CS301
Theory of Computation
Module-5

★ ★ ★ ★ ★ ★ ★ ★ ★

Contents :

* Pumping Lemma for CFL
* Applications of pumping lemma.
* Type1 Formalism : Context-Sensitive Grammar.
  Linear - Bounded Automata [Design Not Req:]
* Type 0 Formalism : Turing Machine (TM) - Basics and formal definition,
* TMs as language acceptors,
* TMs as Transducers,
* Designing Turing Machines.

Prepared by,
Tintu Alphonsa Thomas
Asst. Professor CSE
Amal Jyothi College of Engg
Kanjirappally.

Pumping Lemma for Context free Language.

Let L be a Context free Language, then there exists a positive constant 'n' such that for all x in L with $|x| \geq n$, it is possible to find strings u, v, w, y, and z satisfying the conditions.

$$x = uvwyz$$

$$|vwy| \leq n$$

$$|vy| > 0$$

and for all non negative integers t

$$uv^t wy^t z \in L$$

Application of pumping Lemma:

Pumping Lemma can be used to prove that certain Languages are not Context free.

To prove that a language is not context free using pumping Lemma (for CFL) follow the steps.

Proof is done using the method of <u>Contradiction</u>

→ Assume that L is context free language

→ It has to have a pumping length (say n)

→ All strings longer than 'n' can be pumped $|x| \geq n$

→ Find a string x in L such that $|x| \geq n$

→ Divide x into "uvwyz"

→ Show that $uv^t wy^t z \notin L$ for some 't'

→ Then consider the ways that x can be divided into uvwyz

→ Show that none of these can satisfy all the 3 pumping conditions at the same time.

→ x cannot be pumped = CONTRADICTION

(1) $uv^t wy^t z$ is in L for every $t \geq 0$

(2) $|vy| > 0$

(3) $|vwy| \leq n$

Show that $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context-Free

$\boxed{\text{Step 1:}}$

Assume that L is context free language. L must have a pumping length (say P).

Now for all x in L with $|x| \geq P$, it is possible to find strings u, v, w, y and z satisfying the conditions

$$x = uvwyz$$

$$|vwy| \leq P$$

$$|vy| > 0$$

and for all non negative integers 't'

$$uv^t w y^t z \in L$$

$\boxed{\text{Step2:}}$ We can divide x into parts u, v, w, y, z

consider $P = 10$ so $x = a^4 b^4 c^4$

case 1: v and y each contain only one type of symbol.

$$\underbrace{a}_{u}\underbrace{aa}_{v}\underbrace{abbbbc}_{w}\underbrace{c}_{y}\underbrace{cc}_{z}$$

$uv^t w y^t z$ $(t=2)$

ie, $uv^2 w y^2 z$

∴ aaaaaa bbbb ccccc

$a^6 b^4 c^5 \notin L$

case 2: Either v or y has more than one kind of symbols

$$\underbrace{aa}_{u}\underbrace{aabb}_{v}\underbrace{b}_{w}\underbrace{b}_{y}\underbrace{cccc}_{z}$$

$uv^t w y^t z$ $(t=2)$

ie, $uv^2 w y^2 z$

∴ aa aabb aabb b bb cccc

$\notin L$

∴ It is a contradiction for our assumption that L is context-Free

∴ L is not context-Free language

Show that $L = \{ww \mid w \in \{0,1\}^*\}$ is not context-free

Step 1:

Assume that L is context free and L have a pumping length (say P). Now we can take a string $x \in L$ such that $x = 0^P 1^P 0^P 1^P$ and $|x| > P$

Step 2:

Divide x into parts u, v, w, y, z Eg: P=5 $x = 0^5 1^5 0^5 1^5$

Case 1: vwy does not cross a boundary

B_1, B_2, B_3 Boundary

$$\underbrace{00000\,1}_{u}\,\underbrace{1}_{v}\,\underbrace{1}_{w}\,\underbrace{1}_{y}\,\underbrace{1\,00000\,11111}_{z}$$

$uv^t wy^t z$

Now t=2 $\therefore uv^2wy^2z = 000001111111000001111$

$= \boxed{0^5 1^7 0^5 1^5 \notin L}$

Case 2a: vwy crosses the first boundary

$$\underbrace{000}_{u}\,\underbrace{00}_{v}\,\underbrace{1}_{w}\,\underbrace{11}_{y}\,\underbrace{11\,00000\,11111}_{z}$$

$uv^t wy^t z$

Now t=2 $\therefore uv^2wy^2z = 000\,0000111111100000111111$

$\boxed{0^6 0^7 0^5 1^5 \notin L}$

Case 2b: vwy crosses the third boundary

$$\underbrace{00000\,11111\,000}_{u}\,\underbrace{00}_{v}\,\underbrace{1}_{w}\,\underbrace{11}_{y}\,\underbrace{11}_{z}$$

$uv^t wy^t z$

Now t=2 $\therefore uv^2wy^2z = 0000011111000000011111111$

$\boxed{0^5 1^5 0^7 1^7 \notin L}$

Case 3: vwy crosses the mid-point of the string

$$\underbrace{0000011}_{u}\,\underbrace{11}_{v}\,\underbrace{1}_{w}\,\underbrace{00}_{y}\,\underbrace{000\,11111}_{z}$$

$uv^t wy^t z$

Now t=2 $\therefore uv^2wy^2z = 000001111111000000011111$

$\boxed{0^5 1^7 0^7 1^5 \notin L}$

It is a contradiction to our assumption

$\therefore$ L is not context-Free

Type 1 Formalism - Context Sensitive Grammar

Context sensitive grammar is an unrestricted grammar in which all the productions are of form

$$\alpha \longrightarrow \beta$$

where $\alpha, \beta \in (N \cup T)^*$ and $|\alpha| \le |\beta|$

α, β are strings of non-terminals and terminals

Context sensitive grammars are more powerful than context-free grammars because there are some languages that can be described by CSG but not by Context free grammars and CSL are less powerful than unrestricted grammar.

Context - sensitive Grammar has 4 tuples

$G = (N, T, S, P)$ where.

N - Set of Non terminal

T - Set of Terminals (Σ)

S - Start symbol

P - Finite set of Productions

All the rules in P are of the form

$$\alpha_1 A \alpha_2 \longrightarrow \alpha_1 \beta \alpha_2$$

A is replaced by B based on context of α_1 & α_2

Context Sensitive language

The language that can be defined by context sensitive grammar is called CSL. Properties of CSL are

* Union, Intersection and Concatenation of two CSL is context-sensitive
* Complement of a context-sensitive language is context-sensitive.

Example.

$S \rightarrow abc \mid aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aa \mid aaA$

The language generated by the grammar.

$S \Rightarrow aAbc$

$\Rightarrow abAc$

$\Rightarrow abBbcc$

$\Rightarrow aBbbcc$

$\Rightarrow aabbcc$

The langage generated by grammar is

$\{a^n b^n c^n \mid n \geq 1\}$

Linear Bounded Automata

Linear Bounded Automata has finite amount of memory called tape which can be used to recognize Context Sensitive Languages.
LBA is more powerful than Push down Automata

FA < PDA < LBA < TM.

The input tape is restricted in size. A linear function is used for restricting the length of the input tape. Length of the tape is limited to the length of the input string. The input tape is linearly bounded with end marker. If the tape head reaches the end marker, then the machine stops.

Note:

Many compiler languages lie between context - sensitive and context free languages.

A LBA is a non-deterministic turing machine which has a single tape whose length is not infinite but bounded by end markers.

Formal definition of LBA.

$$\boxed{M = (Q, \Sigma, \Gamma, \delta, q_0, \#, <, >, A)}$$

Q - set of states

Σ - set of input alphabets

Γ - set of tape symbols, including $\#$

δ - transition function

$$(Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{L, R, N\})$$

L-Left
R-Right
N-No change

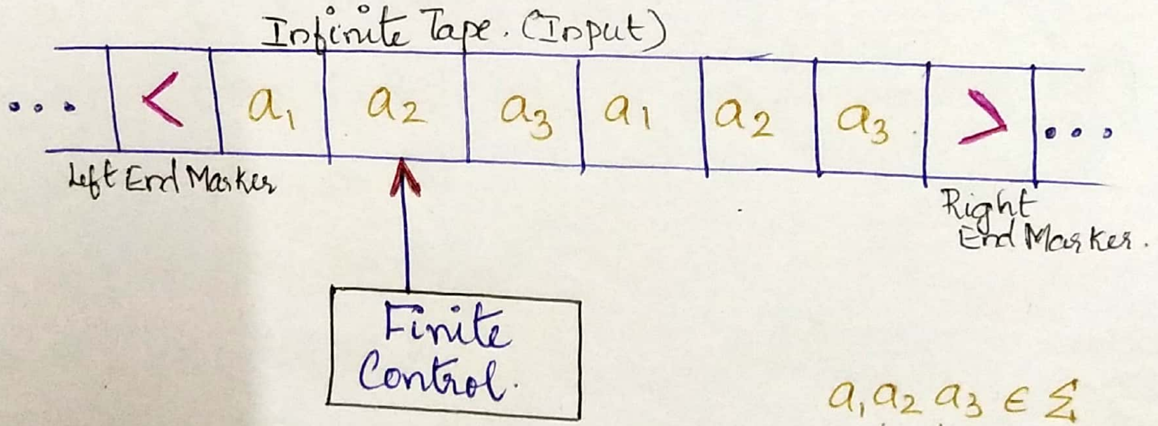
q_0 - start state

$\#$ - blank symbol

$<$ - left end marker in the input tape

$>$ - right end marker in the input tape

A - Accepting / Final state.



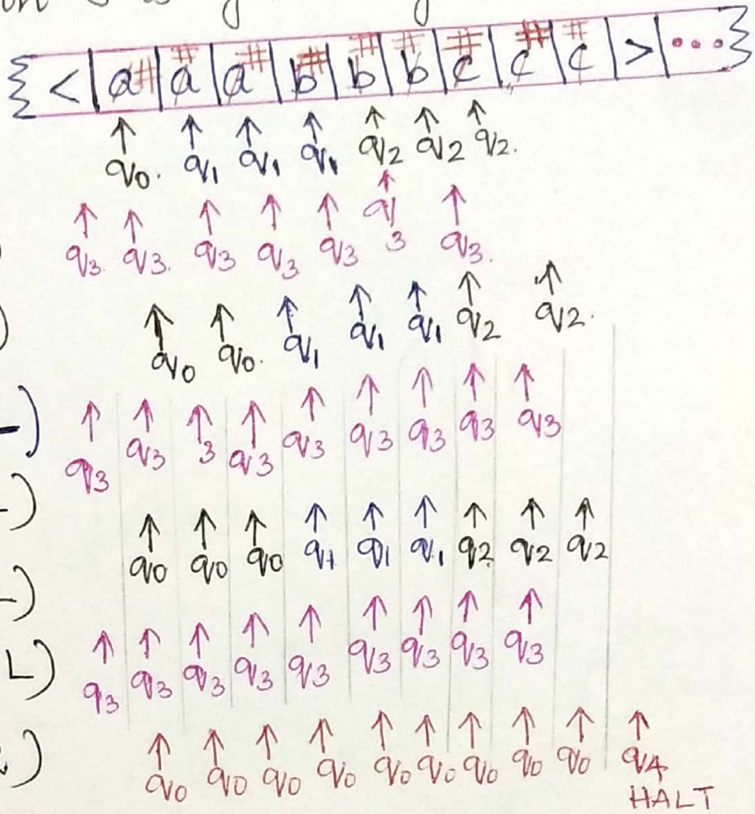
$a_1, a_2, a_3 \in \Sigma$

Design an LBA for the language

$L = \{a^n b^n c^n \mid n \geq 1\}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, <, >, \#, A)$

The transition Function δ is given by.



$\{ \delta(q_0, a) = (q_1, \#, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, \#, R)$

$\delta(q_2, b) = (q_2, b, R)$

$\delta(q_2, c) = (q_3, \#, L)$

$\delta(q_3, b) = (q_3, b, L)$

$\delta(q_3, \#) = (q_3, \#, L)$

$\delta(q_3, a) = (q_3, a, L)$

$\delta(q_3, <) = (q_0, <, R)$

$\delta(q_0, \#) = (q_0, \#, R)$

$\delta(q_1, \#) = (q_1, \#, R)$

$\delta(q_2, \#) = (q_2, \#, R)$

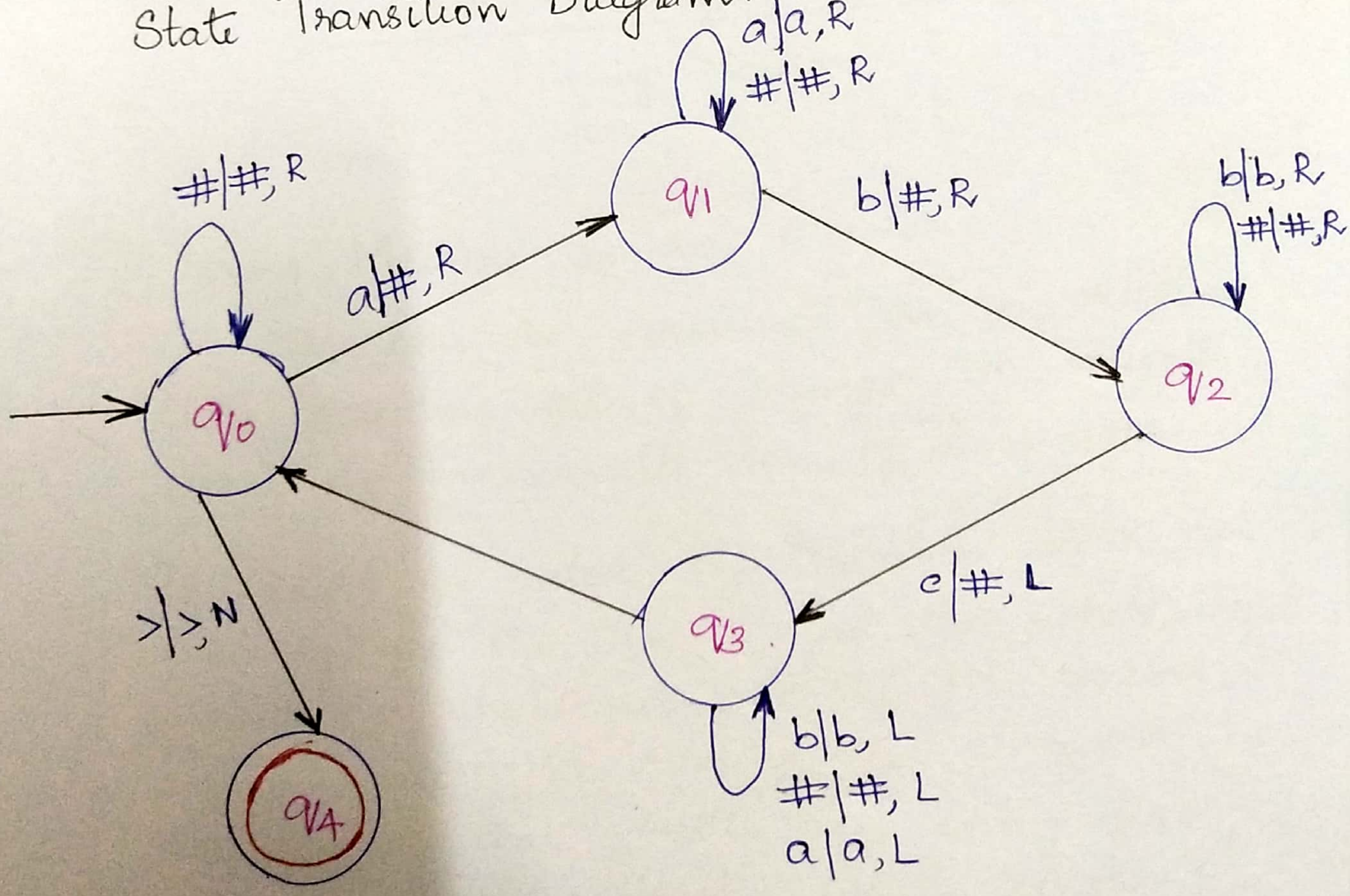
$\delta(q_0, >) = (q_4, >, N) \}$.

