JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CS6503

THEORY OF COMPUTATION

Question Bank

III YEAR A & B / BATCH : 2016 -20

Vision of Institution

To build Jeppiaar Engineering College as an Institution of Academic Excellence in Technical education and Management education and to become a World Class University.

Mission of Institution

M1	To excel in teaching and **learning, research and innovation** by promoting the principles of scientific analysis and creative thinking
M2	To participate in the production, **development and dissemination of knowledge** and interact with **national and international communities**
M3	To equip students with **values, ethics and life skills** needed to enrich their lives and enable them to meaningfully contribute to the **progress of society**
M4	To prepare students **for higher studies and lifelong learning**, enrich them with the **practical and entrepreneurial skills** necessary to excel as future professionals and contribute to **Nation's economy**

Program Outcomes (POs)

PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Vision of Department

To emerge as a globally prominent department, developing ethical computer professionals, innovators and entrepreneurs with academic excellence through quality education and research.

Mission of Department

M1	To create **computer professionals** with an ability to identify and **formulate the engineering problems** and also to provide **innovative solutions** through **effective teaching learning process.**
M2	To **strengthen the core-competence** in computer science and engineering and to create an ability to **interact** effectively with industries.
M3	To produce engineers with good professional skills, **ethical values** and life skills for the **betterment of the society.**
M4	To encourage students towards **continuous and higher level learning** on technological advancements and provide a platform for **employment and self-employment.**

Program Educational Objectives (PEOs)

PEO1	**To address the real time complex engineering problems using innovative approach with strong core computing skills.**
PEO2	**To apply core-analytical knowledge and appropriate techniques and provide solutions to real time challenges of national and global society**
PEO3	**Apply ethical knowledge for professional excellence and leadership for the betterment of the society.**

PEO4	**Develop life-long learning skills needed for better employment and entrepreneurship**

Program Specific Outcomes (PSOs)

Students will be able to

PSO1	An ability to understand the core concepts of computer science and engineering and to enrich problem solving skills to analyze, design and implement software and hardware based systems of varying complexity.
PSO2	To interpret real-time problems with analytical skills and to arrive at cost effective and optimal solution using advanced tools and techniques.
PSO3	An understanding of social awareness and professional ethics with practical proficiency in the broad area of programming concepts by lifelong learning to inculcate employment and entrepreneurship skills.

BLOOM TAXANOMY LEVELS(BTL)

BTL6: Creating.,
BTL 5: Evaluating.,
BTL 4: Analyzing.,
BTL 3: Applying.,
BTL 2: Understanding.,
BTL 1: Remembering

SYLLABUS

THEORY OF COMPUTATION – CS6503 (V SEMESTER)

UNIT I FINITE AUTOMATA

9

Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions –Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA's with and without €-moves –Equivalence of finite Automaton and regular expressions –Minimization of DFA- - Pumping Lemma for Regular sets – Problems based on Pumping Lemma.

UNIT II GRAMMARS

9

Grammar Introduction– Types of Grammar - Context Free Grammars and Languages– Derivations and Languages – Ambiguity- Relationship between derivation and derivation trees – Simplification of CFG – Elimination of Useless symbols - Unit productions - Null productions – Greiback Normal form – Chomsky normal form – Problems related to CNF and GNF.

UNIT III PUSHDOWN AUTOMATA

9

Pushdown Automata- Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Equivalence of Pushdown automata and CFL - pumping lemma for CFL – problems based on pumping Lemma.

UNIT IV TURING MACHINES

9

Definitions of Turing machines – Models – Computable languages and functions –Techniques for Turing machine construction – Multi head and Multi tape Turing Machines - The Halting problem –Partial Solvability – Problems about Turing machine- Chomskian hierarchy of languages.

UNIT V UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS

9

Unsolvable Problems and Computable Functions – Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. MEASURING AND CLASSIFYING COMPLEXITY: Tractable and Intractable problems- Tractable and possibly intractable problems - P and NP completeness - Polynomial time reductions.

Total= 45 Periods

TEXT BOOKS:

1. *Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3)*
2. *John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (UNIT 4,5)*

REFERENCES:

1. Mishra K L P and Chandrasekaran N, "Theory of Computer Science -

Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.

2. Harry R Lewis and Christos H Papadimitriou, "Elements of the Theory of Computation", Second Edition, Prentice Hall of India, Pearson Education, New Delhi, 2003.

3. Peter Linz, "An Introduction to Formal Language and Automata", Third Edition, Narosa Publishers, New Delhi, 2002.

4. Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009

NOTE :REFER NOTES FOR PART B PROBLEMS

Course Outcomes (COs)

C504.1	ı Finite State Automata and Regular Expression for any language
C504.2	ıte the design of Context Free Grammar for any language set
C504.3	ıstrate the push down automaton model for the given language
C504.4	use of Turing machine concept to solve the simple problems
C504.5	n decidability or undesirability of various problems

INDEX

Unit #	Ref. Book
Unit 1	Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3) Mishra K L P and Chandrasekaran N, "Theory of Computer Science - Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.

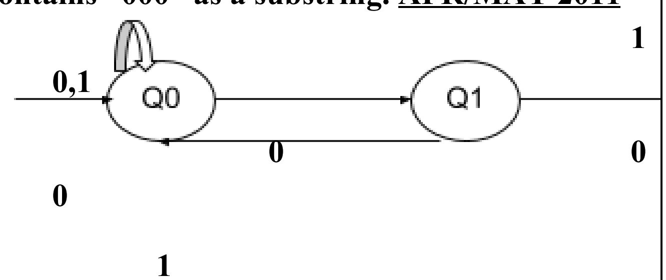
Unit 2	Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3) Mishra K L P and Chandrasekaran N, "Theory of Computer Science - Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.
Unit 3	Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (UNIT 1,2,3) Mishra K L P and Chandrasekaran N, "Theory of Computer Science - Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.
Unit 4	John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (UNIT 4,5)
Unit 5	John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (UNIT 4,5)

