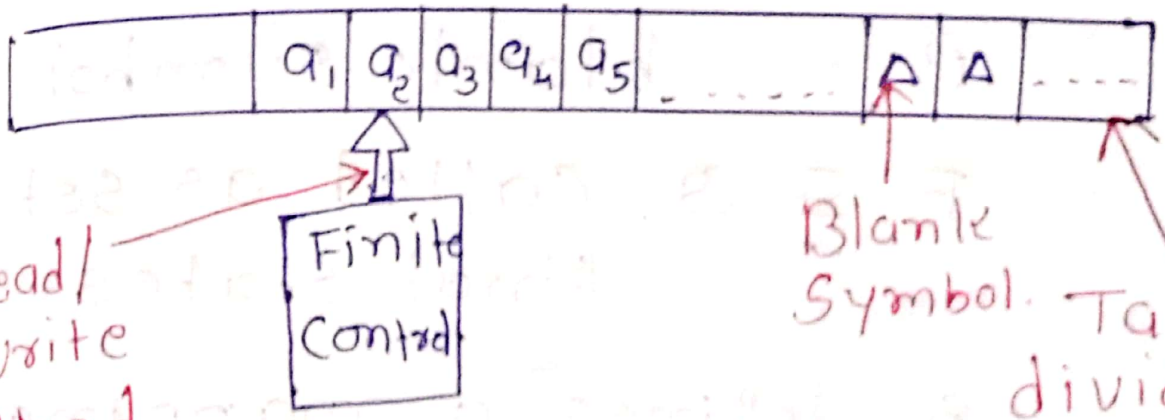
Unit - 9
Turing Machine

Introduction :- [Alan Turing 1936]

* The Turing Machine can be considered as finite control machine connected to an R/W (read/write) head.
* The tape is divided into no. of cells.
* The machine checks the present symbol under the R/W head on the tape and the present state of an automaton in one move.
* Finite control scan one cell at a time.

* Turing Machine Model

OPS.



* only one symbol can be stored by each cell.

* Defination :-

A Turing Machine M is a 7 - tuple variable as given below

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

where,

Q = Finite set of states of TM

Γ = Finite set of symbols called tape symbols.

Σ = Input symbols.

q_0 = initial State

B = blank symbol

$F \subseteq Q$ called as set of final states.

δ defines a mapping from $Q \times \Gamma$ to $Q \times \Gamma$

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

* Representation of Turing Machine:

A TM can be represented by three ways.

(i) Instantaneous Description using Move.

(ii) Transition Table

(iii) Transition Diagram.

(1) Instaneous Description using Move.

→ The arrangement of a TM depends on the present state of the finite control and the next i/p symbol, which depends on the string written on an i/p tape at that instance of time.

→ If the current state is q_0, and the i/p or the next symbol appearing on the tape is X, and if the TM changes its state q_1 in Right direction and replaces the i/p symbol X by B (blank symbol) or Δ

then, it can be represented by following notation.

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

- q_0: i/p state
- X: current i/p
- q_1: output state
- B: Replaced symbol
- R: Direction of tape

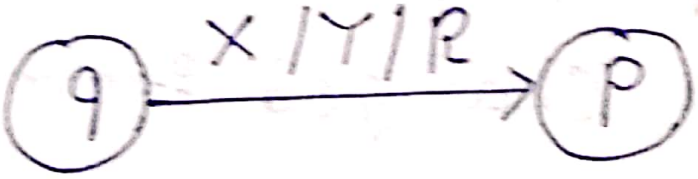
(ii) Transition Diagram :-

→ A transition Diagram of the next-move functions δ of a TM is a graphical representation consisting of a finite no. of nodes and directed-labelled arc between nodes.

→ Each node represents a state of TM and a label on arc from one state to a state represents the information about the required input symbol, the Transition and action on the part of the control of TM.

$\delta(q, X) = (P, Y, R)$

↑ one cell Right

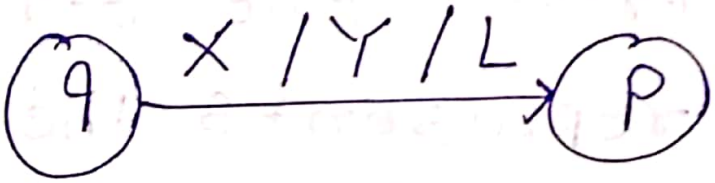


→ X/Y → X is to be replaced by Y

→ R → head is to be moved one cell right.

→ Arrow from q to p means State to be changed from q to P.

$$\delta(q, X) = (p, Y, L)$$



X / Y → X is to be replaced by Y

L → tape head is moved one cell left

arow q to p → State to be changed from q to p

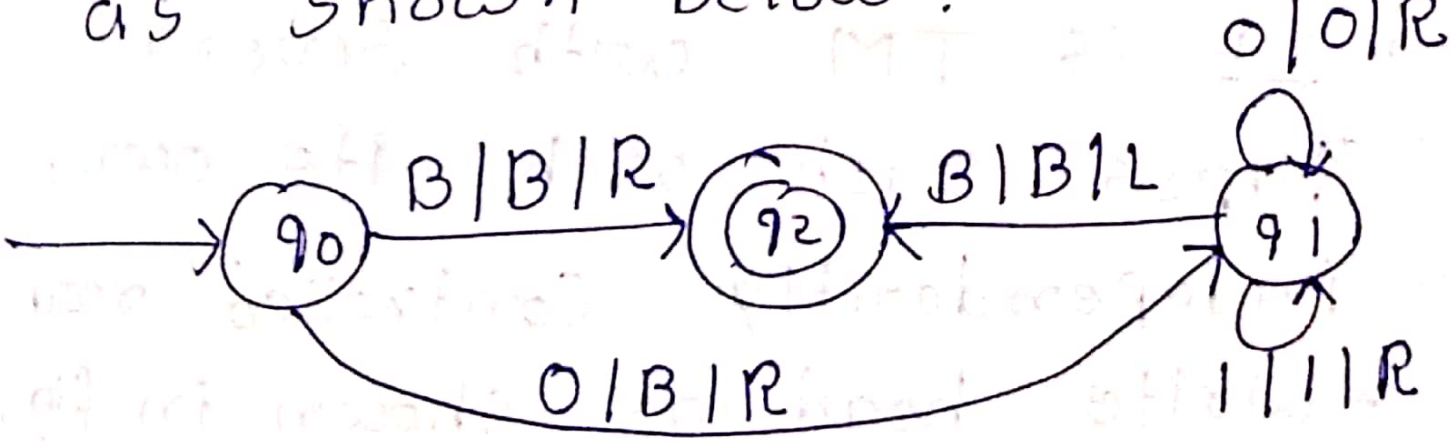
(iii) Transition Table :-

A transition table is a tabular representation of a function like δ that takes two arguments and return a state, replace Symbol and tape

direction. The rows of table correspond to the state, and the columns correspond to the i/p symbols along with B (blank symbol)

or Δ

Ex:- For the transition system shown in fig. can be represented in tabular form as shown below.



Present State	0	1	B
→ q_0	(q_1, B, R)	—	(q_2, B, R)
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	(q_2, B, L)
(q_2)	—	—	—

* Modification / Variants of Turing Machine..

* Two types of Turing machine.

(i) Multitape Turing Machine
(ii) Non deterministic Turing Machine.

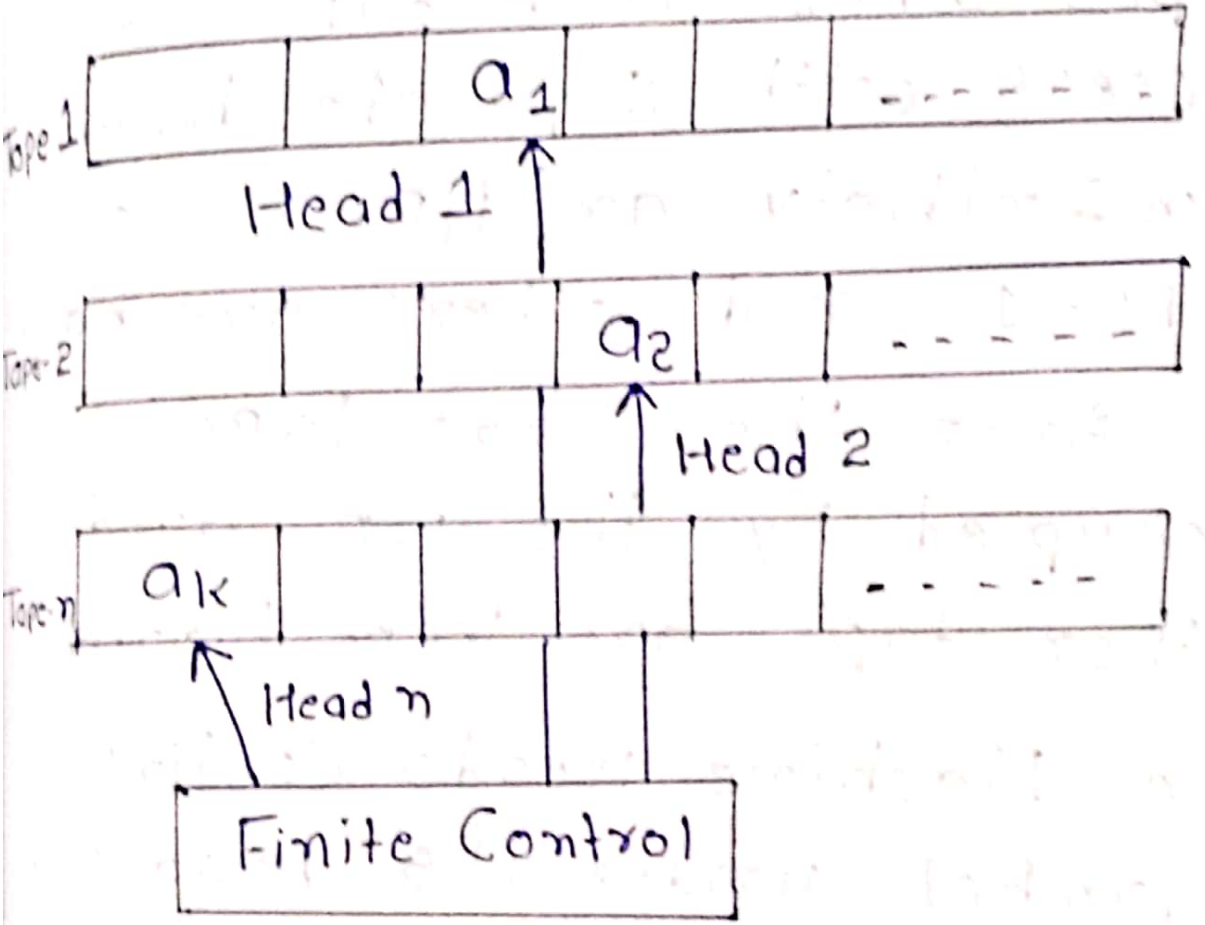
---

* Multitape Turing Machine :-

→ It is TM with several tapes, each with its own, independently controlled read-write head as shown in fig.

→ The formal defination of a multitape turing machine goes beyond the definition of a standard turing machine, Since it requires a modified

transition function.



Multitape Machine

Typically, we define an n-tape machine by

$$\delta : Q \times \{\Gamma_1 \times \Gamma_2 \times \dots \times \Gamma_n\} \rightarrow Q \times \{\Gamma_1 \times \Gamma_2 \times \dots \times \Gamma_n\} \times (\{L, R\}, \dots \times \{L, R\} \text{ n times})$$

i.e.,

$$\delta : (q_1 (a_1, a_2 \dots a_n)) = (p, (b_1, b_2 \dots b_n), (L, R, L, L \dots$$

→ Multitape TM have multiple tapes where each tape is access with seperate head.

→ Initialy an i/p is on tape-1 and others are blan

→ At first the first tape occupied by the i/p and other tape are kept blank.

→ The Machine Reads Continious symbol under its head an the TM print assambled on each tape-2 Moves its head.

→

* Non-Deterministic TM :-

→ A NDTM may proceed according to multiple possibilities at any point while carrying out Computation.

→ When a tape symbol is scanned the machine has a finite no. of choices of next move.

i.e. the transition function δ defines mapping from $Q \times \Gamma$ to finite subset of $\{Q \times \Gamma \times \{L, R\}\}$

→ TM has an equivalent DTM.

→ For, NDTM M, it is always possible to construct a TM M_1 which try all possible sequences of computations of M, and if M enters into an accept state on one of these branches M_1 accepts, else M continues

A non-deterministic TM M is a seven tuple given bela

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

where,

Q = finite set of states

Γ = tape symbols

Σ = non empty subset of Γ

q_0 = initial state

B = blank symbol.

$F \subseteq Q$ called set of final stat

δ defines a mapping from $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R\}$

NOTE :- NDTM means transitio of variable or i/p symbols Can be one or more.

ex:- (q_0, a, a)

(q_0, a, A)

Non-Deterministic TM :-

→ NDTM is a powerful feature.

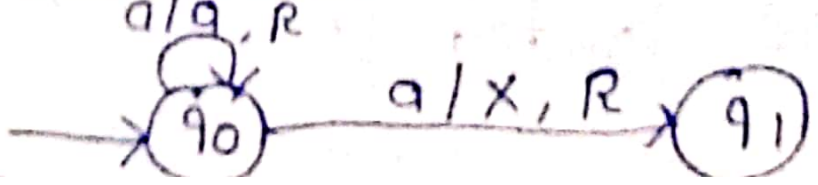
→ NDTM have certain combi-nations of state and symbol under the head, more than one possible choice of behaviour.

→ NDTM does not make a TM more powerful.

→ For every NDTM, there is an equivalent deterministic TM.

→ It is easy to design a NDTM for certain class of problems.

→ A string is said to be accepted by NDTM, if there is atleast one sequence of moves that takes the machine to final state.



Universal Turing Machine :-

→ A general-Purpose computer can be programmed to solve different types of Problem.
→ BY designing reprogrammable TM (all
→ A TM can also behave like a general purpose computer.

→ A general Purpose Computer * solves a problem as given below:

1. A Program is written in a high level language and its machine-code is obtained with the help of Compiler.
2. Machine Code is loaded in main Memory.
3. Input to the program can also be loaded in memory.

1. Program Stored in memory is executed line by line.

Execution involves reading a line of code pointed by IP, decoding the code and executing it.

We can follow a similar approach for a TM. Such, a TM is known as Universal Turing Machine.

→ UTM can solve all sorts of Solvable problems.

Note * A TM is special computer, once we define δ, the machine is restricted to carrying out one particular type of computation.

→ On other hand, computers are general purpose machine, can do different job at diff time

→ TM can not = to general purpose computer

* Power of Various Machines

→ Turing Machine is more powerful than a PDA.

like, $a^n b^n c^n$ can be handled by TM but can not be handled by a PDA.

→ An NPDA is more powerful than a DPDA.

→ DPDA is more powerful tha FA.

FA ≤ DPDA ≤ NPDA ≤ TM.

④ ③ ② ①

Power of TM over FSM.

→ Turing machine is more power-ful than FA.

→ FA has no memory. TM has additional memory in form of a tape.

