COURSE PLAN

A. COURSE OVERVIEW

Degree:	BE	Programme:	CS
Academic Year:	2017-18	Semester:	5
Course Title:	Automata Theory & Computatability	Course Code:	15CS54
L-T-P-S:	4-1-0-0	Duration of USE:	180 Minutes
Total Contact Hours:	52 Hrs	USE Marks:	80 Marks
IA Marks:	20	Assignment	1 / Module
Credits:	4		
Lesson Plan Author:	Ms. Roopa G K	Sign	Date: 06/07/2017
Checked By:	Mr. Harivinod N	Sign	Date: 06/07/2017

B. PREREQUISITES

- Engineering Mathematics (15MAT31/15MAT41)
- Discrete Mathematical Structures (15CS36)
- Data Structures and its applications (15CS33)
- Design and analysis of Algorithms (15CS43)

C. COURSE DESCRIPTION

i) Course Outcomes

At the end of the course, the student will be able to;

1. Acquire fundamental understanding of the core concepts in automata theory and theory of computation.
2. Learn how to identify and translate between different formal language representations.
3. Design Grammars and Automata for different language classes.
4. Develop skills in formal reasoning and reduction of a problem to a formal model.
5. 4Determine the decidability and intractability of Computational problems.

ii) Relevance of the Course

- System Software and Compiler Design(15CS63)
- Operation Research(10CS653)

iii) Applications areas

- Text Processing.
- Artificial Intelligence and Programming Languages.
- Pattern Matching.
- Traffic control Systems.

Prepared by: Roopa G K Checked by: Harivinod N HOD

COURSE PLAN

D1. ARTICULATION MATRIX, CO v/s PO

Mapping of CO to PO												
	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1. Acquire fundamental understanding of the core concepts in automata theory and theory of computation.	3	3	2	2	2	1	-	1	1	1	1	1
2. Learn how to identify and translate between different formal language representations.	3	3	3	2	2	1	-	1	2	2	1	1
3. Design Grammars and Automata for different language classes.	3	3	3	3	3	-	-	-	3	1	1	2
4. Develop skills in formal reasoning and reduction of a problem to a formal model.	3	3	3	3	1	-	-	1	3	2	1	2
5. Determine the decidability and intractability of Computational problems.	3	3	2	3	1	1	-	-	1	2	1	1

Note: Mappings in the Tables D1 (above) and D2 (below) are done by entering in the corresponding cell the Correllation Levels in terms of numbers. For Slight (Low): 1, Moderate (Medium): 2, Substantial (High): 3 and for no correllation: " - ".

D2. ARTICULATION MATRIX, CO v/s PSO

Mapping of CO to PSO			
	PSOs		
COs	1	2	3
1. Acquire fundamental understanding of the core concepts in automata theory and theory of computation.	2	3	2
2. Learn how to identify and translate between different formal language representations.	1	3	-
3. Design Grammars and Automata for different language classes.	1	3	-
4. Develop skills in formal reasoning and reduction of a problem to a formal model.	2	3	-
5. Determine the decidability and intractability of Computational problems.	1	3	-

E. MODULE PLANS

MODULE - I

Title:	**Why study the Theory of Computation, Languages and Strings.**	Appr. Time:	10 Hrs
MO:			Bloom's Level
At the end of the Module, the student will be able to:			
1. Acquire fundamental understanding of the core concepts in automata theory and theory of computation.			L2
2. Understand and design Finite state Machines.			L3
3. Simulate and minimize FSMs.			L4

Lesson Schedule:

Lecture No.	Portion to be covered
1	Strings and Languages
2	A Language Hierarchy, Computation
3	Regular languages

4	Finite State Machines (FSM): Deterministic FSM
5	Designing FSM
6	Non deterministic FSMs
7	From FSMs to Operational Systems
8	Simulators for FSMs, Minimizing FSMs
9	Canonical form of Regular languages
10	Finite State Transducers, Bidirectional Transducers

Remarks:

Application Areas:

- Web Applications
- Gaming Theory.
- Network protocols
- Compiler design

Review Questions (CO):

1	What are the applications of theory of computation? (CO1)
2	Define different terminologies of strings. (CO1)
3	What are the different techniques for defining languages? (CO1)
4	What is deterministic and non deterministic finite state machine? (CO1)
5	Show that NDFSM has an equivalence DFSM. (CO1)
6	Differentiate between DFSM and NDFSM. (CO1)