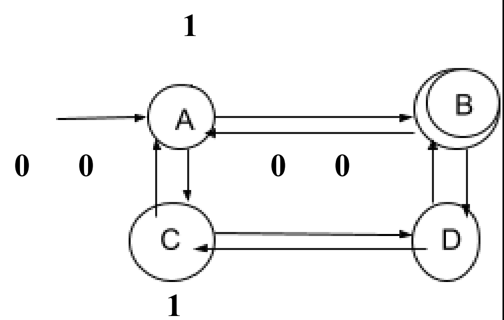
UNIT I FINITE AUTOMATA

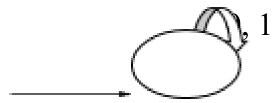
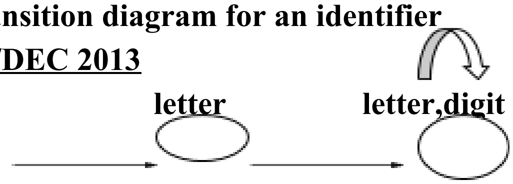
9

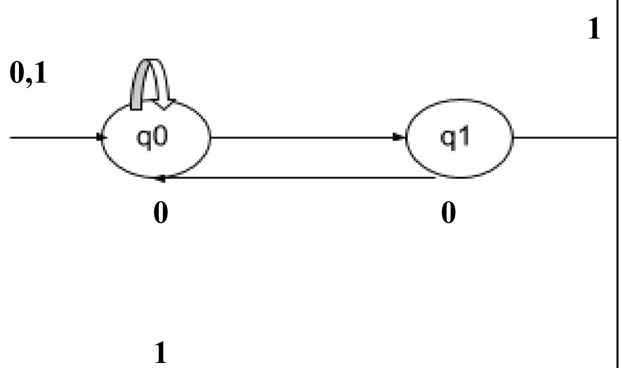
Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions –Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA's with and without €-moves –Equivalence of finite Automaton and regular expressions –Minimization of DFA- - Pumping Lemma for Regular sets – Problems based on Pumping Lemma.

S. No.	Question	Course Outcome	Blooms Taxanomy Level
1	(a) **Finite Automata (FA)** (b) **Transition Diagram** **<u>NOV/DEC 2012</u>** Automata is a 5 tuples denoted by), ∑, δ, q_0, F) where ● Q is a finite set of states ● ∑ is the finite set of input symbols ● δ is a transition function (Q X ∑ → Q)	C504.1	BTL 1

	• q0 is the start state or initial state • F is a set of final or accepting states		
2	**:he Principle of induction.** **<u>NOV/DEC 2012</u>** Refer notes	C504.1	BTL 1
3	**is proof by contradiction? <u>MAY/JUNE 2012</u>** Refer notes	C504.1	BTL 1
4	**: ε-closure(q) with an example.** **<u>MAY/JUNE 2012</u>** Refer notes	C504.1	BTL 1
5	**entiate between proof by contradiction and proof by contrapositive. <u>APR/MAY 2011</u>** ıen C will be proved by assuming ~H and then proving falsehood of falsehood of C. This is proof by contadiction. Proof by contrapositive is proved by assuming ~H and proving ~C.	C504.1	BTL 1
6	**ruct a DFA for the language over {0, 1}* such that it contains "000" as a substring. <u>APR/MAY 2011</u>** 	C504.1	BTL 1
7	**is structural induction?** **<u>NOV/DEC 2011</u>** (X) be a statement about the structures X that are defined by some particular recursive definition. 1. As a basis, Prove S(X) for the basis structure(s) X. 2. For inductive step, take a structure X that the recursive definition says is formed from Y_1,	C504.1	BTL 1

	Y_2,....Y_k. Assume the statements $S(Y_1)$,...,$S(Y_k)$ and use these to prove S(X).		
8	he difference between NFA and DFA. **<u>NOV/DEC 2011</u>** nust emit one and only one vertex/line/edge for each element of the alphabet. NFA do not have to obey this and can have multiple edges labeled with the same letter (repetition) and /or edges labeled with the empty string. DFA and NFA recognize the same languages – the regular languages.	C504.1	BTL 1
9	**1.** Construct deterministic finite automata to recognize odd number of 1's and even number of 0's? **<u>ɿR/MAY 2010</u>** 	C504.1	BTL 1
10	he relations among regular expression, deterministic finite automata, non deterministic finite automaton and finite automaton with epsilon transition. **<u>APR/MAY 2010</u>** Every Regular language defined by a regular expression ia also defined by the finite automata. If a Regular language 'L' is accepted by a NFA then there exists a DFA that accepts 'L'.	C504.1	BTL 1
11	**is inductive proof?** **<u>NOV/DEC 2010</u>** ıent P(n) follows from (a) P(0) and (b) P(n-1) implies P(n) for n>=1	C504.1	BTL 1

	ion (a) is an inductive proof is the basis and Condition (b) is called the inductive step.		
12	**he set of strings accepted by the finite automata.** **<u>NOV/DEC 2010</u>**  (0+1)* or L={ε, 0, 1, 00, 01, 10, 11,...........}	C504.1	BTL 1
13	**is meant by DFA** **<u>MAY/JUNE 2013</u>** -also known as **deterministic finite state machine**—is a <u>finite state machine</u> that accepts/rejects finite strings of symbols and only produces a unique computation (or run) of the automaton for each input string.	C504.1	BTL 1
14	**the term Epsilon transition** **<u>MAY/JUNE 2013</u>** In the <u>automata theory</u>, a **nondeterministic finite automaton with <u>ε</u>-moves (NFA-ε)**(also known as *NFA-<u>λ</u>*) is an extension of <u>nondeterministic finite automaton</u>(NFA), which allows a transformation to a new state without consuming any input symbols	C504.1	BTL 1
15	**the transition diagram for an identifier** **<u>NOV/DEC 2013</u>**  	C504.1	BTL 1
16	**is non deterministic finite automata?** **<u>NOV/DEC 2013</u>** <u>mata theory</u>, a **nondeterministic finite automaton** (NFA), or nondeterministic finite state machine, is a <u>finite state machine</u> that ***(1)*** does not require input symbols for state transitions and ***(2)*** is capable of transitioning to zero or two or more states for a given start state and input symbol	C504.1	BTL 1
17	Deductive Proof. **<u>NOV/DEC 2014</u>**		

	ductive proof consists of a sequence of statements whose truth leads us from some initial statement, called the 'hypothesis' to a 'conclusion' statement. "if H then C" **<u>Ex:</u>** if $x >= 4$ then $2^x >= x^2$	C504.1	BTL 1
18	**1.** Design DFA to accept strings over $\sum = (0,1)$ with two consecutive 0's.**<u>NOV/DEC 2014</u>** 	C504.1	BTL 1
19	s a finite automaton? **<u>NOV/DEC 2015</u>** e automaton (FA) is a simple idealized machine used to recognize patterns within input taken from some character set (or alphabet) C. The job of an FA is to accept or reject an input depending on whether the pattern defined by the FA occurs in the input. Automata is a 5 tuples denoted by), $\sum$, δ, q_0, F) where ● Q is a finite set of states ● $\sum$ is the finite set of input symbols ● δ is a transition function ($Q \times \sum \rightarrow Q$) ● q0 is the start state or initial state ● F is a set of final or accepting states	C504.1	BTL 1
20	Regular Expression for the set of strings over {0.1} that have atleast one. **<u>NOV/DEC 2015</u>** Regualr Expression = (0+1)*1	C504.1	BTL 1

