DHANALAKSHMI COLLEGE OF ENGINEERING, CHENNAI

Department of Computer Science and Engineering

CS6503 – THEORY OF COMPUTATION

2 & 16Mark Questions & Answers

***Year / Semester**: III / V*

***Regulation:** 2013*

***Academic year:** 2017 - 2018*

UNIT IV
PART A

1. Define – Turing Machine

A Turing machine is denoted as M = (Q, Σ, Ɣ, δ, q0, B, F)

Q is a finite set of states.

Σ is set of i/p symbols, not including

B. Ɣ is the finite set of tape symbols.

q0 in Q is called start state.

B in Ɣ is blank symbol.

F is the set of final states.

δ is a mapping from Q X Ɣ to Q X Ɣ X {L, R}.

2. **What is a Turning Machine?**

A finite state machine with storage is called as Turing Machine. Turing machine is a simple mathematical model of a computer. TM has unlimited and unrestricted memory and is a much more accurate model of a general purpose computer. The Turing machine is a FA with an R/W Head. It has an infinite tape divided into cells, each cell holding one symbol.

3. What are the special features of TM?

In one move, TM depending upon the symbol scanned by the tape head and state of the finite control listed as below:

1. Changes state
2. Prints a symbol on the scanned tape cell

3. Moves the R/w head left or right

4. What is multiple tracks Turing machine?

A Turing Machine in which the input tape is divided into multiple tracks where each track is having different inputs is called multiple tracks Turing machine.

5. What is a multidimensional Turing machine?

The Turing Machine which has the useful finite control consists of a k-dimensional array of cells in all 2K directions for some fixed K in a tape cell. Depending on the state and symbol scanned, the device changes state, prints a new symbol and moves its tape head in one of 2K directions, along one of the K axes.

6. Differentiate 2-way FA from TM.

Turing machine can change symbols on its tape, whereas the FA cannot change symbols on tape. Also TM has a tape head that moves both left and right side, whereas the FA doesn't have such a tape head.

7. Define – Instantaneous Description of TM

The ID of a Turing Machine M is denoted as $\alpha 1 q\ \alpha 2$. Here q is the current state of M s in Q; $\alpha 1\ \alpha 2$ is the string in Γ * that is the contents of the tape up to the rightmost nonblank symbol or the symbol to the left of the head, whichever is the rightmost.

8.What are the applications of TM?

TM can be used as:

1. Recognizers of languages

2. Computers of functions on non negative integers

3. Generating devices

9.What is off-line Turing machine?

An Off-line Turing Machine is a multitape TM whose input tape is read only. The Turing Machine is not allowed to move the input tape head off the region between left and right end markers.

10.When is a function f said to be Turing computable? (N/D-09)

A Turing Machine defines a function y = f(x) for strings x,y *, if q0 x |--* q_fy. A

function f is 'Turing Computable' if there exists a Turing Machine that performs a specific function.

11.List out the various representation of TM.
We can describe TM using:

7. Instantaneous description
8. Transition table
9. Transition diagram

12.List out the techniques for Turing machine construction.

The various techniques used for Turing Machine construction are as follows:

1. Storage in finite control
2. Multiple tracks
3. Checking off symbols
4. Shifting over
5. Subroutines

13. What are UTMs or Universal Turing machines? (N/D-13)

Universal TMs are TMs that can be programmed to solve any problem that can be solved by any Turing machine. A specific Universal Turing machine U is

Input to U: The encoding "M "of a Tm M and encoding "w" of a string

w. Behavior: U halts on input "M" "w" if and only if M halts on input w.

14. State the halting problem of Turing Machine.

The halting problem for TMs is: Given any Turing Machine M and an input string w, does M halt on w? This problem is undecidable as there is no algorithm to solve this problem.

15. What is the weak-form of Turing thesis?

A Turing Machine can compute anything that can be computed by a general purpose digital computer which is said to be a weak form of Turing thesis.

16. State the usage of checking off symbol in TM. **(N/D-09)**

Checking off symbols is useful method when a TM recognizes a language with repeated strings and also to compare the length of substrings.

(eg) : { ww | w _ _ * } or $\{a^i b^i \mid i>=1\}$. This is implemented by using an extra track on the tape with symbols Blank or √.

17. How can a TM used as a transducer?

A TM can be used as a transducer. The most obvious way to do this is to treat the entire nonblank portion of the initial tape as input, and to treat the entire blank portion of the tape when the machine halts as output. Or a TM defines a function y=f(x) for strings x ,y _ _* if: $q0X$ | --- q_fY, where q_f is the final state.mathematical concepts and the “real

(1) Various Techniques for turning machine construction.