MODULE - II

Title:	**Regular Expressions (RE)**	Appr. Time:	10 Hrs
MO:			Bloom's Level
At the end of the Module, the student will be able to:			
1. Learn how to identify and translate between different formal language representations.			L2
2. Identify different Formal language Classes and their Relationships			L3
3. Learn how to translate between different models of Computation.			L3

Lesson Schedule:

Lecture No.	Portion to be covered
1	What is a RE?
2	Kleene's theorem
3	Regular Grammars and Regular languages
4	Manipulating and Simplifying REs.
5	Regular Languages (RL)
6	Non regular Languages
7	How many RLs,
8	Closure properties of RLs
9	To show some languages are not RLs
10	To show that a language is regular

Remarks:

Application Areas:

- Development of State-charts model.
- Lexical analysis.
- Searching for regular expressions in files.

Review Questions (CO):

1	Define regular expression. (CO2)
2	State and prove Kleene's theorem. (CO2)
3	Prove that for Every Regular Expression There is an Equivalent FSM. (CO2)
4	Prove that for Every FSM There is an Equivalent Regular Expression. (CO2)
5	What are the applications of regular expressions? (CO2)
6	Define regular grammar. (CO2)
7	Prove that The class of languages that can be defined with regular grammars is exactly the regular languages. (CO2)
8	Prove that there is a countably infinite number of regular languages. (CO2)
9	Prove that every finite language is regular. (CO2)
10	Prove that the regular languages are closed under union, concatenation, and Kleene star. (CO2)
11	State and prove the Pumping Theorem for Regular Languages. (CO2)

MODULE - III

Title:	**Context-Free Grammars(CFG)**	Appr. Time:	10 Hrs
MO:			Bloom's Level
At the end of the Module, the student will be able to:			
1. Design Grammars and Automata for different language classes.			L2
2. Construct Parse tree.			L3
3. Identify ambiguous grammar with the help of parse tree.			L3
4. Obtain Left most and Right most Derivation for the string using the grammar.			L4

Lesson Schedule:

Lecture No.	Portion to be covered
1	Introduction to Rewrite Systems and Grammars
2	CFGs and languages
3	Designing and Simplifying CFGs
4	Proving that a Grammar is correct
5	Derivation and Parse trees, Ambiguity
6	Normal Forms.
7	Pushdown Automata (PDA)
8	Deterministic and Non-deterministic PDAs
9	Non-determinism and Halting, alternative equivalent definitions of a PDA
10	Alternatives that are not equivalent to PDA

Remarks:

Application Areas:

- XML and Document type definitions
- Markup Languages
- YACC Parser Generator

Review Questions (CO):

1	Define CFG. Give example. (CO3)
2	How do you simplify a grammar? (CO3)
3	Define derivation and parse tree. (CO3)
4	How do you decide ambiguity in grammar? (CO3)
5	Why is ambiguity a problem? (CO3)
6	Define different types of normal forms. (CO3)
7	Write algorithm to convert into Chomsky Normal Form. (CO3)
8	Define DPDA and NDPDA. (CO3)
9	Prove that for every CFG there exists an equivalent PDA. (CO3)

MODULE - IV

Title:	**Context-Free and Non-Context-Free Languages**	Appr. Time:	10 Hrs

MO:	Bloom's Level
At the end of the Module, the student will be able to:	
1. Develop skills in formal reasoning and reduction of a problem to a formal model.	L2
2. To understand the basic concepts of context free languages.	L3
3. List various notations of context free languages.	L3
4. Understand the applications of context free languages.	L3

Lesson Schedule:

Lecture No.	Portion to be covered
1	Where do the Context-Free Languages(CFL) fit
2	Showing a language is context-free
3	Pumping theorem for CFL
4	Important closure properties of CFLs
5	Deterministic CFLs
6	Algorithms and Decision Procedures for CFLs
7	Decidable questions, Un-decidable questions
8	Turing machine model, Representation
9	Language acceptability of TM
10	Design of TM, Techniques for TM construction

Remarks:

COURSE PLAN

Application Areas:
• Design of finite state systems. • Searching for regular expressions in files.