→ FA cannot modify its input. A TM can modify its own i/p

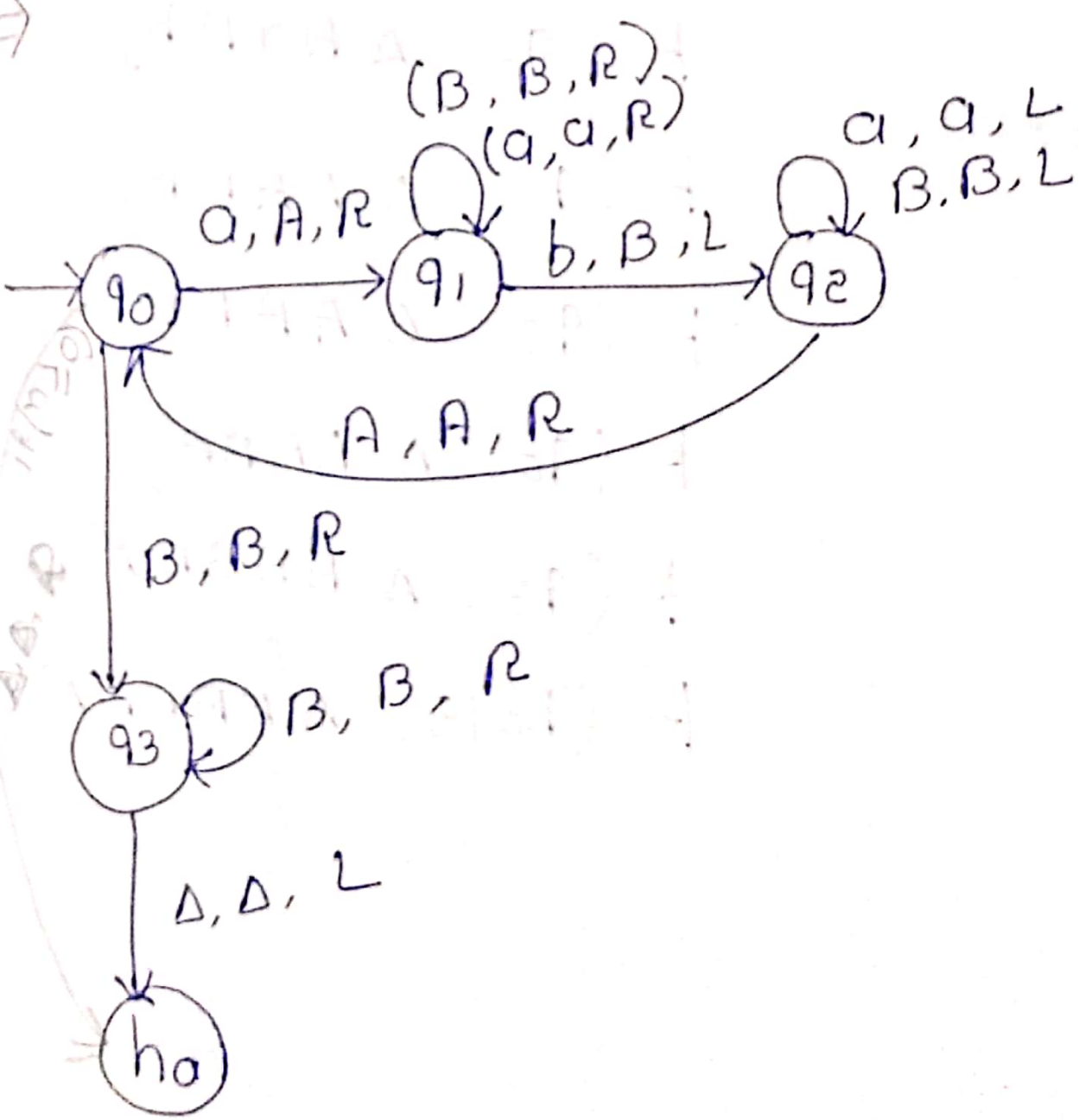
→ FA cannot be used for arithmetic operations. A TM can perform arithmetic ope.

→ A language accepted by FA is Regular. It cannot handle CFG, CSG, RL. but all language can be handled by TM.

Ex-3 Construct a TM for Palindrome:

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

⇒



Tracing :-

$(q_0, \Delta aabb) \vdash (q_1, \Delta Aabb)$

$\vdash (q_2, \Delta AaBb)$

$\vdash (q_0, \Delta AaBb)$

$\vdash (q_1, \Delta AABb)$

$\vdash (q_2, \Delta AABB)$

$\vdash (q_0, \Delta AABB)$

$\vdash (q_3, \Delta AABB\Delta)$

$\vdash (halt, \Delta AABB\Delta)$

Transition Table :-

State	a	b	A	B	Δ
q_0	(q_1, A, R)	-	-	(q_3, B, R)	-
q_1	(q_1, a, R)	(q_2, B, L)	-	(q_1, B, R)	-
q_2	(q_2, a, L)	-	(q_0, A, R)	(q_2, B, L)	-
q_3	$(q_3, -$	-	-	(H, B, R)	(H, Δ, L)
Halt	-	-	-	-	-

Ex-4 Construct TM for the language

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

⇒ Tracing aaabbbccc

$(q_0, \Delta \underline{a}aabbbccc) \vdash$

$(q_1, \Delta A \underline{a}abbbccc)$ R

$(q_2, \Delta AaaB\underline{b}bccc)$ R

$(q_3, \Delta AaaBbb\underline{c}cc)$ L

$(q_0, \Delta AaaBbb\underline{c}cc)$

$(q_1, \Delta AA\underline{a}Bbb\underline{c}cc)$ R

$(q_2, \Delta AAaB\underline{B}b\underline{c}cc)$ R

$(q_3, \Delta AAaBBbC\underline{c}c)$ L

$(q_0, \Delta AAaBBbCcc)$

$(q_1, \Delta AAaBBbCcc)$

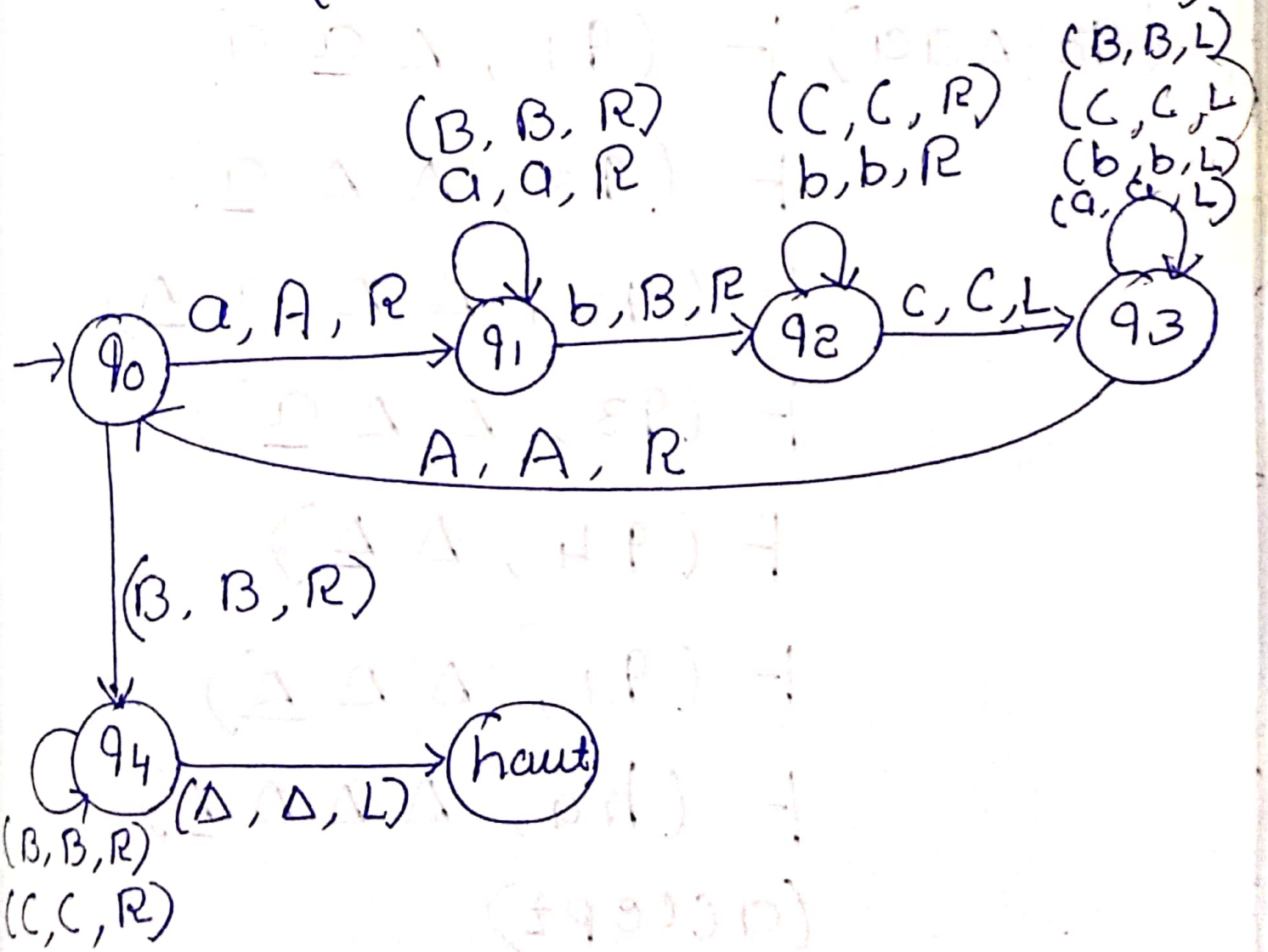
$\vdash$ (q_2, Δ AAABBBCCC)

$\vdash$ (q_3, Δ AAABBBCCC)

$\vdash$ (q_4, Δ AAA BBBCCCΔ)

R

$\vdash$ (Halt , ΔAAABBBCCC Δ)



ex-5 Construct TM for language accepting Palindrome.

⇒ Tracing

$(q_0, \Delta aa) \vdash (q_1, \Delta \underline{a}\, a)$

$\vdash (q_2, \Delta\, \Delta\, \underline{a})$

$\vdash (q_2, \Delta\, \Delta\, a\, \underline{\Delta})$

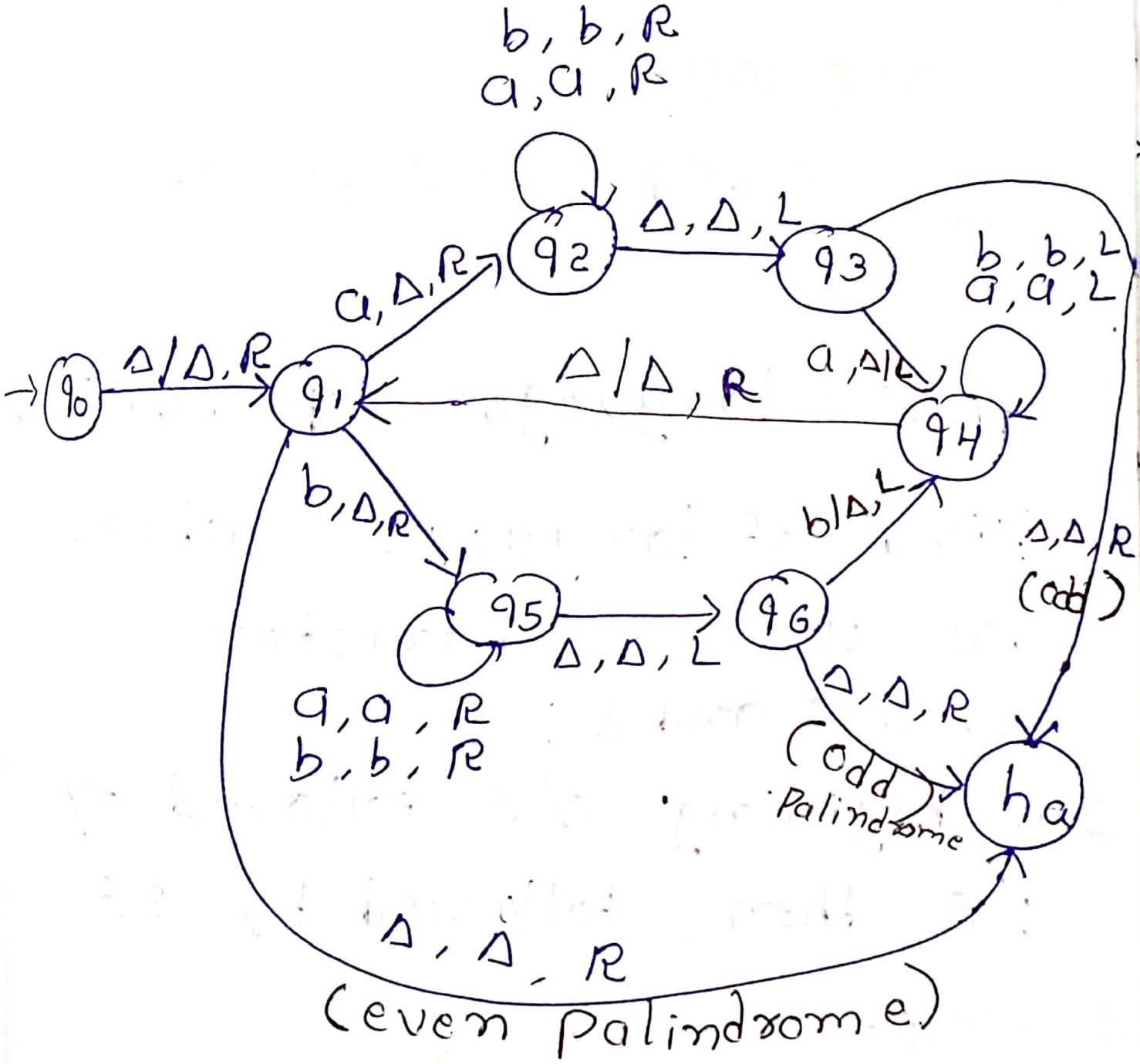
$\vdash (q_3, \Delta\, \Delta\, \underline{a})$