[AU: May- 4, 9, 12, 13, 14 Dec- 11, 12, 13, 14, 15 Marks 12]

Answer:

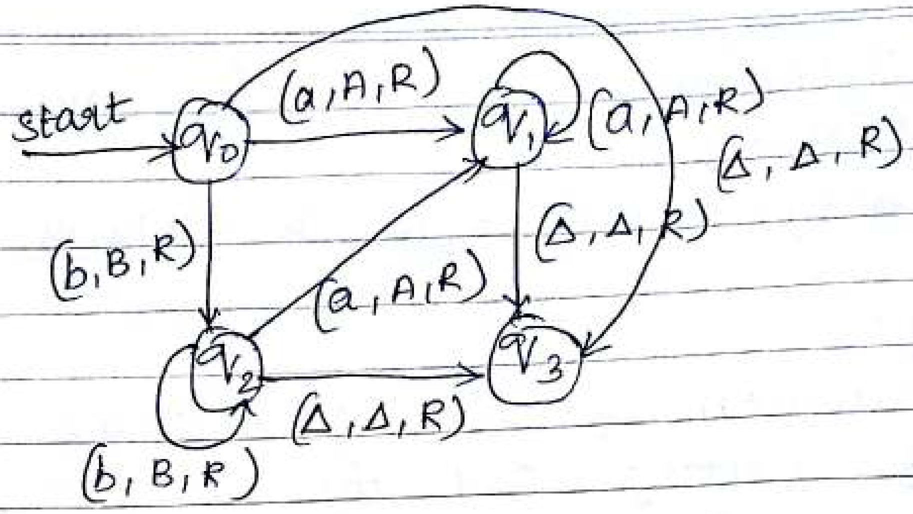
Construction of turning machine is a process of writing out the complete set of states and next move functions. This is a totally conceptual phenomenon. The turning machine can be designed with the help of some conceptual tools. Let us discuss some of these tools.

Storage in finite control :-

The model of turning machine has a finite control. This finite control can be used to hold some amont of information. The finite stores the information in pair of elements such as the current state and the current symbol pointed by the tape head. This is just a conceptual arrangement.

Example: construct a turning machine M for $\Sigma = \{a, b\}$ which will convert lower case letters to upper case.

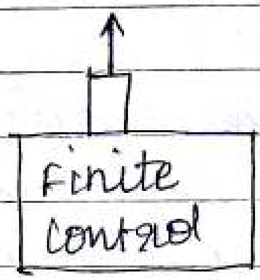
Solution :



Multiple Tracks

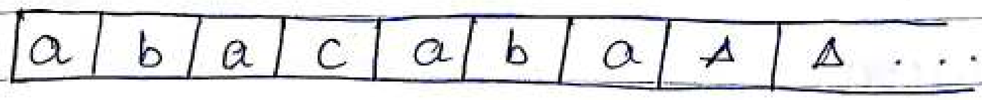
If the input tape is divided into multiple tracks then the input tape will be as follows

#	1	1	1	1	1	\$	
B	B	B	B	1	1	B	....
B	1	1	1	B	B	B	



Example : Construct a turning machine $M = \{Q, \Sigma, \delta, \Gamma, q_0, B, F)$ which recognizes the language
$$L = \{wcw \,/\, w \in (a+b)^+\}$$

Solution:



a	b	a	c	a	b	a	Δ	Δ	...

Read a and mark it

*	b	a	c	a	b	a	Δ	Δ	...
*	b	a	c	a	b	a	Δ	Δ	...
*	b	a	c	a	b	a	Δ	Δ	...
*	b	a	c	a	b	a	Δ	Δ	...

It is same as marked one

*	b	a	c	*	b	a	Δ	Δ	...
*	b	a	c	*	b	a	Δ	Δ	...
*	b	a	c	*	b	a	Δ	Δ	...
*	b	a	c	*	b	a	Δ	Δ	...
*	*	a	c	*	b	a	Δ	Δ	...