Review Questions (CO):	
1	Prove that the Context-Free Languages Properly Contain the Regular Languages. (CO4)
2	Prove that there is a countablv. infinite number of context free languages. (CO4)
3	Prove that The Pumping Theorem for Context-Free Languages. (CO4)
4	Prove that the context-free languages are closed under union. Concatenation, Kleene star. reverse, and Letter substitution. (CO4)
5	Prove that the context-free languages are not closed under intersection, complement, or difference. (CO4)
6	Prove that the context-free languages are dosed under intersection with the regular languages. (CO4)
7	Design a Turing machine to accept a Palindrome. (CO4)
8	Design a TM to recognize a string of the form $a^n b^{2n}$. (CO4)
9	Explain turing machine model. (CO4)

MODULE - V

Title:	**Variants of Turing Machines (TM)**	Appr. Time:	10 Hrs
MO:			Bloom's Level
At the end of the Module, the student will be able to:			
1. Determine the decidability and intractability of Computational problems			
2. Classify a problem with respect to different models of Computation.			
3. Learn various undecidable problems.			

Lesson Schedule:

Lecture No.	Portion to be covered
1	The model of Linear Bounded automata
2	Decidability: Definition of an algorithm
3	Decidability and decidable languages
4	Undecidable languages
5	Halting problem of TM
6	Post correspondence problem
7	Complexity: Growth rate of functions
8	The classes of P and NP
9	Quantum Computation
10	Church-Turing thesis

Remarks:

Application Areas:
• Gaming technology • Simulator design

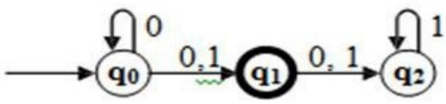
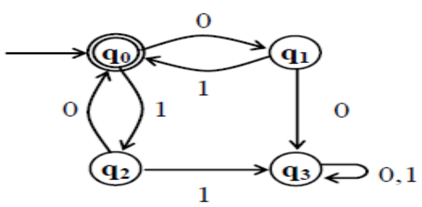
COURSE PLAN

Review Questions (CO):	
1	Explain multitape turing machine. (CO5)
2	Write a note on Post's Correspondence problem. (CO5)
3	Explain briefly the Halting problem of turing machine.(CO5)
4	Prove that If L is a recursive language, L is also recursive.(CO5)
5	Write a note on various types turing machines.(CO5)
6	Describe decidability.(CO5)
7	Define LBA.(CO5)
8	Explain the classes P and NP. (CO5)
9	What is quantum computation? (CO5)
10	Expalin Church-turing thesies. (CO5)

F. INTERNAL ASSESSMENT MODEL QUESTION PAPER (from Modules 1 & 2)

Dept:	Sem / Div:	Course:	Course Code:
Date:	Time: 90 Min.	Max Marks: 40	Elective: Y/N

Note: Answer any 2 FULL questions, each carry equal marks.

QN		Questions	Marks	Bloom's Level	CO no.
1	a	Why do we study theory of computation?	8	L2	CO1
	b	Construct DFSM for the following languages. a) strings of a's and b's ending with 'abb'. b) strings of a's and b's having substring 'aab'. c) $L = \{w \in \{0,1\}^* : w \text{ has odd parity}\}$.	6	L2	CO1
	c	Convert the following NDFSM to DFSM. 	6	L3	CO1
2	a	a) Obtain RE for the following problem.  b) Prove that the regular languages are closed under union, concatenation, and Kleene star.	8	L2	CO2
	b	Write regular expressions for the following. a)strings of a's and b's having odd length. b)strings of 0's and 1's having no two consecutive 0's. c) L= $\{a^{2n}b^{2m} \mid m>=0, n>=0\}$	6	L2	CO2
	c	Prove that for Every Regular Expression There is an Equivalent FSM.	6	L2	CO2
3	a	Define the following with proper examples:	8	L2	CO1

		a) Alphabet b) string c) length of a string d) concatenation of string e) reverse of a string f) substring g) language h)cardinality of a language			
	b	What are the applications of regular expressions?	6	L2	CO2
	c	**Simplify each of the following regular expressions:** **a.** $(a \cup b)^* (a \cup \varepsilon) b^*$. **b.** $(\emptyset^* \cup b) b^*$. **c.** $(a \cup b)^* a^* \cup b$. **d.** $((a \cup b)^*)^*$. **e.** $((a \cup b)^+)^*$.	6	L3	CO2

G. INTERNAL ASSESSMENT EVALUATION

Evaluation	Weightage in Marks	
IA Test – 1	15	
IA Test – 2	15	
IA Test – 3	15	
Attendance	1	5 Marks: To deepen student's understanding and increase his/her confidence in the topics studied. Further, to improve the oral, written skills and engineering aptitudes.
Assignment	2	
Concept Test / Class Test	2	
Seminar		
Quiz & Others		