$\vdash (q_4, \Delta\, \underline{\Delta})$

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

$\vdash (h_a, \Delta\, \Delta\, \Delta\, \underline{\Delta})$

(accept)


b, b, R
a, a, R
q0
q1
q2
q3
q4
q5
q6
ha
Δ/Δ, R
a, Δ, R
Δ, Δ, L
b, b, L
a, a, L
a, Δ/E
Δ/Δ, R
b, Δ, R
b/Δ, L
Δ, Δ, R
(odd)
Δ, Δ, L
q, a, R
b, b, R
Δ, Δ, R
(odd Palindrome)
Δ, Δ, R
(even palindrome)

ex=6 Construct TM for the language

$$L = \{ a^n b^m 0^n 1^n 2^n \mid n \geq 1 \}$$

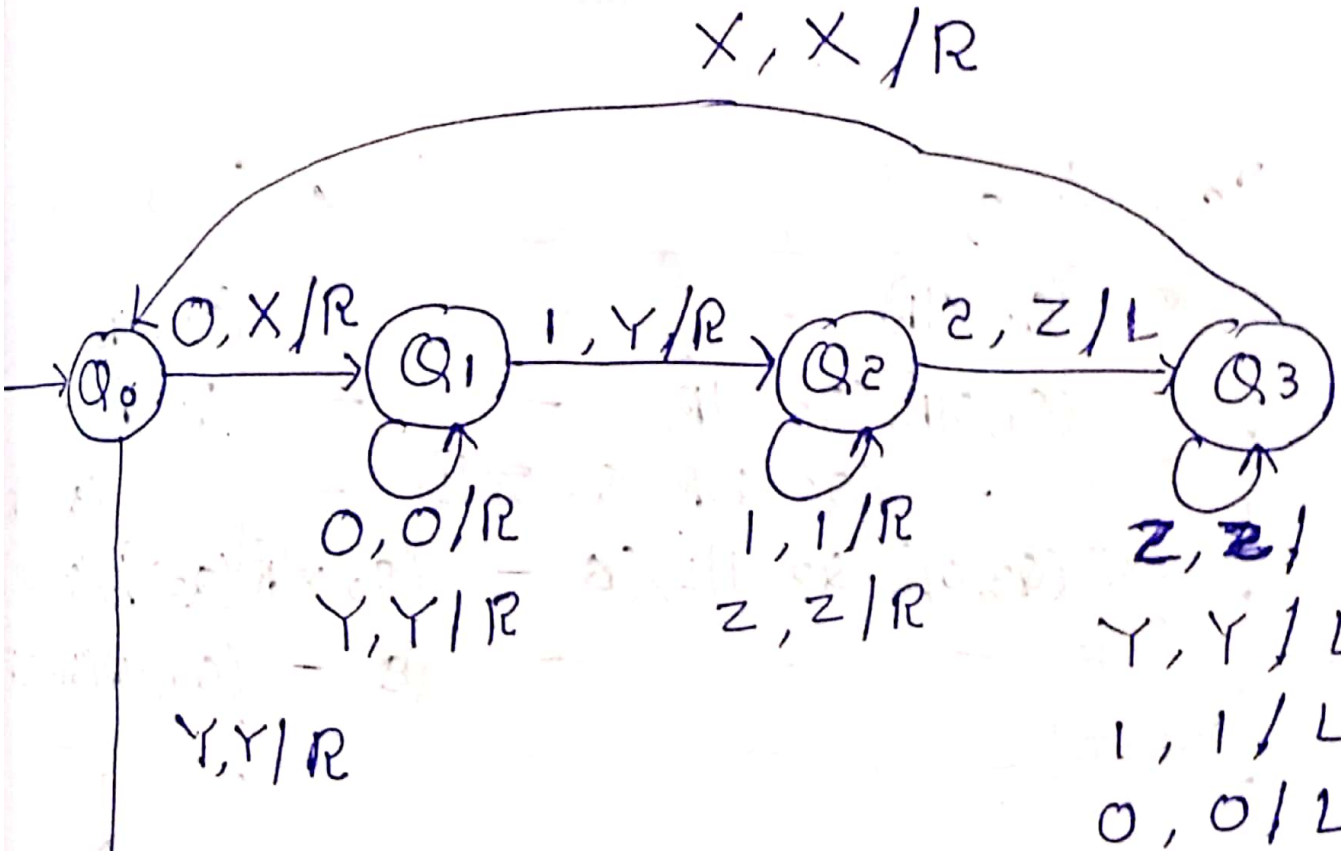
$\Rightarrow$

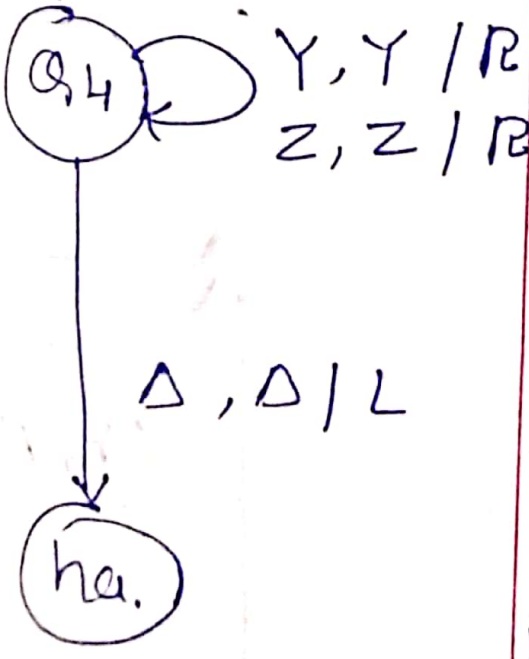
these language represent a kind of language where we use 3 character. i.e. 0, 1 and 2.

In begining 0's followed by 1's then followed by 2's.

Assumption :-

replace 0 by X
1 by Y
2 by Z.





String Tracing

$(q_0, \Delta 0011220\Delta)$

$\vdash (q_1, \Delta X 011220\Delta)$

$\vdash (q_1, \Delta X 0\underline{0}11220\Delta)$

$\vdash (q_1, \Delta X 0Y1220\Delta)$

$\vdash (q_2, \Delta X 0Y\underline{1}220\Delta)$

$\vdash (q_3, \Delta X 0Y1Z20\Delta)$

$\vdash (q_0, \Delta X 0Y1Z20\Delta)$

$\vdash (q_1, \Delta XXY1Z20\Delta)$

$\vdash (q_1, \Delta XXY1Z20\Delta)$

$\vdash (q_2, \Delta XXYYZ20\Delta)$

$\vdash (q_3, \Delta XXYYZZ0\Delta)$

Transition Table

State	0	1	2	X	Y	Z
Q_0	$(Q_1, X\|R)$	–	–	–	$(Q_4, Y\|R)$	–
Q_1	$(Q_1, 0\|R)$	$(Q_2, Y\|R)$	–	–	$(Q_1, Y\|R)$	–
Q_2	–	$(Q_2, 1\|R)$	$(Q_3, Z\|L)$	–	–	$(Q_2, Z\|$
Q_3	$(Q_3, 0\|L)$	$(Q_3, 1\|L)$	–	$(Q_0, X\|R)$	$(Q_3, Y\|L)$	$(Q_3, Z\|L$
Q_4	–	–	–	–	$(Q_4, Y\|R)$	$(Q_4, Z\|$
Halt	–	–	–	–	–	–

logic

Δ 0 0 1 1 2 2 Δ

↓ – ↓ ↓

X 0 Y 1 Z 2 Δ

R R R R

L L L L

X X Y Y Z Z Δ

halt

1) Construct a TM for Language $L = \{0^{2n}1^n \mid n \geq 0\}$

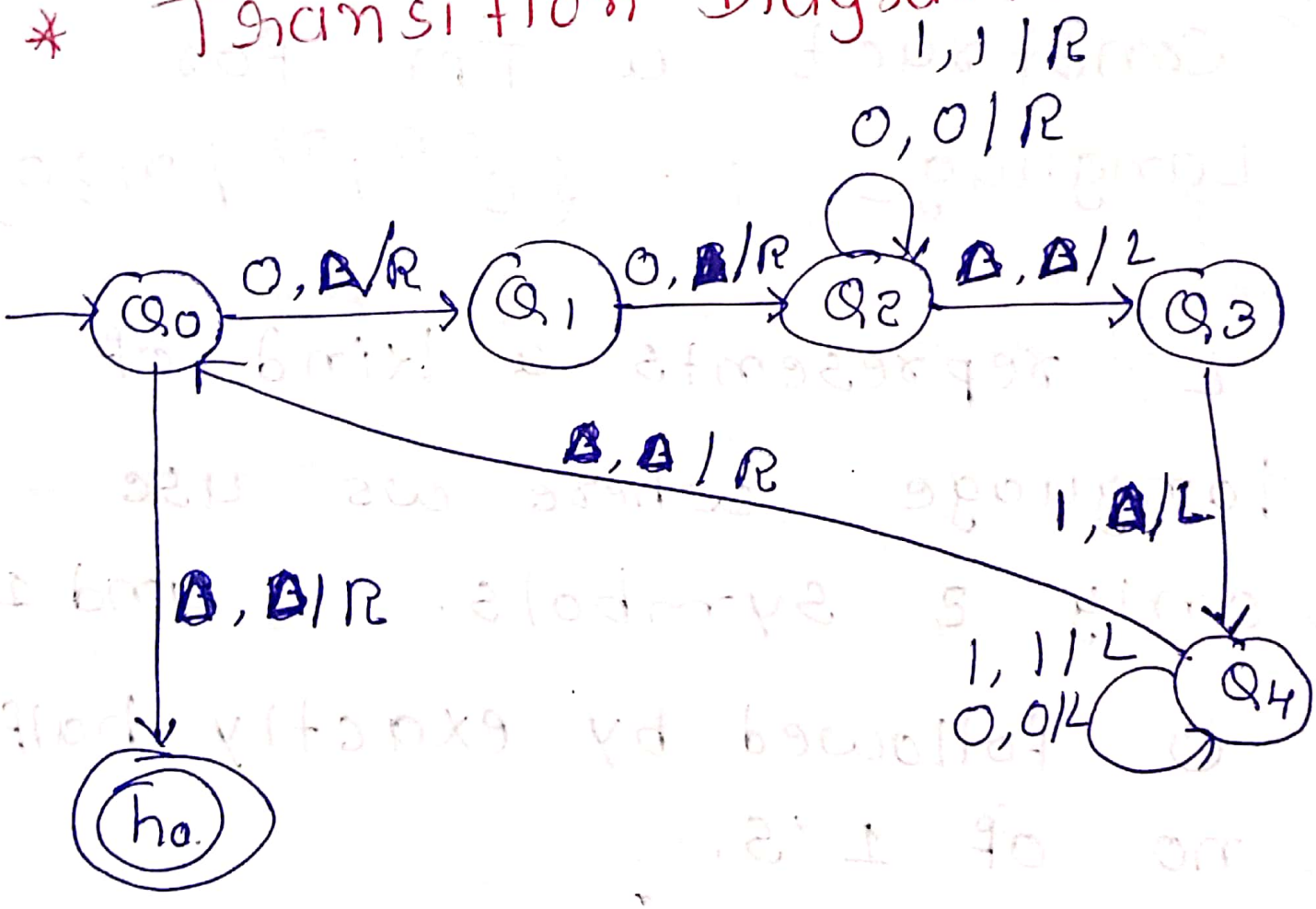
⇒ L represents a kind of language where we use only 2 symbols. 0 and 1.

⇒ 0 followed by exactly half no. of 1's.

Input :- 001
output :- YES

Input :- ∧
output :- YES.

* Transition Diagram.



Blank Symbol $B = \Delta$

* Transition Table.

State	0	1	Δ
q_0	$(q_1, \Delta/R)$	–	$(h_a, \Delta/R)$
q_1	$(q_2, \Delta/R)$	–	–
q_2	$(q_2, 0/R)$	$(q_2, 1/R)$	$(q_3, \Delta/L)$
q_3	–	$(q_4, \Delta/L)$	–
q_4	$(q_4, 0/L)$	$(q_4, 1/L)$	$(q_0, \Delta/R)$
h_a	–	–	–

Tracing Input String

"00011"

$(q_0, \Delta \underline{0}00011\Delta)$

$\vdash (q_1, \Delta\Delta \underline{0}0011\Delta)$

$\vdash (q_2, \Delta\Delta\Delta \underline{0}011\Delta)$

$\vdash (q_2, \Delta\Delta\Delta 0\underline{0}11\Delta)$

$\vdash (q_3, \Delta\Delta\Delta 00\underline{1}1\Delta)$

$\vdash (q_4, \Delta\Delta\Delta 001\underline{1}\Delta)$

$\vdash (q_3, \Delta\Delta\Delta 00\underline{1}\Delta\Delta)$

$\vdash (q_4, \Delta\Delta\Delta 00\underline{\Delta}\Delta\Delta)$

$\vdash (q_0, \Delta\Delta\Delta 00\underline{1}\Delta\Delta)$

$\vdash (q_0, \Delta\Delta\Delta\Delta \underline{0}1\Delta\Delta)$

$\vdash (q_1, \Delta\Delta\Delta\Delta 0\underline{1}\Delta\Delta)$

$\vdash (q_2, \Delta\Delta\Delta\Delta\Delta \underline{1}\Delta\Delta)$

$\vdash (q_3, \Delta\Delta\Delta\Delta\Delta 1\Delta\Delta)$

$\vdash (q_4, \Delta\Delta\Delta\Delta\Delta\Delta\Delta\Delta)$

$\vdash (q_0, \Delta\Delta\Delta\Delta\Delta\Delta\Delta\Delta) \vdash$ halt (Acc)

00 = ΔΔ

1 - Δ

ex-8 Construct a TM for

$L = \{a^i b^j c^k / i < j < k$

$i \geq 1\}$

⇒ Approch

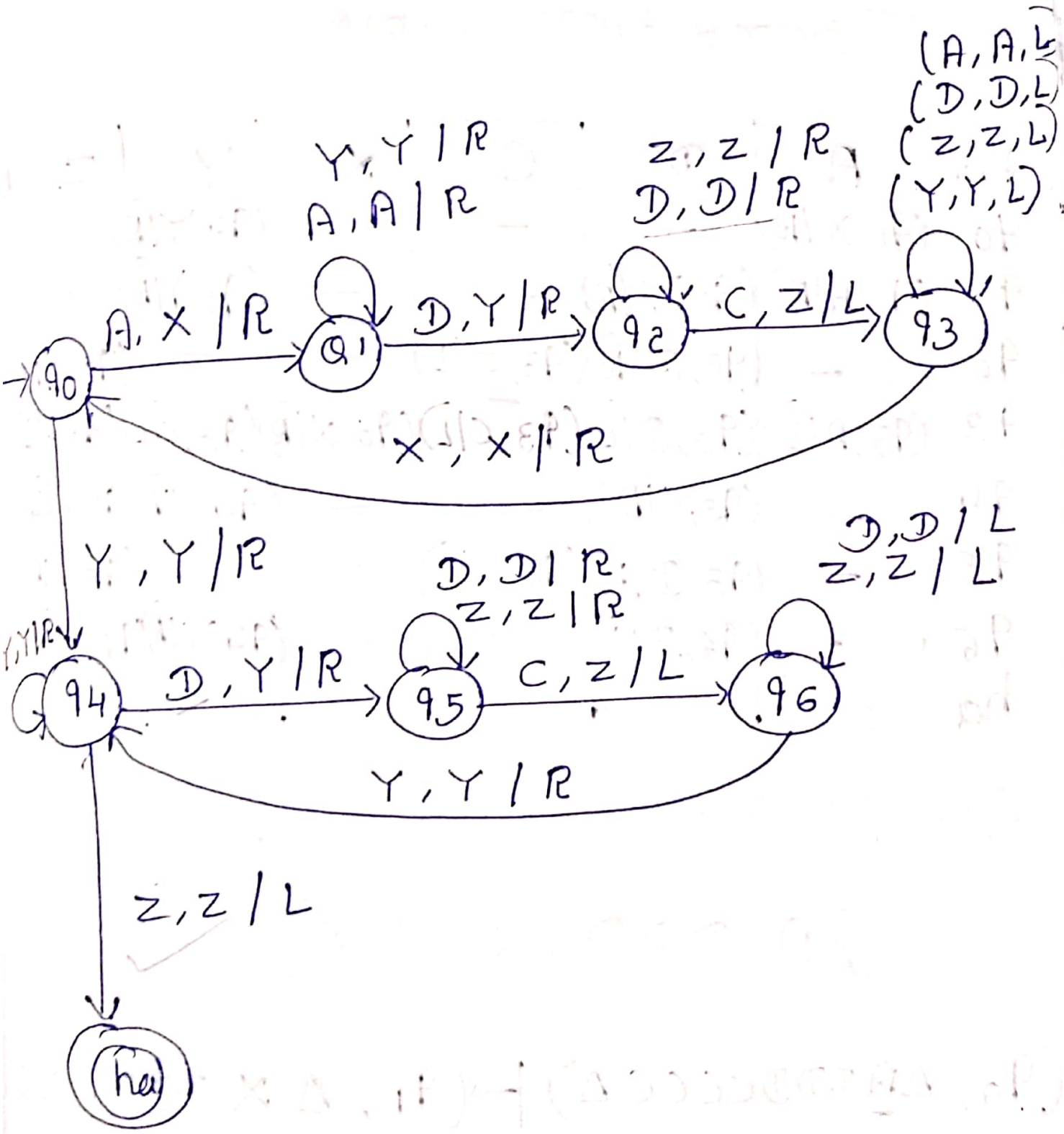
→ Comparing two elements by making D and C as a single element.