Mark it

*	*	a	c	*	b	a	Δ	Δ	...
*	*	a	c	*	b	a	Δ	Δ	...
*	*	a	c	*	*	a	Δ	Δ	...

mark it if it is same as marked

*	*	a	c	*	*	a	Δ	...
				↑				

*	*	a	c	*	*	a	Δ	Δ	...
			↑						

*	*	a	c	*	*	a	Δ	Δ	...
		↑							

*	*	a	c	*	*	a	Δ	Δ	...
	↑								

*	*	*	c	*	*	a	Δ	Δ	...
		↑							

*	*	*	c	*	*	a	Δ	Δ	...
			↑						

*	*	*	c	*	*	a	Δ	Δ	...
						↑			

*	*	*	c	*	*	*	Δ	Δ	...
					↑				

*	*	*	c	*	*	*	Δ	Δ	...
				↑					

*	*	*	c	*	*	*	Δ	Δ	...
			↑						

*	*	*	c	*	*	*	Δ	Δ	...
		↑							

Subroutine:

In the high level languages use of subroutines built the modularity in the program development process. The same type of concept can be introduced in construction of TM.

Example: construct a TM for subroutine
$f(a, b) = a * b$
where a and b are unary numbers.

solutions:

$$\underbrace{1\ \ 1}_{a}\ \ \$\ \ \underbrace{1\ \ 1\ \ 1}_{b}\ \ \ldots$$

1 1 $ 1 1 1 Δ

X 1 $ 1 1 1 Δ

X 1 $ 1 1 1 Δ

X 1 $ Y 1 1 Δ 1 Δ

X 1 $ Y 1 1 Δ 1 Δ

X 1 $ Y Y 1 Δ 1 Δ

X 1 $ Y Y 1 Δ 1 1 Δ

X 1 $ Y Y 1 Δ 1 1 Δ

x 1 $ y y 1 Δ 1 1 Δ

x 1 $ y y y Δ 1 1 1 Δ

x 1 $ y y y Δ 1 1 1

x 1 $ 1 1 1 Δ 1 1 1

x x $ 1 1 1 Δ 1 1 1

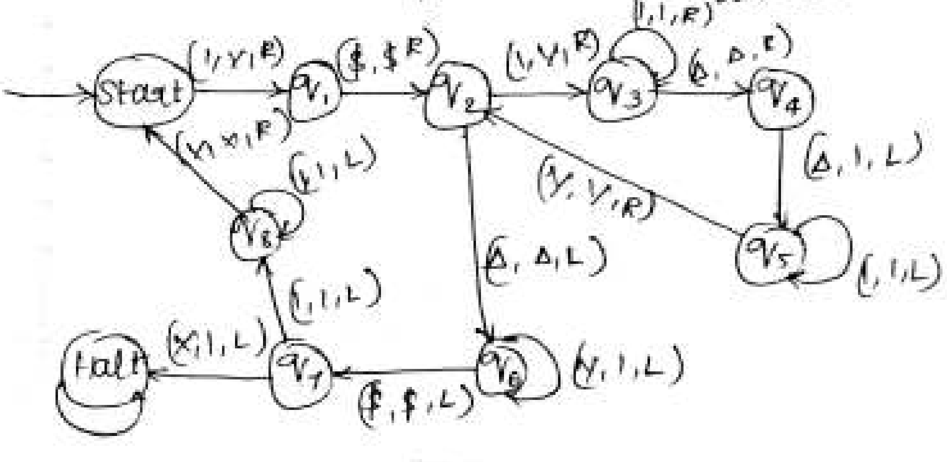
x x $ 1 1 1 Δ 1 1 1

x x $ y 1 1 Δ 1 1 1 1

x x $ y 1 1 Δ 1 1 1 1

x x $ y y 1 Δ 1 1 1 1 1

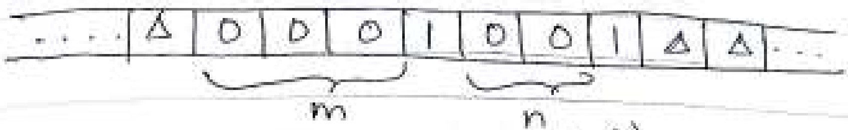
$\underbrace{1\ 1}_{a}\ \$\ \underbrace{1\ 1\ 1}_{b}\ \Delta\ \underbrace{1\ 1\ 1\ 1\ 1\ 1}_{answer}\ \Delta$