H. QUESTIONS APPEARED IN THE PREVIOUS YEARS

Course:	Automata Theory and Computability	Course Code:	15CS54

Module	Sl. No.	Questions	Appeared in year
I	1	Why do we study theory of computation? (CO1)	-
	2	Define DFSM and NDFSM. Write notations of FSM. (CO1)	-
	3	Differentiate between DFSM and NDFSM. (CO1)	-
	4	Obtain NDFSM for : a) $L = \{w \in \{a,b\}^* : w$ is made up of an optional a followed by aa followed by zero or more b's} b) $L = \{w \in \{a,b\}^* : w = aba$ or $\|w\|$ is even$\}$. c) $L = \{w \in \{a, b\}^* :$ the fourth from the last character is a$\}$. (CO1)	-
	5	Obtain DFSM to accept a) strings of a's and b's having exactly one 'a'. b) strings of a's and b's starting with 'ab'. c)strings of a's and b's ending with 'abb'.	-

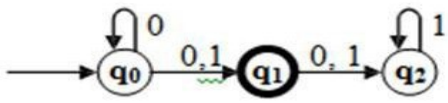
		d)strings of a's and b's having substring 'aab'. e) strings of 0's and 1's having 3 consecutive 0's Let $L = \{w \in \{a, b\}^* : w$ contains no more than one b$\}$. f) strings of a's and b's starting and ending with 'a' d)strings o $L = \{w \in \{0, 1\}^* : w$ has odd parity$\}$. (CO1)	
	6	Construct DFSM for the following languages. $L = \{w \in \{a, b\}^* :$ every a region in w is of even length$\}$. a) $L = \{w \in \{a, b\}^* :$ every a is immediately followed by a b$\}$. b) $L = \{w \in \{a, b\}^* :$ no two consecutive characters are the same$\}$. c) FLOAT = $\{w : w$ is the string representation of a floating point number$\}$. d) Let $L = \{w \in \{a - z\}^* :$ all five vowels, a, e, i, o, and u, occur in w in alphabetical e) f) g) order$\}$ (CO1)	-
	7	Write a DFSM for simple communication protocol. (CO1)	-
	8	Construct DFSM to accept strings of a's and b's having i)even no of a's and even no of b's. ii)odd no of a's and odd no of b's (CO1)	-
	9	Define the following with proper examples: a) Alphabet b) string c) length of a string di) concatenation of string e) reverse of a string f) substring g) prefix of a string h) suffix of a string i) language j) cardinality of a language (CO1)	-
	10	Define the language and write Lexicographic Enumeration for the following and i) All a's Precede All b's ii) Strings That End in a iii) The Empty Language iv) A Halting Problem language (CO1)	-
	11	Write a note on functions of languages. (CO1)	-
	12	Consider the language $L = \{1^n 2^n : n > 0\}$. Is the string 122 in L? (CO1)	-
	13	Let $L_1 = \{a^n b^n : n > 0\}$. Let $L_2 = \{c^n : n > 0\}$. For each of the following strings, state whether or not it is an element of L_1L_2: **a.** ε **b.** aabbcc. c. abbcc. **d.** aabbcccc. (CO1)	-
	14	Let $L1$ ={peach, apple, cherry} and L2 = {pie, cobbler, e}. List the elements of L_1L_2 in lexicographic order. (CO1)	-
	15	Let $L = \{w \in \{a, b\}^* : \|w\| \equiv_3 0\}$. List the first six elements in a lexicographic enumeration of L. (CO1)	-

COURSE PLAN

	16	Write a note on hierarchy of language classes. (CO1)	-

17 Convert the following NDFSM to DFSM. -

a)



b)

δ	a	b
q0	q0, q1	q0
q1	φ	q2
q2	φ	φ

c)

δ	0	1
q0	q0, q1	q1
q1	q2	q2
q2	φ	q2

d)

δ	a	b
q0	q0, q1	q0, q3
q1	φ	q2
q2	φ	φ
q3	q4	φ
q4	φ	φ

e)

