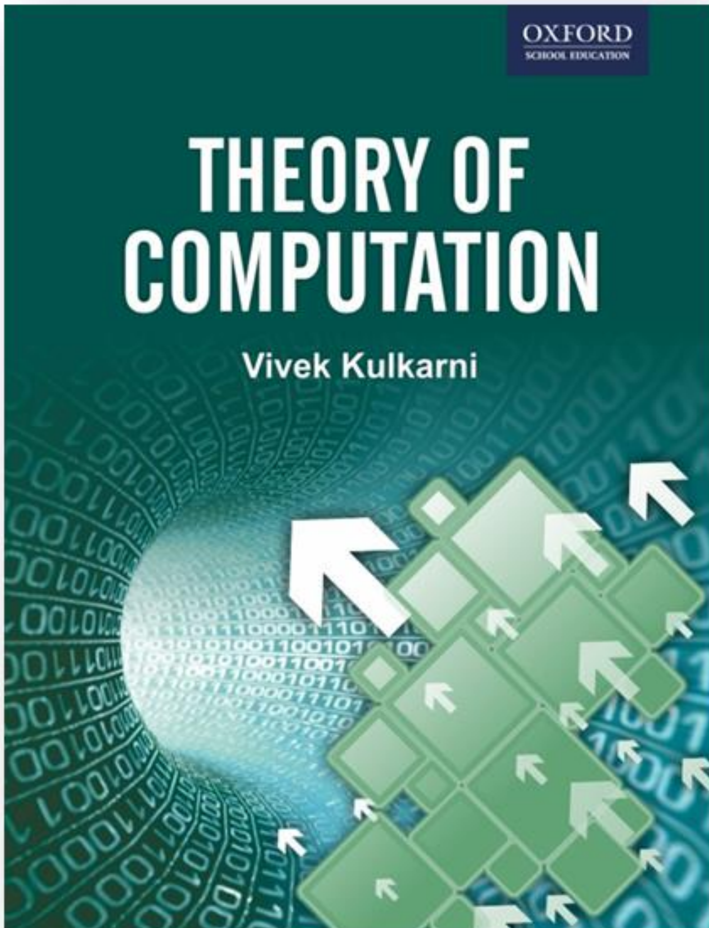


Author: Vivek Kulkarni
(vivek_kulkarni@yahoo.com)

Chapter-4: Turing Machines

Solutions for Review Questions

Q.1 Compare FSM and TM.

Solution:

Refer to the section 4.1.

Q.2 Explain the halting problem.

Solution:

Refer to the section 4.14.

Q.3 Design a TM for multiplying two unary numbers. Show the step-wise functioning of the TM for the input sequences:

(i) 111×1111 (ii) 111×11

Solution:

Refer to the example 4.10 from the book.

Q.4 Design a TM to recognize an arbitrary string divisible by 4, from $\sum = \{0, 1, 2\}$.

Solution:

We can see that the input is a ternary (base-3) number. As we need to check the divisibility, the problem is solvable using a FSM. We need four states pertaining to four values of remainders namely, 1, 2, 3, and 0. Zero remainder indicates that the number is divisible by 4. The STF and MAF for the FSM are,

Q \ ∑	0	1	2
q0	q0	q1	q2
q1	q3	q0	q1
q2	q2	q3	q0
q3	q1	q2	q3

STF: $Q \times \sum \rightarrow Q$

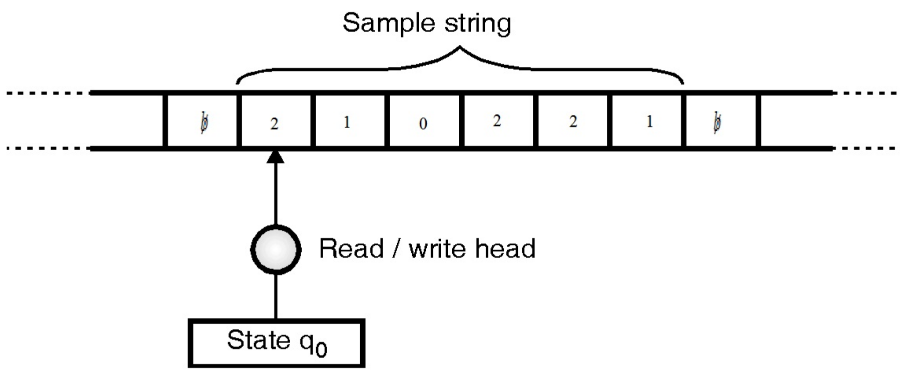
Q \ ∑	0	1	2
q0	Y	N	N
q1	N	Y	N
q2	N	N	Y
q3	N	N	N

MAF: $Q \times \sum \rightarrow \Delta$

The output *Y* and *N* is generated to indicate whether the number is divisible by 4 or not, respectively.