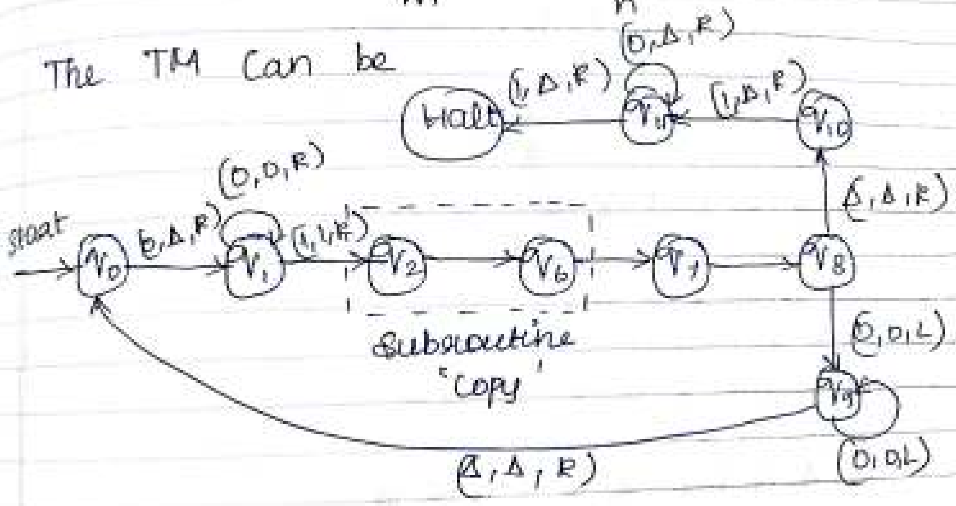
Q) Design a turing machine M to implement the function multiplication using the subroutine "copy"

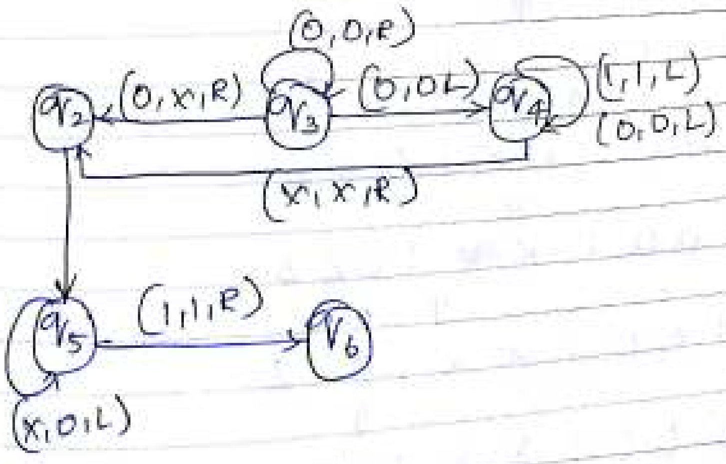
AU: May-4, Dec-14 Marks-12

<u>solution</u> :



The TM can be





DHANALAKSHMI

Δ 0 0 0 1 0 0 1 Δ . . .

Δ Δ 0 0 1 0 0 1 Δ Δ Δ . . .

Δ Δ 0 0 1 0 0 1 Δ Δ Δ . . .

Δ Δ 0 0 1 0 0 1 Δ Δ Δ . . .

Δ Δ 0 0 1 0 0 1 Δ Δ Δ . . .

Δ Δ 0 0 1 X 0 1 Δ Δ Δ . .

Δ Δ 0 0 1 X 0 1 Δ Δ Δ . . .

Δ Δ 0 0 1 X 0 1 Δ . . .

Δ Δ 0 0 1 X 0 1 0 Δ Δ . .

Δ Δ 0 0 1 X 0 1 0 Δ Δ . . .

Δ Δ 0 0 1 X 0 1 0 Δ Δ . . .

Δ Δ 0 0 1 X X 1 0 Δ Δ . . .

Δ Δ 0 0 1 X X 1 0 Δ Δ

Δ Δ 0 0 1 X X 1 0 Δ Δ

Δ Δ 0 0 1 X X 1 0 Δ

Page No.

Δ Δ 0 0 1 X X 1 0 0 Δ

Δ Δ 0 0 1 X X 1 0 0 Δ

Δ Δ 0 0 1 X X 1 0 0 Δ

Δ Δ 0 0 1 0 0 1 0 0 Δ

Δ Δ 0 0 1 0 0 1 0 0 Δ

Δ Δ Δ 0 1 0 0 1 0 0 Δ

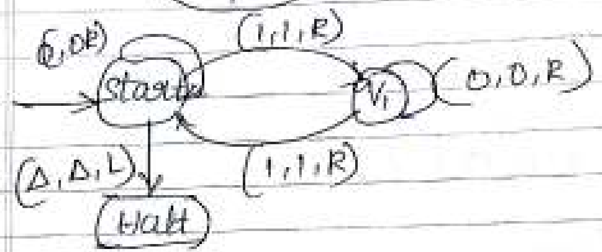
Δ Δ Δ Δ 1 0 0 1 0 0 Δ

Thus finally

Δ Δ Δ Δ Δ Δ Δ 0 0 0 0 0 0 0 Δ Halt.