HALT

State Transition Table

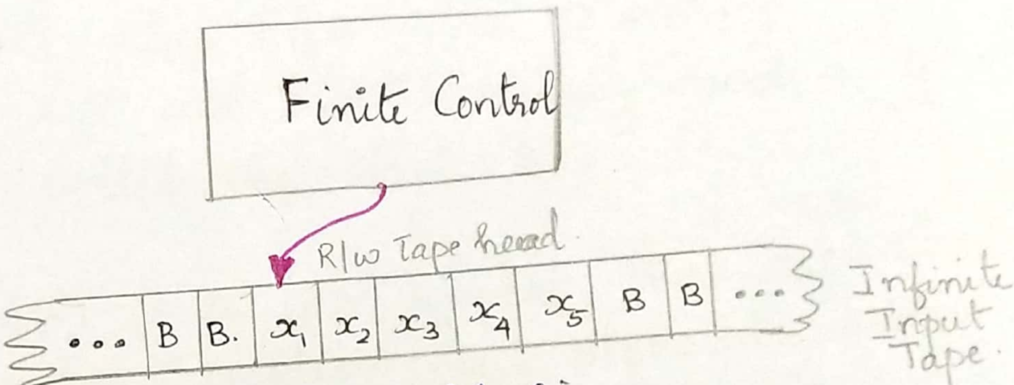
Q \ Γ	a	b	c	<	>	#
$\rightarrow q_0$	$(q_1, \#, R)$	-	-	-	HALT $(q_4, >, N)$	$(q_0, \#, R)$
q_1	(q_1, a, R)	$(q_2, \#, R)$	-	-	-	$(q_1, \#, R)$
q_2	-	(q_2, b, R)	$(q_3, \#, L)$	-	-	$(q_2, \#, R)$
q_3	(q_3, a, L)	(q_3, b, L)	-	$(q_0, <, R)$	-	$(q_3, \#, L)$
$*q_4$	-	-	-	-	-	-

State Transition Diagram



Turing Machines

A Turing Machine is an accepting device which accepts the languages (recursively enumerable set) generated by type 0 grammars. It was invented in 1936 by Alan. Turing.



Turing Machine.

Turing Machine as an Acceptor.

TM can be considered as an accepting device accepting set of strings. TM accepts recursively enumerable set of languages.

It should be noted, by defintion, it is not necessary for the TM to read the whole input. If $\omega_1\omega_2$ is the input and the TM reaches a final state after reading ω_1. $\omega_1\omega_2$ will be accepted; for that matter any string $\omega_1\omega_j$ will be accepted. Usually while constructing a TM, we make sure that the whole of the input is read.

Notation for the Turing Machine

Turing machine consist of a finite control having finite set of states. The input tape is divided into cells each cell hold a input symbol /alphabet (Σ).

Initially the input string is placed on the tape. and all other tape cells extending infinitely to the left and right, initially hold a special symbol called the blank.(B).

Blank (B) is a tape symbol, there are other tape symbols besides input alphabets and Blank.

There is a tape head that is always positioned at one of the tape cells. The turing machine scans each cells in the tape.

In one move, the turing Machine will

1. change the state [optionally the m/e stays at current state also]

2. Write a tape symbol in the cell scanned. This tape symbol replaces whatever symbol was in that cell. [optionally the symbol written may be the same as the symbol currently there]

③ Move the tape head Left or Right.

Formal definition of Turing Machine.

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

Q : Finite set of states of the finite control

Σ : Finite set of input symbols

Γ : Tape symbols, $(\Sigma \subseteq \Gamma)$

δ : Transition Function : The arguments of δ is defined as,

$$\boxed{\delta(q, X) = (p, Y, D)}$$

$q \in Q$
$X \in \Gamma$

1. p is the next state in Q
2. Y is the symbol in Γ, written in the cell being scanned, replacing whatever symbol was there.
3. D is a direction either R/L, telling us the direction in which the head moves.

q_0 : The start state : $q_0 \in Q$

B : Blank symbol $B \in \Gamma$, $B \notin \Sigma$

F : Final or accepting states : $F \subseteq Q$

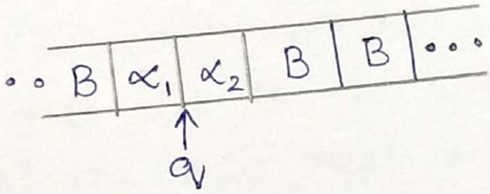
Transition Function:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, L\}$$

Instantaneous Description of a TM

An ID of a TM is a string of the form
$\boxed{\alpha_1 q \alpha_2}$, $\alpha_1, \alpha_2 \in \Gamma^*$, $q \in Q$.

This means that at that particular instance
$\alpha_1\alpha_2$ is the content of the tape of the TM.
q is the current state and the tape head
points to the first symbol of α_2.



contents of tape and head position of ID $\alpha_1 q \alpha_2$

The relationship between ID can be described

If $X_1 \ldots X_{i-1} q X_i X_{i+1} \ldots X_m$ is an ID.
and $\delta(q, X_i) = (P, Y, R)$ then the next ID is
$X_1 \ldots X_{i-1} Y p X_{i+1} \ldots X_m$

If $\delta(q, X_j) = (p, Y, L)$ then the next ID
$X_1 \ldots X_{i-2} p X_{i-1} Y X_{i+1} \ldots X_m$.

It can be denoted as.
$X_1 \ldots X_{i-1} q X_i X_{i+1} \ldots X_m \vdash X_1 \ldots X_{i-2} p X_{i-1} Y X_{i+1} \ldots X_m$

$q_0 X_1 \ldots X_m$ is the initial ID. Initially, the tape head points to the leftmost cell containing the input. If $q X_1 \ldots X_m$ is an ID and $\delta(q, X_1) = (p, Y, L)$, machine halts. ie, moving off the left end of the tape is not allowed.

* An input will be accepted if the TM reaches a final state.

* After going to a final state, TM halts, ie it makes no more moves.

* A string w will not be accepted by the TM, if it reaches an ID $\alpha_1 r \alpha_2$ from which it cannot make a next move; $\alpha_1, \alpha_2 \in \Gamma^*$ $r \in Q$ and r is not a final state or while reading w, the TM gets into a loop and is never able to halt.

Scanned by CamScanner

Type 0 Formalism: Turing Machine

Type 0 grammars are called unrestricted Grammars

A. type 0 grammar is a 4 tuple
$G = (N, T, S, P)$

N - set of Non terminals
T - set of Terminals
S - Start symbol
P - Set of production rules.

$P: \alpha \rightarrow \beta$

$\alpha \in (N \cup T)^* N (N \cup T)^*$

$\beta \in (N \cup T)^*$

The LHS of the production rule can be a string consisting of Terminals and Non-Terminals with atleast one Non-terminal on it.
They are called phrase structured Grammar.

The language generated by unrestricted grammars are called unrestricted Language.
The automata which accepts Type 0 languages are called Turing Machine.

Design a Turing Machine to accept the lang

$L = \{0^i 1^i \mid i \geq 1\}$.

$T = (Q, \Sigma, \Gamma, \delta, q_0, B, A)$ where.

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

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

$\Gamma = \{X, Y, B, 0, 1\}$.

q_0 = Start state

B = Blank symbol.

q_4 = Accepting State.

Transition Function δ:

$\delta(q_0, 0) = (q_1, X, R)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, 1) = (q_2, Y, L)$

$\delta(q_2, 0) = (q_2, 0, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_2, Y) = (q_2, Y, L)$

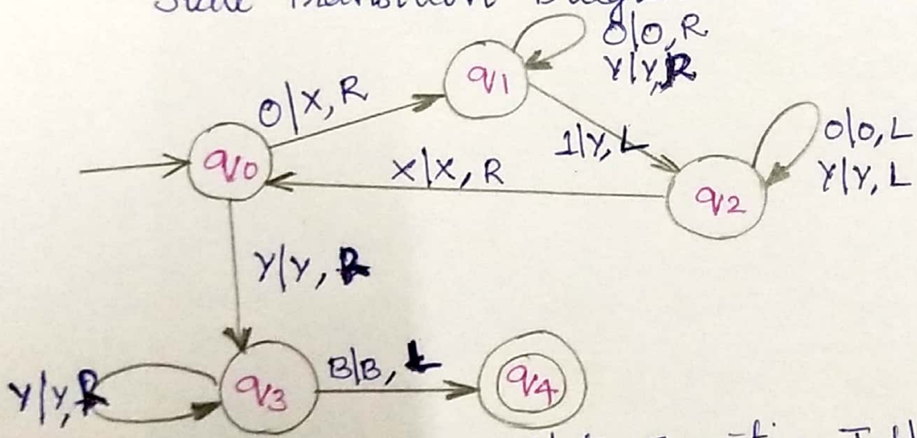
$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_0, Y) = (q_3, Y, R)$

$\delta(q_3, Y) = (q_3, Y, R)$

$\delta(q_3, B) = (q_4, B, N)$

State Transition Diagram.



State Transition Table

Q \ Γ	0	1	X	Y	B
→ q_0	(q_1, X, R)	-	-	(q_3, X, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_2, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	(q_4, B, L)
* q_4					

Computation of T on 000111.

$(q_0, \underset{\uparrow}{0}00111)$

$\vdash (q_1, X\underline{0}0111)$

$\vdash (q_1, X0\underline{0}111)$

$\vdash (q_1, X00\underline{1}11)$

$\vdash (q_2, X0\underline{0}Y11)$

$\vdash (q_2, X0\underline{0}Y11)$

$\vdash (q_2, X\underline{0}0Y11)$

$\vdash (q_2, \underline{X}00Y11)$

$\vdash (q_0, X\underline{0}0Y11)$

$\vdash (q_1, XX\underline{0}Y11)$

$\vdash (q_1, XX0\underline{Y}11)$

$\vdash (q_1, XX0Y\underline{1}1)$

$\vdash (q_2, XX0\underline{Y}Y1)$

$\vdash (q_2, XX\underline{0}YY1)$

$\vdash (q_2, X\underline{X}0YY1)$

$\vdash (q_0, XX\underline{0}YY1)$

$\vdash (q_1, XXX\underline{Y}Y1)$

$\vdash (q_1, XXXY\underline{Y}1)$

$\vdash (q_1, XXXYY\underline{1})$

$\vdash (q_2, XXXY\underline{Y}Y)$

$\vdash (q_2, XXX\underline{Y}YY)$

$\vdash (q_2, XX\underline{X}YYY)$

$\vdash (q_0, XXX\underline{Y}YY)$

$\vdash (q_3, XXXY\underline{Y}Y)$

$\vdash (q_3, XXXYY\underline{Y}B)$

$\vdash (q_3, XXXYYY\underline{B})$

$\vdash (q_4, XXXYY\underline{Y})$

$q_4 \in A$

$\therefore$ string accepted

T.M Halts

Design a TM for $L = \{a^n b^n c^n \mid n \geq 1\}$

$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

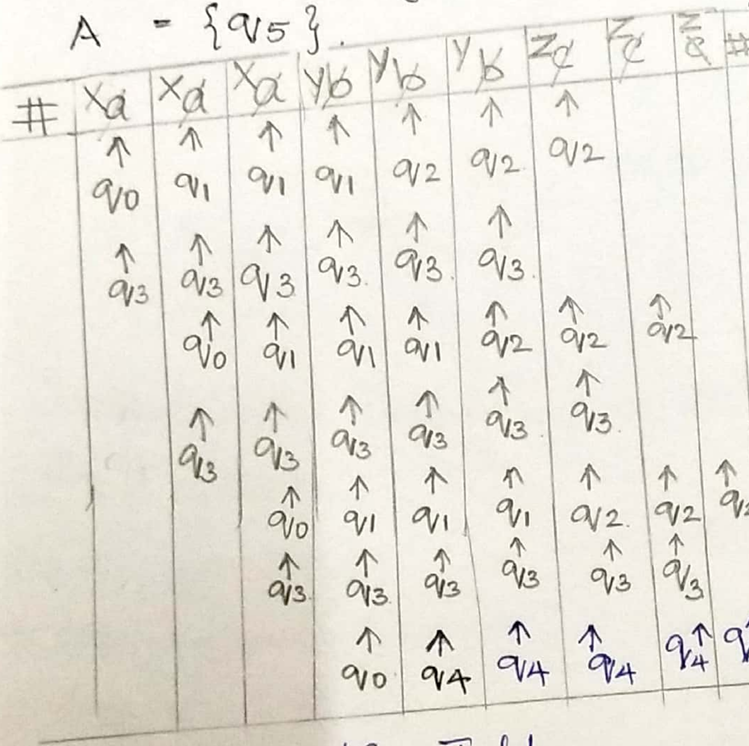
$\Sigma = \{a, b, c\}$

$\Gamma = \{\#, a, b, c, X, Y, Z\}$.

q_0 - start state

$\#$ - Blank symbol.