21	a Non-deterministic finite automata to accept strings containing the substring 0101. **<u>MAY/JUNE 2016</u>** **Refer Notes**	C504.1	BTL 1
22	the pumping lemma for regular languages. **<u>MAY/JUNE 2016</u>Refer Notes**	C504.1	BTL 1
23	the languages described by DFA and NFA.) = { w / δ‟(q0,w) is in F}.It is the set of strings w that take the start state q0 to one of the accepting states.)= { w / δ‟(q0,w)∩F≠ }.It is the set of strings w such that δ‟(q0,w)contains at least one accepting state.	C504.1	BTL 1
24	extended transition function for a DFA. e extended transition function δ‟: Qε∑ * εQ is defined as follows. δ'(q, ε) = q (ε - Empty) Suppose w is a string of form xa(w= xa), wε∑*and q Q , then δ'(q, w)= δ(δ'(q, x),a)	C504.1	BTL 1
25	regular expression for the set of all strings having odd number of 1's RE= 1(0+11)*	C504.1	BTL 1
26	he regular expression for the set of all strings ending in 00. Regular expression = (0+1)*00	C504.1	BTL 1
27	two states are equivalent and distinguishable? y that two states p and q are equivalent iff for each input string x , δ(p,x) is an accepting state iff δ(q,x) is an accepting state. p is distinguishable from q if there exists an x such that δ(p,x) is in F and v is not in F or vice versa.	C504.1	BTL 1
28	are the applications of regular expression?	C504.1	BTL 1

	ır expression in UNIX, Lexical analysis, Pattern searching		
29	**regular expressions for the following.** ıry numbers that are multiple of 2. (0/1)*. rings of a‟s and b‟s with no consecutive a‟s .b* (abb*)(a / ε) (iii) Strings of a‟s and b‟s containing consecutive a‟s. (a/b)*aa(a/b)*	C504.1	BTL 1
30	**State Arden's theorem.** Let P and Q be two regular expressions over ∑. If P does not contain null string ε over ∑ then R=Q+RP, it has the solution R=QP*	C504.1	BTL 1
	PART B		
1	**in the different forms of proof with examples. (8) NOV/DEC 2012**	C504.1	BTL 1
2	**that, if L is accepted by an NFA with ε-transitions, then L is accepted by an NFA without ε-transitions. (8) NOV/DEC 2012, NOV/DEC 2013**	C504.1	BTL 1
3	**that if n is a positive integer such that n mod 4 is 2 or 3 then n is not a perfect square. (6) NOV/DEC 2012**	C504.1	BTL 1
4	**ruct a DFA that accepts the following** (i) **L={ x € {a,b}:$\|x\|_a$ = odd and $\|x\|_b$ = even}. (10) NOV/DEC 2012** (ii) **Binary strings such that the third symbol from the right end is 1. (10) MAY/JUNE 2012** (iii) **All strings w over {0,1} such that the number of 1's in w is 3 mod 4. (8) NOV/DEC 2011** (iv) **Set of all strings with three consecutive 0's.(10) NOV/DEC 2010**	C504.1	BTL 1

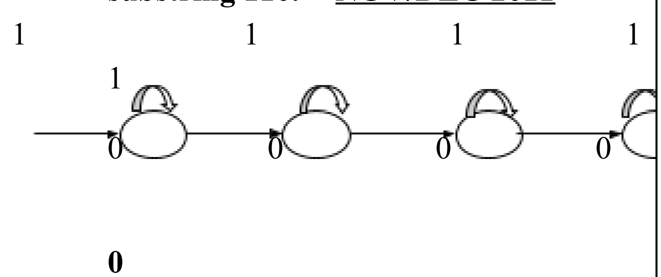
5	**by induction on n that** $\sum_{i=0}^{n} i = n(n+1)/2$ **(6) <u>MAY/JUNE 2012</u>**	C504.1	BTL 1
6	**ruct an NFA without ε-transitions for the NFA give below. (8) <u>MAY/JUNE 2012</u>** 	C504.1	BTL 1
7	**ruct an NFA accepting binary strings with two consecutive 0's. (8) <u>MAY/JUNE 2012</u>**		
8`	**that a connected graph G with n vertices and n-1 edges (n>2) has at least one leaf. (6) <u>APR/MAY 2011</u>** ● G has n vertices & (n-1) edges. ● Therefore ∑ deg(V) = 2(n-1) which is impossible ● Therefore deg(V) = 1 for at least one vertex and this vertex is a leaf.		
9	**that there exists a DFA for every ε-NFA. (8) <u>APR/MAY 2011</u>** Refer Notes		
10	**guish NFA and DFA with examples. <u>MAY/JUNE 2013</u>**		

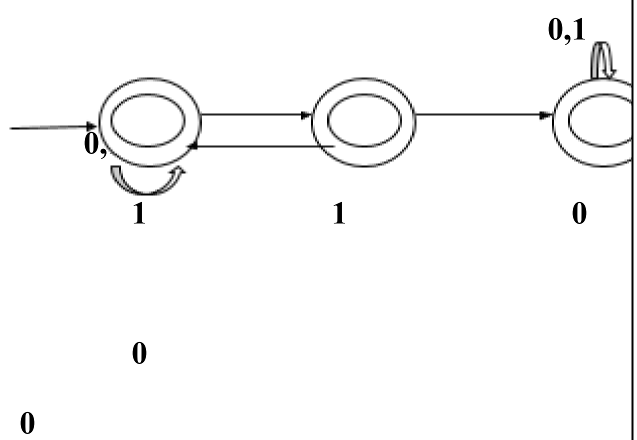
UNIT II GRAMMARS

9

Grammar Introduction– Types of Grammar - Context Free Grammars and Languages– Derivations and Languages – Ambiguity- Relationship between derivation and derivation trees – Simplification of CFG – Elimination of Useless symbols - Unit productions - Null productions – Greiback Normal form – Chomsky normal form – Problems related to CNF and GNF.