③ Construct TM for language consisting of strings having any number of 0's and only even number of 1's over the input set $\Sigma = \{0, 1\}$.

Solution :



Input: 1 1 0 1 0 1 Δ

head ↑ q_1

1 1 0 1 0 1 Δ

↑ start

1 1 0 1 0 1 Δ

↑ start

1 1 0 1 0 1 Δ

↑ q_1

1 1 0 1 0 1 Δ

↑ q_1

1 1 0 1 0 1 Δ

↑ start

1 1 0 1 0 1 Δ

↑ Halt.

Thus this i/p is accepted by TM.

④ Construct TM for the language $L = \{a^n b^n\}$ where $n \geq 1$.

Solution:

a a b b Δ
↑

A a b b Δ
↑

A a b b Δ
↑

A a B b Δ
↑

A a B b Δ
↑

A a B b Δ
↑

A A B b Δ
↑

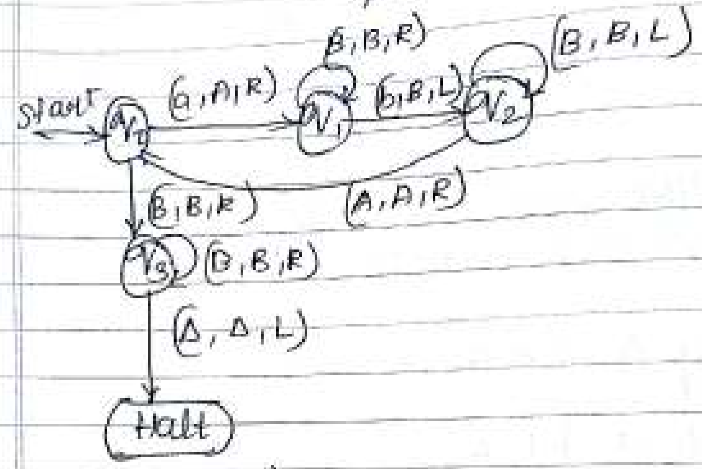
A A B B Δ
↑

A A B B Δ
↑

A A B B A

A A B B Δ
↑

A A B B Δ
↑ .



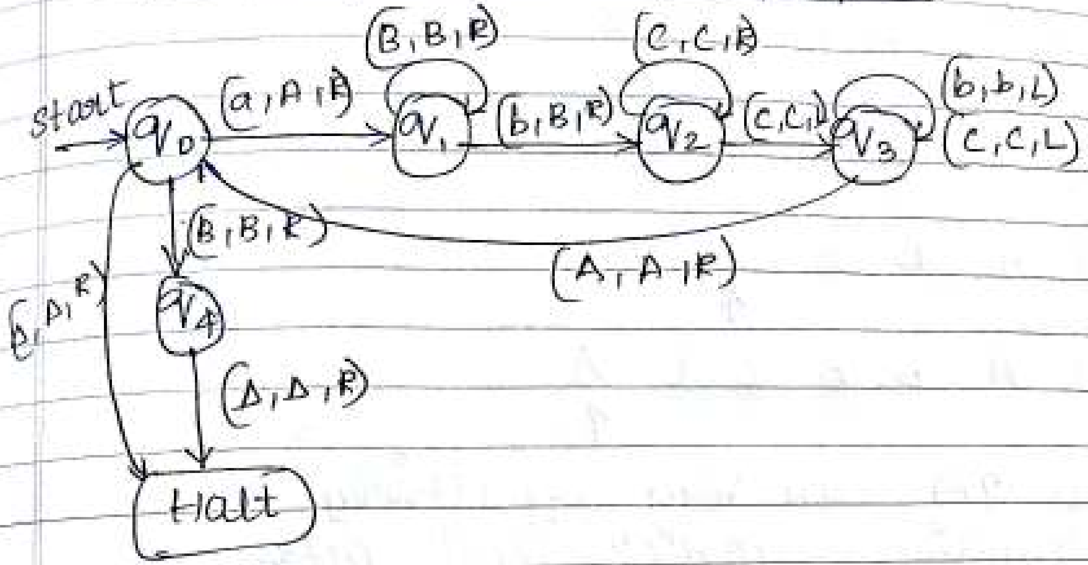
~~To~~ Transition table.