$A = \{q_5\}$.



Transition Function δ:

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, Y, R)$

$\delta(q_2, b) = (q_2, b, R)$

$\delta(q_2, c) = (q_3, Z, L)$

$\delta(q_3, b) = (q_3, b, L)$

$\delta(q_3, Y) = (q_3, Y, L)$

$\delta(q_3, a) = (q_3, a, L)$

$\delta(q_3, X) = (q_0, X, R)$

$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_2, Z) = (q_2, Z, R)$

$\delta(q_3, Z) = (q_3, Z, L)$

$\delta(q_0, Y) = (q_4, Y, R)$

$\delta(q_4, Y) = (q_4, Y, R)$

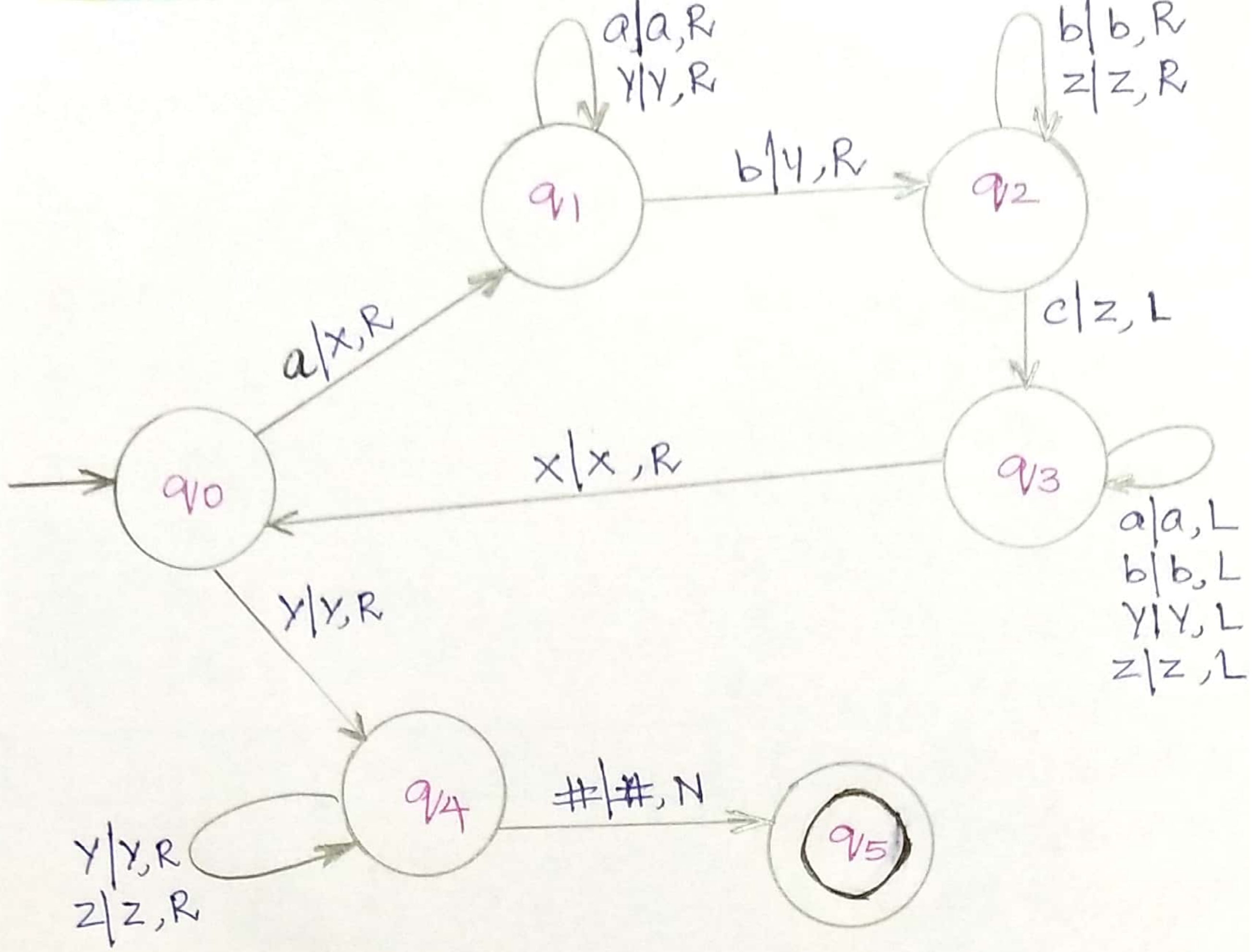
$\delta(q_4, Z) = (q_4, Z, R)$

$\delta(q_4, \#) = (q_5, \#, N)$ Halt

q_5 (HALT)

State Transition Table.

Q \ Γ	a	b	c	X	Y	Z	#
$\rightarrow q_0$	(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)	$(q_5, \#, N)$
$*q_5$							



Instantaneous Description of 'aabbcc'

$(q_0, aabbcc)$

$\vdash (q_0, aabbcc)$

$\vdash (q_0, Xabbcc)$

$\vdash (q_1, Xabbcc)$

$\vdash (q_2, XaYbcc)$

$\vdash (q_2, XaYbcc)$

$\vdash (q_3, XaYbZc)$

$\vdash (q_3, XaYbZc)$

$\vdash (q_3, XaYbZc)$

$\vdash (q_3, XaYbZc)$

$\vdash (q_0, XaYbZc)$

$\vdash (q_1, XXYbZc)$

$\vdash (q_1, XXYbZc)$

$\vdash (q_1, XXYYZc)$

$\vdash (q_2, XXYYZc)$

$\vdash (q_3, XXYYZZ)$

$\vdash (q_3, XXYYZZ)$

$\vdash (q_3, XXYYZZ)$

$\vdash (q_3, XXYYZZ)$

$\vdash (q_0, XXYYZZ)$

$\vdash (q_4, XXYYZZ)$

$\vdash (q_4, XXYYZZ)$

$\vdash (q_4, XXYYZZ)$

$\vdash (q_4, XXYYZZ\#)$

$\vdash (q_5, XXYYZZ\#)$ Halt.

Design a T.M for $L = \{a^n b^{2n} \mid n \geq 1\}$

$T = \{Q, \Sigma, \Gamma, \delta, q_0, \#, A\}$

$Q = \{$

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

$\Gamma = \{\#, a, b, X, Y\}.$

$q_0 = q_0.$

$\#$ = Blank symbol.

$A = \{q_5\}.$

Transition Fn δ:

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, Y, R)$

$\delta(q_2, b) = (q_3, Y, R)$

$\delta(q_3, b) = (q_4, b, L)$

$\delta(q_4, Y) = (q_4, Y, L)$

$\delta(q_4, a) = (q_4, a, L)$

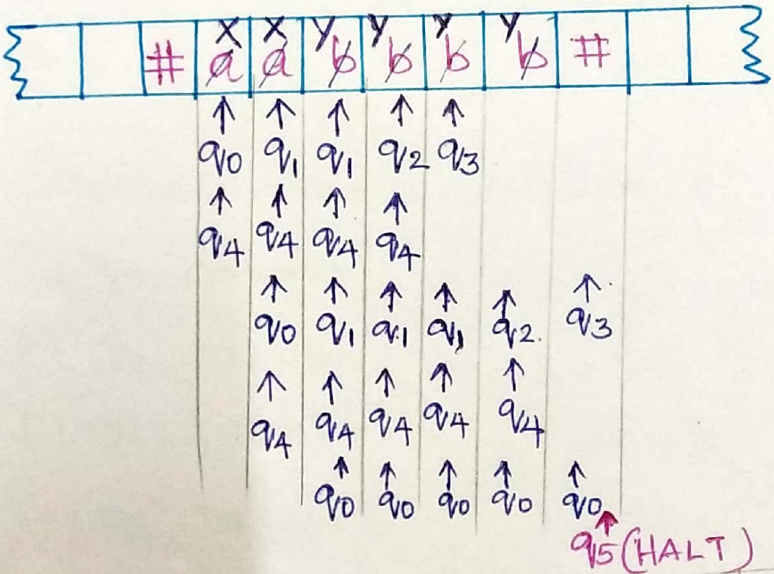
$\delta(q_4, X) = (q_0, X, R)$

$\delta(q_1, Y) = (q_1, Y, R)$

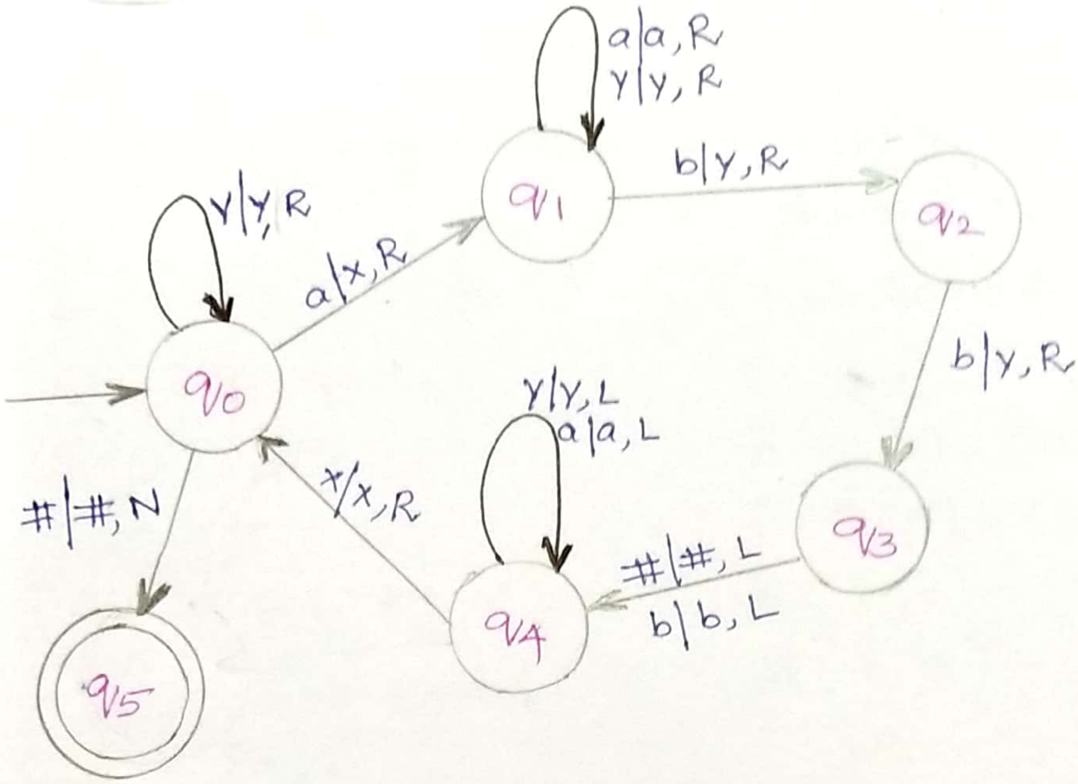
$\delta(q_3, \#) = (q_4, \#, L)$

$\delta(q_0, Y) = (q_0, Y, R)$

$\delta(q_0, \#) = (q_6, \#, N)$



Q \ Γ	a	b	X	Y	#
→ q_0	(q_1, X, R)			(q_0, Y, R)	$(q_5, \#, N)$ HALT
q_1	(q_1, a, R)	(q_2, Y, R)		(q_1, Y, R)	-
q_2	-	(q_3, Y, R)		-	-
q_3	-	(q_4, b, L)		.	$(q_4, \#, L)$
q_4	(q_4, a, L)	-	(q_0, X, R)	(q_4, Y, L)	
* q_5		.			



Instantaneous Description of 'aabbbb'

$(q_0, aabbbb)$

$\vdash (q_0, aabbbb)$

$\vdash (q_1, Xabbbb)$

$\vdash (q_1, Xabbbb$

$\vdash (q_2, XaYbbb$

$\vdash (q_3, XaYYbb$

$\vdash (q_4, XaYYbb$

$\vdash (q_4, XaYYbb$

$\vdash (q_4, XaYYbb$

$\vdash (q_4, XaYYbb$

$\vdash (q_0, XaYYbb$

$\vdash (q_1, XXYYbb)$

$\vdash (q_1, XXYYbb)$

$\vdash (q_1, XXYYbb)$

$\vdash (q_2, XXYYYb)$

$\vdash (q_3, XXYYYY\#)$

$\vdash (q_4, XXYYYY\#)$

$\vdash (q_4, XXYYYY\#)$

$\vdash (q_4, XXYYYY\#)$

$\vdash (q_4, XXYYYY\#)$

$\vdash (q_4, XXYYYY\#)$

$\vdash (q_0, XXYYYY\#)$

$\vdash (q_0, XXYYYY\#)$

$\vdash (q_0, XXYYYY\#)$

$\vdash (q_0, XXYYYY\#)$

$\vdash (q_0, XXYYYY\#)$

$\vdash (q_5, XXYYYY\#)$ Halt.

Design a Turing Machine $L = \{ w \mid w \in \{a,b\}^*$ such that $n_a(w)$ is ODD. $\}$

$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

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

$\Gamma = \{\#, a, b\}$

$q_0 = \{q_0\}$

$\#$ = Blank symbol

$A = \{q_2\}$.

Transition Function δ:

$\delta(q_0, a) = (q_1, \#, R)$

$\delta(q_0, b) = (q_0, \#, R)$

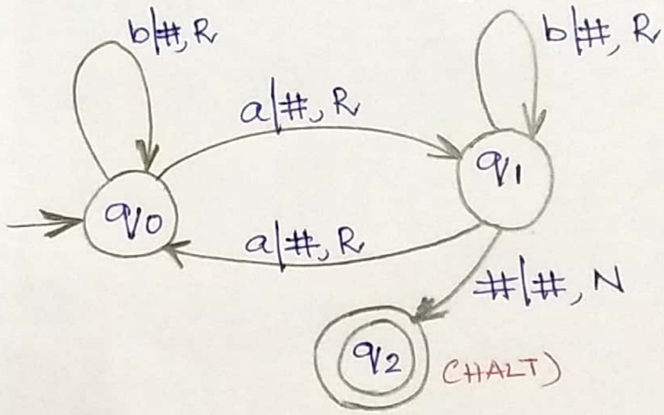
$\delta(q_1, a) = (q_0, \#, R)$

$\delta(q_1, b) = (q_1, \#, R)$

$\delta(q_1, \#) = (q_2, \#, N)$

$[q_2 \in A]$

State Transition Diagram.