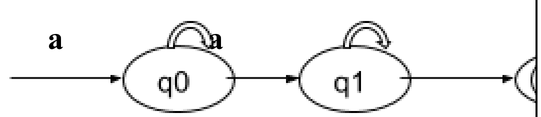
S. No.	Question	Course Outcome	Blooms Taxanomy Level
1	**egular expressions for the following** **t of all strings of 0 and 1 ending in 00** **t of all strings of 0 and 1 beginning with 0 and ending with 1. <u>NOV/DEC 2012</u>** $RE1=(0+1)^{+}00$ $RE2=0(0+1)^{+}1$	C504.2	BTL 1

2	**entiate regular expression and regular language.** **<u>NOV/DEC 2012</u>** Refer notes	C504.2	BTL 1
3	**ruct NFA for the regular expression a*b*.** **<u>MAY/JUNE 2012</u>** Refer notes	C504.2	BTL 1
4	**ılar set is closed under complementation?Justify.** **<u>MAY/JUNE 2012</u>** <u>e under complement</u> ● If L is a regular language, over alphabet ∑ Complement of L=∑* - L ● Let L = L(A) for DFA A = (Q, ∑, δ, q0, F) ● Complement of L = L(B) where DFA B = (Q, ∑, δ, q0, Q-F) ● B is similar to A except accepting states of A have become non-accepting states of B and vice-versa. A string w is in L(B) iff δ' (q0,w) is in Q-F which occurs iff w is not in L(A).	C504.2	BTL 1
5	**that the complement of a regular language is also regular.<u>APR/MAY 2011</u>** ement of L_1 is constructed from L_1 by reversing the states and the arrows in automata.	C504.2	BTL 1
6	**by pumping lemma, that the language 0^n1^n is not regular.<u>APR/MAY 2011</u>**	C504.2	BTL 1
7	**ruct a DFA for the following:** (a) **All strings that contain exactly 4 zeros.** (b) **All strings that don't contain the substring 110. <u>NOV/DEC 2011</u>**  	C504.2	BTL 1

			
8	**set of strings over the alphabet {0} of the form 0n where n is not a prime is regular? Prove or disprove.** **<u>NOV/DEC 2011</u>** ove this language is not regular, examine the complement because the set of regular languages is closed under complement. e that the set is regular. Let p be the pumping length of the language. Then, according to the pumping lemma, break the string $s=0^p$ into s=xyz where y has positive length. $s=xy^iz=0^{p+(i-1)\|y\|}$ must also be in the set for any i. In particular let i=p+1.Then $xy^{p+1}z=0^{p+p\|y\|}$ must be in the set so p+p|y| = p(1+|y|) must be prime. ve have a contradiction and the set cannot be regular.	C504.2	BTL 1
9	**= {w:w ε {0,1}* w does not contain 00 and is not empty}. Construct a regular expression that generates L.** **<u>APR/MAY 2010</u>** r Expression = (0+1)(1+10)*(101+1)*	C504.2	BTL 1
10	**or disprove that the regular languages are closed under concatenation and complement.** **<u>APR/MAY 2010</u>** <u>e under concatenation</u>	C504.2	BTL 1

	• Since L and M are regular languages,they have regular expressions L=L(R) and M=L(S) • Then L.M = L(R.S), by definition of regular expression osure under complement • If L is a regular language, over alphabet ∑ Complement of L=∑* - L • Let L = L(A) for DFA A = (Q, ∑, δ, q0, F) • Complement of L = L(B) where DFA B = (Q, ∑, δ, q0, Q-F) • B is similar to A except accepting states of A have become non-accepting states of B and vice-versa. • A string w is in L(B) iff δ' (q0,w) is in Q-F which occurs iff w is not in L(A).		
11	**he regular expression for set of all strings ending in 00.<u>NOV/DEC 2010</u>** (0+1)*00	C504.2	BTL 1
12	**pumping lemma for regular set. <u>NOV/DEC 2010, NOV/DEC 2013,NOV/DEC 2014</u>** be a regular set. Then there is a constant n such that if Z is a string in L and |Z| >=n, Z can be written as Z=UVW such that |V|>=1 and |UV|<=n and for all i>=0 UV^iW is in L.	C504.2	BTL 1
13	**is a regular expression ? <u>MAY/JUNE 2013</u>** **ular expression** (abbreviated **regex** or **regexp**) is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations	C504.2	BTL 1
14	**any four closure properties of regular languages <u>MAY/JUNE 2013</u>** ion, intersection, complement, difference	C504.2	BTL 1

15	**ruct NFA equivalent to the regular expression (0+1)01** **<u>NOV/DEC 2013</u>** Refer notes	C504.2	BTL 1
16	**or disprove that (r+s)* = r* + s*** **<u>NOV/DEC 2014</u>** Refer notes	C504.2	BTL 1
17	**be the grammar with** **S→aB|bA** **A→a|aS|bAA** **B→b|bS|aBB**	C504.2	BTL 1
18	do you mean by null production and unit production? Give an example. **<u>MAY / JUNE 2016</u>** Refer notes	C504.2	BTL 1
19	uct a CFG for set of strings that contain equal number of a's and b's over ∑ = {a,b}. **<u>MAY / JUNE 2016</u>** **S->A|B** **A→aA|€** **B→bB|€**	C504.2	BTL 1
20	**What is unambiguity?** Refer notes	C504.2	BTL 1
21	**Mention the application of CFG.** Refer notes	C504.2	BTL 1
22	**ruct the Context free grammar representing the set of palindromes over (0+1)*** **<u>NOV/DEC 2015</u>** **S->0S0|1S1| ε**	C504.2	BTL 1
23	**is meant by Context Free Grammar (CFG)?** **<u>NOV/DEC 2016</u>**	C504.2	BTL 1
24	**Chomsky normal form theorem.** **<u>NOV/DEC 2016</u>**	C504.2	BTL 1
25	**Define Regular Expression.** Refer Notes	C504.2	BTL 1