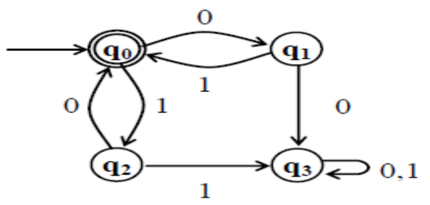
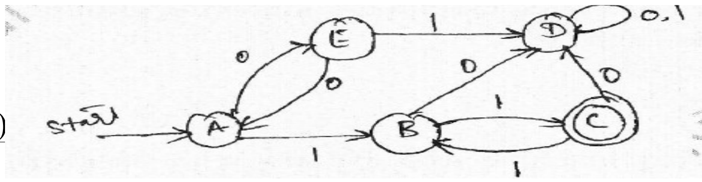
δ	0	1
p	p,q	p
q	φ	q
r	p,r	r

(CO1)

	18	Write the algorithms for simulating DFSM and NDFSM. (CO1)	-
II	1	Define regular expression. (CO2)	-

2 Write regular expressions for the following. -

a) strings of a's and b's of length <=2
b) strings of a's and b's having even legth.
c)strings of a's and b's having odd length.
d)strings of 0's and 1's having no two consecutive 0's.
e)strings of a's and b's starting with 'a' and ending with 'b'
f)to accept words with 2 or more letters but beginning and ending with

		the same letter. (CO2)	
	3	Obtain RE for the language a) L= $\{a^n b^m \mid m+n \text{ is even}\}$ b) L= $\{a^n b^m \mid m>=1, n>=1, nm>=3\}$ c)L= $\{a^{2n} b^{2m} \mid m>=0, n>=0\}$ d)L= $\{a^n b^m \mid n>=4, m<=3\}$ e)L= {w :|w| mod 3=0 where w ε (a,b)*} (CO2)	-
	4	State and prove Kleene's theorem.	-
	5	Prove that for Every Regular Expression There is an Equivalent FSM. (CO2)	-
	6	Prove that for … t Regular Expression. (CO2)	-
	7	 Obtain RE for the following problem. (CO2)	-
	8	 Obtain RE for the following problem. **Simplify each of the following regular expressions:** **a.** (a ∪ b)* (a ∪ ε) b*. **b.** (∅* ∪ b) b*. **c.** (a ∪ b)*a* ∪ b. **d.** ((a ∪ b)*)*. **e.** ((a ∪ b)⁺)*. (CO2	-
	9	What	-
	10	(CO2)	-
	11	Define regular grammar. (CO2)	-
	12	**Show a regular grammar for each of the following languages:** **a.** {w ∈ {a, b}*: w contains an even number of a's and an odd number of b's}. **b.** {w ∈ {a, b}*: w does not end in aa}. **c.** {w ∈ {a, b}*: w contains the substring abb}. **d.** {w ∈ {a, b}*: if w contains the substring aa then |w| is odd}. **e.** {w ∈ {a, b}*: w does not contain the substring aabb}. (CO2)	-
	13	Prove that the class of languages that can be defined with regular grammars is exactly the regular languages. (CO2)	-
	14	Prove that there is a countably infinite number of regular languages. (CO2)	-
	15	Prove that every finite language is regular. (CO2)	-
	16	Prove that the regular languages are closed under union, concatenation, and Kleene star. (CO2)	-
	17	State and prove the Pumping Theorem for Regular Languages. (CO2)	-

	18	Let $L = \{a^nb^m : n > m\}$. Let L be PalEven $= \{ww^R : w \in \{a, b\}^*\}$. Prove that the following language is not regular. a) b) c) d) ((	-
III	1	D	-
	2	Define Context-Free Grammar (CFG) and also obtain the CFG's,.for the following languages: (CO3)	-
	3	Consider the CFG with productions E → E*T | T T → T*F |F F → (E) | 0 | 1 Write LMD, RMD and parse tree for the string “0-((1*0)-0)” (CO3)	-
	4	Show that the following grammar is ambiguous. S → SbS |a	-
	5	Obtain grammar to generate integers and write the derivation for unsigned integer 1965. (CO3)	-
	6	Consider the grammar: S → aS |aSbS | ε Is the above grammar ambiguous? Show that the string aab has two - i) Parse trees ii) Left most derivations iii)	-
	7	Define CFG. Write CFG for the following languages. (CO3)	-
	8	How do you simplify a grammar? (CO3)	-
	9	Define derivation and parse tree. (CO3)	-
	10	How do you decide ambiguity in grammar? (CO3)	-
	11	Why is ambiguity a problem? (CO3)	-
	12	Define	-
	13	Write	-
	14	Define	-
	15	Prove	-
	16		-
	17	Define ... g CFG to CN S → A A → a B → A	-
	18	What i	-

i) $L_1 = \{a^n ww^R b^n \mid w \in \{0,1\}^* \text{ and } n \geq 2\}$

ii) $L_2 = \{a^k b^m c^n \mid m + n = k \text{ and } m, n \geq 1\}$

iii) $L_3 = \{w \in \{a\}^* \mid |w| \bmod 3 \neq |w| \bmod 2\}$.