	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, B, R)	$(Halt, Δ, L)$
Halt	–	–	–	–	–

Thus TM does not lead to HALT state for such a invalid input.

⑤ Construct TM for $L = \{a^n b^n c^n \mid n \geq 0\}$

<u>solution</u> :



a a b b c c Δ
↑

A a b b c c Δ
↑

A a b b c c Δ
↑

A a B b c c Δ
↑

A a B b C c Δ
↑

A a B b C c Δ
↑

A a B b C c Δ
↑

A A B b C C Δ

A A B B C C Δ

A A B B C C Δ

A A B B C C Δ

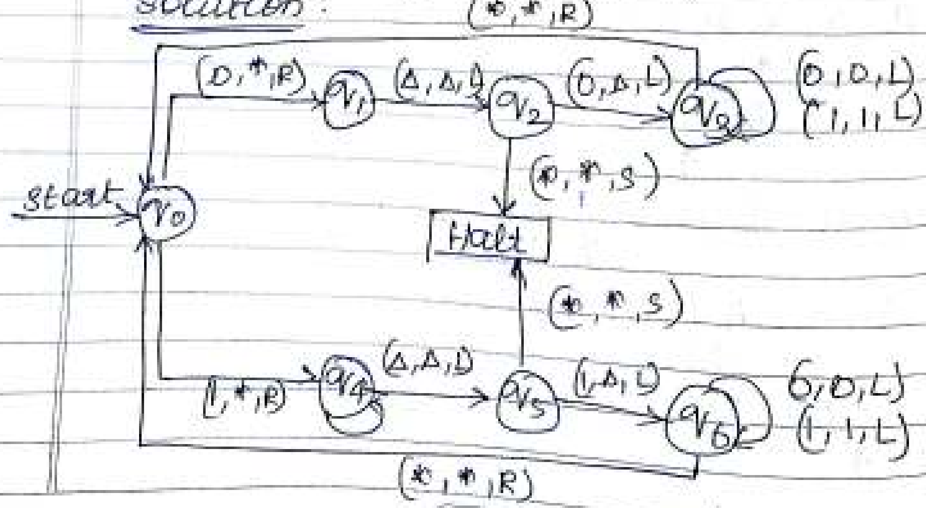
A A B B C C Δ

A A B B C C Δ

Thus TM can very effectively recognizes $a^n b^n c^n$ and more powerful than PDA

⑥ construct a TM for checking the Palindrome of a string of odd palindrome for $\Sigma = \{0,1\}$.

solution:



Page No.

Scanned by CamScanner

0 0 1 0 0 Δ

* 0 1 0 0 Δ

* 0 1 0 0 Δ

* 0 1 0 Δ Δ

* 0 1 0 Δ

* 0 1 0 Δ

* * 1 0 Δ

* * 1 0 Δ

* * 1 Δ Δ

* * 1 Δ Δ

* * * Δ Δ

* * * Δ Δ

* * * Δ Δ

DHANALAKSHMI

(7) Construct TM for checking well formness of the parenthesis.

<u>solution</u>:

Δ ( ) ( ) Δ

Δ ( ) ( ) Δ

Δ (* ( ) Δ

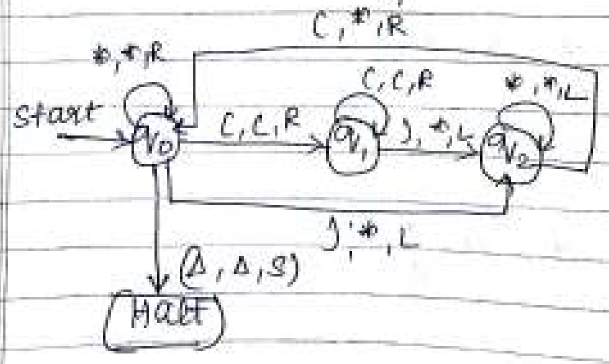
Δ * * ( ) Δ

Δ * * ( ) Δ

Δ * * (* Δ

Δ * * * * Δ

Δ * * * * Δ



Transition Table.

State	(	)	*	Δ
q_0	$(q_1, (, R)$	$(q_2, *, L)$	$(q_0, *, R)$	$(Halt, \Delta, S)$
q_1	$(q_1, (, R)$	$(q_2, *, L)$	–	–
q_2	$(q_2, *, R)$	–	$(q_0, *, L)$	–
Halt	–	–	–	–

Page No.