26	**What is null and unit production?** Refer Notes	C504.2	BTL 1
	PART B		
1	**Prove that there exists an NFA with ε-transitions that accepts the regular expression γ.** **(10) <u>MAY/JUNE 2012, NOV/DEC 2010</u>**	C504.2	BTL 1
2	**Which of the following languages is regular? Justify.(Using Pumping Lemma)** (i) **$L=\{a^n b^m \mid n,m>=1\}$** (ii) **$L=\{a^n b^n \mid n>=1\}$** **(8) <u>MAY/JUNE 2012</u>** (iii) **$L=\{a^m b^n \mid m>n\}$** **(10) <u>NOV/DEC 2012</u>** (iv) **$L=\{a^n b^n \mid n>=1\}$** **(6) <u>NOV/DEC 2010</u>** (v) **$L=\{0^{n^2} \mid$ n is an integer, n>=1}** **(6) <u>NOV/DEC 2014</u>**	C504.2	BTL 1
3	**Obtain the regular expression for the finite automata. (8) <u>MAY/JUNE 2012</u>** 	C504.2	BTL 1
4	**Prove any two closure properties of regular languages.(8)<u>NOV/DEC 2012, NOV/DEC 2011, APRIL/MAY 2010</u>**	C504.2	BTL 1
5	**ruct a minimized DFA from the regular expression** (i) **(b/a)*baa** **(10) <u>MAY/JUNE 2012</u>** (ii) **0*(01)(0/111)*** **(16) <u>NOV/DEC 2012</u>**	C504.2	BTL 1

	(iii) **(x+y)x(x+y)*. Trace for a string w=xxyx.** **(16) <u>NOV/DEC 2011</u>** (iv) **(a+b)(a+b)* and trace for a string baaaab.** **(16) <u>APR/MAY 2010</u>** (v) **(b/a)*baa** **(16) <u>NOV/DEC 2010</u>** (vi) **10+(0+11)0*1** **(16) <u>NOV/DEC 2014</u>**		
6	**Construct a regular expression for the following DFA using kleene's theorem. (10) <u>APR/MAY 2011</u>** 0 1 →*A A B B C B A B	C504.2	BTL 1
7	**Construct a ε-NFA for the following regular expression. (6) <u>APR/MAY 2011</u>** (i) **(0+1)*(00+11)(0+1)***	C504.2	BTL 1
8	**(i) What is the purpose of normalization? Construct the CNF and GNF for the following grammar and explain the steps. (10) <u>MAY/JUNE 2016</u>** **S→aAa | bBb |€** **A→C|a** **B→C|b** **C→CDE | €** **D→A|B|ab** **(ii) Constuct a CFG for the regular expression (011+1)(01). (6) <u>MAY/JUNE 2016</u>**	C504.2	BTL 1

UNIT III PUSHDOWN AUTOMATA

9

Pushdown Automata- Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Equivalence of Pushdown automata and CFL - pumping lemma for CFL – problems based on pumping Lemma.

S. No.	Question	Course Outcome	Blooms Taxanomy Level
1	**is ambiguous grammar? <u>NOV/DEC 2012, MAY/JUNE 2013</u>** Refer notes	C504.3	BTL 1
2	**are the diiferent types of language accepted by a PDA and define them? <u>NOV/DEC 2012</u>** ● Accepted by null state ● Accepted by final state	C504.3	BTL 1
3	**y the use of context free grammar. <u>MAY/JUNE 2012</u>** Refer notes	C504.3	BTL 1

4	**parse tree with an example.** **<u>MAY/JUNE 2012</u>** Refer notes	C504.3	BTL 1
5	**ruct a CFG over {a,b} generating a language consisting of equal number of a's and b's.** **<u>APR/MAY 2011</u>** S→ aSbS | bSaS | SS	C504.3	BTL 1
6	**language of Deterministic PDA and Non – deterministic PDA same? <u>APR/MAY 2011</u>** nguage is not same. This language of NPDA is a superset of the language of DPDA.	C504.3	BTL 1
7	**grammar below ambiguous S→ SS | (S) | S(S)S | E?<u>NOV/DEC 2011</u>** ● It is ambiguous ● The sentence such as E(E)E can have more than one LMD (or) RMD (or) Parse tree.	C504.3	BTL 1
8	**rt the following grammar into an equivalent one with no unit productions and no useless symbols S→ABA A→aAA|aBC|bB B→ A|bB|Cb C→CC|Cc <u>NOV 2011</u>** Refer notes	C504.3	BTL 1
9	**der the following grammar G with productions** **<u>APR/MAY 2010</u>** **S→ ABC | BaB** **A→aA|BaC|aaa** **B→bBb|a** **C→CA|AC** **CFG with no useless variables that generates the same language.** l C is non-generative, after removing productions with C we have, S→ BaB	C504.3	BTL 1

	A→ aA|aaa B→bBb|a CFG with no usless variables 3aB A→ aA|aaa B→bBb|a}		
10	**:he definition of Pushdown automata.** **<u>APR/MAY 2010</u>** ıdown automaton consists of 7 tuples P = (Q, ∑, Γ, δ, q0, Z0, F) Q – A finite non empty set of states ∑ - A finite set of input symbols Γ – A finite non empty set of stack symbols δ - The transition function is given by δ:Q X (∑ U {ε}) X Γ → Q X Γ* q0 – q0 in Q is the start state Z0 - Initial start symbol of the stack F - F in Q, set of accepting states or final states.	C504.3	BTL 1
11	**down the context free grammar for the language L = { $a^n b^n$ |n>=1} <u>NOV/DEC 2010, NOV/DEC 2013</u>** S→ aSb |ab	C504.3	BTL 1
12	**grammar E→E+E|id is ambigupus ? Justify.** **<u>NOV/DEC 2010</u>** ● It is ambiguous, The sentence such as id+id+id can have more than one LMD (or) RMD (or) Parse tree.	C504.3	BTL 1
13	**is a CFG ? <u>MAY/JUNE 2013</u>** **:text-free grammar (CFG)** is a <u>formal grammar</u> in which every <u>production rule</u> is of the form *V* → *w* where *V* is a *single* <u>nonterminal</u> symbol, and *w* is a string of <u>terminals</u> and/or nonterminals (*w* can be	C504.3	BTL 1