ID for baaba.

$(q_0, baaba)$

$\vdash (q_0, \underline{b}aaba)$

$\vdash (q_0, \#\underline{a}aba)$

$\vdash (q_1, \#\#\underline{a}ba)$

$\vdash (q_0, \#\#\#\underline{b}a)$

$\vdash (q_0, \#\#\#\#\underline{a})$

$\vdash (q_1, \#\#\#\#\#\underline{\#})$

$\vdash (q_2, \#\#\#\#\#\#)$

T Halts, $q_2 \in A$

$\therefore$ string accepted

State Transition Table.

Q \ Γ	a	b	#
→ q_0	$(q_1, \#, R)$	$(q_0, \#, R)$	–
q_1	$(q_0, \#, R)$	$(q_1, \#, R)$	$(q_2, \#, N)$
* q_2	–	–	–

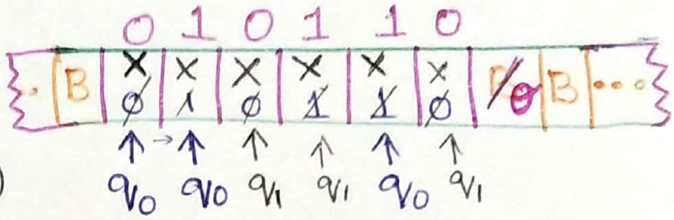
Example 1:

Design a TM to print 'Θ' if the number of 1's is ODD and print 'E' if the number of 1's is even over the string. $\Sigma = \{0, 1\}$

[Deterministic TM]

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

where δ is defined as.



$\delta(q_0, 0) = (q_0, X, R)$

$\delta(q_0, 1) = (q_1, X, R)$

$\delta(q_1, 0) = (q_1, X, R)$

$\delta(q_1, 1) = (q_0, X, R)$

$\delta(q_1, B) = (q_2, \Theta, -)$

$\delta(q_0, B) = (q_2, E, -)$

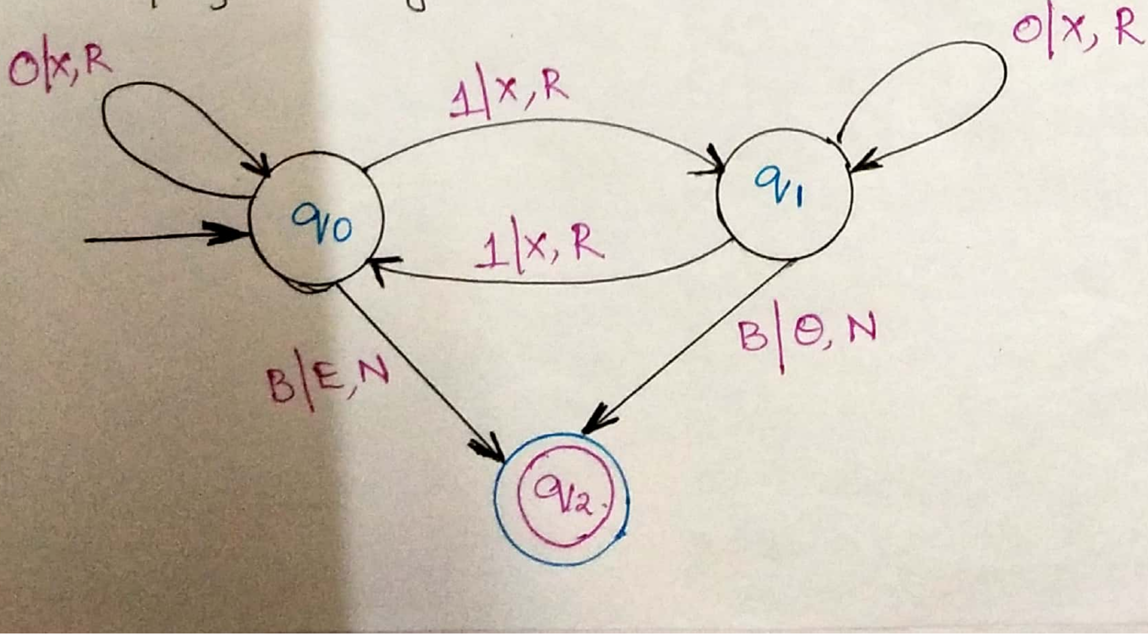
[When '1' comes as i/p change the state, when '0' comes no change in state]

$Q = \{q_0, q_1\}$

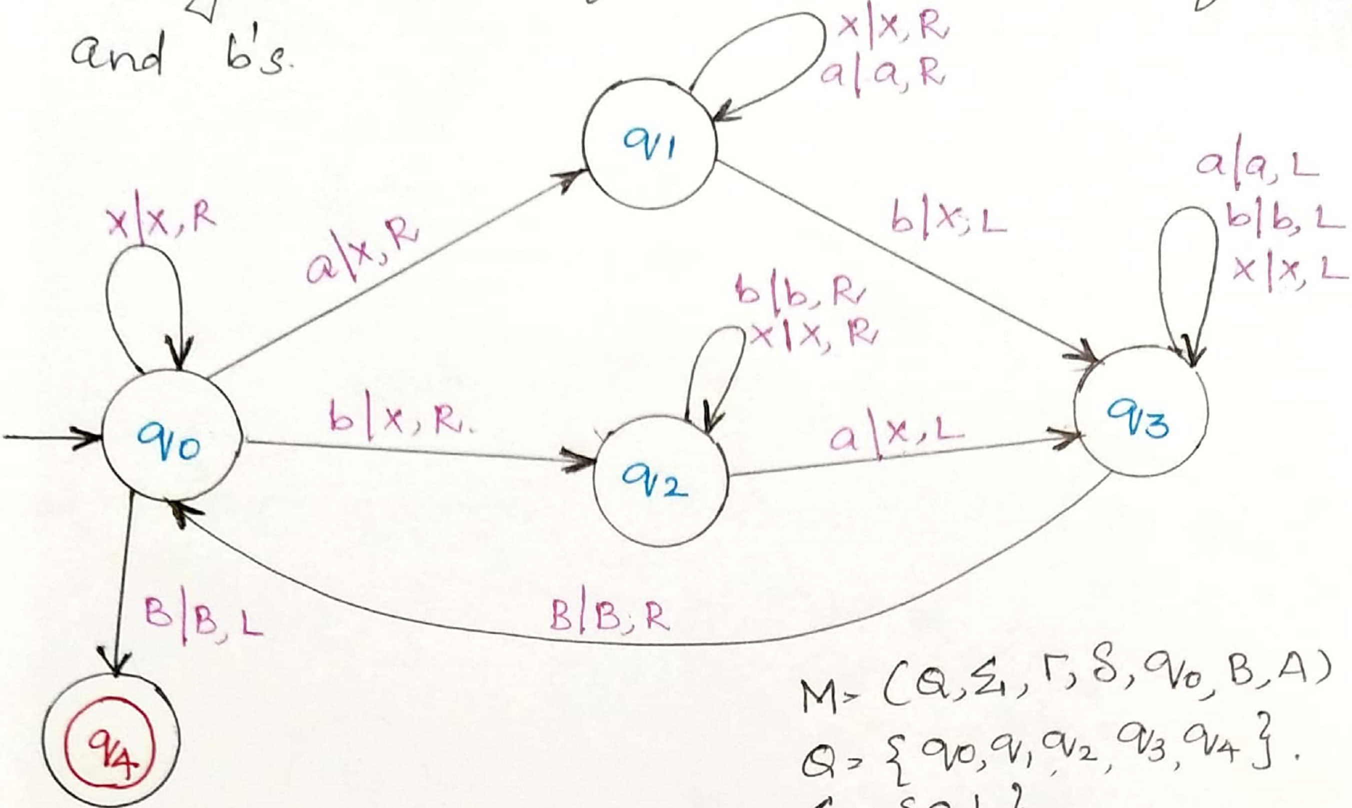
$\Sigma = \{0, 1\}$

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

$F = \{H\}$ Halting state.



Turing Machine for equal number of a's and b's.



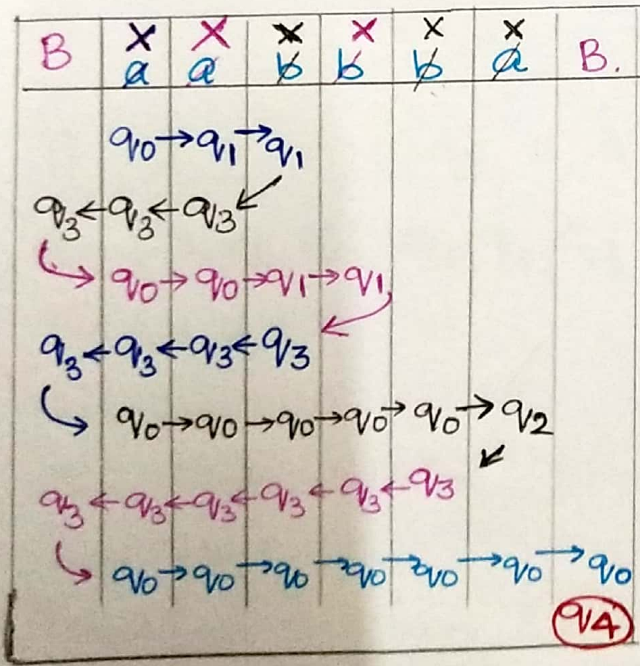
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, A)$

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

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

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

$A = \{q_4\}$



The transition fn δ is:

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_3, X, L)$

$\delta(q_1, X) = (q_1, X, R)$

$\delta(q_3, a) = (q_3, a, L)$

$\delta(q_3, b) = (q_3, b, L)$

$\delta(q_3, X) = (q_3, X, L)$

$\delta(q_3, B) = (q_0, B, R)$

$\delta(q_0, X) = (q_0, X, R)$.

$\delta(q_0, b) = (q_2, X, R)$

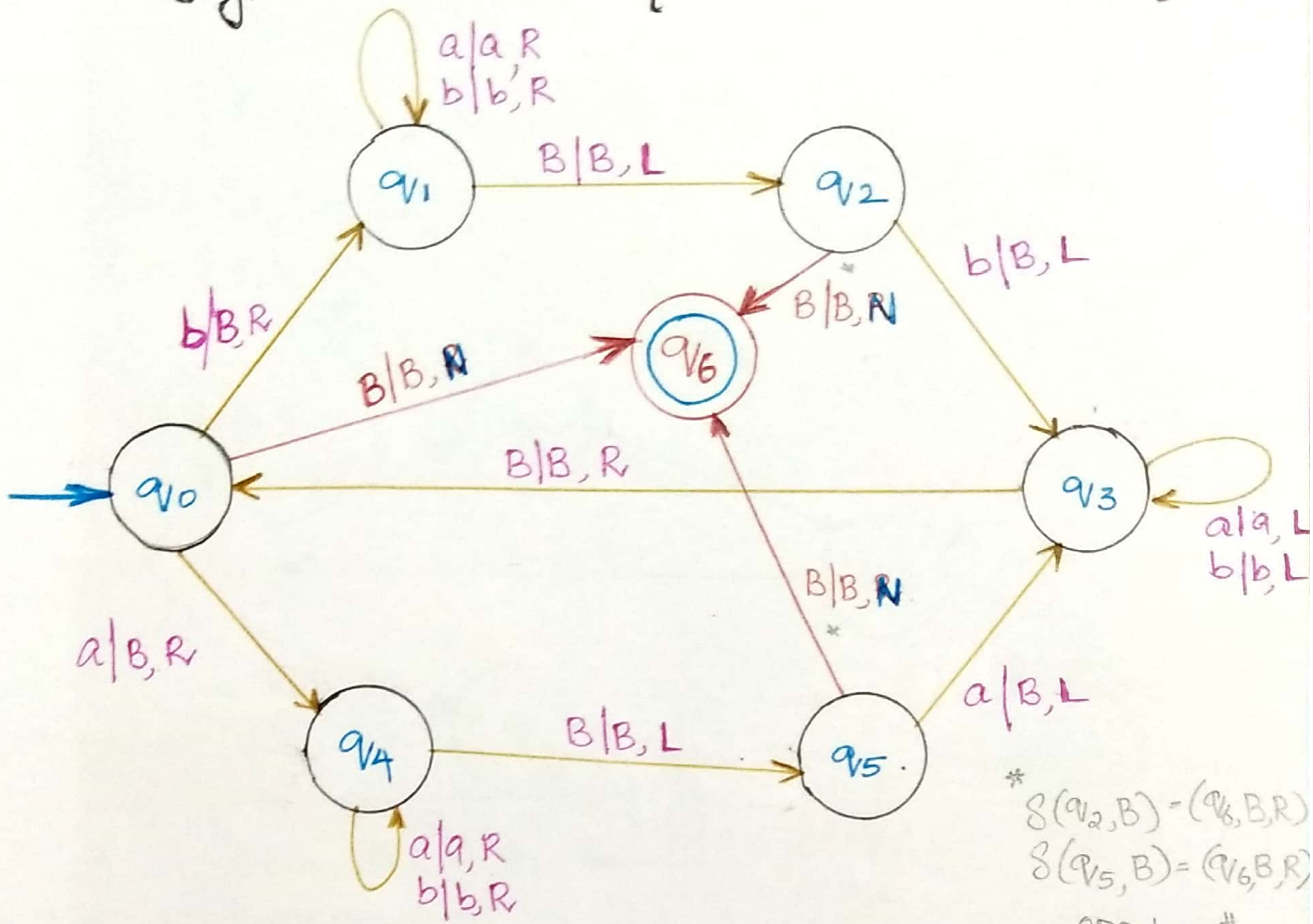
$\delta(q_2, b) = (q_2, b, R)$

$\delta(q_2, X) = (q_2, X, R)$

$\delta(q_3, a) = (q_3, X, L)$

$\delta(q_0, B) = (q_4, B, L)$

Design a T.M $L = \{ ww^R \mid w \in \{a,b\}^* \}$



* $\delta(q_2,B) = (q_6,B,R)$
$\delta(q_5,B) = (q_6,B,R)$
ODD Length Palindrome.