We need to convert these tables to a functional matrix for the resultant TM. The direction is always moving towards right, as this is a FSM.

Let us assume the below initial configuration for the TM.



The functional matrix for the TM is,

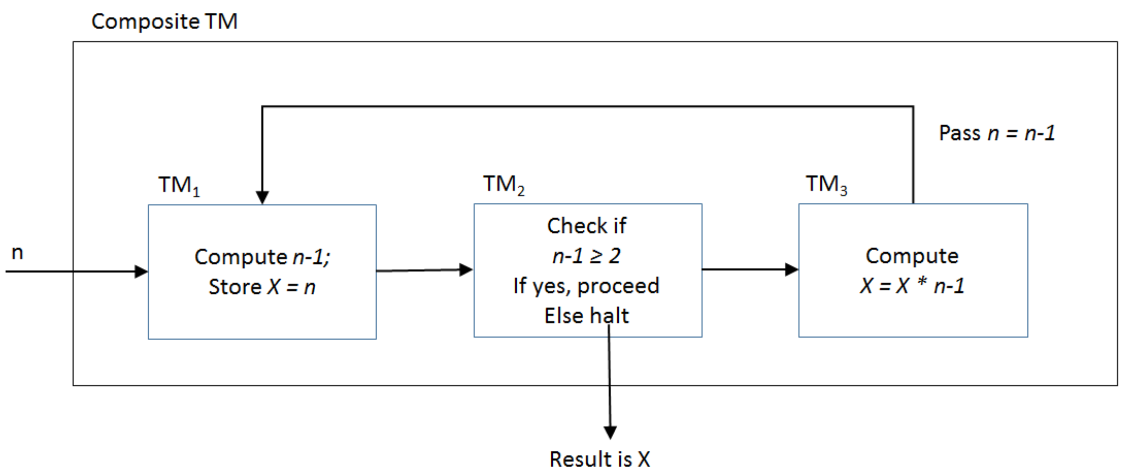
Q \ ∑	0	1	2	ƀ
q0	Y q0 R	N q1 R	N q2 R	q4 L
q1	N q3 R	Y q0 R	N q1 R	q4 L
q2	N q2 R	N q3 R	Y q0 R	q4 L
q3	N q1 R	N q2 R	N q3 R	q4 L

One can see that the TM keeps on generating the output *Y* or *N* depending on the transition. q4 is the halt state. The TM enters into the halt state upon reaching the end of the input. TM moves one position left to point to a cell on the tape which has the last output which is valid for the input string.

Q.5 Design a Turing machine to compute *n*! (Factorial *n*). Show the step-wise functioning of the TM for the input $n = 3$.

Solution:

We need solve this on the similar lines as that of the example 4.14 from the book. We need to build the composite TM as shown below,



Q.6 Design a TM to convert a binary-coded decimal number into a unary number. Validate the design for:

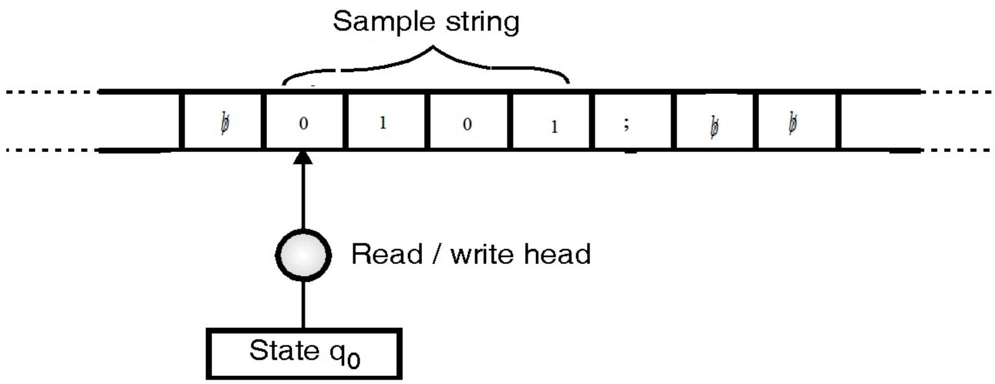
(i) 1001 (ii) 0000

Solution:

Binary coded decimal (BCD) is the type of binary encoding for the decimal number using fixed number (4 bits) of bits. As we can see from the below table that in fact 4 bits are sufficient to denote the decimal numbers from 0 to 9.