(i) $L = \{0^n1^n \mid n \geq 1\}$

(ii) $L = \{$String l of a's and b's with equal number of a's and b's $\}$

Convert each of the following grammars to Chomsky normal form:

a. $S \to aSa$
$S \to B$
$B \to bbC$
$B \to bb$
$C \to \varepsilon$
$C \to cC$

b. $S \to ABC$
$A \to aC \mid D$
$B \to bB \mid \varepsilon \mid A$
$C \to Ac \mid \varepsilon \mid Cc$
$D \to aa$

c. $S \to aTVa$
$T \to aTa \mid bTb \mid \varepsilon \mid V$
$V \to cVc \mid \varepsilon$

COURSE PLAN

		S → aBa| abba | A → ab | AA B → aB | a. (CO3)	
	19	Convert the following CFG to CNF. S → aB | bA A → a | aS | bAA B → b | aS aBB. (CO3)	-
	20	Design a PDA for the following language : L = {ww^R{w ε {a,b} *}. Also, draw the transition diagram for the constructed PDA. Write the instantaneous description (ID) for the string 'abbbba'. (CO3)	-
	21	Convert the following CFG to a PDA that accepts the same language by empty stack: E → E+E | E*E | I I → Ia |Ib | I0 |I1 |a |b (CO3)	-
	22	Define a $L(M) = \{\omega C \omega^R \mid \omega \in (a+b)^*\}$ ng with transition diagram for the following language: L = {a"b" | n >=0 }. (CO3)	-
	23	Define PDA. Design PDA to accept the language by a final state and also give the graphical representation of PDA.(CO3)	-
	24	Convert the following CF'G to PDA: (CO3) S → aABB | aAA A → aBB |a B → bBB| A C → a	-
	25	Construct PDA for the language and simulate this PDA L= {$a^i b^j c^k$ | j = i + k, i, k> =0 } (CO3)	-
	26	Define PDA. Explain the language accepted by PDA. (CO3)	-
	27	Explain the PDA with two stocks. (CO3)	-
	28	Construct PDA for the language and simulate this PDA L= {$a^n b^{2n}$ | n> =1 } (CO3)	-
	29	Obtain a PDA to accept string of balanced parentheses. (CO3)	-
IV	1	Show that L: {$a^n b^n c^n$ | n >= 0} is not context free. (CO4)	-
	2	Prove that the context free languages are closed under union, concatenation and reversal. (CO4)	-
	3	Prove that the context free languages are not closed under difference, intersection and compliment. (CO4)	-
	4	Prove that the Context-Free Languages Properly Contain the Regular Languages. (CO4)	-
	5	Prove that there is a countably infinite number of context free languages.	-
	6	Prove that The Pumping Theorem for Context-Free Languages. (CO4)	-
	7	Define a Turing machine. Also, design a Turing machine to accept the set of all palindromes over {0,1}*. Write the transition diagram for the constructed Turing machine and write the sequenceof ID's for the input string '1001'. (CO4)	-
	9	Design a turing machine that perfonns the following function: and also write its transition diagram. (CO4)	-

$$q_0\omega \vdash^* q_f\omega\omega \text{ for any } \omega \in \{1\}^*$$

	10	Define a turing machine and explain the working of basic turing machine. (CO4)	-
	11	Design a turing machine to accept the set of all palindromes over {a.b}*. Also indicate the moves made by turing machine for the string 'aba'. (CO4)	-
	12	Construct a TM for L= { $a^nb^nc^n$ | n>=1 }. Give the graphical representation for obtained TM. (CO4)	-
V	1	Explain multitape turing machine. (CO5)	-
	2	Write a note on Post's Correspondence problem. (CO5)	-
	3	Explain briefly the Halting problem of turing machine. (CO5)	-
	4	Prove that If L is a recursive language, L is also recursive. (CO5)	-
	5	Write a note on various types turing machines. (CO5)	-
	6	Describe decidability.(CO5)	-
	7	Define LBA.(CO5)	-
	8	Explain the classes P and NP. (CO5)	-
	9	What is quantum computation? (CO5)	-
	10	Expalin Church-turing thesies. (CO5)	-