset{\uparrow q_3}{X}bB \vdash aXX\underset{\uparrow q_3}{b}B \vdash aXXb\underset{\uparrow q_3}{B} \vdash aXX\underset{\uparrow q_4}{b}bB \vdash aX\underset{\uparrow q_4}{X}bbB$

$\vdash a\underset{\uparrow q_1}{X}XbbB \vdash \underset{\uparrow q_1}{a}XXbbB \vdash X\underset{\uparrow q_2}{X}XbbB \vdash XX\underset{\uparrow q_2}{X}bbB \vdash XXX\underset{\uparrow q_2}{b}bB$

$\vdash XXXb\underset{\uparrow q_2}{b}B \vdash XXXbb\underset{\uparrow q_2}{B} \vdash XXXb\underset{\uparrow q_4}{b}aB \vdash XXX\underset{\uparrow q_4}{b}baB \vdash XX\underset{\uparrow q_4}{X}bbaB$

$\vdash X\underset{\uparrow q_1}{X}XbbaB \vdash \underset{\uparrow q_1}{X}XXbbaB \vdash \underset{\uparrow q_1}{B}XXXbbaB \vdash B\underset{\uparrow q_5}{X}XXbba \ldots \vdash BXXX\underset{\uparrow q_5}{b}ba \vdash BXXX\underset{\uparrow q_6}{b}baB$ (Accept)

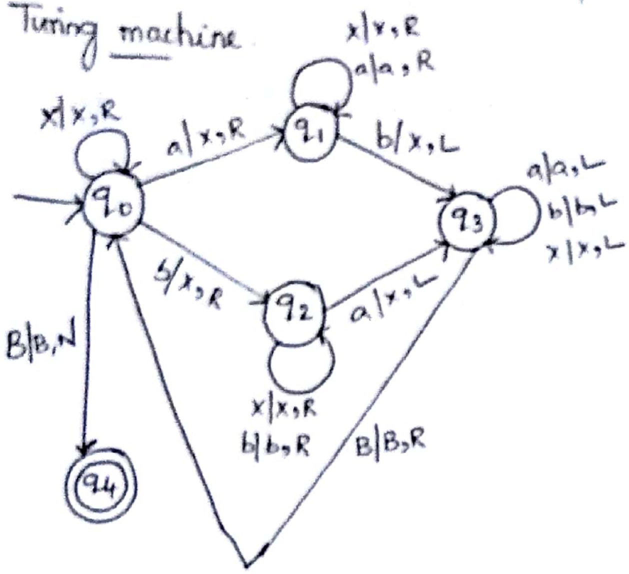
5) Design a Turing Machine to check if there are equal number of a's and b's over {a,b}

Solution:

Procedure:

1. process the first symbol.
2. If it is 'a', move right to locate the first 'b' and replace both by x.
3. If 'b' occurs, replace it by x and move forward to reach the first 'a' replace 'a' by x.
4. perform step 2 and 3 until all inputs are processed.

Turing machine



The turing machine, M is given by

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Where,

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, x, B\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_0\}$

Transition function δ is given by.

δ \ $\Sigma \cup \Gamma$	a	b	x	B
$\rightarrow q_0$	(q_1, x, R)	(q_2, x, R)	(q_0, x, R)	(q_4, B, N)
q_1	(q_1, a, R)	(q_3, x, L)	(q_1, x, R)	–
q_2	(q_3, x, L)	(q_2, b, R)	(q_2, x, R)	–
q_3	(q_3, a, L)	(q_3, b, L)	(q_3, x, L)	(q_0, B, R)
$* q_4$	ϕ	ϕ	ϕ	ϕ

Instantaneous description for w=abba

$\vdash BabbaB$ ↑ q_0 $\quad \vdash BxbbaB$ ↑ q_1 $\quad \vdash BxxbaB$ ↑ q_3

$\vdash BxxbaB$ ↑ q_3 $\quad \vdash BxxbaB$ ↑ q_0 $\quad \vdash BxxbaB$ ↑ q_0

$\vdash BxxbaB$ ↑ q_0 $\quad \vdash BxxxaB$ ↑ q_2 $\quad \vdash BxxxxB$ ↑ q_3

$\vdash BxxxxB$ ↑ q_3 $\quad \vdash BxxxxB$ ↑ q_3 $\quad \vdash BxxxxB$ ↑ q_3

$\vdash BxxxxB$ ↑ q_0 .... $\vdash BxxxxB$ ↑ q_0

$\vdash BxxxxB$ ↑ q_4 (Accept).