Decimal Digit	BCD 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

We need read the binary encoding and convert it to its decimal equivalent using the unary number format. Let us use symbol *a* for the unary representation. For example, decimal 3 whose BCD encoding is *0011* needs to be represented as *aaa*. Below is the initial configuration of the TM.



We will read the BCD input from MSB to LSB which is from left to right. The result which is the unary encoding is written to the right of the semicolon ';'. The functional matrix for the TM can be given as below.

Q \ ∑	1	0	;	b̸	a
q0	0 R	q1 R	q4 R	--	--
q1	0 R	q2 R	q12 R	--	--
q2	0 R	q3 R	q16 R	--	--
q3	0 R	q20 R	q18 R	--	--
q4	--	--	--	a q5 R	R
q5	--	--	--	a q6 R	R
q6	--	--	--	a q7 R	R
q7	--	--	--	a q8 R	R
q8	--	--	--	a q9 R	R
q9	--	--	--	a q10 R	R
q10	--	--	--	a q11 R	R
q11	--	--	--	a q19 L	R
q12	--	--	--	a q13 R	R
q13	--	--	--	a q14 R	R
q14	--	--	--	a q15 R	R
q15	--	--	--	a q19 L	R
q16	--	--	--	a q17 R	R
q17	--	--	--	a q19 L	R
q18	--	--	--	a q19 L	R
q19	L	L	L	q0 R	L
q20 (halt)	--	--	--	--	--

q4 to q11 are responsible for writing 8 (eight times *a*) for the first *1* (out of 4-bit BCD) read. For the second *1* read, q12 to q15 write 4 times *a*. Similarly, third one represents 2 and fourth one represents 1. This is represented in unary by writing respectively two *a*'s and one *a* in the result area. When TM halts, the unary number to the right of the semicolons represents the decimal equivalent for the BCD input.

The inputs '1001' and '0000' gets changed to 'aaaaaaaaa' and '' respectively representing decimal 9 and 0.

Q.7 What is a universal Turing machine?

Solution:

Refer to the section 4.9.

Q.8 Design a TM to find the GCD of two given numbers.

Solution:

Refer to the example 4.11 from the book.

Q.9 Write a short note on halting problem.

Solution:

Refer to the section 4.14.

Q.10 Design a TM to compare two numbers, which will produce the output *L* if the first number is less than second number, output *G* if the first number is greater than second number, and output *E* otherwise.

Solution:

Refer to the example 4.8 from the book.

Q.11 Design a Turing machine, which computes the 2's complement of a given binary number.

Solution:

Refer to the example 4.3 from the book.

Q.12 Construct a TM for the language, $L = \{a^m b^n \mid m \geq n, n \geq 1\}$.

Solution:

Refer to the example 4.5 from the book; replace *0* by *a* and *1* by *b*.

Q.13 Construct a TM for checking if a given set of parentheses are well-formed.

Solution:

Refer to the example 4.2 from the book.

Q.14 Write a short note on solvability and semi-solvability.

Solution:

Refer to the section 4.13.

Q.15 Write a short note on recursive TM.

Solution:

Refer to the section 4.8; recursive TM is also called as iterative TM.

Q.16 Consider the Turing machine $M = \{(q_0, q_1, q_2, q_f), (0, 1), (0, 1, B), \delta, q_0, B, (q_f)\}$. Describe the language, $L(M)$, if δ consists of the following sets of rules:

$\delta\ (q_0, 0) = (q_1\ 1\ R)$;

$\delta\ (q_1, 1) = (q_0\ 0\ R)$;

$\delta\ (q_1, B) = (q_f\ B\ R)$.