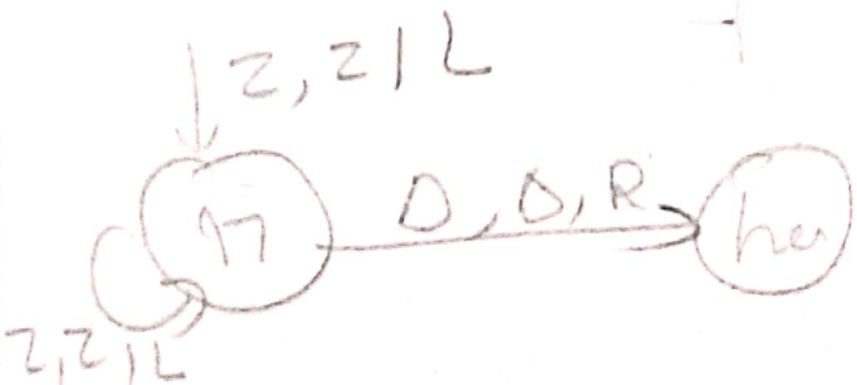
→ After comparing D & C.

→ if |A| is greater than |(D, C)|, then it is not accepted, o

→ if |C| is greater than |D|, then it is not accepted

→ else it is accepted.





*Like comparator.

* Transition Table.

State	A	D	C	X	Y	Z	Δ
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	–	$(q_2, D/R)$	$(q_3, Z/L)$	–	–	$(q_2, Z/R)$	
q_3	$(q_3, A/L)$	$(q_3, D/L)$	–	$(q_0, X/R)$	$(q_3, Y/L)$	$(q_3, Z/L)$	
q_4	–	$(q_5, Y/R)$	–	–	$(q_4, Y/R)$	$(q_5, Z/L)$	
q_5	–	$(q_5, D/R)$	$(q_6, Z/L)$	–	–	$(q_5, Z/R)$	
q_6	–	$(q_6, D/L)$	–	–	$(q_4, Y/R)$	$(q_6, Z/L)$	
h_a	–	–	–	–	–	–	

ex 9 Construct TM for Concatenation of two string of unary numbers.

⇒ The TM will be construct for Concatenate two string of unary no. The strings are placed as follows.

⇒ The two strings are separated by $, we will replace that $ by 1.

Let us assume that on i/p tape the input is :—

Action (next)

1 1 1 $ 1 1 1 Δ	Move Right till $
1 1 1 $ 1 1 1 Δ	Move Right
1 1 1 $ 1 1 1 Δ	if current symbol is 1 move left and convert $ to 1
1 1 1 1 1 1 1 Δ	Now move Right and convert 1 to $
1 1 1 1 $ 1 1 Δ	Move Right
1 1 1 1 $ 1 1 Δ	if current symbol is 1 move left and convert $ to 1 and move Right
1 1 1 1 1 1 1 Δ	Convert 1 to $ and move Right
1 1 1 1 1 $ 1 Δ	~~convert 1 to $ and~~ move Right.
1 1 1 1 1 $ 1 Δ	move Right.

1 1 1 1 1 $ 1 Δ

↑

if from left current symbol is 1 move left convert $ to 1

1 1 1 1 1 1 1 Δ

↑

Move Right convert 1 to $

1 1 1 1 1 $ Δ

↑

move Right

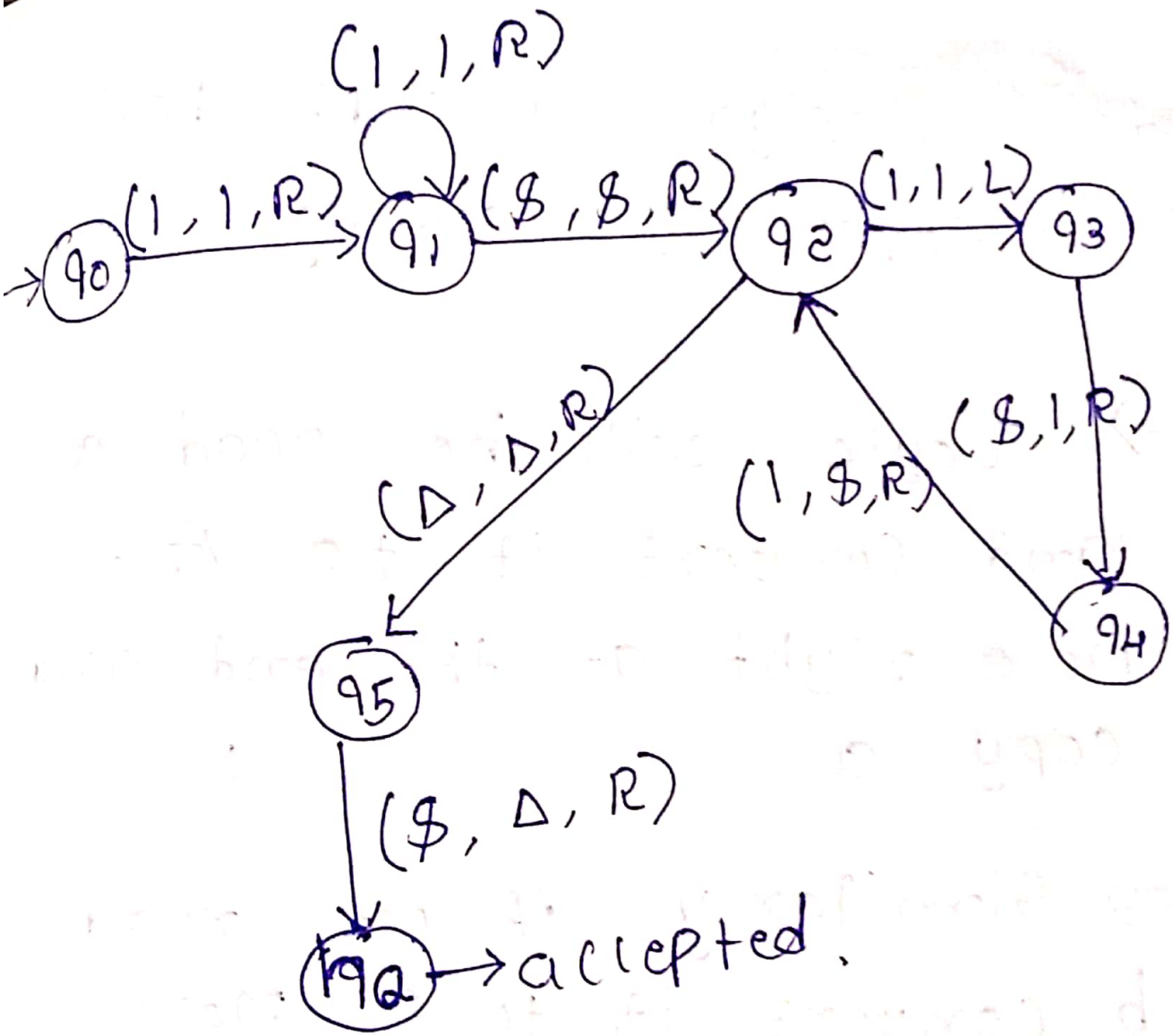
1 1 1 1 1 $ Δ

↑

Since it is end of i/p we will move left convert $ to Δ and halt.

1 1 1 1 1 Δ

* Diagram :-



State	1	Δ	Δ	
q_0	$(q_1, 1, R)$	–	–	
q_1	$(q_1, 1, R)$	$(q_2, Δ, R)$	–	
q_2	$(q_3, 1, L)$	–	$(q_5, Δ, R)$	
q_3	$(q_4, 1, R)$	$(q_4, 1, R)$	–	
q_4	$(q_2, Δ, R)$	–	–	
q_5	–	$(q_a, Δ, R)$	–	
h_a	–	–	–	

ex-10

Design a TM for copy a string.

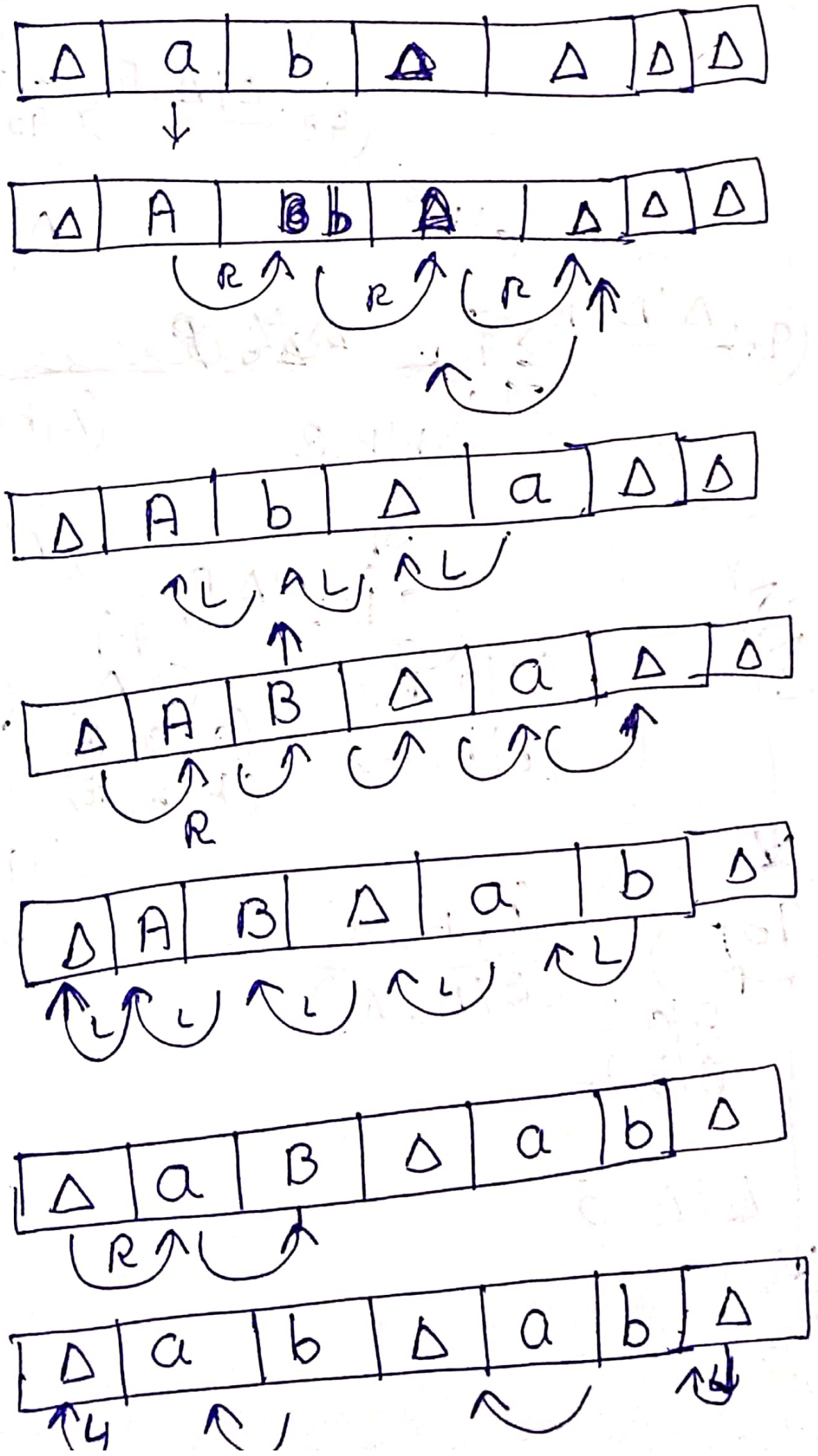
⇒ Logic .. will be read a and convert it to A. Move right at the end and copy a.

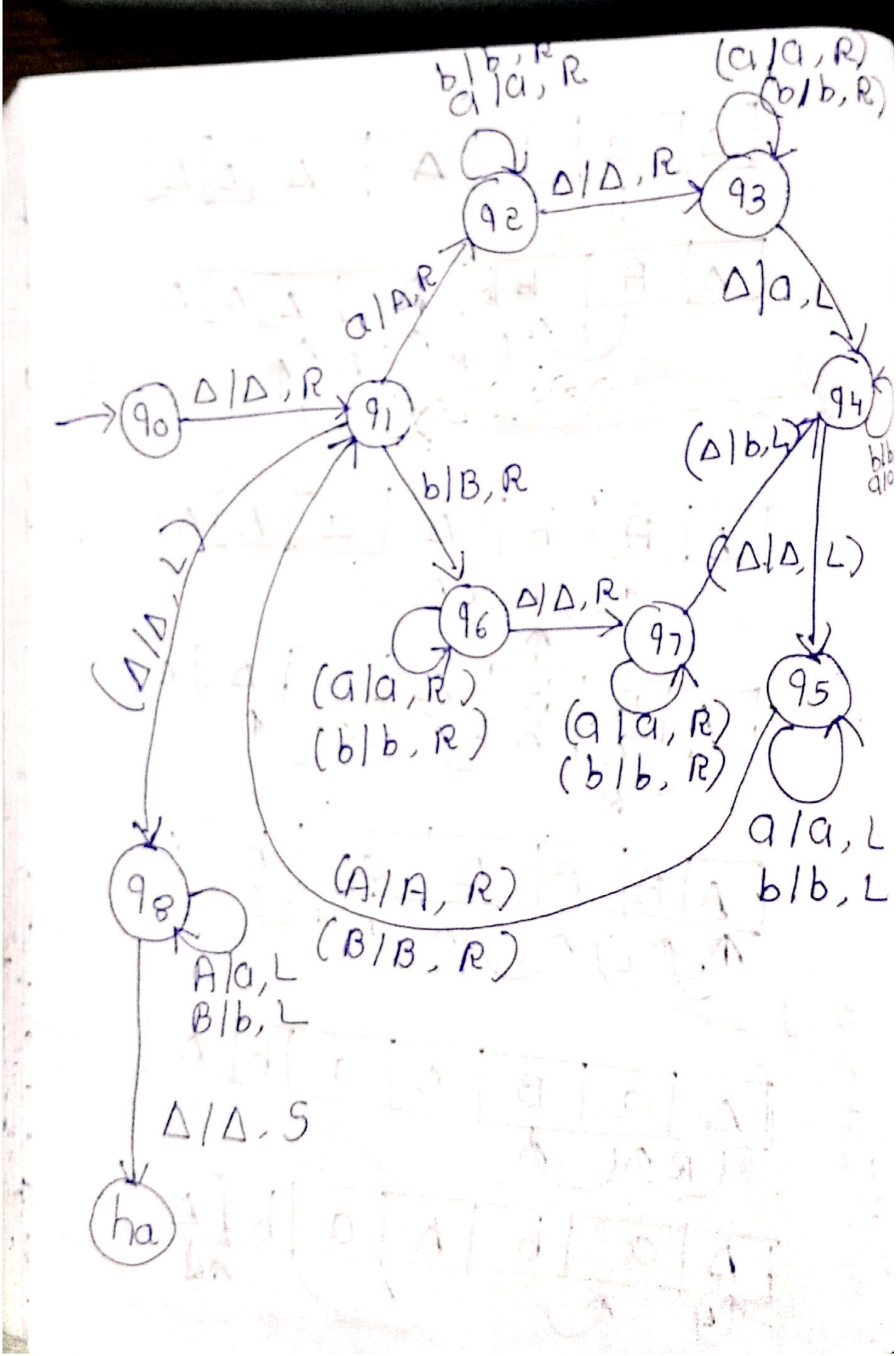
⇒ Similarly if we read b convert it to B. Move Right at end and copy b.