$T = \{Q, \Sigma, \Gamma, \delta, q_0, B, A\}$

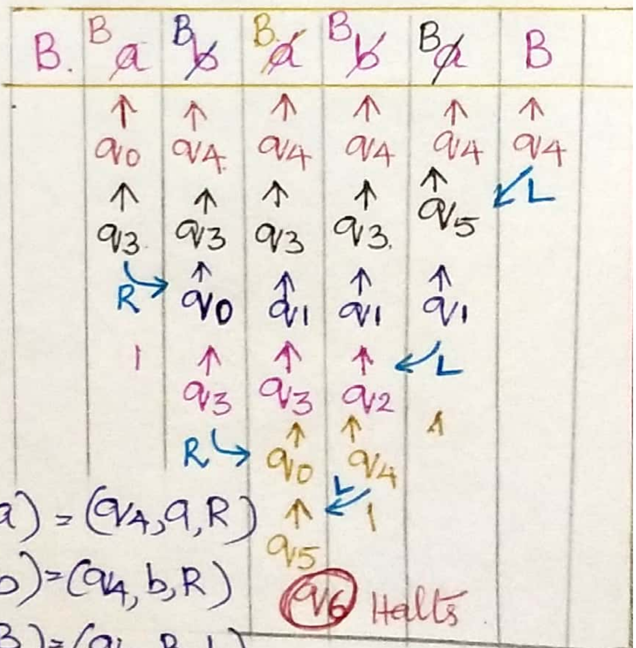
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

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

$q_0 = \{q_0\}$

B - Blank symbol

$A = \{q_6\}$



Transition Function δ:

$\delta(q_0,a) = (q_4,B,R)$ $\quad$ $\delta(q_4,a) = (q_4,a,R)$

$\delta(q_0,b) = (q_1,B,R)$ $\quad$ $\delta(q_4,b) = (q_4,b,R)$

$\delta(q_1,a) = (q_1,a,R)$ $\quad$ $\delta(q_4,B) = (q_5,B,L)$

$\delta(q_1,b) = (q_1,b,R)$ $\quad$ $\delta(q_5,a) = (q_3,B,L)$

$\delta(q_1,B) = (q_2,B,L)$ $\quad$ $\delta(q_3,B) = (q_0,B,R)$

$\delta(q_2,b) = (q_3,B,L)$ $\quad$ $\delta(q_0,B) = (q_6,B,N)$

$\delta(q_3,a) = (q_3,a,L)$ $\quad$ $\delta(q_2,B) = (q_6,B,N)$

$\delta(q_3,b) = (q_3,b,L)$ $\quad$ $\delta(q_5,B) = (q_6,B,N)$

T.M for $L = \{a^n b^m c^m \mid m, n \geq 1\}$.

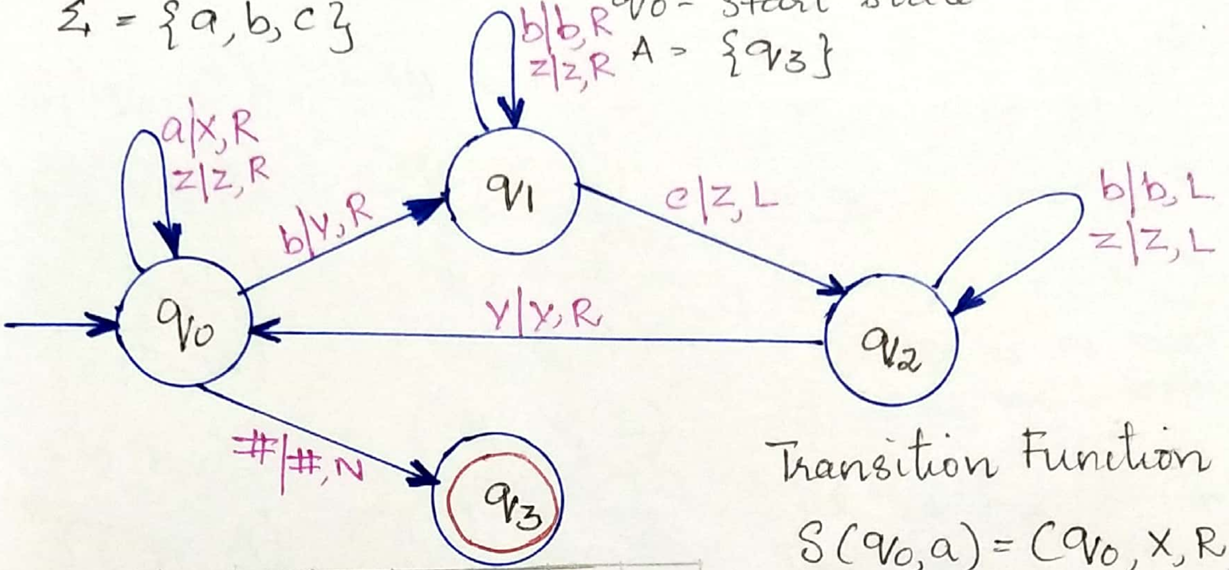
$T = \{Q, \Sigma, \Gamma, \delta, q_0, \#, A\}$

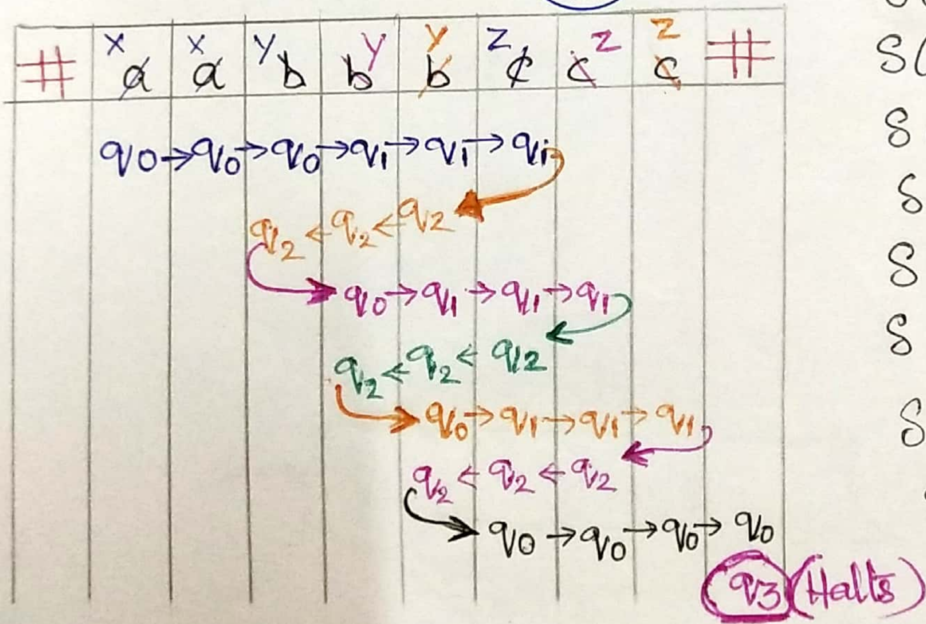
$Q = \{q_0, q_1, q_2, q_3\}$. $\Gamma = \{a, b, c, X, Y, Z, \#\}$.

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

q_0 - start state

$A = \{q_3\}$





Transition Function δ

$\delta(q_0, a) = (q_0, X, R)$

$\delta(q_0, b) = (q_1, Y, R)$

$\delta(q_1, b) = (q_1, b, R)$

$\delta(q_1, c) = (q_2, Z, L)$

$\delta(q_2, b) = (q_2, b, L)$

$\delta(q_2, Y) = (q_0, Y, R)$

$\delta(q_0, Z) = (q_0, Z, R)$

$\delta(q_1, Z) = (q_1, Z, R)$

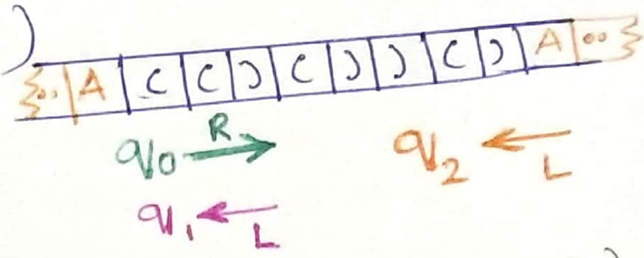
$\delta(q_2, Z) = (q_2, Z, L)$

$\delta(q_0, \#) = (q_3, \#, N)$

Design a T.M for paranthesis checking.

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

Let 'A' be a end marker.



$\delta(q_0, () = (q_0, (, R)$

$\delta(q_0,)) = (q_1, X, L)$

$\delta(q_1, () = (q_0, X, R)$

$\delta(q_0, X) = (q_0, X, R)$

$\delta(q_1, X) = (q_1, X, L)$

$\delta(q_1, A) = (H, N, -)$

[If more ')' appears]

$\delta(q_0, A) = (q_2, A, L)$

[There is no ')']

$\delta(q_2, X) = (q_2, X, L)$

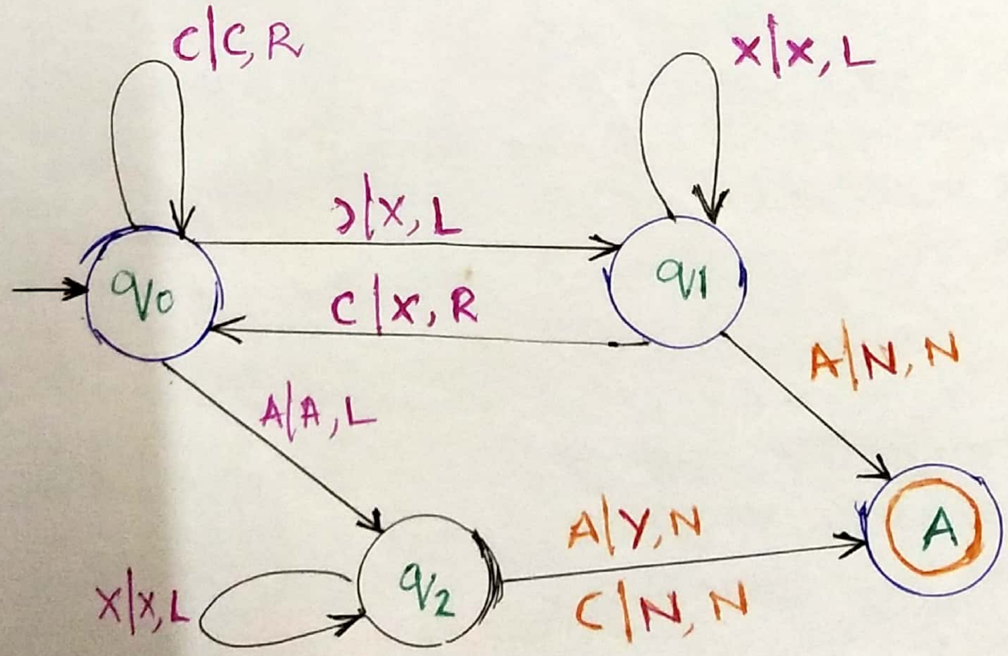
$\delta(q_2, A) = (H, Y, -)$

[There is no-more '(']

ie it is well-formed.

$\delta(q_2, () = (H, N, -)$

[If more '(' appears]