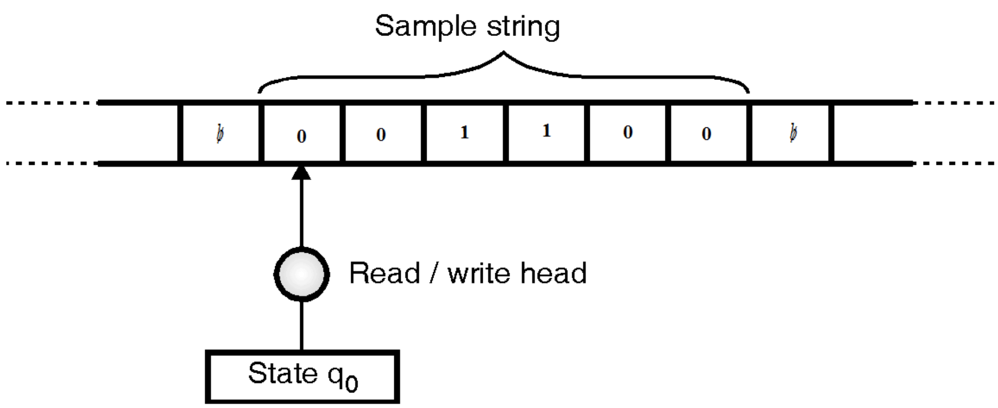
Solution:

The TM accepts the language of the form *0 (10)**. Also, it replaces every *0* by *1* and every *1* by *0*.

Q.17 Design a Turing machine that accepts the language $\{0^n\ 1^n\ 0^n \mid n \geq 1\}$. Also, give the transition function and the transition diagram.

Solution:

The initial configuration of the TM is as follows,



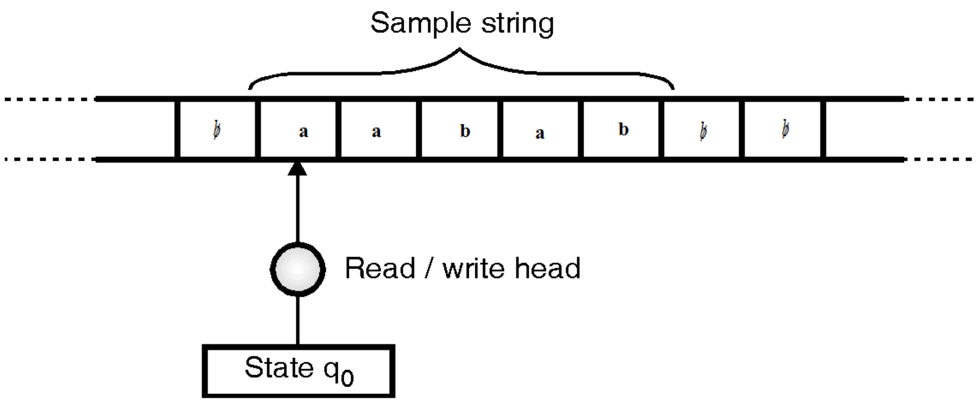
All the initial set of *0*'s from $\{0^n\ 1^n\ 0^n \mid n \geq 1\}$ are replaced by all *a*'s, all *1*'s are replaced by *b*'s and later set of *0*'s are replaced by all *c*'s for checking the acceptance of the string. At the end if we are left with only *a*'s, *b*'s and *c*'s on the tape then the input string is acceptable to the TM which generates output *A* and so it belongs to $\{0^n\ 1^n\ 0^n \mid n \geq 1\}$. Otherwise, the input is invalid and TM generates the output *E* to indicate the error. The simplified functional matrix is shown below.

Q \ ∑	0	1	a	b	c	ƀ
q0	a q1 R	--	--	q4 R	E q5 N	E q5 N
q1	R	b q2 R	--	R	q4 R	E q5 N
q2	c q3 L	R	--	--	R	E q5 N
q3	L	L	q0 R	L	L	--
q4	E q5 N	E q5 N	--	R	R	A q5 N
q5 (halt)	--	--	--	--	--	--

Q.18 Construct a Turing machine that accepts the language ($a^*\ b\ a^*\ b$).

Solution:

This is a regular language. Hence, the TM accepting the same just need to move towards right. The initial configuration of the TM is as below.



As per the language the minimum length string required is '*bb*'. Also, the input string may start with either *a* or *b* but always needs to end with symbol *b*. The functional matrix is as shown below.

Q \ ∑	a	b	ƀ
q0	R	q1 R	--
q1	R	q2 R	--
q2	--	--	A q3 N
q3 (halt)	--	--	--

The states q1 and q2 are reached on reading one *b* each. The TM accepts the string by writing the output *A* onto the tape.

Q.19 Design Turing machines that recognize the following languages:

(a) $\{0^n\ 1^n\ 0^n \mid n \geq 1\}$

(b) $\{WW^R \mid W \text{ is in } (0 + 1)^*\}$

(c) The set of strings with equal number of 0's and 1's

Solution:

(a) Refer to the answer for the Q.17 above.