Finally we get S Δ S.

→ eliminate Δ and shift S ahead so SS Δ remains on tape.

$$\frac{\Delta ab}{\text{original}} \quad \frac{\Delta ab}{\text{Pasted}}$$


Δ a b Δ Δ Δ Δ
Δ A B b A Δ Δ Δ
R R R R
Δ A b Δ a Δ Δ
L L L
Δ A B Δ a Δ Δ
R
Δ A B Δ a b Δ
L L L L
Δ a B Δ a b Δ
R
Δ a b Δ a b Δ
L L


q0
q1
q2
q3
q4
q5
q6
q7
q8
ha
Δ/Δ, R
a/A, R
b/B, R
b/b, R
a/a, R
Δ/a, L
Δ/b, L
Δ/Δ, L
(a/a, R)
(b/b, R)
A/a, L
B/b, L
Δ/Δ, S
(A/A, R)
(B/B, R)
a/a, L
b/b, L

Transition Table

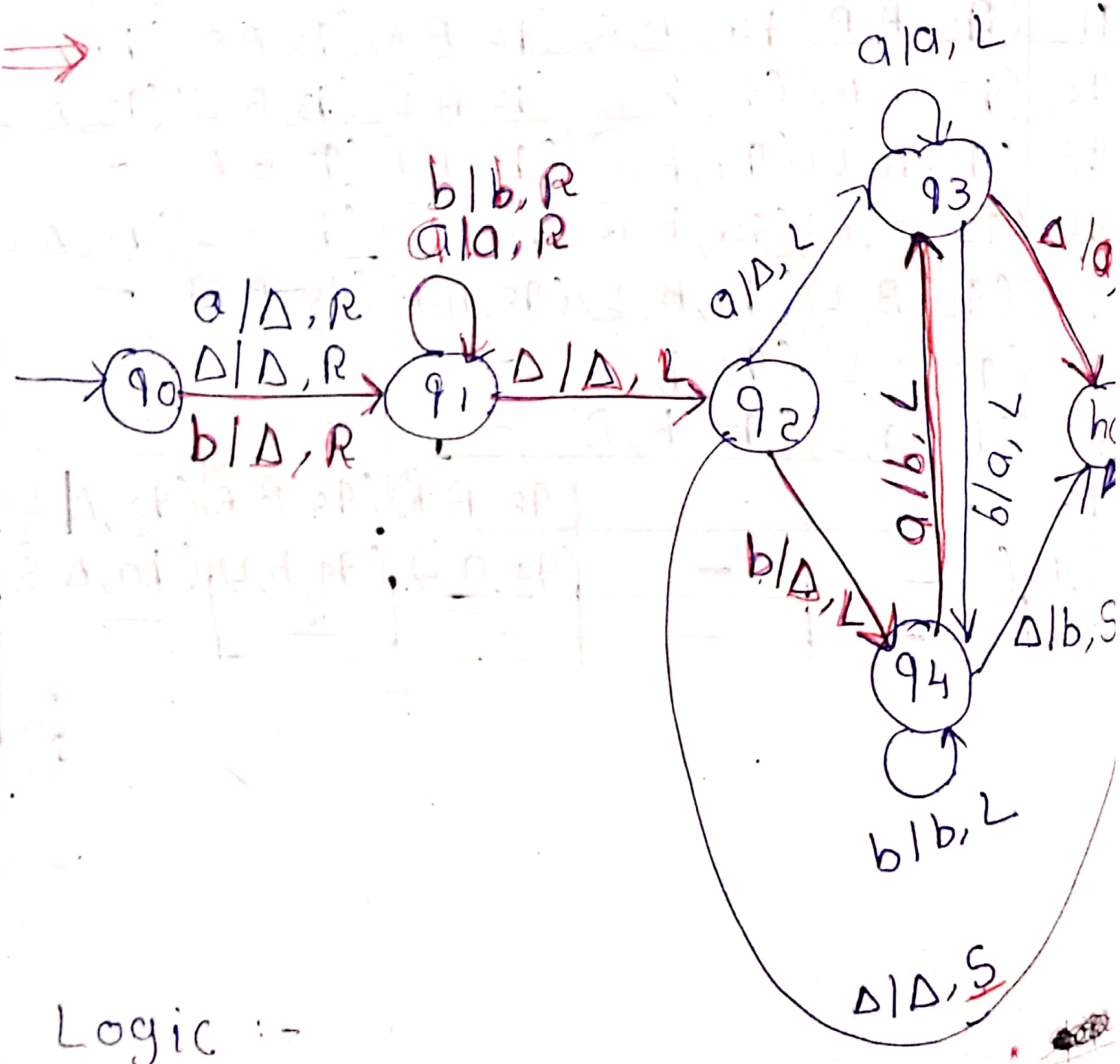
DPS.

State	a	b	A	B	Δ
q_0	–	–	–	–	(q_1, Δ, R)
q_1	(q_2, A, R)	(q_6, B, R)	–	–	(q_8, Δ, L)
q_2	(q_2, a, R)	(q_2, b, R)	–	–	(q_3, Δ, R)
q_3	(q_3, a, R)	(q_3, b, R)	–	–	(q_4, a, L)
q_4	–	–	–	–	(q_5, Δ, L)
q_5	(q_5, a, L)	(q_5, b, L)	(q_1, A, R)	(q_1, B, R)	–
q_6	(q_6, a, R)	(q_6, b, R)	–	–	(q_7, Δ, R)
q_7	(q_7, a, R)	(q_7, b, R)	–	–	(q_4, b, L)
q_8	–	–	(q_8, a, L)	(q_8, b, L)	(h_a, Δ, S)
h_a	–	–	–	–	–

DPS.

Ex 12

Design a TM for delete a symbol.

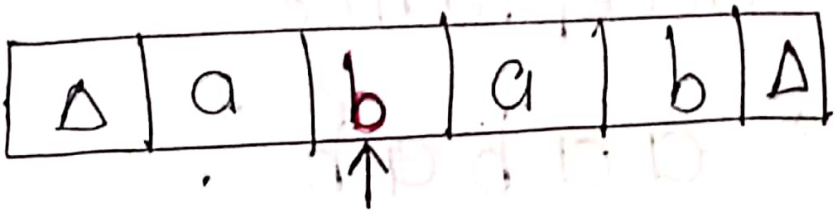


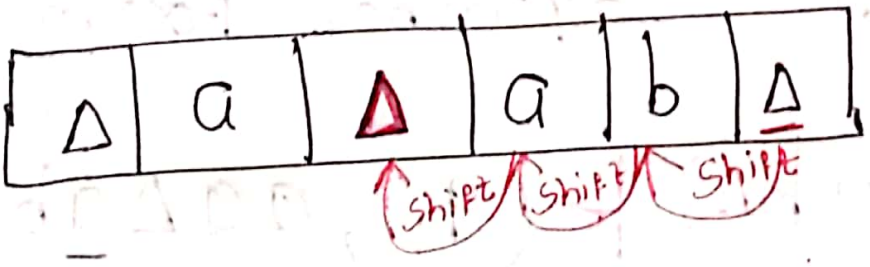
Logic :-

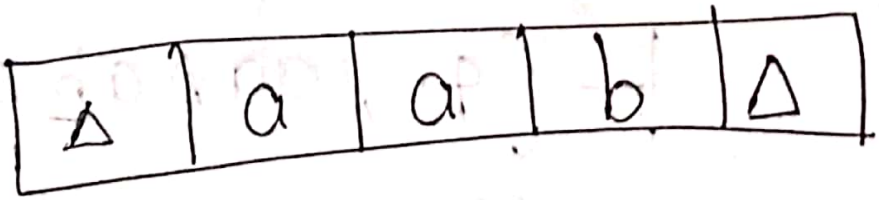
1. Replace the Symbol you want to delete by Δ.
2. Keep moving Right till you reach

DPS.

Logic :- output :- ΔaabΔ.







Δ / a, Shift
a / b, Shift
b / Δ, Shift.

---

3. one by one keep shifting every symbol one step to left until you reach Δ.

Ex-2.2.3

String acceptance

"aabab"

delete 3rd symbol from Right

$(q_0, aa\underline{b}ab\Delta) \vdash (q_1, aa\Delta\underline{a}b\Delta)$

$\vdash (q_1, aa\Delta a\underline{b}\Delta)$

$\vdash (q_1, aa\Delta ab\underline{\Delta})$

$\vdash (q_2, aa\Delta a\underline{b}\Delta)$

$\vdash (q_4, aa\Delta\underline{a}\Delta\Delta)$

$\vdash (q_3, aa\underline{\Delta}b\Delta\Delta)$

$\vdash (h_a, aaab\Delta)$

String accepted with 3rd symbol delete from Right.

DPS.

String acceptance

"ababaa"

delete 3rd symbol from left side.

$(q_0, ababaa) \vdash (q_1, aba\Delta\underline{a}a\Delta)$

$\vdash (q_1, aba\Delta a\underline{a}\Delta)$

$\vdash (q_1, aba\Delta aa\underline{\Delta})$

$\vdash (q_2, aba\Delta a\underline{a}\Delta)$

$\vdash (q_3, aba\Delta\underline{a}\Delta\Delta)$

$\vdash (q_3, aba\underline{\Delta}a\Delta\Delta)$

$\vdash (q_3, abaaa\Delta)$

ha

String accepted with 3rd symbol delete from left side.

Ex=13

Construct a TM for a language equal no. of a's and b's in it over the input set $\Sigma = \{a, b\}$.

Or

$\{X \in \{a,b\}^* / n_a(x) = n_b(x)\}$

⇒ logic

→ if we get a we will replace it by A and move right in search of b.

→ if we get b we will replace it by B and move left.

→ we will move to the left most Δ and again repeat the same process of marking a by A and b by B.

⇒ we may have string starting with a or b, so, from initial state only we will have two path for a and b.

$\Rightarrow$ Δ b a a b Δ — Convert b to B and move right in search of a.

Δ B a a b Δ — a has to be converted to A and move left

Δ B A a b Δ — Skip B and move left.

Δ B A a b Δ — Keep Δ as it and move Right

Δ B A a b Δ — Move right

Δ B A a b Δ — Move right

Δ B A a b Δ — Convert a to A and move right.

Δ B A A b Δ — Convert b to B and move left.

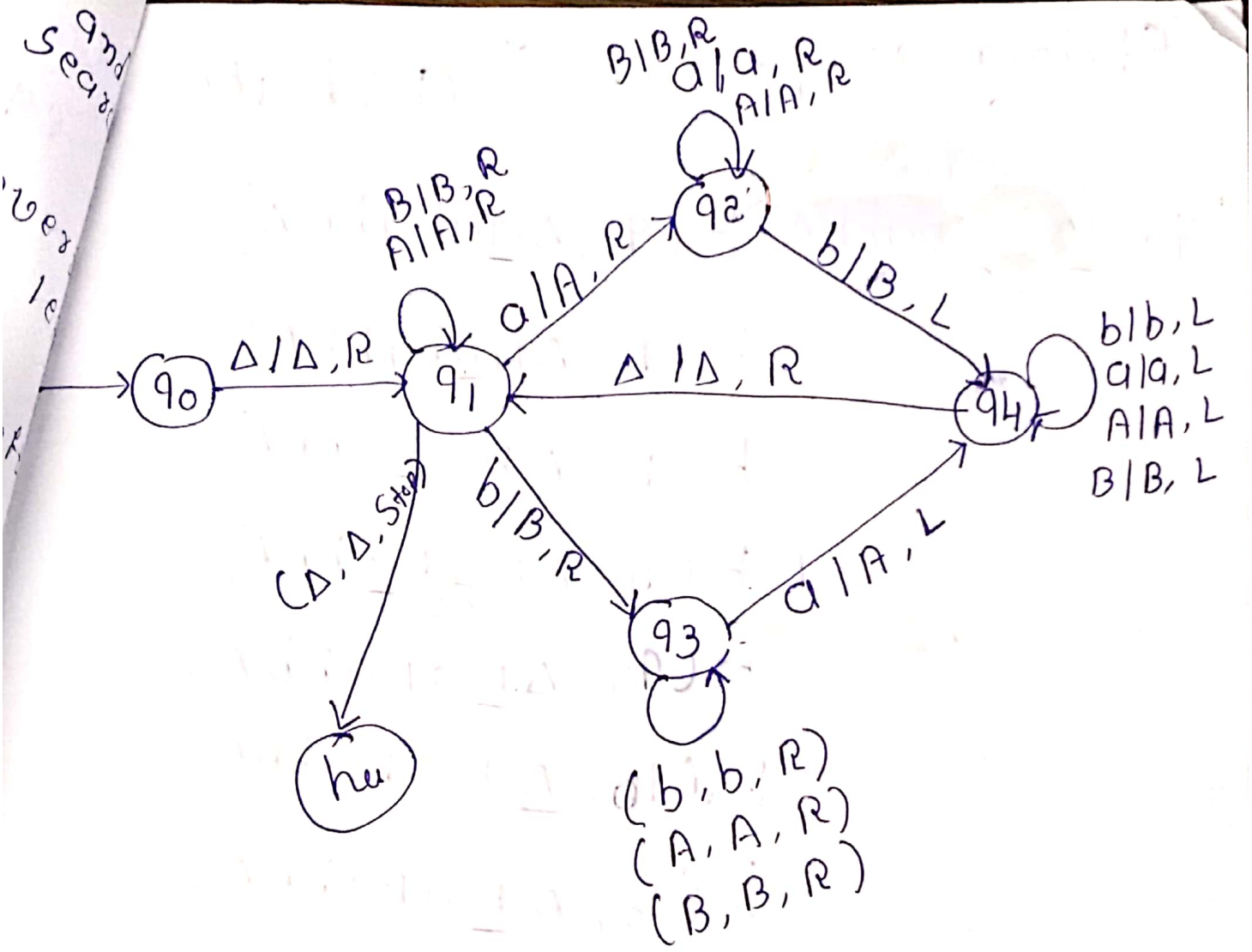
Δ B A A B Δ — Move left

Δ B A A B Δ — Move left

Δ B A A B Δ — Move left

Δ B A A B Δ — if Δ comes move to right by ignoring all A's and B's

Δ B A A B Δ — Δ is reached go to halt



Transition Table :-

State	a	b	A	B	Δ
q_0	–	–	–	–	(q_1, Δ, R)
q_1	(q_2, A, R)	(q_3, B, R)	(q_1, A, R)	(q_1, B, R)	(ha, Δ, S)
q_2	(q_2, a, R)	(q_4, B, L)	(q_2, A, R)	(q_2, B, R)	–
q_3	(q_4, A, L)	(q_3, b, R)	(q_3, A, R)	(q_3, B, R)	–
q_4	(q_4, a, L)	(q_4, b, L)	(q_4, A, L)	(q_4, B, L)	(q_1, Δ, R)
ha	–	–	–	–	–