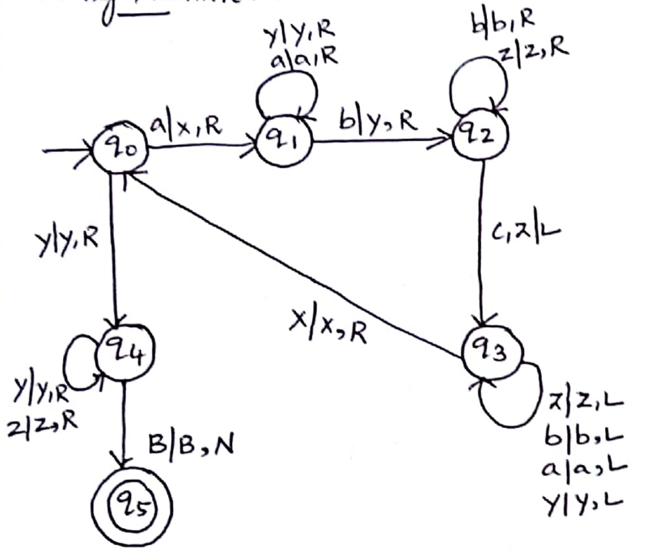
b) Design a TM that recognizes strings of the form $a^n b^n c^n \mid n \geq 1$ over $\Sigma = \{a, b, c\}$

Solution:

procedure:

1. process the leftmost 'a' and replace it by 'x'
2. Move right until the leftmost 'b' is reached. Replace it by 'y'.
3. Move right until the leftmost 'c' is reached. Replace it by 'z'.
4. Move left to reach the leftmost 'a' and perform steps 1, 2, and 3 $(n-1)$ times.
5. Halt if there are 'n' number of x, y, z.

Turing Machine:



Turing machine, M is given by,

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Where,

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, x, y, z, B\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_5\}$

Transition function δ is given by,

δ \ $\Sigma \cup \Gamma$	a	b	c	x	y	z	B
$\rightarrow q_0$	(q_1, x, R)	-	-	-	(q_4, y, R)	-	-
q_1	(q_1, a, R)	(q_2, y, R)	-	-	(q_1, y, R)	-	-
q_2	-	(q_2, b, R)	(q_3, z, L)	-	-	(q_2, z, R)	-
q_3	(q_3, a, L)	(q_3, b, L)	-	(q_0, x, R)	(q_3, y, L)	(q_3, z, L)	-
q_4	-	-	-	-	(q_4, y, R)	(q_4, z, R)	(q_5, B, N)
$*q_5$	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

. Instantaneous Description for $w = a^2 b^2 c^2$

$w = aabbcc$

$\vdash B\underset{q_0}{\underset{\uparrow}{a}}abbcc$ $\vdash Bx\underset{q_1}{\underset{\uparrow}{a}}bbccB$

$\vdash Bxa\underset{q_1}{\underset{\uparrow}{b}}bccB$ $\vdash Bxay\underset{q_2}{\underset{\uparrow}{b}}ccB$

$\vdash Bxayb\underset{q_2}{\underset{\uparrow}{c}}cB$ $\vdash Bxay\underset{q_3}{\underset{\uparrow}{b}}zcB$

$\vdash Bxay\underset{q_3}{\underset{\uparrow}{b}}zcB$ $\vdash Bxa\underset{q_3}{\underset{\uparrow}{y}}bzcB$

$\vdash B\underset{q_3}{\underset{\uparrow}{x}}aybzcB$ $\vdash Bx\underset{q_0}{\underset{\uparrow}{a}}ybzcB$ $\vdash Bxx\underset{q_1}{\underset{\uparrow}{y}}bzcB$ $\vdash Bxxy\underset{q_1}{\underset{\uparrow}{b}}zcB$ $\vdash Bxxyy\underset{q_2}{\underset{\uparrow}{z}}cB$

$\vdash Bxxyyz\underset{q_2}{\underset{\uparrow}{c}}B$ $\vdash Bxxyy\underset{q_3}{\underset{\uparrow}{z}}zB$ $\vdash Bxx\underset{q_0}{\underset{\uparrow}{y}}yzzB$ $\vdash Bxxy\underset{q_4}{\underset{\uparrow}{y}}zzB$ $\vdash Bxxyyzz\underset{q_4}{\underset{\uparrow}{B}}$ $\vdash Bxxyyzz\underset{q_5}{\underset{\uparrow}{B}}$

(Accept)