(b) The language expressed is nothing a set of binary palindromes. Refer to the example 4.7 from the book for the solution.

(c) Refer to the example 4.6 from the book.

Q.20 Define the Turing machine, explain its working, and give the applications of the same.

Solution:

Refer to the sections 4.1, 4.2, and 4.3.

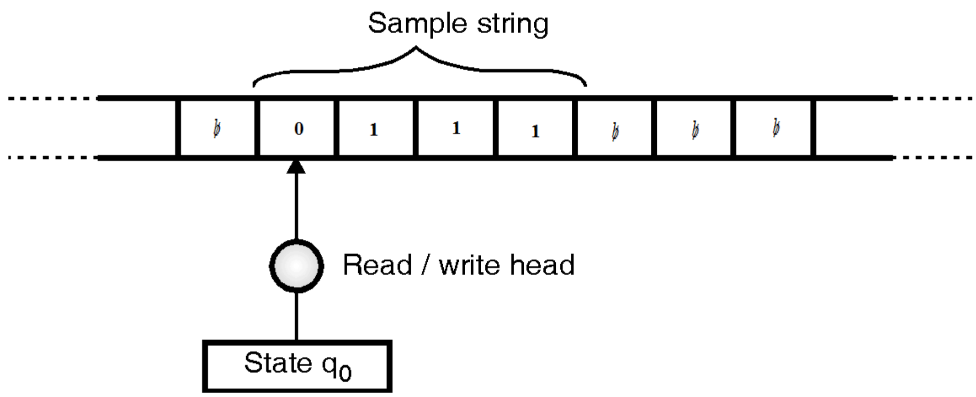
Q.21 Construct Turing machines that recognize the languages:

1) $L = \{0^n 1^m \mid n, m >= 0\}$

2) $L = \{x \in \{0, 1\}^* \mid x \text{ ends in } 00\}$

Solution:

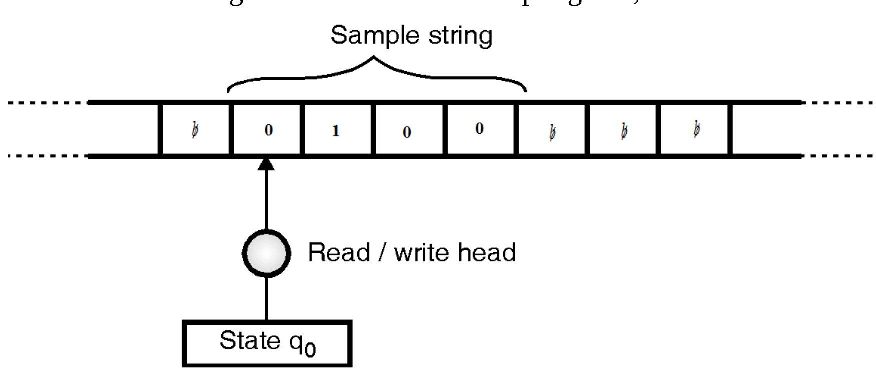
1) The language, $L = \{0^n 1^m \mid n, m >= 0\}$ represents the regular language denoted by, *0*1**. The initial configuration for the TM accepting L is,



The functional matrix for the TM is,

Q \ ∑	0	1	b̸
q0	R	q1 R	A q2 N
q1	--	R	A q2 N
q2 (halt)	--	--	--

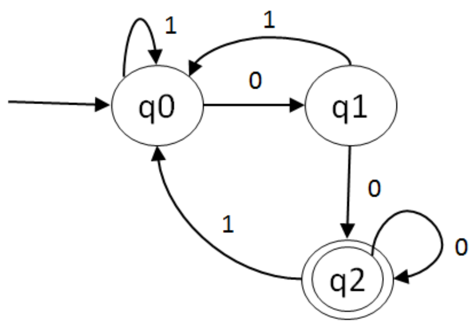
2) The language, $L = \{x \in \{0, 1\}^* \mid x \text{ ends in } 00\}$ represents the regular language denoted by, *(0+1)*00*. The initial configuration for the TM accepting L is,



The functional matrix for the TM is,

Q \ ∑	0	1	ƀ
q0	q1 R	R	--
q1	q2 R	q0 R	--
q2	R	q0 R	A q3 N
q3 (halt)	--	--	--

The TM is equivalent to the DFA as drawn below,



Q.22 Define and explain multi-tape Turing machine.