	empty). A formal grammar is considered "context free" when its production rules can be applied regardless of the context of a nonterminal. No matter which symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side.		
14	**are the different ways of language acceptances by a PDA and define them?** <u>**NOV/DEC 2015**</u> ● Language accepted by a Empty store ● Language accepted by a Final state.	C504.3	BTL 1
15	**Define pushdown Automaton** <u>**MAY/JUNE 2016**</u> A pushdown Automaton is a ε•- NFA with stack data structure where it can push strings into and pop strings out of stack. It consists of 7-tuples P = (Q,Σ,Γ,δ,q_0, z_0 , F),Where Q is a finite set of states,Σ•is a finite set of input symbols,Γis a finite set of nonempty stack alphabets, δ•is a transition function,q_0 is an initial state, Z_0 is the initial start symbol of the stack , F is the set of accepting states.δ•is defined as $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to Q \times \Gamma^*$	C504.3	BTL 1
16	**What are the different ways of language acceptances by a PDA and** define t**hem** There are two ways of language acceptances, 1)Acceptance by final state .L(M)= {w | (q_0,w,Z_0) Γ•*(q, ε,γ) for some q in F and γ•in Γ•*}	C504.3	BTL 1
17	**are NFA and PDA**<u>**NOV/DEC 2013**</u> **Refer notes**	C504.3	BTL 1
18	**he general forms of CNF.** <u>**NOV/DEC 2014**</u> **A→BC** **A→a**	C504.3	BTL 1
19	**that CFLs are closed under substitutions** <u>**NOV/DEC 2014**</u> **Refer notes**	C504.3	BTL 1

20	**rt the following CFG to a PDA.<u>NOV/DEC 2015</u>** **S→aAA, A→aS|bS|a**	C504.3	BTL 1
21	Does a Push down Automata have memory? Justify. **<u>MAY/JUNE 2016</u>** Refer Notes	C504.3	BTL 1
22	When is Push Down Automata (PDA) said to be deterministic? **<u>NOV/DEC 2016</u>**	C504.3	BTL 1
23	What are the conventional notations of Push Down Automata? **<u>NOV/DEC 2016</u>**	C504.3	BTL 1
24	ne main application of pumping Lemma in CFL's	C504.3	BTL 1
25	Compare Deterministic and Non deterministic PI it true that non deterministic PDA is more power that of deterministic PDA? Justify your answer.	C504.3	BTL 1
26	Design the equivalence of PDA and CFG	C504.3	BTL 1
<u>PART – B</u>			
1	**Consider the following grammar for list structures:** **S→a|^|(T) T→T,S|S** **Find left most derivation, rightmost derivation and parse tree for (((a,a),^(a)),a)(10) <u>NOV/DEC 2012</u>**	C504.3	BTL 1
2	**Construct the PDA accepting the language** 1. **L={(ab)^n|n>=1} by empty stack.(6) <u>NOV/DEC 2012</u>** 2. **L={a^{2n}b^n|n>=1} Trace your PDA for the input with n=3. (10) <u>NOV/DEC 2012</u>** 3. **L={ww^R|w is in (a+b)*} (10)<u>MAY/JUNE 2012</u>**	C504.3	BTL 1

	4. **$L=\{0^n1^{2n}\}$ by empty stack(8)<u>APR/MAY 2011</u>** 5. **$L=\{ww^Rw \mid w$ is in $\{0+1\}^*\}$(10)<u>NOV/DEC 2010</u>**		
3	**Find the PDA equivalent to the given CFG with the following productions** 1. **S→A, A→BC, B→ba, C→ac (6) <u>NOV/DEC 2012</u>** 2. **S→aSb|A, A→bSa|S| ε (10) <u>NOV/DEC 2011</u>**	C504.3	BTL 1
4	**Is the following grammar is ambiguous? Justify your answer.** 1. **E→ E+E |E*E | id (6) <u>MAY/JUNE 2012</u>** 2. **E→ E+E|E*E|(E)|a (4) <u>APRIL/MAY 2011</u>**	C504.3	BTL 1
5	**Find the context free languages for the following grammars.** 1. **S→aSbS|bSaS| ε <u>MAY/JUNE 2012</u>** 2. **S→aSb|ab** } **(10)** 3. **S→aSb|aAb, A→bAa, A→ba (6) <u>NOV/DEC 2011</u>**	C504.3	BTL 1
6	**that if there exists a PDA that accepts by final state then there exists an equivalent PDA that accepts by Null state. (8) <u>APRIL/MAY 2011</u>**	C504.3	BTL 1
7	**Is NPDA (Nondeterministic PDA) and DPDA (Deterministic PDA) equivalent? Illustrate with an example. (8) <u>NOV/DEC 2011</u>**	C504.3	BTL 1
8	**What are the different types of language acceptances by a PDA and define them. Is it true that the language accepted by a PDA by these different types provides different languages? (8) <u>NOV/DEC 2011</u>**	C504.3	BTL 1
9	**Construct PDA for the language**	C504.3	BTL 1

	L = {ww^R | W in (a+b)*)} **<u>MAY/JUNE 2013,NOV/DEC 2013, MAY/JUNE 2016</u>**		

UNIT IV TURING MACHINES

9

Definitions of Turing machines – Models – Computable languages and functions –Techniques for Turing machine construction – Multi head and Multi tape Turing Machines - The Halting problem –Partial Solvability – Problems about Turing machine- Chomskian hierarchy of languages.

S. No.	Question	Course Outcome	Blooms Taxanomy Level
1	**:he pumping lemma for CFLs.** **<u>NOV/DEC 2012</u>** Refer notes	C504.4	BTL 1
2	**are the applications of Turing Machine?** **<u>NOV/DEC 2012</u>** Refer notes	C504.4	BTL 1
3	**pumping lemma for CFL.** **<u>MAY/JUNE 2012</u>** Refer notes	C504.4	BTL 1
4	**is chomsky normal form?** **<u>MAY/JUNE 2012</u>** Refer notes	C504.4	BTL 1

5	**is the height of the parse tree to represent a string of length 'n' using Chomsky normal form?** **<u>APR/MAY 2011</u>** n+1	C504.4	BTL 1
6	**ruct a Turing machine to compute 'n mod 2' where n is represented in the tape in unary form consisting of only 0's.** **<u>APR/MAY 2011</u>** 0 B q0 (q1,B,R) (q2,B,R) q1 (q0,B,R) (q3,B,R) q2 - - representing even no q3 - - representing odd no	C504.4	BTL 1
7	**a TM that accepts the language of odd integers written in binary.** **<u>NOV/DEC 2011</u>** ept odd valued binary strings, we only have to look at the last bit. The TM moves right until it reads a blank, moves left one space and acepts if and only if there is a 1 on the tape.	C504.4	BTL 1
8	**he two normal forms and give an example.** **<u>NOV/DEC 2011</u>** 1. Chomsky normal form A→BC |a 2. Greibach normal form X→bYXZ|a	C504.4	BTL 1
9	**rt the following grammar G in greibach normal form.<u>APR/MAY 2010</u>** **S→ABb|a A→aaA|B B→bAb** ● No ε production in the given grammar ● Eliminating unit production A→B we have S→ABb|a A→aaA|bAb B→bAb ● Eliminating useless variables A & B (non generating) S→a	C504.4	BTL 1
	a Turing machine with no more than three states that accepts the language a(a+b)*. Assume ∑ = {a,b} **<u>APR/MAY 2010</u>**	C504.4	BTL 1