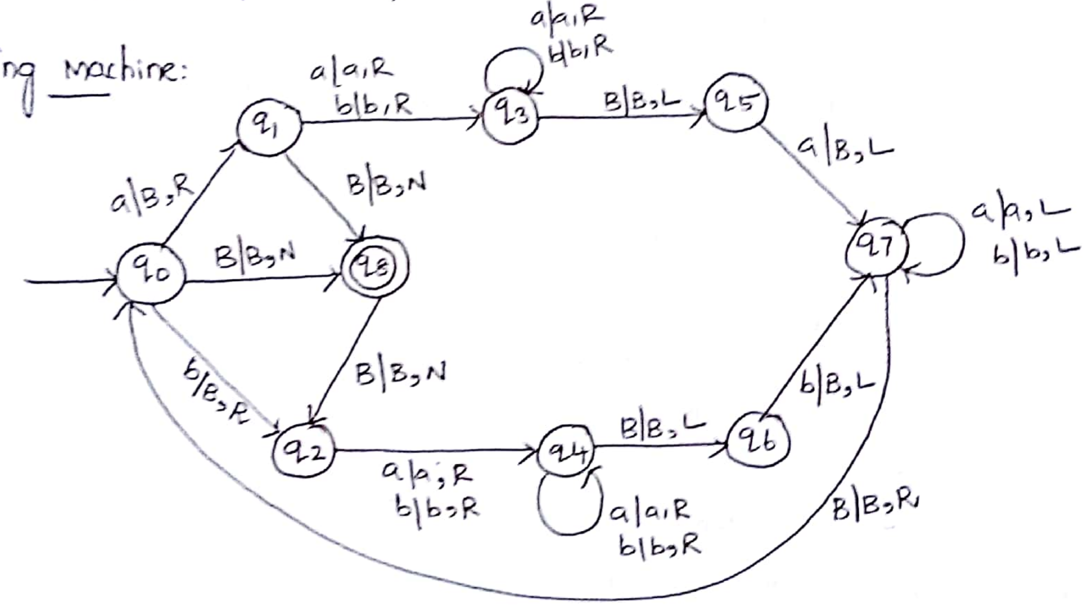
7) Design a TM which recognizes palindrome over $\Sigma = \{a, b\}$ (6)

Solution:

Procedure:

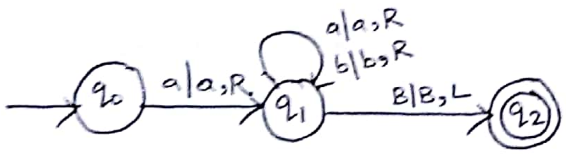
1. If there is no input, reach the final state and Halt.
2. If the input = 'a', then traverse forward to process the last symbol = 'a'. Convert both a's to B.
3. Move left to read the next symbol.
4. If the input = 'b', replace it by B and move right to process its equivalent 'B' at the rightmost end.
5. Convert the last 'b' to 'B'.
6. Move left and process steps 2-5 until there are no more input to process.

Turing Machine:



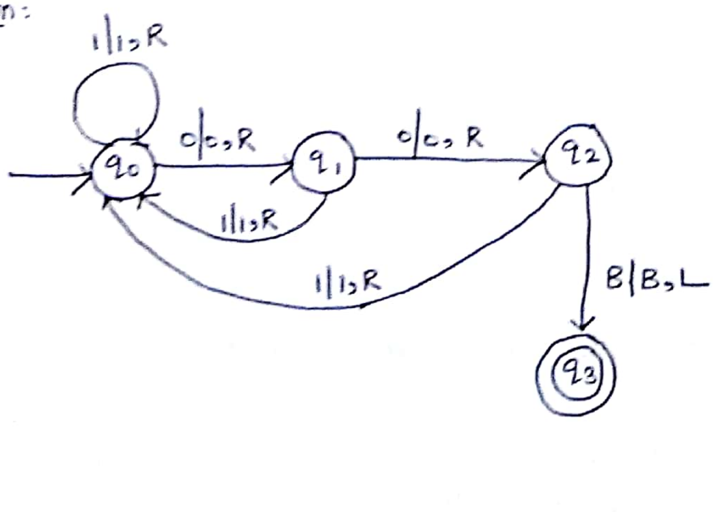
8) Design a TM that accepts $a(a+b)^*$ over $\Sigma = \{a, b\}$

Solution:



9) Construct a TM for $L = \{x \in (0,1)^* \mid x \text{ ends in } 00\}$

Solution:

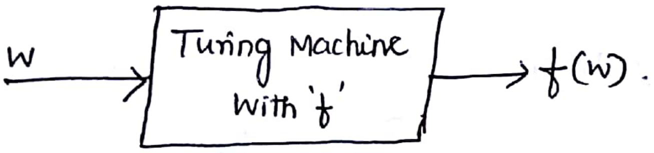


Turing Machines for Computing functions. (Transducer)

Turing machine is capable of performing several operations / functions. It is capable of performing computations such as,

- Addition
- Subtraction
- Multiplication
- Division
- 1's and 2's complementation.
- Comparing two numbers.
- Squaring a number
- GCD of numbers
- finding binary equivalent.