Solution:

Refer to the section 4.10.

Q.23 Define recursive and recursively enumerable languages.

Solution:

Refer to the section 4.15.

Q.24 Explain the following for Turing machine:

i) Power of Turing machine over finite state machine

ii) Universal Turing machine

Solution:

i) Power of Turing machine over finite state machine: Refer to the section 4.1.

ii) Universal Turing machine: Refer to the section 4.9.

Q.25 Design a Turing machine to replace '110' by '101' in a binary input string.

Solution:

Refer to the example 4.4 from the book.

Q.26 Write a short note on Post's correspondence problem (PCP).

Solution:

Refer to the section 4.18.

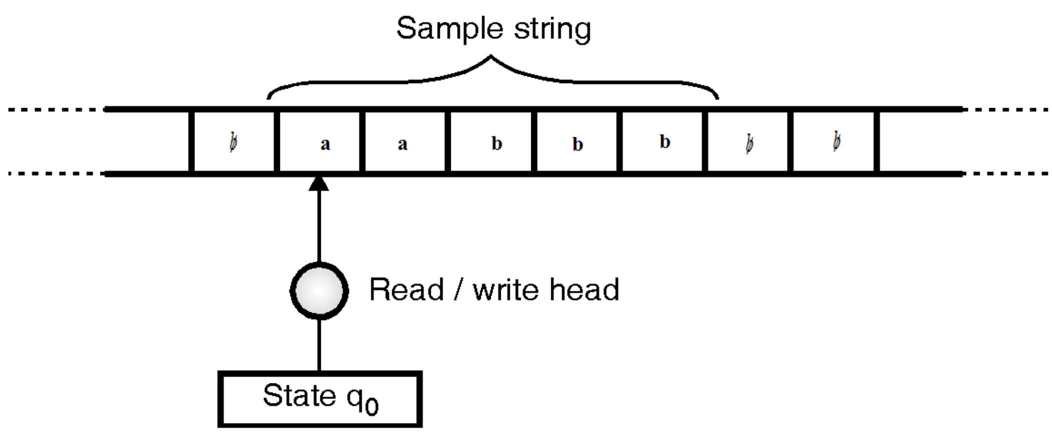
Q.27 Draw transitions tables for Turing machines that accept the following languages:

(a) $\{a^i b^j \mid i < j\}$

(b) The language of balanced string of parentheses

Solution:

(a) The initial configuration of the TM is,



We are going to replace all *a*'s by *'s and all *b*'s by #'s. We keep on matching *a*'s with *b*'s this way. At the end there should be more number of *b*'s should be left unchanged to # indicating number of *b*'s is more than that of *a*'s. TM generates the output *A* if the string is acceptable.The functional matrix for the TM is,

Q \ ∑	a	b	*	#	b̸
q0	* q1 R	q4 R	--	q3 R	--
q1	R	# q2 L	--	R	--

q2	L	--	q0 R	L	--
q3	--	# q4 R	--	R	--
q4	--	R	--	--	A q5 N
q5 (halt)	--	--	--	--	--

(b) Refer to the example 4.2 from the book.

Q.28 Draw a transition table for a Turing machine that accepts the language of all non-palindromes over $\{a, b\}$.

Solution:

Refer to the example 4.20 from the book.

Q.29 Define the following:

(1) Multi-track Turing machine

(2) Multi-tape Turing machine

(3) Recursively enumerable language

(4) Recursive language

Solution:

(1) Multi-track Turing machine: Refer to the section 4.12.

(2) Multi-tape Turing machine: Refer to the section 4.10.

(3) Recursively enumerable language: Refer to the section 4.15.

(4) Recursive language: Refer to the section 4.15.

Q.30 Design a Turing machine to compute the function n^2.

Solution:

Refer to the example 4.14 from the book.

Q.31 Design a TM to accept the language, $L = \{x \in \{0, 1\}^* \mid x$ contains equal number of 0's and 1's$\}$. Simulate the operation for the string '110100'.

Solution:

Refer to the example 4.6 from the book.

Q.32 Design a Turing machine to find the value of $\log_2 n$, where n is any binary number and a perfect power of 2.

Solution:

Refer to the example 4.13 from the book.

Q.33 Determine the solution for the following instance of Post's correspondence problem.

	A	***B***
i	w_i	x_i
1	01	0
2	110010	0
3	1	1111
4	11	01

Solution:

The instance of the PCP has no solution.