Scanned by CamScanner

Design a T.M $= \{a^n b^m a^m b^n \mid n, m \geq 1\}$

$M = \{Q, \Sigma, \Gamma, \delta, q_0, \#, A\}$

$Q =$

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

$\Gamma = \{a, b, X, Y, Z, P, \#\}$

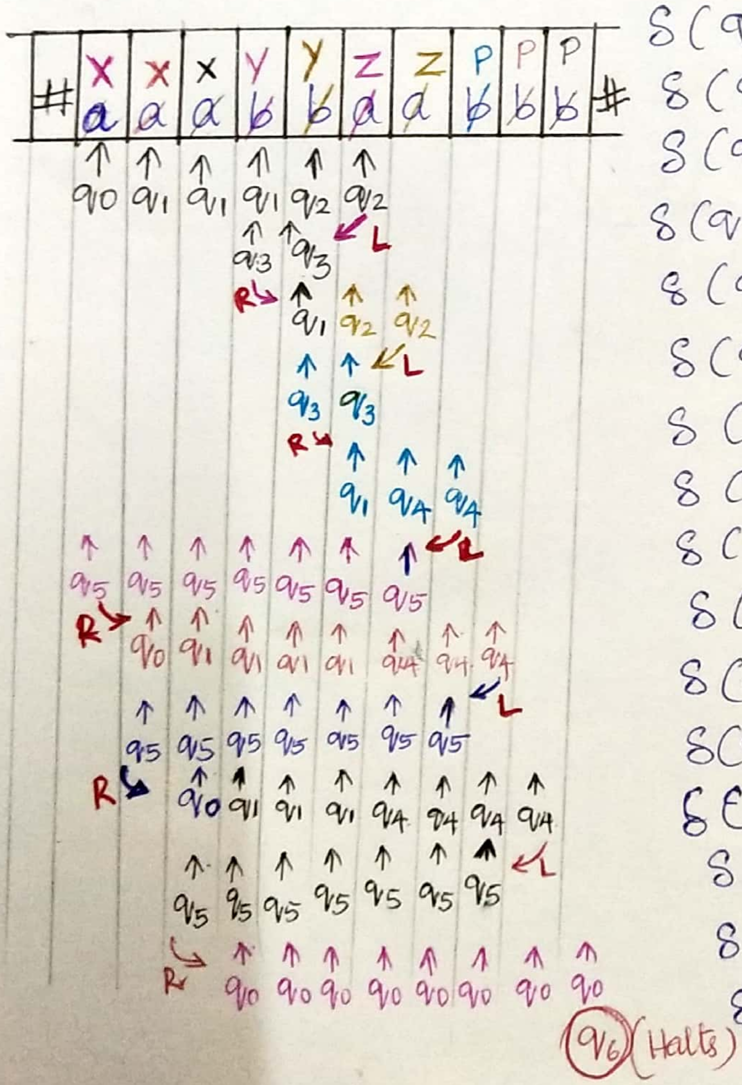
$q_0 - q_0.$

$\# = \#$

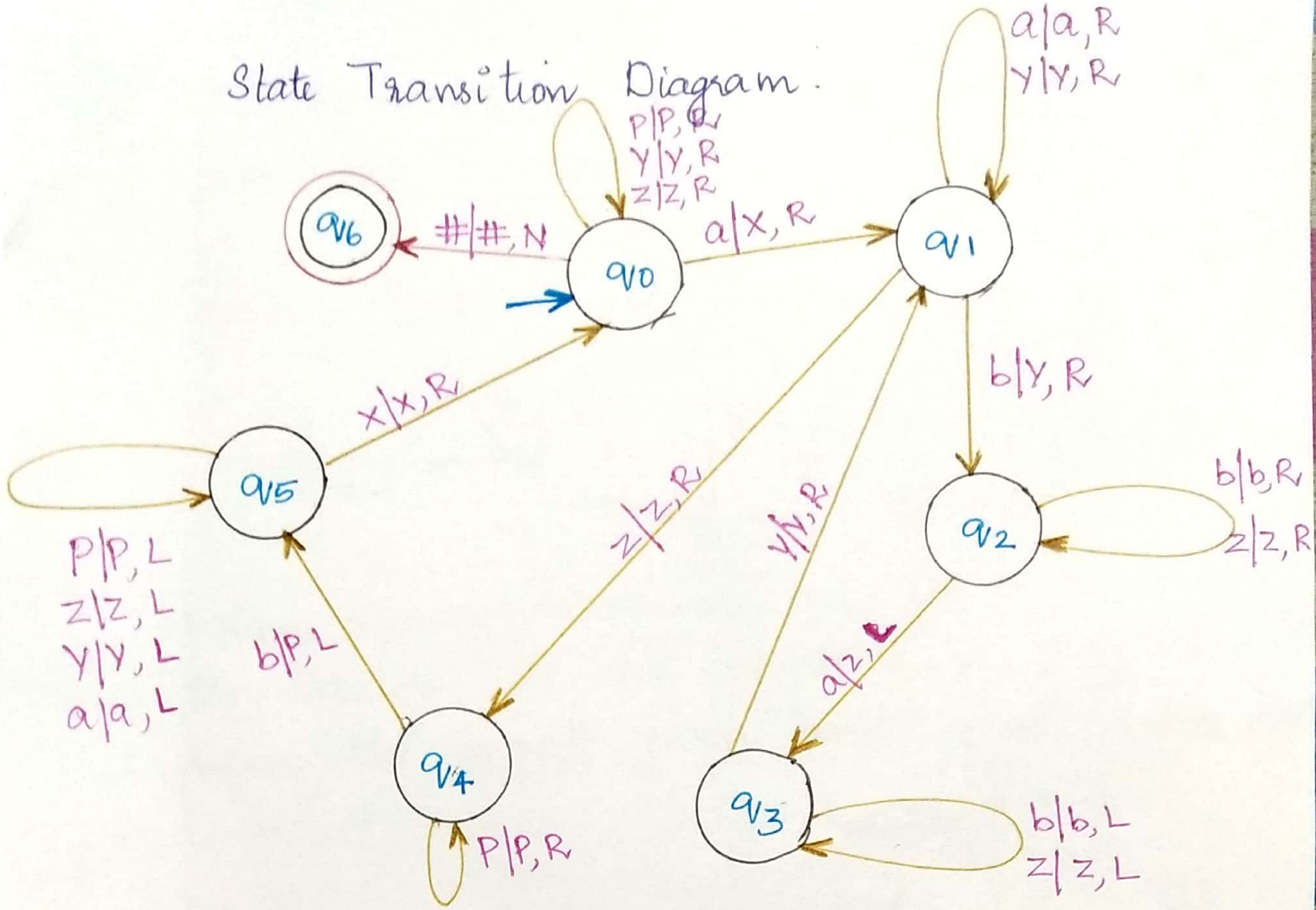
$A = \{q_6\}.$

Transition Function δ:

$\delta(q_0, a) = (q_1, X, R)$
$\delta(q_1, a) = (q_1, a, R)$
$\delta(q_1, b) = (q_2, Y, R)$
$\delta(q_2, b) = (q_2, b, R)$
$\delta(q_2, a) = (q_3, Z, L)$
$\delta(q_3, b) = (q_3, b, L)$
$\delta(q_3, Y) = (q_1, Y, R)$
$\delta(q_2, Z) = (q_2, Z, R)$
$\delta(q_3, Z) = (q_3, Z, L)$
$\delta(q_1, Z) = (q_4, Z, R)$
$\delta(q_4, Z) = (q_4, Z, R)$
$\delta(q_4, b) = (q_5, P, L)$
$\delta(q_5, P) = (q_5, P, L)$
$\delta(q_5, Z) = (q_5, Z, L)$
$\delta(q_5, Y) = (q_5, Y, L)$
$\delta(q_5, a) = (q_5, a, L)$
$\delta(q_5, X) = (q_0, X, R)$
$\delta(q_1, Y) = (q_1, Y, R)$
$\delta(q_4, P) = (q_4, P, R)$
$\delta(q_0, Y) = (q_0, Y, R)$
$\delta(q_0, Z) = (q_0, Z, R)$
$\delta(q_0, P) = (q_0, P, R)$
$\delta(q_0, \#) = (q_6, \#, N)$ Halt.

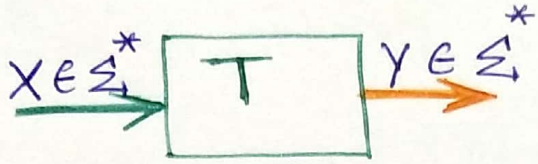


State Transition Diagram.



Turing Machine works like Acceptor as well as Transducer.

Turing Machine as Transducers.



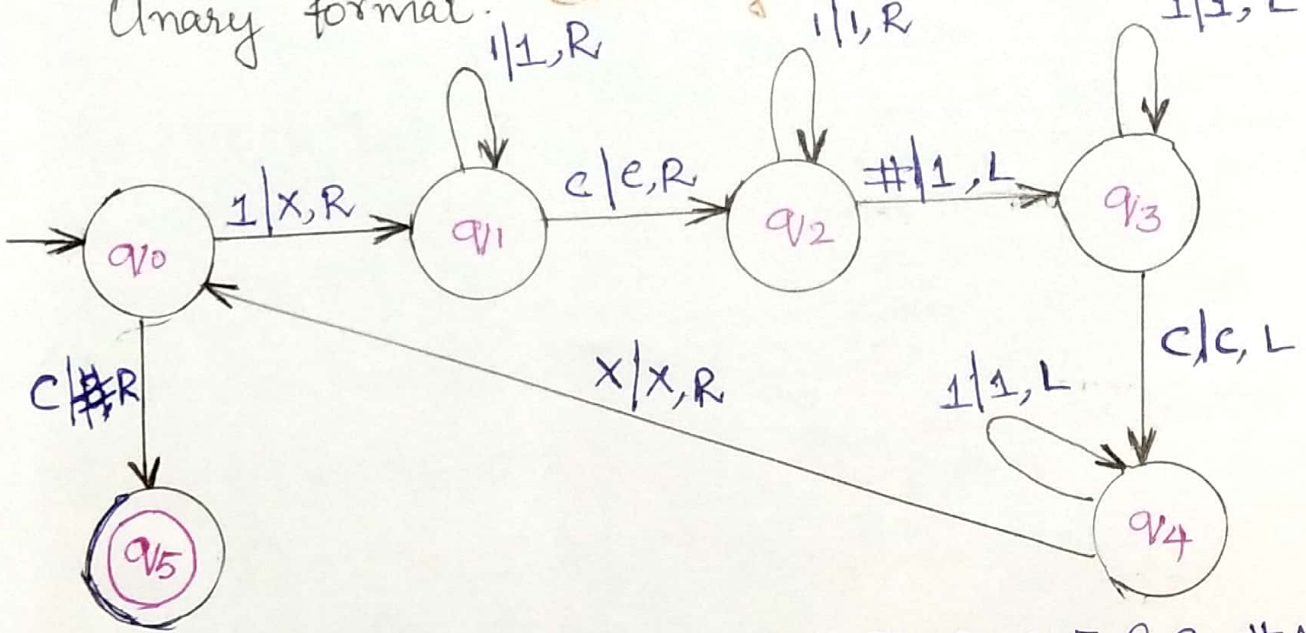
$y = f(x)$

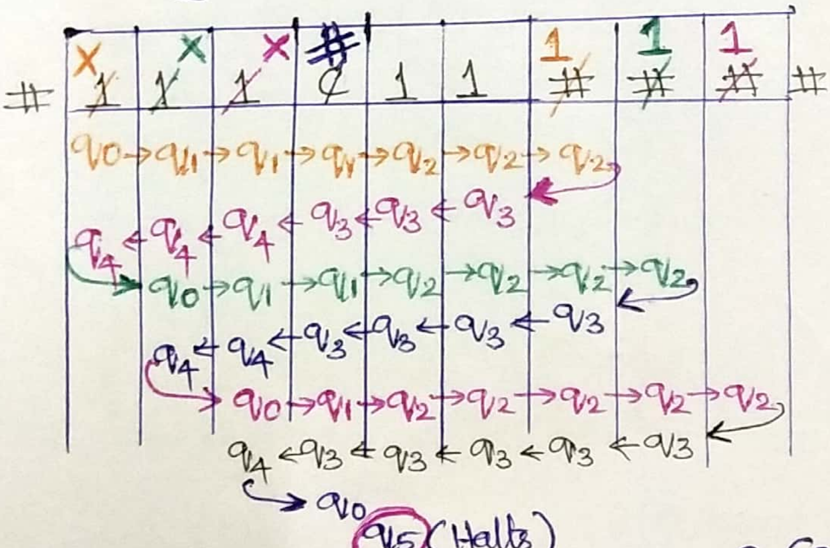
When turing Machines are used as transducers, it will compute a function upon the input given and gives back a result.

The input to the computation is a set of symbols on the tape.
At the end of the computation, whatever remains on the tape is the output

All common mathematical functions are Turing computable
Turing Machine is an abstract model of our modern day computer.

Design a T.M to add two numbers. Assume that numbers are represented as Unary format. (University Question - June 2018)





$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

$\Sigma = \{1\}$

$\Gamma = \{1, X, \#\}$

q_0 = q_0 start state

$A = \{q_5\}$

$\#$ = Blank symbol.

Transition Function δ:

$\delta(q_0, 1) = (q_1, X, R)$

$\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_1, c) = (q_2, C, R)$

$\delta(q_2, 1) = (q_2, 1, R)$

$\delta(q_2, \#) = (q_3, 1, L)$

$\delta(q_3, 1) = (q_3, 1, L)$

$\delta(q_3, c) = (q_4, c, L)$

$\delta(q_4, 1) = (q_4, 1, L)$

$\delta(q_4, X) = (q_0, X, R)$

$\delta(q_0, c) = (q_5, \#, R)$

Note:

Unary representation of 3 is 111 OR 000

Unary representation of 5 is 11111 OR 00000.

Design a Turing Machine to find the 2's compliment of a string over $\Sigma = \{0,1\}$

I/P: 11100 I/P: 11010

O/P: 00100 O/P. 00110

Go to the last symbol move in the left directⁿ until the First '1' appears then reverse all other symbols in the left

If I/P = 0000 then replace the first # by '1' @ q_1

T =

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

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

$\Gamma = \{0,1,\#\}$.