String acceptance "aababb"

$(q_0, \Delta aababb\Delta) \vdash (q_1, \Delta a ababb\Delta)$

$\vdash (q_2, \Delta A a babb\Delta)$

$\vdash (q_2, \Delta A a b abb\Delta)$

$\vdash (q_4, \Delta A a B abb\Delta)$ (L)

$\vdash (q_4, \Delta A a B abb\Delta)$

$\vdash (q_0, \Delta A a B a bb\Delta)$

$\vdash (q_0, \Delta A a B abb\Delta)$

$\vdash (q_1, \Delta A a B abb\Delta)$

$\vdash (q_2, \Delta A A B abb\Delta)$

$\vdash (q_2, \Delta A A B a bb\Delta)$

$\vdash (q_2, \Delta A A B a b b\Delta)$

$\vdash (q_4, \Delta A A B a B b\Delta)$ (L)

$\vdash (q_4, \Delta A A B a B b\Delta)$ (L L L)

$\vdash (q_1, \Delta A A B a B b\Delta)$

$\vdash (q_1, \Delta A A B a B b\Delta)$ (R R)

$\vdash (q_2, \Delta A A B A B b\Delta)$ (R)

$\vdash (q_2, \Delta AABABb\Delta)$ (R)

$\vdash (q_4, \Delta AABABB\Delta)$ (L)

$\vdash (q_4, \Delta AABABB\Delta)$ (L L L L)

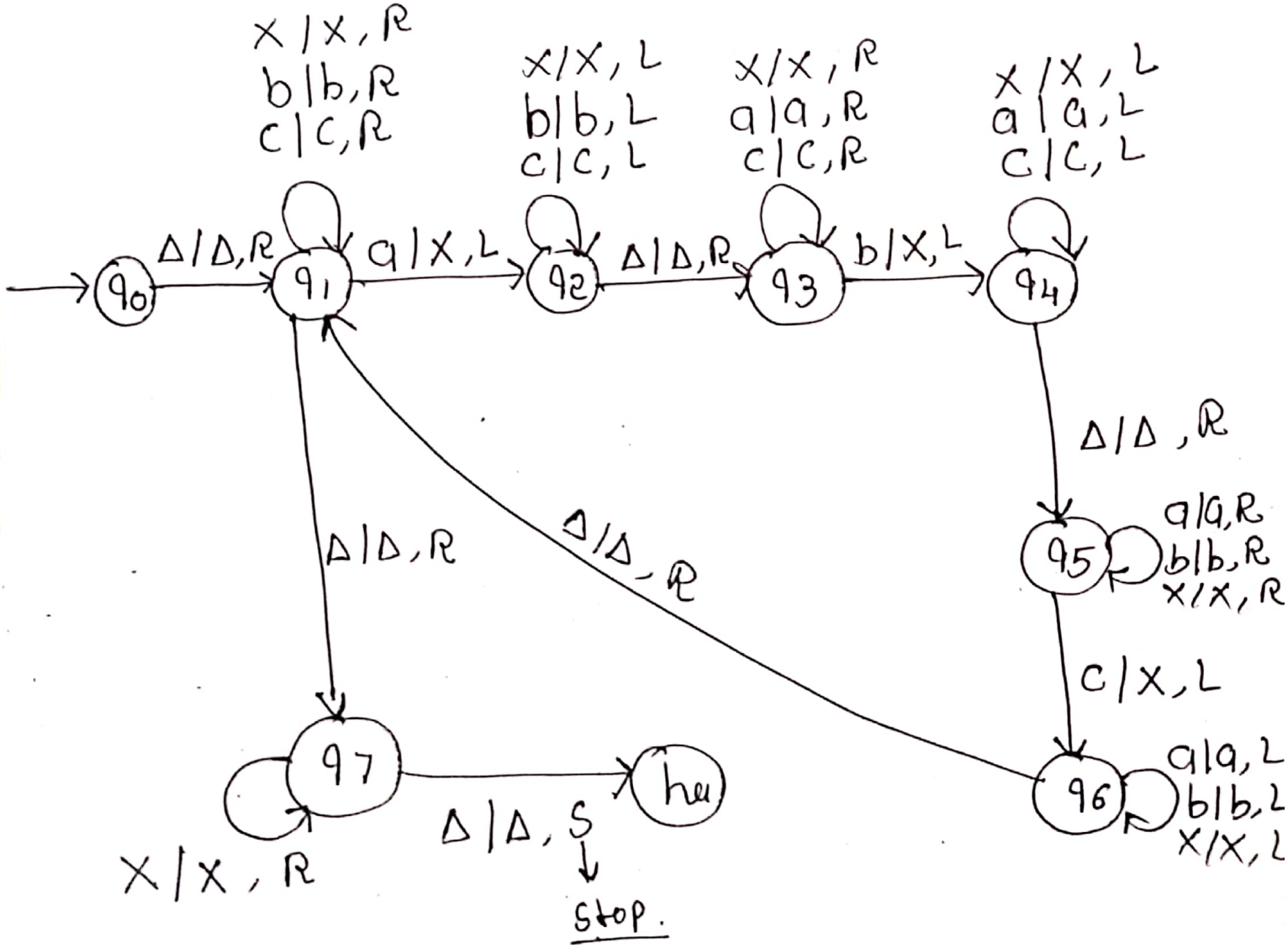
$\vdash (q_1, \Delta AABABB\Delta)$ (R)

$\vdash (h_a, \Delta AABABB\Delta)$

accepted.

Ex-14 Construct a TM for a lang having equal no. of a's a b's and c's in it over th input set $\Sigma = \{a, b, c\}$

OR Design TM $\{x \in \{a,b,c\}^* n_a(x) = n_b(x) = n_c(x)\}$



String acceptance "ababcc"

$(q_0, \underline{\Delta}ababcc\Delta) \vdash (q_1, \Delta \underline{a} babcc\Delta)$

$\vdash (q_2, \Delta a \times babcc\Delta)$

$\vdash (q_2, \Delta \times \underline{b} abcc\Delta)$

$\vdash (q_3, \Delta \times b \underline{a} bcc\Delta)$

$\vdash (q_3, \Delta \times \underline{b} abcc\Delta)$

$\vdash (q_4, \Delta \underline{X} \times abcc\Delta)$

$\vdash (q_4, \Delta \times \underline{X} abcc\Delta)$

$\vdash (q_5, \Delta \underline{X} \times abcc\Delta)$

$\vdash (q_5, \Delta \times \underline{X} abcc\Delta)$

$\vdash (q_5, \Delta \times \times \underline{a} bcc\Delta)$

$\vdash (q_5, \Delta \times \times a \underline{b} cc\Delta)$

$\vdash (q_5, \Delta \times \times ab \underline{c} c\Delta)$

$\vdash (q_6, \Delta \times \times a \underline{b} \times c\Delta)$

$\vdash (q_6, \Delta \times \times \underline{a} b \times c\Delta)$

$\vdash (q_6, \Delta \times \underline{X} \times ab \times c\Delta)$

$\vdash (q_6, \Delta \underline{X} \times ab \times c\Delta)$

$\vdash (q_6, \underset{R}{\Delta} \times \times ab \times C \Delta)$

$\vdash (q_1, \Delta \underline{\times} \times ab \times C \Delta)$

$\vdash (q_1, \Delta \times \underline{\times} ab \times C \Delta)$

$\vdash (q_1, \Delta \times \times \underline{a} b \times C \Delta)$

$\vdash (q_2, \Delta \times \underset{R}{\underline{\times}} \times b \times C \Delta)$

$\vdash (q_2, \Delta \underline{\times} \times \times b \times C \Delta)$

$\vdash (q_2, \underline{\Delta} \times \times \times b \times C \Delta)$

$\vdash (q_3, \Delta \underset{R}{\underline{\times}} \times \times b \times C \Delta)$

$\vdash (q_3, \Delta \times \underline{\times} \times b \times C \Delta)$

$\vdash (q_3, \Delta \times \times \underline{\times} b \times C \Delta)$

$\vdash (q_3, \Delta \times \times \times \underline{b} \times C \Delta)$

$\vdash (q_4, \Delta \times \times \underset{R}{\underline{\times}} \times \times C \Delta)$

$\vdash (q_4, \Delta \times \underline{\times} \times \times \times C \Delta)$

$\vdash (q_4, \Delta \underline{\times} \times \times \times \times C \Delta)$

$\vdash (q_4, \underline{\Delta} \times \times \times \times \times C \Delta)$

$\vdash (q_5, \Delta \underline{\times} \times \times \times \times C \Delta)$

$\vdash (q_5, \Delta \underline{X} X X X C \Delta)$

$\vdash (q_5, \Delta X \underline{X} X X C \Delta)$

$\vdash (q_5, \Delta X X \underline{X} X C \Delta)$

$\vdash (q_5, \Delta X X X \underline{X} C \Delta)$

$\vdash (q_5, \Delta X X X X \underline{C} \Delta)$

$\vdash (q_6, \Delta X X X X \underline{X} X \Delta)$

$\vdash (q_6, \underline{\Delta} X X X X X X \Delta)$

$\vdash (q_1, \Delta \underline{X} X X X X X \Delta)$

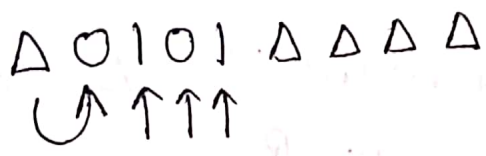
$\vdash (q_7, \Delta X \underline{X} X X X X \Delta)$

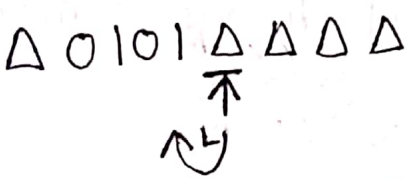
$\vdash (q_7, \Delta X X X X X X \Delta)$

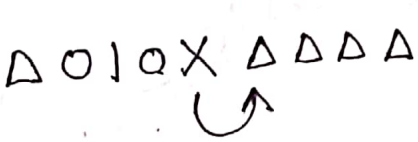
$\vdash (h_a, \Delta X X X X X X \Delta)$

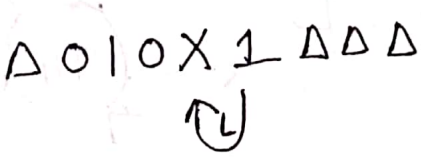
String accepted.

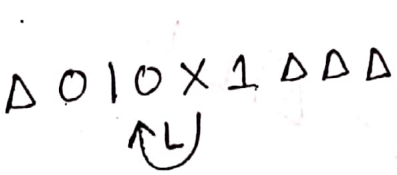
ex:-15 Design TM to reversing a string.

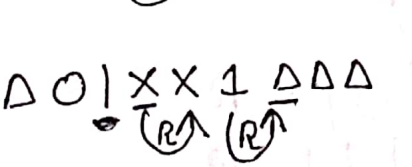

(read all and Right)

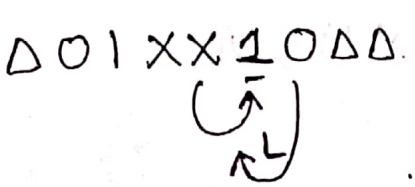

(shift left when Δ come)

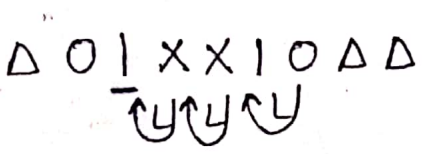

(convert 1 to X) Shift Right)

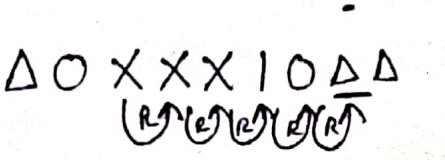

(convert Δ to 1 and Shift left)

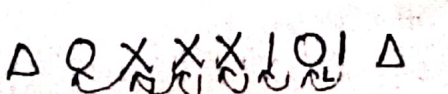

(Shift left)

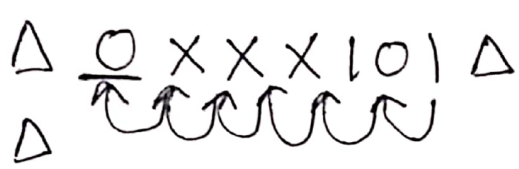

(convert 0 to X Shift Right)


(convert Δ to 0 and Shift left)

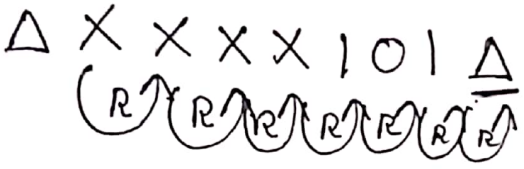

(Shift left)


(convert 1 to X Shift Right)

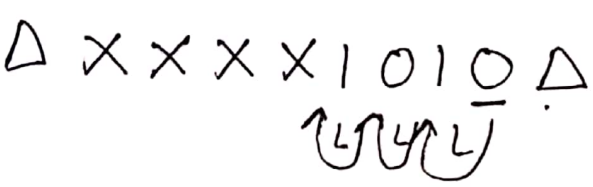

(convert Δ to 1, Shift left



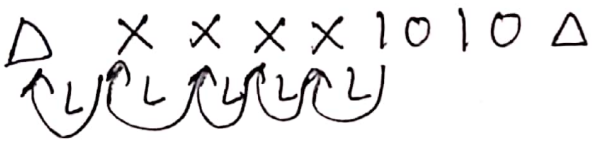
Shift left



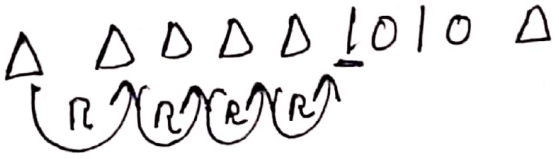
Convert 0 to X and move Right till Δ come.