A Turing machine M can compute a function, f from the subset of Σ^*. The machine starts the computation corresponding to the input string, w in IT domain of 'f' and halts with the output string f(w).



When 'w' is an input string that cannot be computed by Turing machine, M with function f, then the TM should not accept W.

Representation:

The function f is Turing computable by the machine,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

with the transition function δ of the form,

$$(q_0, w) \overset{*}{\underset{M}{\vdash}} (q_f, f(w))$$

where
- $q_0 \rightarrow$ initial state ($q_0 \in Q$)
- $w \rightarrow$ Input string, where $w \in \Sigma$
- $q_f \rightarrow$ Final state ($q_f \in F$)
- $f(w) \rightarrow$ output string after computation.

For numerical function $w = 0^n$

$$f(w) = 0^{f(n)}$$

Problem:

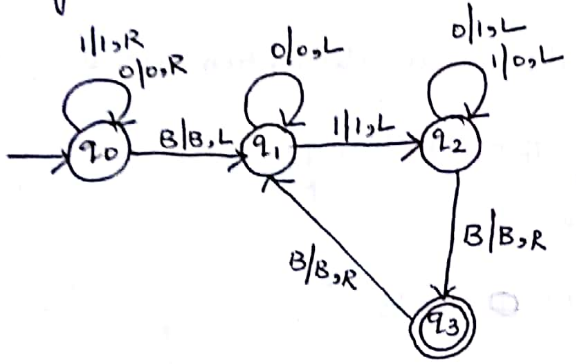
1) Design a TM to perform 2's complement of a number over $\Sigma = \{0, 1\}$

Solution:

Procedure:

1. Traverse right and locate the rightmost bit.
2. If the bit = '0', perform no replacement and move left
3. If the bit = '1', perform no change and move left.
4. If the next bit symbol = '0', replace it by '1' and move left.
5. Else if the next bit = '1', replace it by '0' and move left.
6. perform Step 4 and 5 until all the I/p symbols are processed.
7. Halt the machine.

Turing Machine:



Turing machine, M is given by,

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Where,

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, B\}$

$q_0 = \{q_0\}$

$B = \{B\}$

$F = \{q_3\}$

Transition table, δ

Q \ $\Sigma \cup \Gamma$	0	1	B
$\rightarrow q_0$	$(q_0, 0, R)$	$(q_0, 1, R)$	(q_1, B, L)
q_1	$(q_1, 0, L)$	$(q_2, 1, L)$	(q_3, B, R)
q_2	$(q_2, 1, L)$	$(q_2, 0, L)$	(q_3, B, R)
$* q_3$	ϕ	ϕ	ϕ

Instantaneous description for $w = 10100$

$\vdash \underset{q_0}{\uparrow}$ B10100B $\vdash$ B10100B ($\uparrow q_0$) $\vdash$ B10100B ($\uparrow q_0$)

$\vdash$ B10100B ($\uparrow q_0$) $\vdash$ B10100B ($\uparrow q_0$) $\vdash$ B10100B ($\uparrow q_0$)

$\vdash$ B10100B ($\uparrow q_1$) $\vdash$ B10100B ($\uparrow q_1$) $\vdash$ B10100B ($\uparrow q_1$)

$\vdash$ B10100B ($\uparrow q_2$) $\vdash$ B11100B ($\uparrow q_2$) $\vdash$ B01100B ($\uparrow q_2$)

$\vdash$ B01100B ($\uparrow q_3$) (Halt).

2's complement of $w = 10100$

10100

1's ⇒ 01011

$\quad\quad\quad\;\;$ 1

2's ⇒ 01100

2) Design a TM to compute addition of two unary numbers.

Solution:

Procedure:

1. Read the symbols of the first input with no replacement and move right.
2. When the symbol = '#' replace it by '0' and move right.
3. Traverse right side until the rightmost '0'
4. Replace the rightmost '0' by 'B'.
5. Stop the machine.

Turing machine, M is given by

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

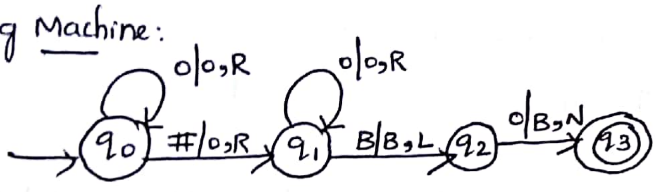
where, $Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, \#\}$

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

$q_0 = \{q_0\}$, $B = \{B\}$, $F = \{q_3\}$.