Transition Function δ

$\delta(q_0, 0) = (q_0, 0, R)$

$\delta(q_0, 1) = (q_0, 1, R)$

$\delta(q_0, \#) = (q_1, \#, L)$

$\delta(q_1, 0) = (q_1, 0, L)$

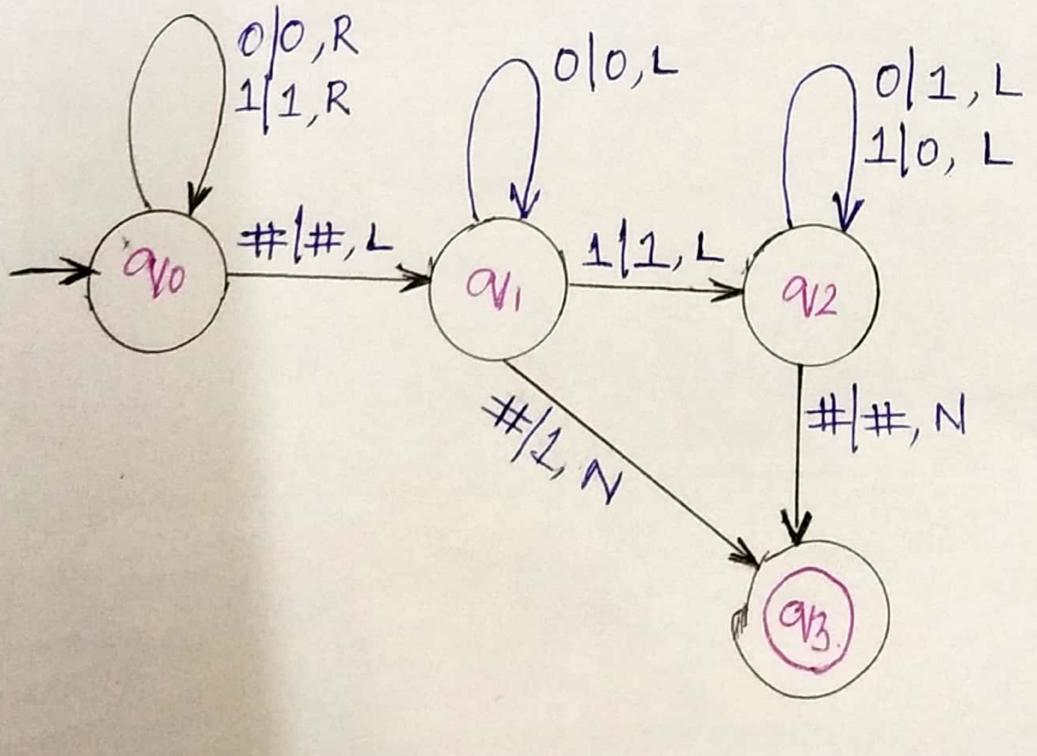
$\delta(q_1, 1) = (q_2, 1, L)$

$\delta(q_2, 0) = (q_2, 1, L)$

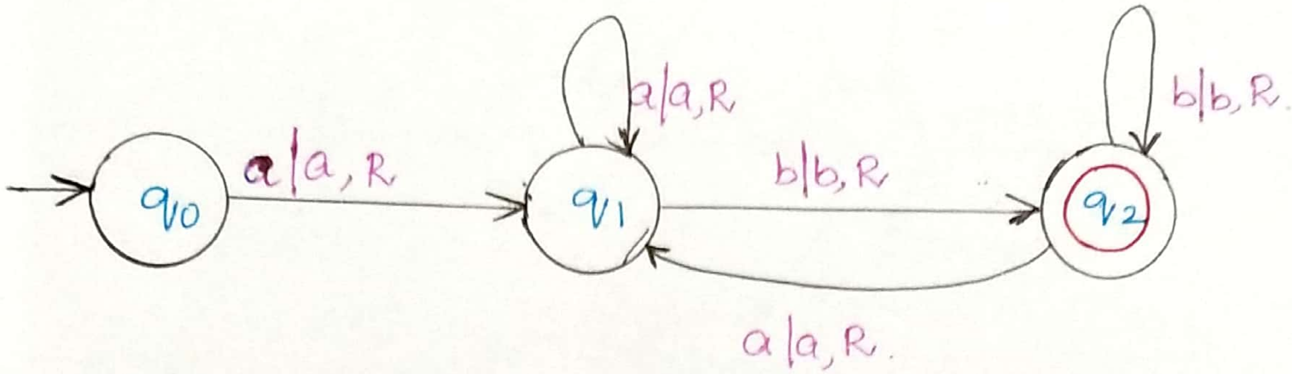
$\delta(q_2, 1) = (q_2, 0, L)$

$\delta(q_2, \#) = (q_3, \#, N)$

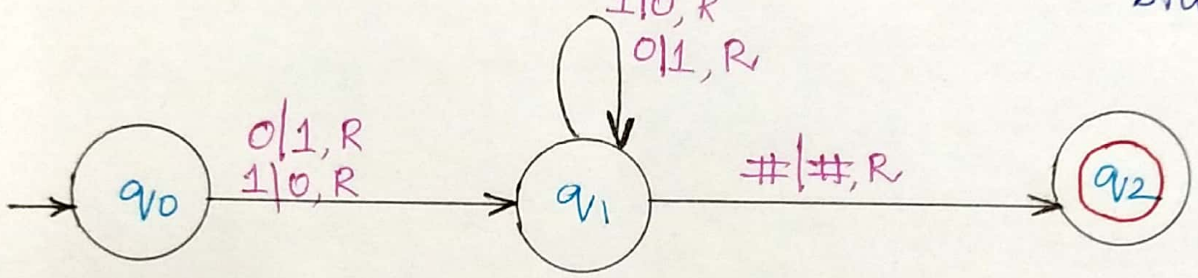
$\delta(q_1, \#) = (q_3, 1, N)$



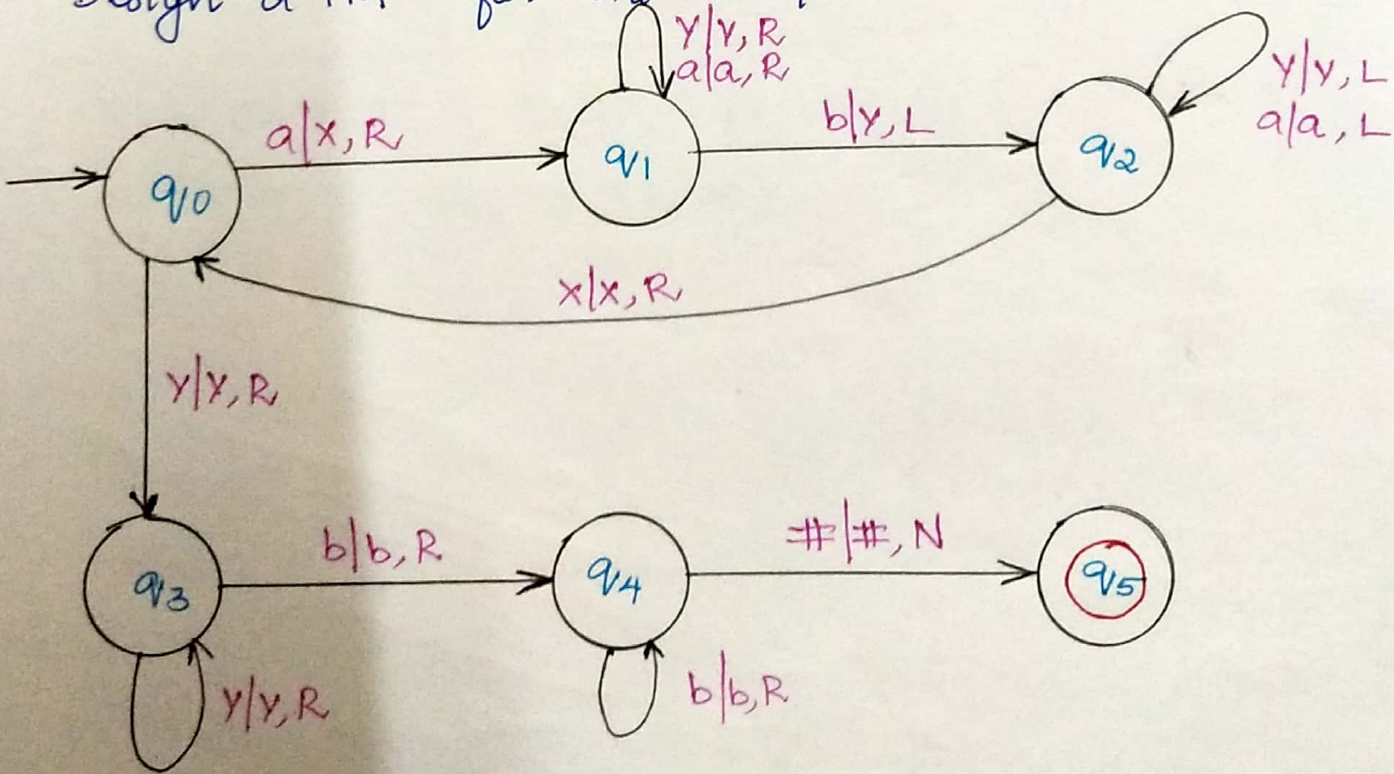
Design a Turing Machine which accepts all strings starting with 'a' and ending with 'b'. eg ab, aab....



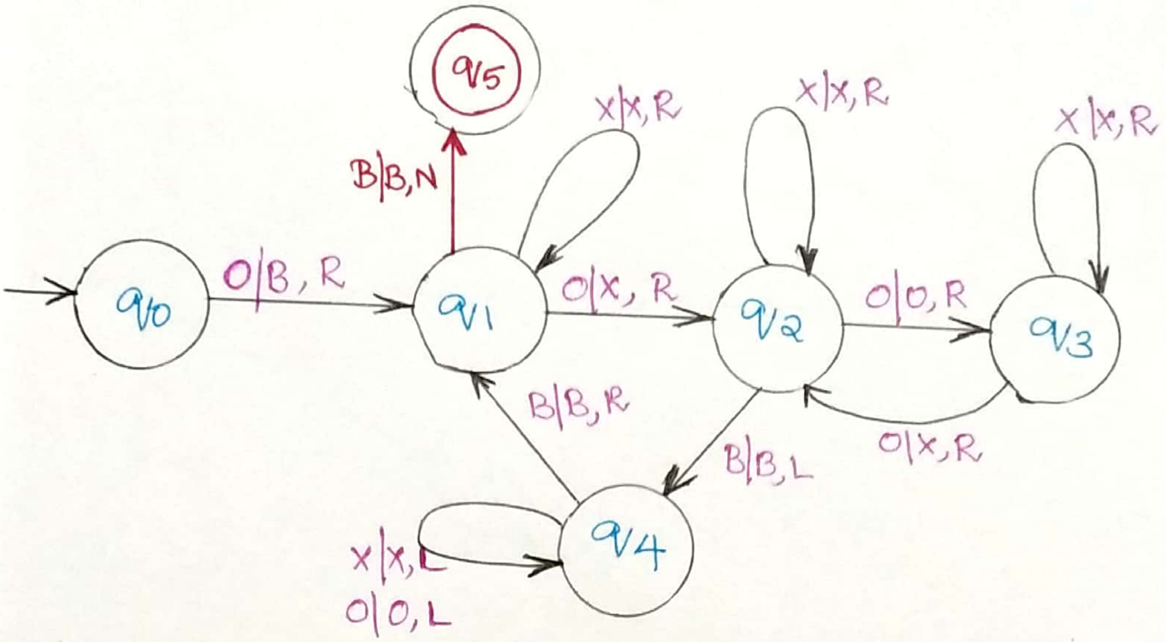
Turing Machine to find the compliment of a string.



Design a TM for the $L = \{ a^m b^n \mid m < n \}$



$L = \{ 0^{2^n} \mid n \geq 0 \}$



B	$\overset{B}{0}$	$\overset{x}{0}$	$\overset{x}{0}$	$\overset{x}{0}$	$\overset{x}{0}$	$\overset{x}{0}$	0^{x}	$\overset{x}{0}$	B

$q_0 \to q_1 \to q_2 \to q_3 \to q_2 \to q_3 \to q_2 \to q_3 \to q_2$

$q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4$

$\to q_1 \to q_2 \to q_2 \to q_3 \to q_3 \to q_3 \to q_2 \to q_2$

$q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4$

$\to q_1 \to q_1 \to q_1 \to q_1 \to q_2 \to q_2 \to q_2 \to q_2$

$q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4 \leftarrow q_4$

$\to q_1 \to q_1 \to q_1 \to q_1 \to q_1 \to q_1 \to q_1 \to q_1$

q_5 Accept

$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

$\Sigma = \{0\}$

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

q_0 - Start state

q_5 - Accept state

Transition Function δ:

$\delta(q_0, 0) = (q_1, B, R)$

$\delta(q_1, 0) = (q_2, x, R)$

$\delta(q_1, x) = (q_1, x, R)$

$\delta(q_2, x) = (q_2, x, R)$

$\delta(q_2, 0) = (q_3, 0, R)$

$\delta(q_3, x) = (q_3, x, R)$

$\delta(q_3, 0) = (q_2, x, R)$

$\delta(q_2, B) = (q_4, B, L)$

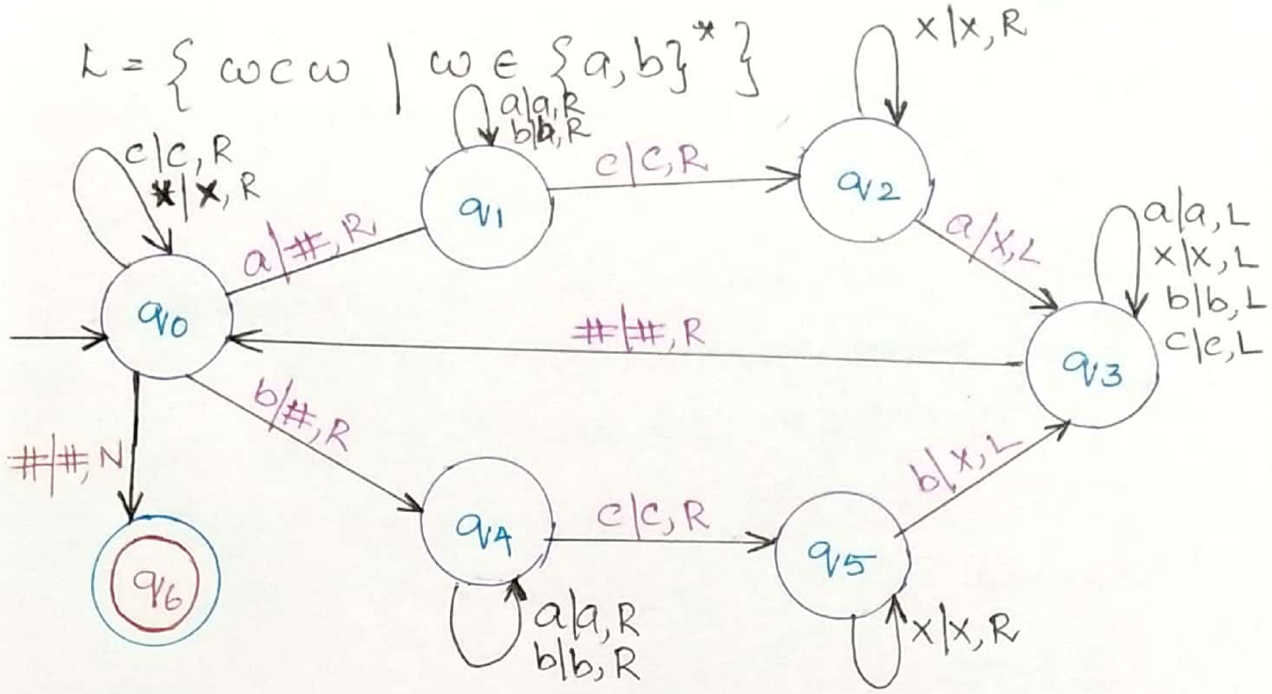
$\delta(q_4, x) = (q_4, x, L)$

$\delta(q_4, 0) = (q_4, 0, L)$

$\delta(q_4, B) = (q_1, B, L)$

$\delta(q_1, B) = (q_5, B, N)$

$L = \{ wcw \mid w \in \{a,b\}^* \}$



$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

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

$\Gamma = \{\#, a, b, c, x\}$

q_0 = Start state

q_6 = Accept state

Transition Function : δ

$\delta(q_0, a) = (q_1, \#, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_1, b, R)$

$\delta(q_1, c) = (q_2, c, R)$

$\delta(q_2, x) = (q_2, x, R)$

$\delta(q_2, a) = (q_3, x, L)$

$\delta(q_3, a) = (q_3, a, L)$

$\delta(q_3, b) = (q_3, b, L)$

$\delta(q_3, x) = (q_3, x, L)$

$\delta(q_3, c) = (q_3, c, L)$

$\delta(q_3, \#) = (q_0, \#, R)$

$\delta(q_0, b) = (q_4, \#, R)$

$\delta(q_4, a) = (q_4, a, R)$

$\delta(q_4, b) = (q_4, b, R)$

$\delta(q_4, c) = (q_5, c, R)$

$\delta(q_5, x) = (q_5, x, R)$

$\delta(q_5, b) = (q_3, x, L)$

$\delta(q_0, x) = (q_0, x, R)$

$\delta(q_0, c) = (q_0, c, R)$

$\delta(q_0, \#) = (q_6, \#, N)$

Design a Turing Machine which converts Unary number system to binary Number.

I/p : 111 O/p 11.