10	=(Q, ∑, Γ, δ, q0, B,{q2}) Q – {q0,q1,q2} ∑ - {a,b} Γ – (a,b,B} q0 – Initial state q2 - Final state δ - Transition function given as follows δ(q0,a) = (q1,a,R) δ(q1,a) = (q1,a,R) δ(q1,b) = (q1,b,R) δ(q1,B) = (q2,B,R)		
11	**is Turing machine?** **<u>NOV/DEC 2010</u>** denoted by M=(Q, ∑, Γ, δ, q0, B,F) Q – A finite non empty set of states ∑ - A finite set of input symbols Γ – A finite non empty set of tape symbols δ - The transition function is given by δ:Q X Γ → Q X Γ X {L,R,S} q0 – Initial state B ε Γ – Blank Symbols F - Final state	C504.4	BTL 1
12	are the required fields of an instantaneous description of a Turing machine? **<u>NOV/DEC 2016</u>**	C504.4	BTL 1

13	e primary objectives of Turing Machine. **NOV/DEC 2016**	C504.4	BTL 1
14	**e language L={$a^nb^nc^n$ | n>=1} is context free?Justify.NOV/DEC 2010**	C504.4	BTL 1
15	**is meant by Greibach Normal Form ? MAY/JUNE 2013** ext-free grammar is in **Greibach normal form** (GNF) if the right-hand sides of all production rules start with a terminal symbol, optionally followed by some variables. A non-strict form allows one exception to this format restriction for allowing the empty word (epsilon, ε) to be a member of the described language	C504.4	BTL 1
16	**e closure properties of Context Free Languages MAY/JUNE 2013, NOV/DEC 2013** , intersection, kleene closure, substitution, homomorphism	C504.4	BTL 1
17	**ut the different techniques for turing machine construction.NOV/DEC 2013** **Refer notes**	C504.4	BTL 1
18	**G be the grammar S→aB|bA A→a|aS|bAA B→b|bS|aBB. For the string aaabbabbba, Find (a) LMD (b) RMD NOV/DEC 2014** Refer Notes	C504.4	BTL 1
19	**Diagonalization (Ld) Language. NOV/DEC 2014** Ld={wi | wi ∑ L(Mi)}	C504.4	BTL 1
20	**a Turing Machine. OV/DEC 2015** Refer notes	C504.4	BTL 1
21	**is a multitape turing machine? NOV/DEC 2015**Refer notes	C504.4	BTL 1

22	ıre the differences between a Finite automata and a Turing machine? **<u>MAY/JUNE 2016</u>**	C504.4	BTL 1
23	s Turing Machine? **<u>MAY/JUNE 2016</u>**	C504.4	BTL 1
24	s multitape Turing machine? Explain in one move. What are the actions take place in TM?	C504.4	BTL 1
25	ıre the applications of Turing Machine?	C504.4	BTL 1
26	ıre the techniques for TM construction?	C504.4	BTL 1
PART – B			
1	**Convert the following grammar into CNF** **S→cBA, S→A, A→cB, A→AbbS, B→aaa (6) <u>NOV/DEC 2012</u>** **S→a|AAB, A→ab|aB| ε, B→aba| ε (8) <u>APR/MAY 2011</u>** **S→A|CB, A→C|D, B→1B|1, C→0C|0, D→2D|2 (16) <u>APR/MAY 2010</u>** **S→aAD A→aB|bAB B→ b D→ d (6) <u>NOV/DEC 2014</u>**	C504.4	BTL 1
2	**State and prove the pumping lemma for CFL. What is its main application? Give two examples. (10)** **<u>NOV/DEC 2012, NOV/DEC 2011,</u> MAY/JUNE**	C504.4	BTL 1
3	**Design a Turing machine for the following** **Reverses the given string {abb}. (8) <u>NOV/DEC 2012</u>** **L={$1^n0^n1^n$|n>=1} (10) <u>MAY/JUNE 2012</u>** **L={$a^nb^nc^n$} (8) <u>APR/MAY 2011</u>**	C504.4	BTL 1

	To perform proper subtraction (8) <u>APR/MAY 2011</u> **To move an input string over the alphabet A ={a} to the right one cell. Assume that the tape head starts somewhere on a blank cell to the left of the input string. All other cells are blank, labeled by ^. The machine must move the entire string to the right one cell, leaving all remaining cells blank. (10) <u>APR/MAY 2010</u>** **$L=\{1^n0^n \| n>=1\}$ (8) <u>NOV/DEC 2010</u>** **$L=\{ww^R \|$ w is in $(0+1)^*\}$ (8) <u>NOV/DEC 2010</u>** **mplement the function "MULTIPLICATION" using the subroutine "COPY". (12) <u>NOV/DEC 2014</u>** **$L=\{0^n1^n \| n>=1\}$ (10) <u>NOV/DEC 2015</u>**		
4	**Write briefly about the programming techniques for TM. (8) <u>NOV/DEC 2012,MAY/JUNE 2013, NOV/DEC 2015</u>**	C504.4	BTL 1
5	**Find Greibach normal form for the following grammar** (i) **$S \rightarrow AA \| 1$, $A \rightarrow SS \| 0$ (10) <u>MAY/JUNE 2012</u>** (ii) **$S \rightarrow a \| AB$, $A \rightarrow a \| BC$, $B \rightarrow b$, $C \rightarrow b$ (4) <u>APR/MAY 2011</u>** (iii) **$S \rightarrow AA \| 0$, $A \rightarrow SS \| 1$ (8) <u>NOV/DEC 2010</u>** (iv) **$A_1 \rightarrow A_2A_3$, $A_2 \rightarrow A_3A_1 \| b$, $A_3 \rightarrow A_1A_2 \| a$ (10) <u>NOV/DEC 2014</u>**	C504.4	BTL 1
6	**Explain the different models of Turing machines. (10) <u>NOV/DEC 2011</u>**	C504.4	BTL 1

7	**Discuss the various techniques for Turing Machine Construction (16) <u>NOV/DEC 2016</u>**	C504.4	BTL 1
8	**(i) Write about Multi tape Turing Machines. (10) <u>NOV/DEC 2016</u>** **(ii) Explain highlight the implications of halting problems (6) <u>NOV/DEC 2016</u>**	C504.4	BTL 1
9	**Describe the Chomsky hierarchy of languages. <u>NOV/DEC 2015</u>**	C504.4	BTL 1

UNIT V UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS

9

Unsolvable Problems and Computable Functions – Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. MEASURING AND CLASSIFYING COMPLEXITY: Tractable and Intractable problems- Tractable and possibly intractable problems - P and NP completeness - Polynomial time reductions.