Turing Machine:



Transition table δ,

Q \ $\Sigma \cup \Gamma$	0	#	B
$\rightarrow q_0$	$(q_0, 0, R)$	$(q_0, 0, R)$	–
q_1	$(q_1, 0, R)$	–	(q_2, B, L)
q_2	(q_3, B, N)	–	–
$* q_3$	ϕ	ϕ	ϕ

Instantaneous description $w = 2, 3$, $2+3=5$

$\vdash B00\#000B$ ($\uparrow q_0$) $\vdash B00\#000B$ ($\uparrow q_0$) $\vdash B00\#000B$ ($\uparrow q_0$)

$\vdash B00\#000B$ ($\uparrow q_0$) $\vdash B000000B$ ($\uparrow q_1$) $\vdash B000000B$ ($\uparrow q_1$)

$\vdash B000000B$ ($\uparrow q_1$) $\vdash B000000B$ ($\uparrow q_2$) $\vdash B00000BB$ ($\uparrow q_3$)

$[2+3 = 00 + 000 \Rightarrow 00000]$ (Halt)

3) Design a TM to compute subtraction of two unary numbers given by,

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

Solution:

Procedure:

1. Replace the leftmost '0' by B and move right
2. Replace the rightmost '0' by B and move left
3. perform steps 1 and 2, (n-1) times
4. Halts the machine since remaining '0' by B and halt.

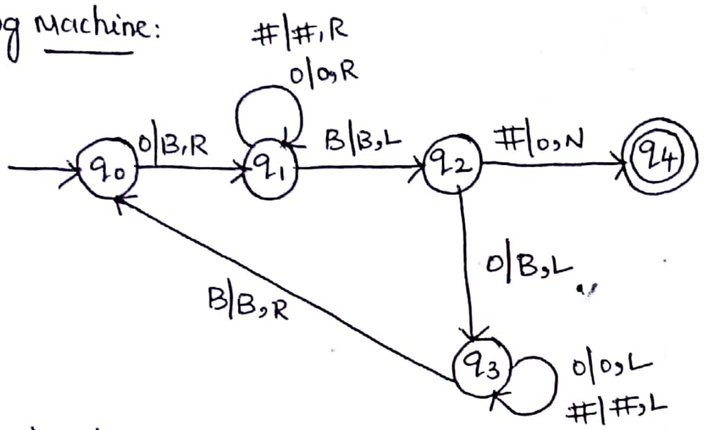
Scanned by CamScanner

Solution:

procedure:

1. Replace the leftmost '0' by B and move right.
2. Replace the rightmost '0' by B and move left.
3. perform steps 1 and 2, (n-1) times.
4. Halt the machine since remaining '0' by B and halt.

Turing Machine:



Turing Machine, M is given by,
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Where $Q = \{q_0, q_1, q_2, q_3, q_4\}$
$\Sigma = \{0, \#\}$
$\Gamma = \{0, \#, B\}$.
$q_0 = \{q_0\}$
$B = \{B\}$
$F = \{q_4\}$

δ is given by,

$\Sigma \cup \Gamma$ / Q	0	#	B
$\rightarrow q_0$	(q_1, B, R)	-	-
q_1	$(q_1, 0, R)$	$(q_1, \#, R)$	(q_2, B, L)
q_2	(q_3, B, L)	$(q_4, 0, N)$	-
q_3	$(q_3, 0, L)$	$(q_3, \#, L)$	(q_0, B, R)
$*q_4$	ϕ	ϕ	ϕ

Instantaneous description W = 000#0 [3-1]

$\vdash$ B000#0B (q_0 at first 0) $\vdash$ BB00#0B (q_1) $\vdash$ BB00#0B (q_1)

$\vdash$ BB00#0B (q_1) $\vdash$ BB00#0B (q_1) $\vdash$ BB00#0B (q_1)

$\vdash$ BB00#0B (q_2) $\vdash$ BB00#BB (q_3) $\vdash$ BB00#BB (q_3)

$\vdash$ BB00#BB (q_3) $\vdash$ BB00#BB (q_0) $\vdash$ BBB0#BB (q_1) $\vdash$ BBB0#BB (q_1)

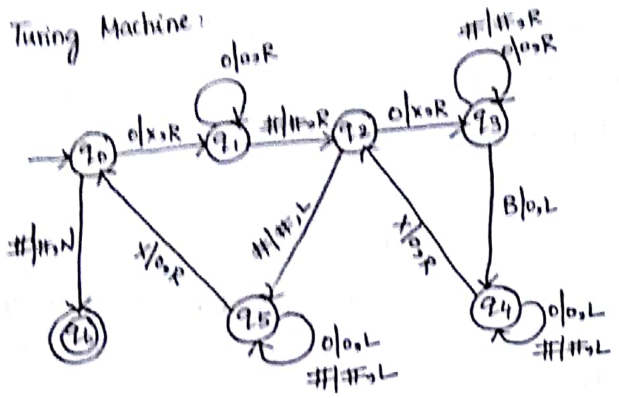
$\vdash$ BBB0#BB (q_2) $\vdash$ BBB00BB (q_4) Accept.