To differentiate the numbers in input and O/p. In O/p 0's can be represented by Y and one 1's can be represented by Z

∴ I/p : 111 O/p : ZZ

$(q_0, \#\#\#\underline{1}11\#)$

$\vdash (q_1, \#\#\underline{\#}X11\#)$

$\vdash (q_0, \#\#Z\underline{X}11\#)$

$\vdash (q_0, \#\#ZX\underline{1}1\#)$

$\vdash (q_1, \#\#Z\underline{X}X1\#)$

$\vdash (q_1, \#\#\underline{Z}XX1\#)$

$\vdash (q_1, \#\underline{\#}YXX1\#)$

$\vdash (q_0, \#Z\underline{Y}XX1\#)$

$\vdash (q_0, \#ZY\underline{X}X1\#)$

$\vdash (q_0, \#ZYX\underline{X}1\#)$

$\vdash (q_0, \#ZYXX\underline{1}\#)$

$\vdash (q_1, \#ZYX\underline{X}X\#)$

$\vdash (q_1, \#ZY\underline{X}XX\#)$

$\vdash (q_1, \#Z\underline{Y}XXX\#)$

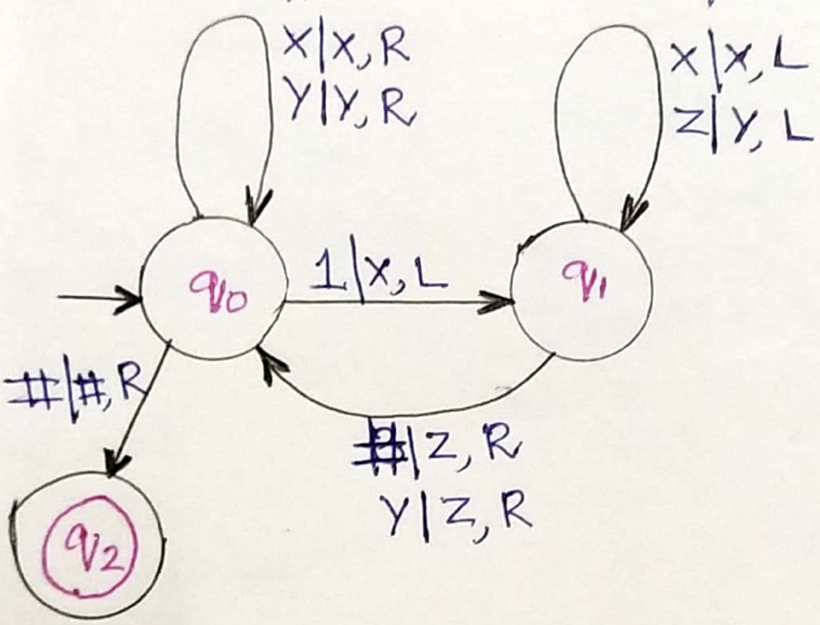
$\vdash (q_0, \#ZZ\underline{X}XX\#)$

$\vdash (q_0, \#ZZX\underline{X}X\#)$

$\vdash (q_0, \#ZZXX\underline{X}\#)$

$\vdash (q_0, \#ZZXXX\underline{\#})$

$q_2 \in A$ Halts



$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

$\Sigma = \{1\}$

$\Gamma = \{\#, X, 1, Z, Y\}$

q_0 - Start state

q_2 - Accept state.

$\delta(q_0, 1) = (q_1, X, L)$

$\delta(q_0, X) = (q_0, X, R)$

$\delta(q_0, Y) = (q_0, Y, R)$

$\delta(q_1, X) = (q_1, X, L)$

$\delta(q_1, Z) = (q_1, Y, L)$

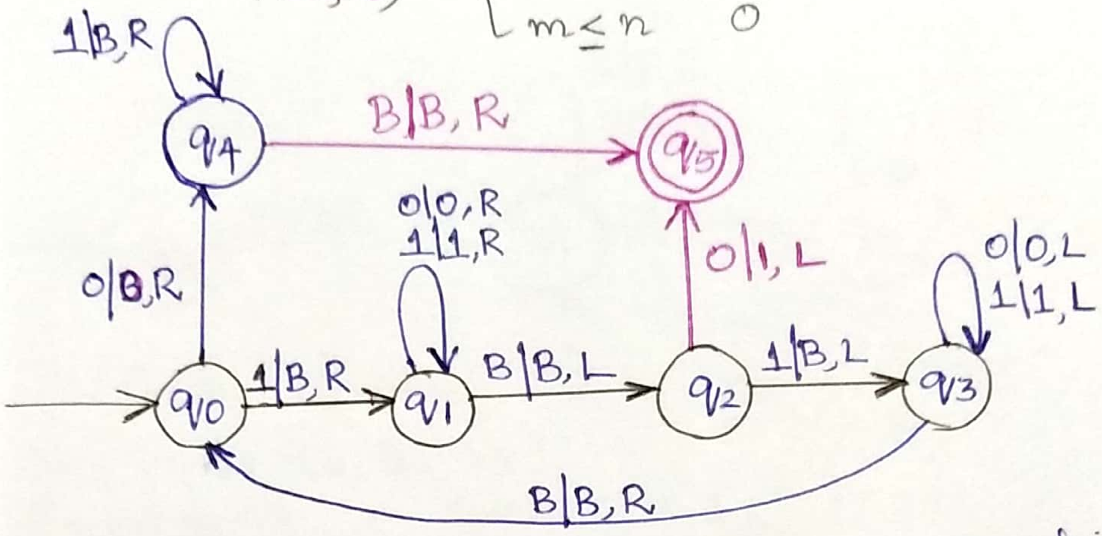
$\delta(q_1, Y) = (q_0, Z, R)$

$\delta(q_1, \#) = (q_0, Z, R)$

$\delta(q_0, \#) = (q_2, \#, R)$

create a turing machine which performs the substraction operation.

$f(m,n) = \begin{cases} m>n & o/p. \\ m\leq n & 0 \end{cases}$



$T = (Q, \Sigma, \Gamma, \delta, q_0, B, A)$

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

$\Sigma = \{1\}$

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

q_0 - Start state

A = q_5

B - Blank symbol.

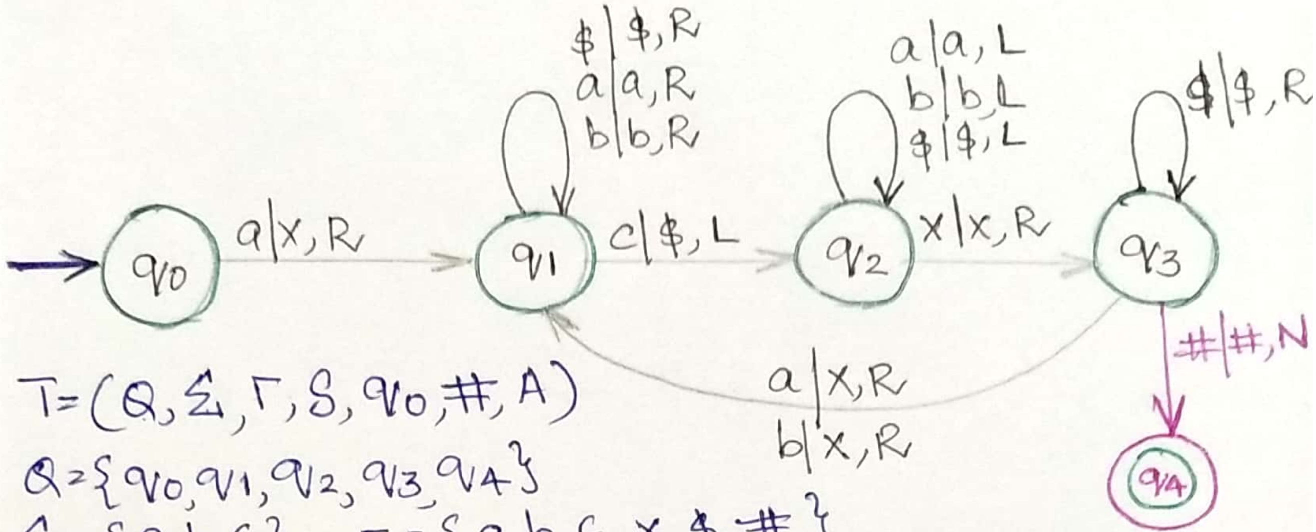
Transition Function δ:

$\delta(q_0, 1) = (q_1, B, R)$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, 1) = (q_1, 1, R)$
$\delta(q_1, B) = (q_2, B, L)$
$\delta(q_2, 1) = (q_3, B, L)$
$\delta(q_3, 0) = (q_3, 0, L)$
$\delta(q_3, 1) = (q_3, 1, L)$
$\delta(q_3, B) = (q_0, B, R)$
$\delta(q_2, 0) = (\underline{q_5}, 1, L)$ Halt $m>n$
$\delta(q_0, 0) = (q_4, B, R)$
$\delta(q_4, 1) = (q_4, B, R)$
$\delta(q_4, B) = (\underline{q_5}, B, R)$ Halt $m\leq n$

Turing Machine halts if it enters a state q scanning a tape symbol x, and there is no move in this situation ie $\delta(q, x)$ is undefined.

TM halts when it reaches an accepting state

Design a Turing Machine for
$L = \{a^n b^m c^k \mid k = n+m \ \& \ n, m > 0\}$



$T = (Q, \Sigma, \Gamma, \delta, q_0, \#, A)$

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

$\Sigma = \{a, b, c\}$ $\quad \Gamma = \{a, b, c, X, \$, \#\}$

Transition Function is defined as:

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_1, b, R)$

$\delta(q_1, \$) = (q_1, \$, R)$

$\delta(q_1, c) = (q_2, \$, L)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, b) = (q_2, b, L)$

$\delta(q_2, \$) = (q_2, \$, L)$

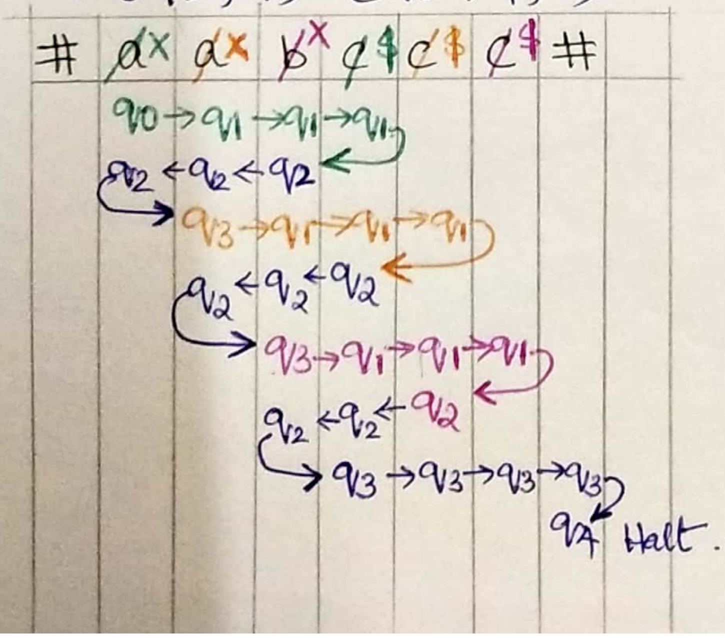
$\delta(q_2, X) = (q_3, X, R)$

$\delta(q_3, \$) = (q_3, \$, R)$

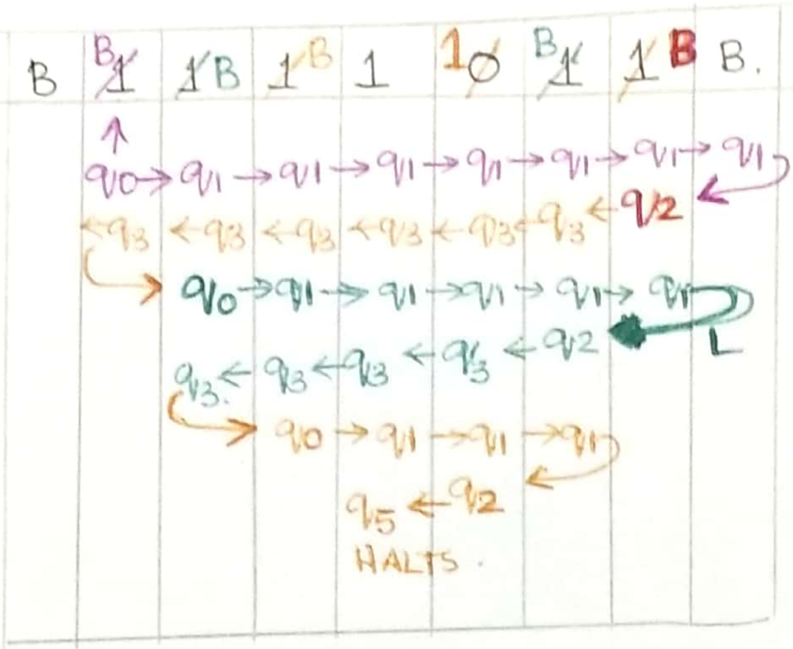
$\delta(q_3, a) = (q_1, X, R)$

$\delta(q_3, b) = (q_1, X, R)$

$\delta(q_3, \#) = (q_4, \#, N)$

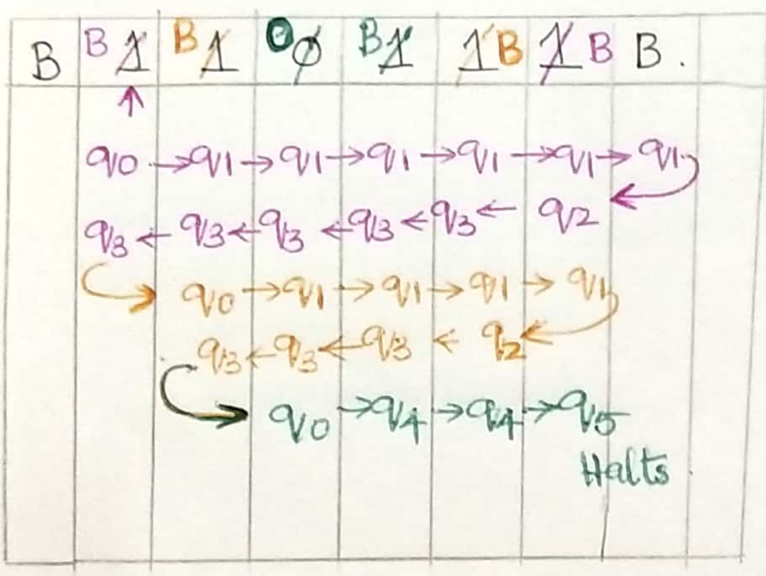


Case 1:



$m > n$

case 2:



$m \leq n = 0$