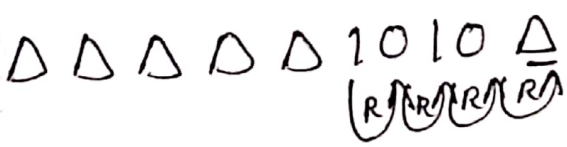
Convert Δ to 0 and Shift left



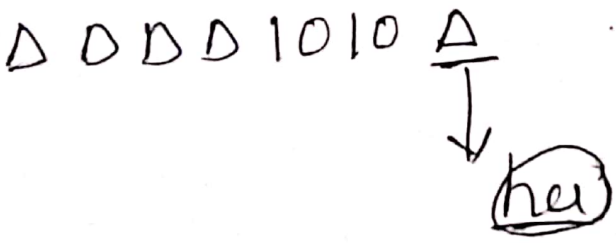
during left move Convert X to Δ till reach Δ



Shift Right.

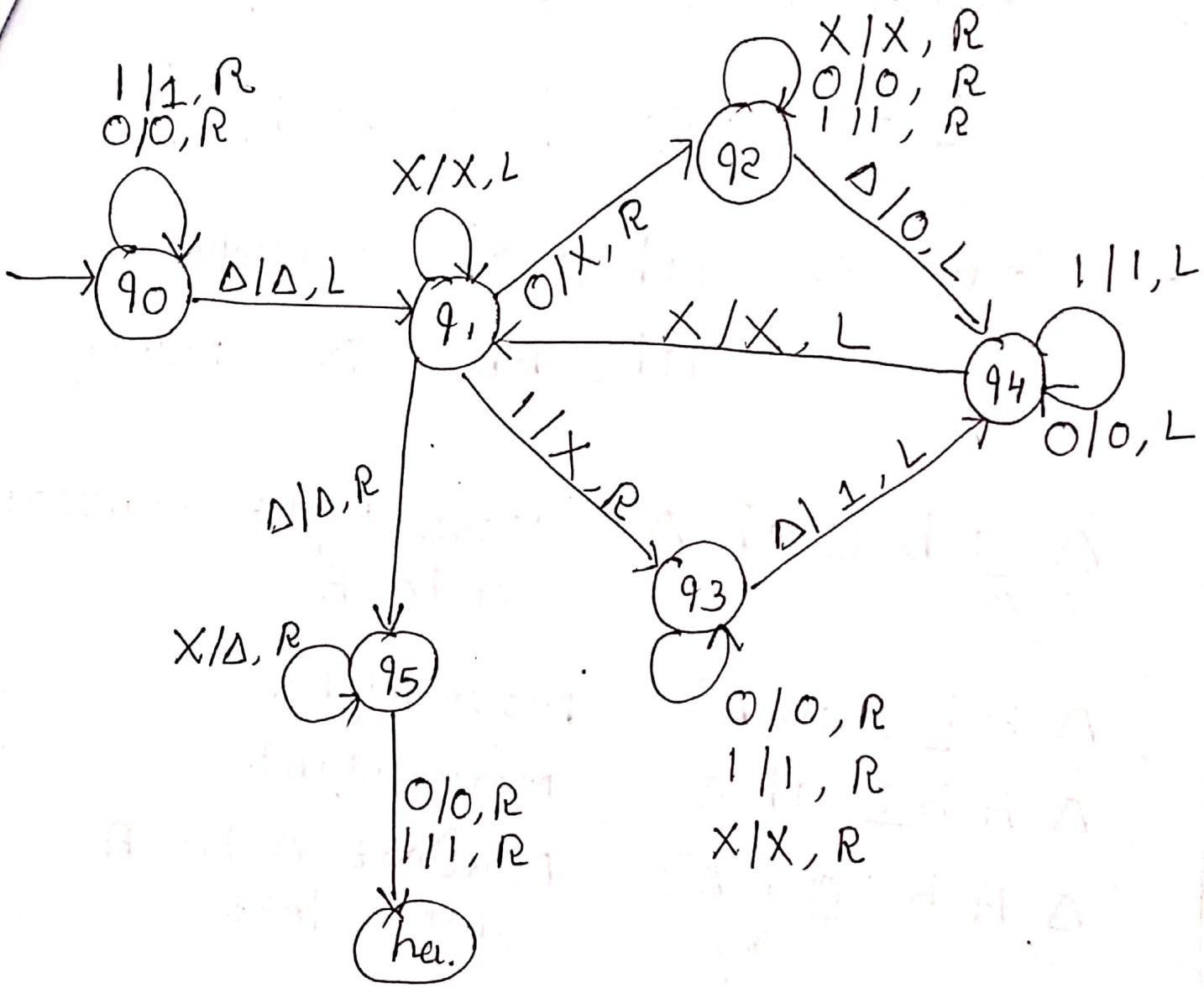


during Right move ignore 1 and 0 reach Δ.



String accepted.

Transition Diagram (TM)



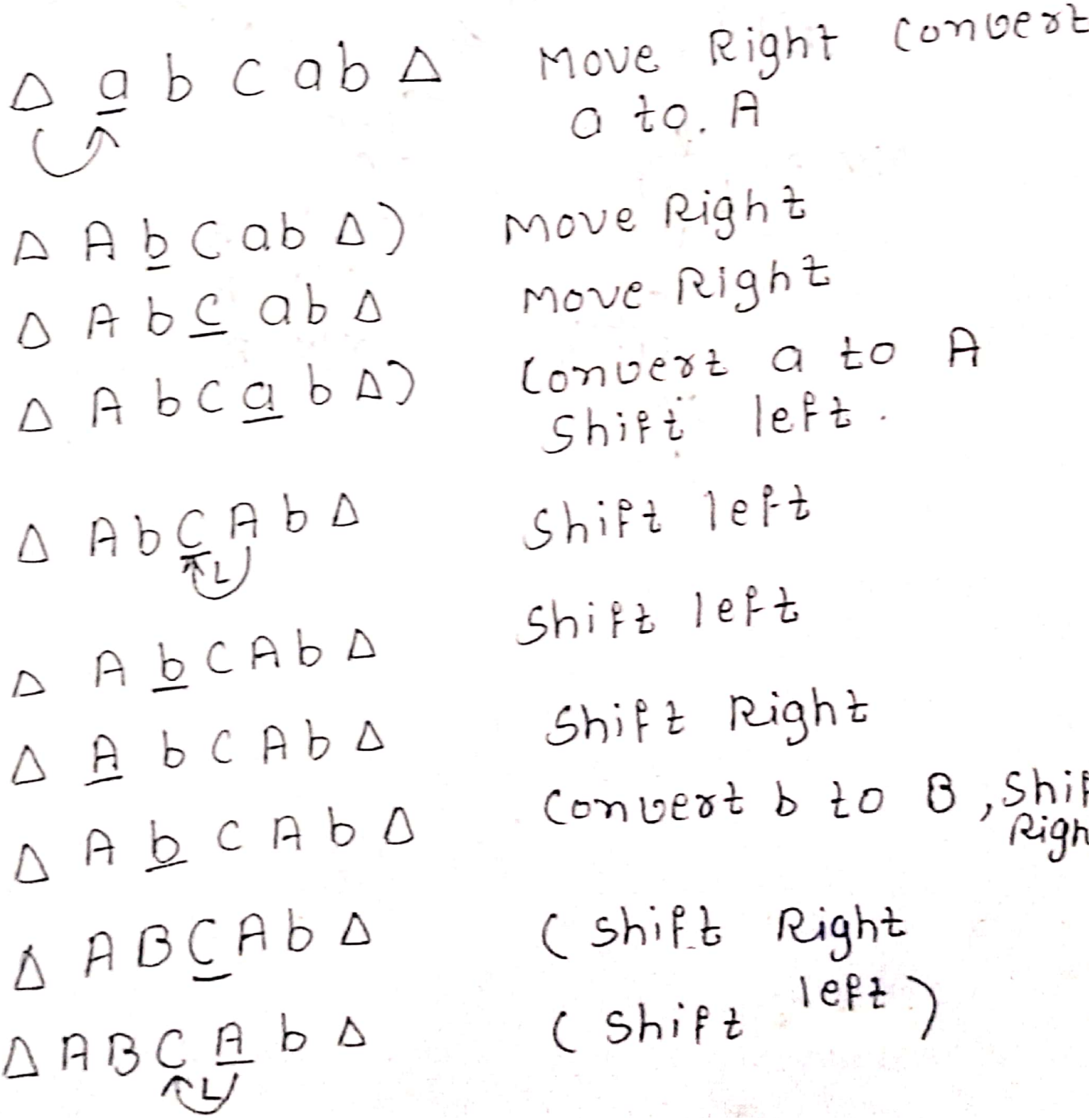
ex-16

Construct a TM for language

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

$\Rightarrow$ logic :- abcab ✓ accept

abcbb ✗ reject



Δ A B C A b Δ Shift left

Δ A B C A b Δ Shift left till A

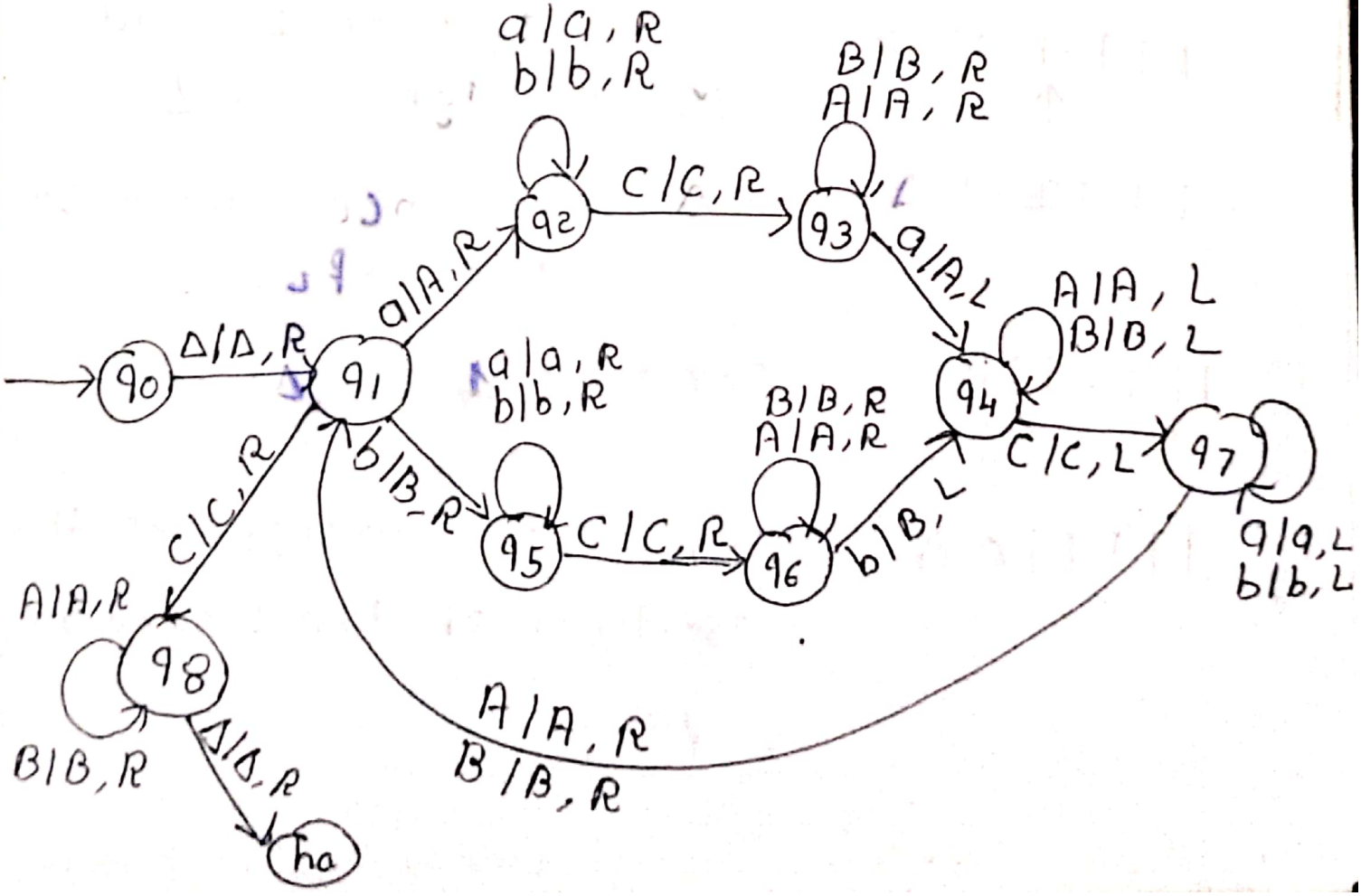
Δ A B C A b Δ Shift Right till b.

Δ A B C A B Δ Convert b to B
move left till C.

Δ A B C A B Δ Shift Right till Δ.

ha ABCAB Δ accepted.

* Transition Diagram :-



ex-16] Construct TM for the addition function for unary no. System

⇒ The unary no. is made up of only one character.

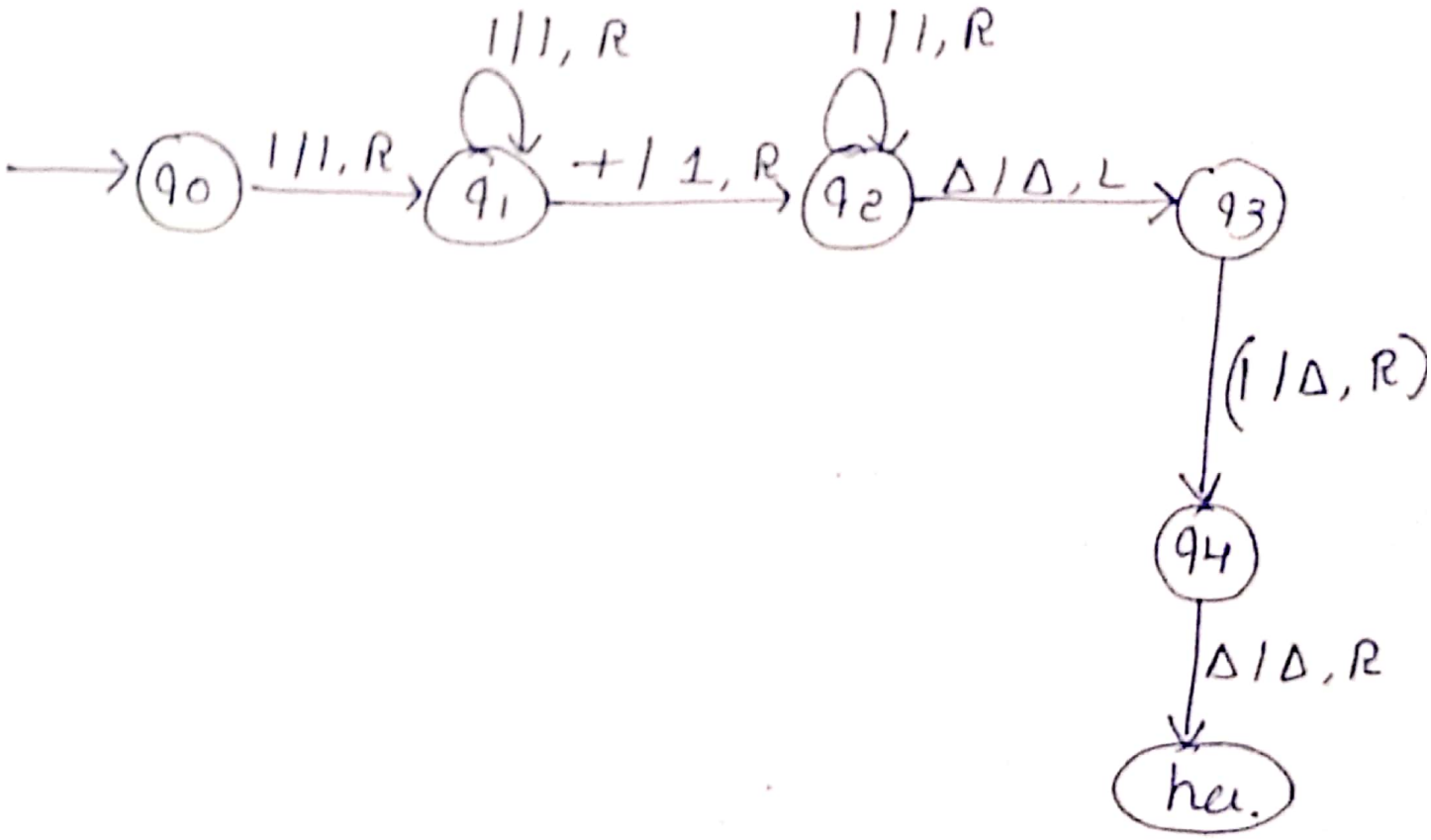
For example :- 2 + 3
11 + 111
= 11111

Logic :-

The input is 3 + 2.