4) Compute a TM that performs multiplication of two unary numbers.

Solution: procedure:

1. Read the leftmost '0' replace it by 'x' and move right to process the immediate symbol after '#'.
2. Replace the symbol '0' by x and move right reach the first 'B' after '#'.
3. Replace 'B' by '0' and move left until the nearest 'X' is reached.
4. Replace 'x' by '0' and move right to process the next symbol of the multiplicand.
5. Perform Step 2, 3 and 4 until all the symbols of the multiplicand are processed.
6. Move left to replace the symbol of the multiplier 'x' by '0'
7. Perform steps 1 to 6 until all the symbols of the multiplier are processed.

Turing Machine:



Turing machine M is given by
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
Where,
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
$\Sigma = \{0, \#\}$
$\Gamma = \{0, \#, x, B\}$
$q_0 = \{q_0\}$
$B = \{B\}$
$F = \{q_6\}$

Transition function δ is given by,

	0	#	X	B
q_0	(q_1, x, R)	$(q_6, \#, N)$	-	-
q_1	$(q_1, 0, R)$	$(q_2, \#, R)$	-	-
q_2	(q_3, X, R)	$(q_5, \#, L)$	-	-
q_3	$(q_3, 0, R)$	$(q_3, \#, R)$	-	$(q_4, 0, L)$
q_4	$(q_4, 0, L)$	$(q_4, \#, L)$	$(q_2, 0, R)$	-
q_5	$(q_5, 0, L)$	$(q_5, \#, L)$	$(q_0, 0, R)$	-
q_6	ϕ	ϕ	ϕ	ϕ

Example: Instantaneous description of $W = 2 \times 2 = 4 \Rightarrow$ B00#00#BB.

⊢ B00#00#B (q_0) ⊢ BX0#00#B (q_1) ⊢ BX0#00#B (q_1) ⊢ BX0#00#B (q_2)

⊢ BX0#X0#B (q_3) ⊢ BX0#X0#B (q_3) ⊢ BX0#X0#B (q_3) ⊢ BX0#X0#0B (q_4)

⊢ BX0#X0#0B (q_4) ⊢ BX0#X0#0B (q_4) ⊢ BX0#00#0B (q_2) ⊢ BX0#0X#0B (q_3)

⊢ BX0#0X#0B (q_3) ⊢ BX0#X0#0B (q_3) ⊢ BX0#X0#00B (q_4) ⊢ BX0#0X#00B (q_4)

⋮

⊢ B00#00#$\underbrace{0000}$B (q_6)

Result of 2x2.

5) Design a TM that perform multiplication operation using "copy" subroutine.

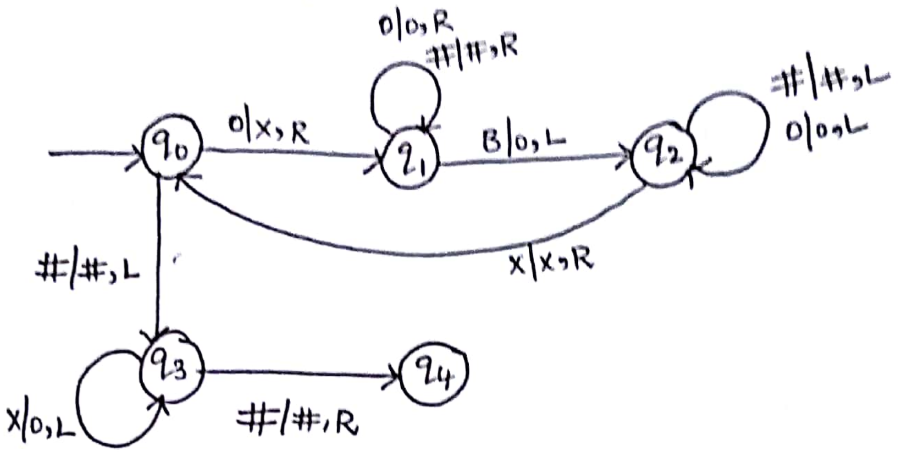
Solution:

Let the inputs be 0^x and 0^y

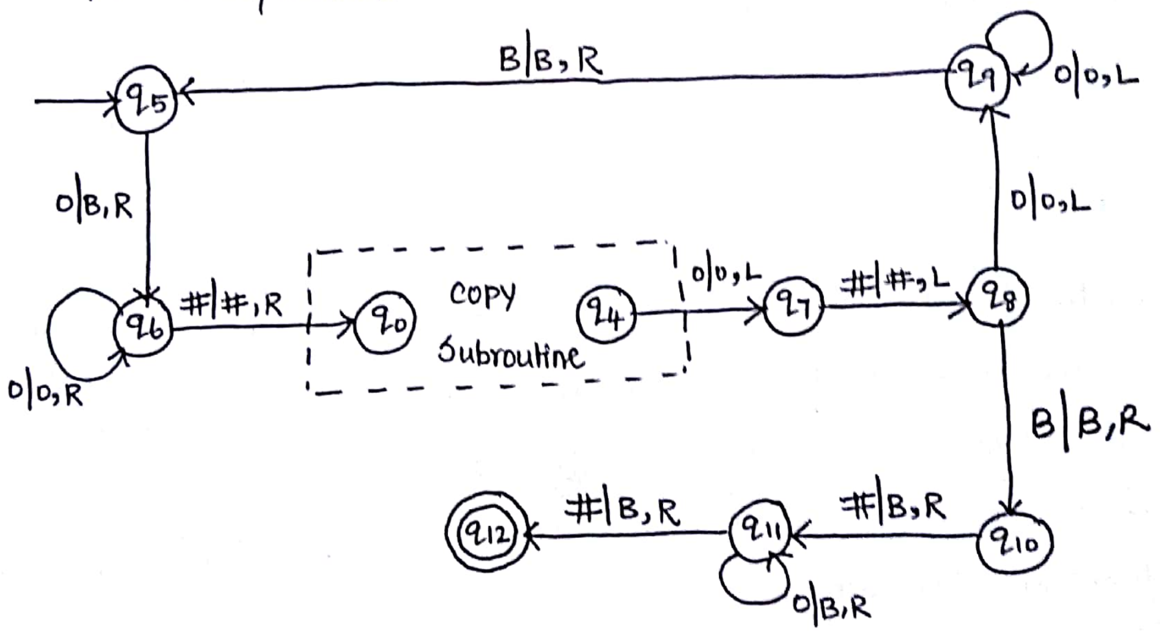
The inputs are stored as $B0^x \# 0^y \# BB$.

Turing Machine:

Subroutine for copy operation.



TM for Multiplication:



Turing Machine constructions

A turing machine is also as powerful as a conventional computer. The following are the different technique of constructing ~~an~~ TM.

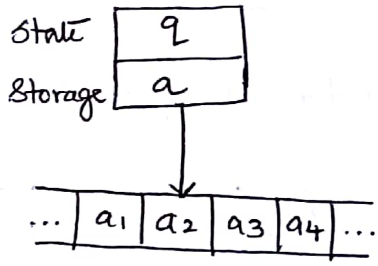
1. Storage in the finite control (or) State
2. Multiple tracks
3. Subroutines
4. Checking off symbols.

Storage in The finite control:

⇒ The states of turing machine are to remember / store a symbol. This is done by the finite control.

⇒ The finite control can also be used to hold a finite amount of information along with the task of representing a position in The program.

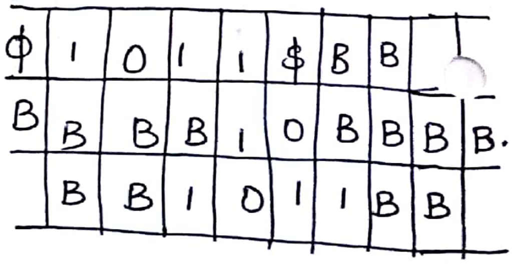
⇒ The state is written as a pair of elements, one for control and the other storing a symbol.



Multiple tracks:

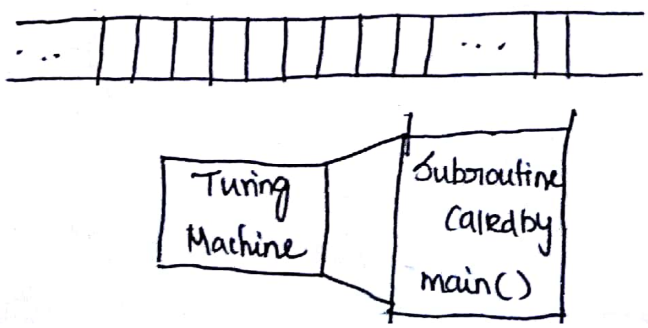
⇒ It is also possible that a turing machine, input tape can be divided into several tracks.

⇒ Each track can hold one symbol, and the tape alphabet of the TM consists of tuples with One component for each track.



Subroutines:

⇒ Subroutines are subfunctions that can be used to execute repeated tasks for number of times depending on The application.



⇒ In such case, the turing machine has to be designed that handles subroutines.

⇒ The subroutine has two states (i) Initial state (ii) Return state.