PART - A

S. No.	Question	Course Outcome	Blooms Taxanomy Level
1	**we say a problem is decidable? Give an example of undecidable problem.** **NOV/DEC 2012** Refer notes	C504.5	BTL 1
2	**is recursively enumerable language? NOV/DEC 2012, MAY/JUNE 2012, NOV/DEC 2010, MAY/JUNE 2013, NOV/DEC 2013** s the language accepted by a Turing Machine.	C504.5	BTL 1
3	**on the difference between P and NP problems. MAY/JUNE 2012** Refer notes	C504.5	BTL 1

4	**o prove that the Post Correspondence problem is Undecidable.** **<u>NOV/DEC 2011</u>** ıce a modified PCP and reduce the same to the original PCP. reduce L_u to the modified PCP. ıain of reduction infers if original L_u is known to be undecidable then conclude that PCP is undecidable.	C504.5	BTL 1
5	**that any PSPACE-hard language is also NP-hard.** **<u>NOV/DEC 2011</u>** ve must show that the language is not in NP. This is trivial since NP is a subset of PSPACE and therefore, anything outside of PSPACE is also outside of NP. ve must show that any problem in NP can be reduced to any PSPACE-hard language. Thus, any PSPACE-hard problem is also NP-hard.	C504.5	BTL 1
6	**Rice's theorem.** **<u>APR/MAY 2010</u>** non-trivial property of the RE language is undecidable. perty is trivial if it is either empty such that it is satisfied by no language or is all RE languages, or else it is non-trivial.	C504.5	BTL 1
7	**that the collection of all Turing machines is countable.<u>APR/MAY 2010</u>** ● If for a set there is an enumerator, then the set is countable. ● Any Turing Machine can be encoded with a binary string of 0's and 1's. ● An enumeration procedure for the set of Turing Machine strings: Repeat ● Generate the next binary string of 0's and 1's in proper order ● Check if the string describes a Turing Machine If Yes: Print string on output tape If No: Ignore string	C504.5	BTL 1
8	**on the difference between decidable and undecidable problems.** **<u>NOV/DEC 2010</u>** ● Decidable Problem – Existence of an algorithm ● Undecidable Problem – No algorithm for solving it.	C504.5	BTL 1
9	**is universal turing machine<u>NOV/DEC 2013</u>** **sal Turing machine** (**UTM**) is a <u>Turing machine</u> that can simulate an arbitrary Turing machine on arbitrary input. The universal machine essentially achieves this	C504.5	BTL 1

	by reading both the description of the machine to be simulated as well as the input thereof from its own tape		
10	multiple turing machine. **NOV/DEC 2014** tended TM model has more number of tapes. A move is based on the state and on the vector of symbols scanned by the hand on each of the tapes.	C504.5	BTL 1
11	**xample for NP-complete problems. NOV/DEC 2014** Traveling Salesman Problem	C504.5	BTL 1
12	**when a problem is said to be decidable and give an example of an undecidable problem. NOV/DEC 2015**	C504.5	BTL 1
13	**is a universal Language L_u? NOV/DEC 2015** Refer notes	C504.5	BTL 1
14	is a Recursively Enumerable language said to be Recursive? **MAY/JUNE 2016**	C504.5	BTL 1
15	y whether "Tower of Hanoi" problem is tractable or intractable. Justify your answer. **MAY/JUNE 2016**	C504.5	BTL 1
16	Universal Turing Machine **NOV/DEC 2016**	C504.5	BTL 1
17	NP-hard and NP-complete problems. **NOV/DEC 2016**	C504.5	BTL 1
18	a recursively enumerable language is said to be recursive.	C504.5	BTL 1
19	are and Contrast recursive and recursively enumerable languages.	C504.5	BTL 1
20	vhen a problem is said to be decidable and give an example of an undecidable problem	C504.5	BTL 1
21	ue that the language accepted by a non deterministic Turing Machine is different from recursively enumerable language?	C504.5	BTL 1
22	wo properties of recursively enumerable sets which are undecidable.	C504.5	BTL 1
23	the classes of P and NP		

		C504.5	BTL 1
24	a language is said to be recursively enumerable?	C504.5	BTL 1
25	Time and Space Complexity of TM.	C504.5	BTL 1
26	guish between PCP and MPCP. What are the concepts used in UTMs?	C504.5	BTL 1
PART – B			
1	**If L_1 and L_2 are recursive language then $L_1 U L_2$ is a recursive language.(6)NOV/DEC 2012**	C504.5	BTL 1
2	**Prove that the halting problem is undecidable.(10) NOV/DEC 2012, NOV/DEC 2010**	C504.5	BTL 1
3	**State and prove the Post's correspondence problem. (10)NOV/DEC 2012, NOV/DEC 2010**	C504.5	BTL 1
4	**Write a note on NP problems. (6) NOV/DEC 2012**	C504.5	BTL 1
5	**Explain undecidability with respect to post correspondence problem. (8) MAY/JUNE2012**	C504.5	BTL 1
6	**and prove Post Correspondence Problem and Give example. (16) NOV/DEC 2014**	C504.5	BTL 1
7	**is Post Correspondence problem (PCP)? Explain with the help of an example. (16) MAY/JUNE 2016**	C504.5	BTL 1
8	**hat are tractable problems? Compare it with intractable problems. (10) NOV/DEC 2016** **) Outline the concept of polynomial time reductions. (6) NOV/DEC 2016**	C504.5	BTL 1
9	**Prove that for two recursive languages L_1 and L_2 their union and intersection is recursive. NOV/DEC 2013**	C504.5	BTL 1
10	**State and explain RICE theorem.** **Prove that "MPCP reduces to PCP".**	C504.5	BTL 1