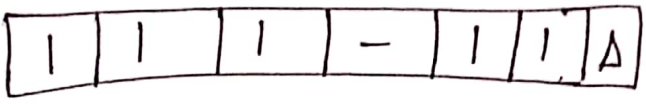
111+11Δ ↑	Move right up to + sign.
111+11Δ ↑	Convert + to 1 and Move Right till Δ
1111±11Δ ↑	Δ has encountered so move left.
111111Δ ↑	Convert 1 to Δ
111111ΔΔ	Tape now consists of the addition of two unary no.

Transition Diagram:-



ex-18 Construct TM for performing Subtraction of two unary no. $f(a-b)=c$ where a is always greater than b.

$\Rightarrow$



Logic

$\underset{\uparrow}{1}11-11\Delta$ Move Right up to −

$111\underset{\uparrow}{-}11\Delta$ Move Right and convert 1 to Δ.

$111-\underset{\uparrow}{\Delta}1\Delta$ Move left.

$111\underset{\uparrow}{-}\Delta1\Delta$ Move left

$\underset{\uparrow}{1}11-\Delta1\Delta$ Convert 1 to Δ

$11\underset{\uparrow}{\Delta}-\Delta1\Delta$ Move Right.

$11\Delta\underset{\uparrow}{-}\Delta1\Delta$ Move Right till 1.

$11\Delta-\Delta\underset{\uparrow}{1}\Delta$ Convert 1 to Δ and move left

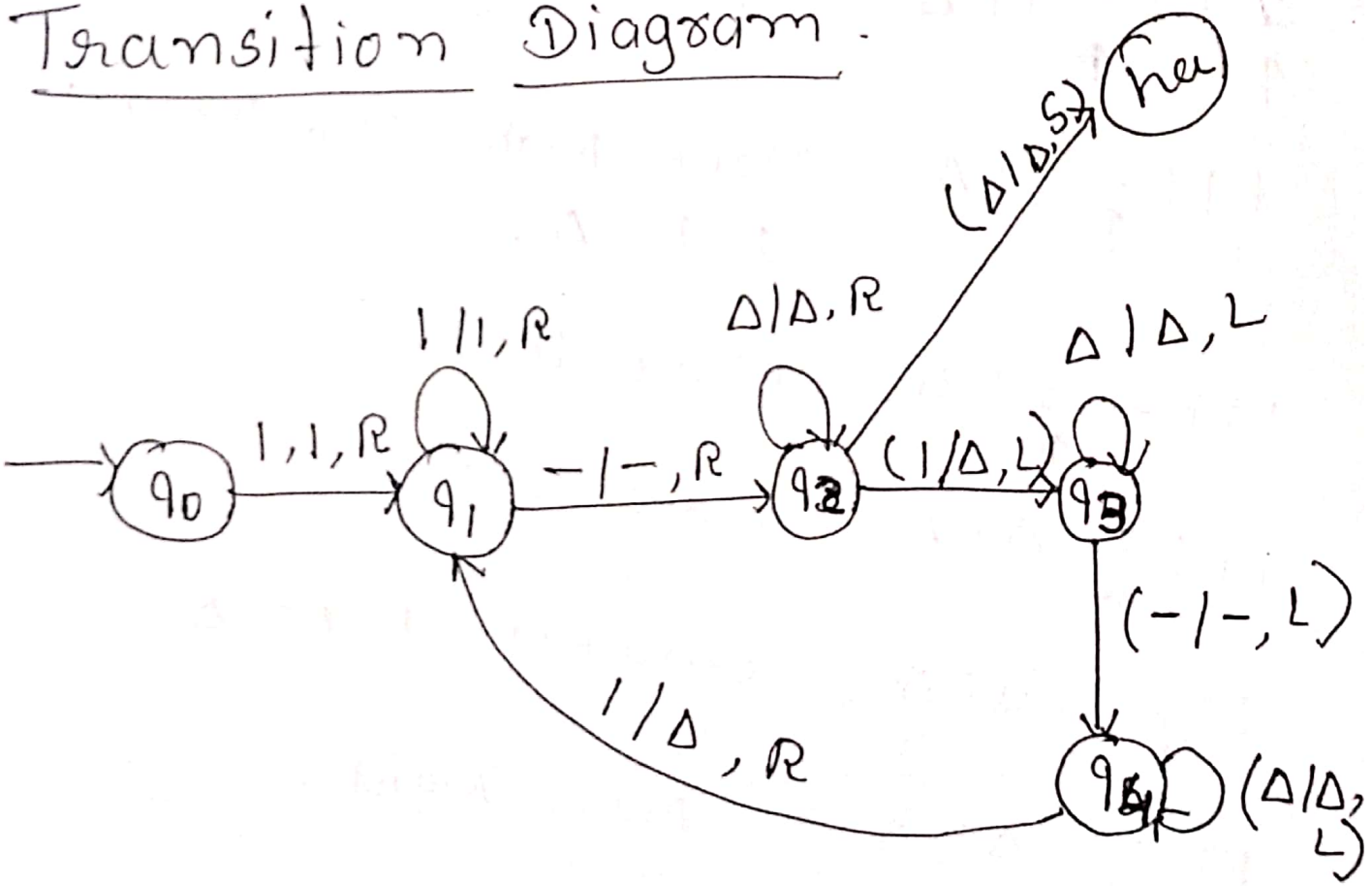
1 1 Δ − Δ Δ Δ Move left till 1
↑

1 1 Δ − Δ Δ Δ Convert 1 to Δ
↑ and move Right

1 Δ Δ − Δ Δ Δ Move Right till Δ
↑

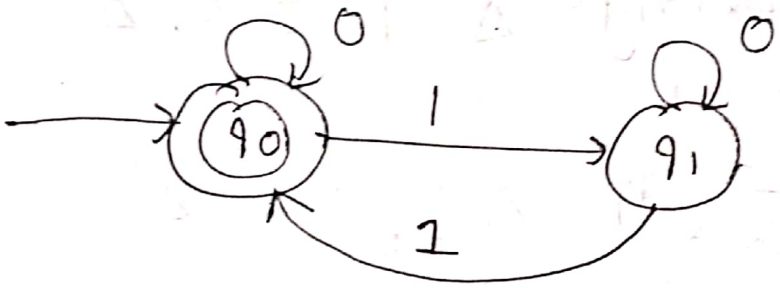
1 Δ Δ − Δ Δ Δ Halt.
↑

Transition Diagram.

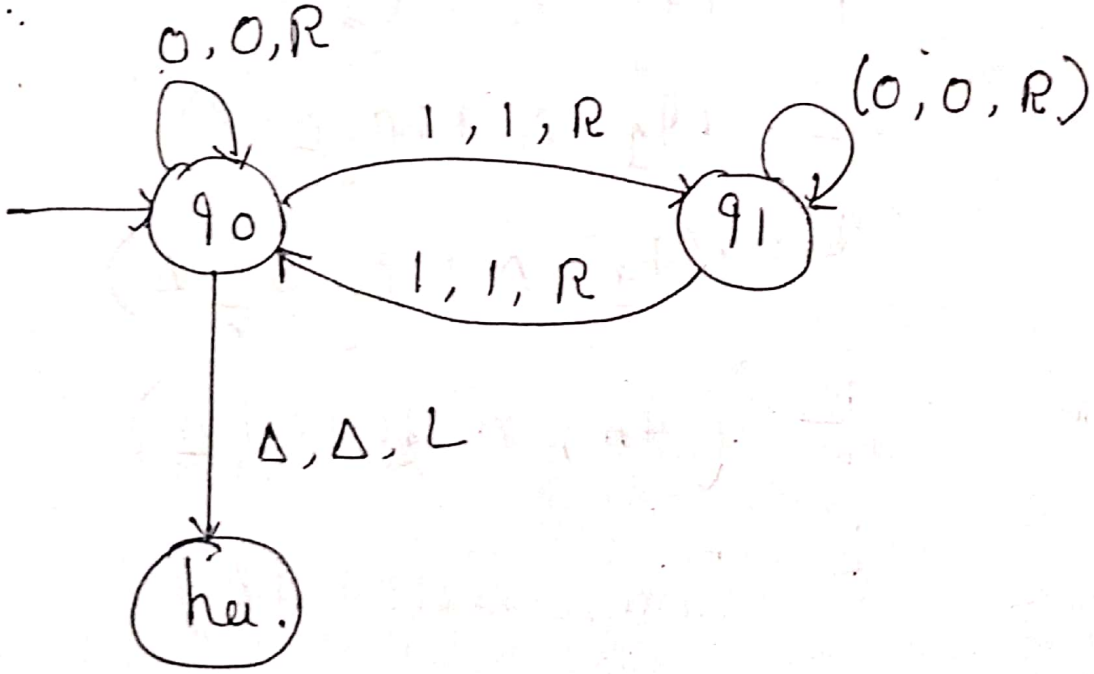


Ex-19 Construct TM for language Consisting of strings having any no. of 0's and only even no. of 1's over set $\Sigma = \{0,1\}$

$\Rightarrow$ FA



$\Rightarrow$ TM



Transition Table.

State	0	1	Δ
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	(H, Δ, L)
q_1	$(q_1, 0, R)$	$(q_0, 1, R)$	—
halt	—	—	—

Tracing String Δ 11010 1 Δ

$(q_0, \Delta 11010 1 \Delta) \vdash (q_1, \Delta 1 1010 1 \Delta)$

$\vdash (q_0, \Delta 11 010 1 \Delta)$

$\vdash (q_0, \Delta 110 1 0 1 \Delta)$

$\vdash (q_1, \Delta 1101 0 1 \Delta)$

$\vdash (q_1, \Delta 11010 1 \Delta)$

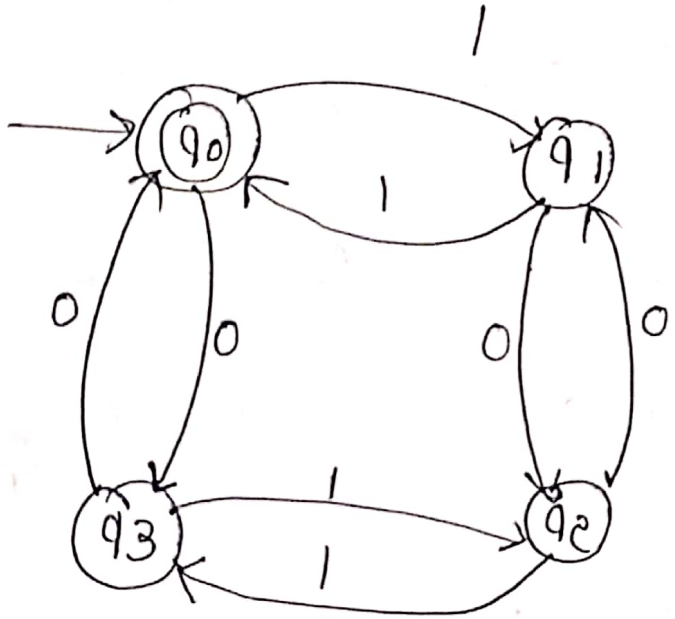
$\vdash (q_0, \Delta 110101 \Delta)$

$\vdash (h_a, \Delta 110101 \Delta)$

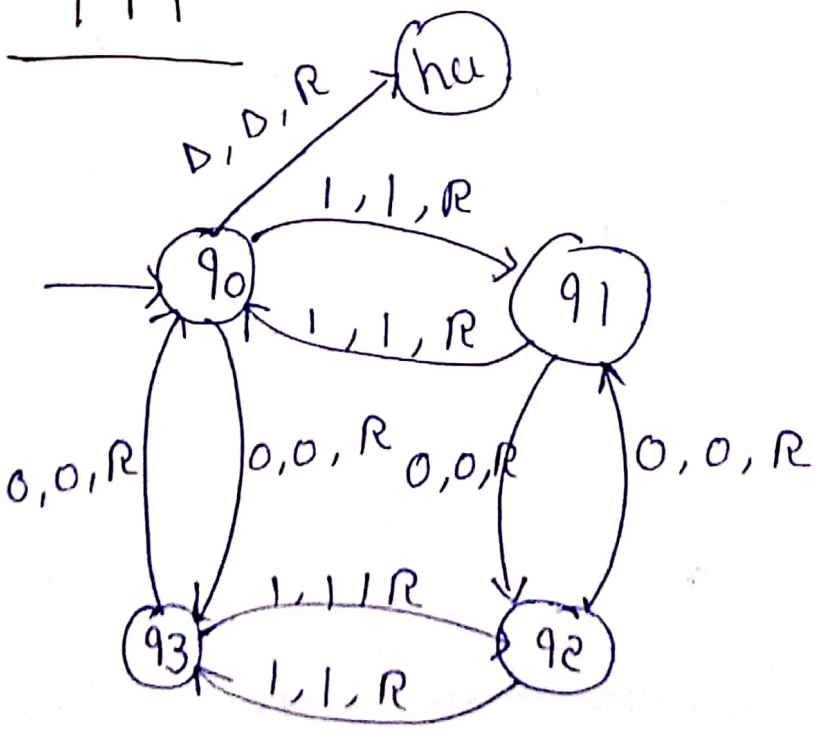
accepted.

Q.20 Construct a TM for language of even no. of 1's and even no. of 0's over $\Sigma = \{0,1\}$

$\Rightarrow$ FA



$\Rightarrow$ TM



String acceptance "101011"

$(q_0, \Delta \underset{\uparrow}{1}01011) \vdash (q_1, \Delta 1 \underset{\uparrow}{0}1011\Delta)$

$\vdash (q_2, \Delta 10 \underset{\uparrow}{1}011\Delta)$

$\vdash (q_3, \Delta 101 \underset{\uparrow}{0}11\Delta)$

$\vdash (q_0, \Delta 1010 \underset{\uparrow}{1}1\Delta)$

$\vdash (q_1, \Delta 10101 \underset{\uparrow}{1}\Delta)$

$\vdash (q_0, \Delta 101011 \underset{\uparrow}{\Delta})$

$\vdash (h_a, \Delta 101011\Delta)$

accepted.