⇒ When the main function is executed, the subroutine is called. The TM reaches the initial state and follows a series of execution using the transition rules of the subroutine.

⇒ After processing, the TM reaches the return state that returns back to the main function after a temporary halt.

Checking off symbols:

⇒ It is a useful technique for visualization how a TM recognizes languages defined by repeated string such as.

$$L = \{ww \mid w \text{ in } \Sigma^*\}$$

⇒ Here we use one track of the tape to mark the some symbols have been read without changing them.

Ex: $W = abb$

1^{st} track	a	a	b	#	a	a	b	#	← Input symbol
2^{nd} track	✓								← Status

↑ checking off symbol.

Definition: Multitape turing machine.

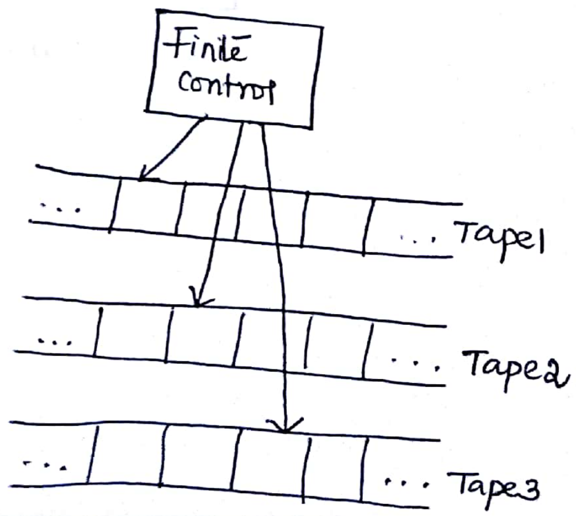
A multitape turning has a finite control with some finite number of tape. Each tape is infinite in both directions. It has its own initial state and some accepting states.

Initially,

⇒ Finite set of input symbols is placed on the first tape.

⇒ All the other cells of all the tapes hold the blank.

⇒ The control head of 1^{st} tape is at the left end of the input.



In one move, the multitape TM can

- change state
- Print a new symbol on each of the cells scanned by its tape heads.
- Move each of its tape heads, independently, one cell to the left or right or keep it stationary.

Definition : Non-deterministic TM.

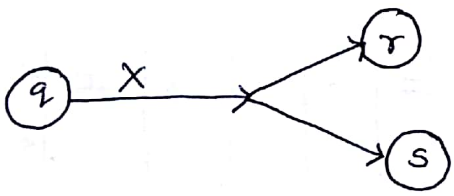
A non-deterministic turing machine is a device with a finite control and a single one-way infinite tape. For a given state and a tape symbol scanned by the tape head, the machine has a finite number of choices for the next move

For $\delta(q, x)$, there is a set of triples like.

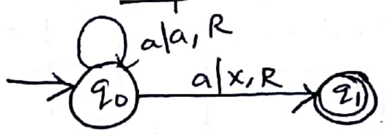
$$\{(q_1, Y_1, D_1), (q_2, Y_2, D_2), \ldots (q_K, Y_K, D_K)\}$$

where K is an integer.

M accepts an input 'w' if there is any sequence of choices of move that leads from the initial ID with w as input, to an ID with an accepting state.



Example :



Problems : (Subroutine)

Design a TM to implement the function "multiplication". and TM will start with $0^m 1 0^n$ on its tape and will end with 0^{mn} on the tape.

Solution : [Similar problem of multiplication with copy subroutine]

Problem (Multiple track)

Design the TM to check whether the given input is prime or not using multiple tracks.

Solution:

Procedure

1) The input greater than 2 is placed on the 1st track and also the same input is placed on the 3rd track.
2) The TM write a number 2 on the 2nd track.
3) The number on the 2nd track is subtracted from the 3rd track as many as possible till getting the remainder.
4) If the remainder is zero and the number on the 1st track is not a prime.
5) If the remainder is non-zero then increase the number on the 2nd track by 1.
6) If the 2nd track equal to the 1st track then the number is prime, because it should be divided by itself.

Example: consider the number 8.

1st	8	8	8	8	8	...
2nd	2	2	2	2	2	
3rd	8	6	4	2	0	⇒ The number is not prime.

Consider the number = 7.

Level 1:

1st	7	7	7	7
2nd	2	2	2	2
3rd	7	5	3	1

Level 2:

1st	7	7	7
2nd	3	3	3
3rd	7	4	1

Level 3:

1st	7	7
2nd	4	4
3rd	7	3

Level 4:

1st	7	7
2nd	5	5
3rd	7	2

Level 5:

1st	7	7
2nd	6	6
3rd	7	1

Level 6:

7
7
→ 7

The given number 7 is prime.