FORMAL LANGUAGES AND AUTOMATA THEORY
(CS501PC)
COURSE PLANNER

I. COURSE OVERVIEW:

Formal languages and automata theory deals with the concepts of automata, formal languages, grammar, computability and decidability. The reasons to study Formal Languages and Automata Theory are Automata Theory provides a simple, elegant view of the complex machine that we call a computer. Automata Theory possesses a high degree of permanence and stability, in contrast with the ever-changing paradigms of the technology, development, and management of computer systems. Further, parts of the Automata theory have direct bearing on practice, such as Automata on circuit design, compiler design, and search algorithms; Formal Languages and Grammars on compiler design; and Complexity on cryptography and optimization problems in manufacturing, business, and management. Last, but not least, research oriented students will make good use of the Automata theory studied in this course.

II. PREREQUISITE:

- A course on “Discrete Mathematics”
- A course on “Data Structures”

III. COURSE OBJECTIVES:

1	To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
2	To introduce the fundamental concepts of formal languages, grammars and automata theory
3.	Classify machines by their power to recognize languages.
4.	Employ finite state machines to solve problems in computing.
5.	To understand deterministic and non-deterministic machines.
6.	To understand the differences between decidability and undecidability.

IV. COURSE OUTCOMES:

Course Outcomes	Description	Bloom’s Taxonomy Levels
CO1	Able to understand the concept of abstract machines and their power to recognize the languages.	L2:Understand
CO2	Able to employ finite state machines for modeling and solving computing problems.	L3:Apply
CO3	Able to design context free grammars for formal languages.	L6:Create
CO4	Able to distinguish between decidability and undecidability.	L4: Analyze
CO5	Able to gain proficiency with mathematical tools and formal methods.	L2:Understand

V. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex	**3**	Lectures, Assignments / Mid Test
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.	**3**	Lectures, Assignments / Mid Test
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.	**2**	Lectures, Assignments / Mid Test
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.	**2**	Lectures, Assignments / Mid Test
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.	**1**	Lectures, Assignments / Mid Test
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	**-**	Personality development seminar

	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.	**1**	Student Seminars
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.	**2**	Assignments / Mid Test

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) - : None

VI. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSO)		Level	Proficiency assessed by
PSO1	**Foundation of mathematical concepts:** To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.	**3**	Assainment, Mid Exam, Extrenal exam
PSO2	**Foundation of Computer System:** The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.	**2**	Assainment, Projects
PSO3	**Foundations of Software development:** The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.	**2**	Assainment, Mid Exam, Extrenal exam

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) - : None

VII. SYLLABUS:

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems. **Nondeterministic Finite Automata:** Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines.

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions. Chomsky Normal form Griebech Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications.

Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine.

UNIT - V

Types of Turing machine: Turing machines and halting.

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXT BOOKS:

T1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.

T2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.

5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

VIII. LESSON PLAN:

Sl No.	Week	Unit No.	Topics to be covered	Content to be covered under each topic	Link for Lecture handouts PDF/PPT	Link for Small Projects/ Numericals(if any)	Course Learning Outcomes	Teaching Methodology	References
1	1	1	Introduction to Automata Theory and Fundamental Concepts; Basic	Introduction ,Applications	https://drive.google.com/file/d/17BwFX-GEm8MD8ymNxjrjIOQeI5Jq9hLj/view?usp=sha	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
2		1	Basic Fundamentals /Central Concepts of Automata Theory (part-	Introduction	https://drive.google.com/file/d/1tuTuB4im8KzHHk-WA22OBDU7vZjuEMPU/view?	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
3		1	Introduction to Finite Automata (FA): Model of FA, Formal Definition of FA, Example of FA, Transition Table of FA, Transition Diagram of	Grammar Regular Expressions	https://drive.google.com/file/d/1re-y3JpR7fEnjKRC1sFYbE904t2Qbzod/view?usp=s https://drive.google.com/file/d/1NbzLEXP4302nuyIEh4DpqabjRRg-SWnu/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
4	2	1	Deterministic Finite Automata (DFA): Definition of DFA, How a	Definintion How a DFA Processes String Simpler Notations for	https://drive.google.com/file/d/1bELNdFW-Ocv4xN_L6G0QDOiFHsMh7W32/view?usp=	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
5		1	Nondeterministic Finite Automata (NFA): Introduction to NFA,	Strings- Empty Strings, Length of the string, Power of Alphabet, Concatenationo	https://drive.google.com/file/d/10uHEBB4_RFkzTBzyVkoxDY7MVv1yZ-If/view?usp=sha	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
6		1	Nondeterministic Finite Automata (NFA)- Equivalence of NFA &	. Definintion	https://drive.google.com/file/d/1qlriBD0JDHruS0ZNCDsvU_oBu5atx6WN/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2

7	3	1	Finite Automata with Epsilon Transitions (NFA-ϵ): Uses of ϵ-	How a DFA Processes String	https://drive.google.com/file/d/14Oi2nmMe922qxmXUHbFoCqtA2uqtDUJX/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
8		1	Equivalence of NFA with and without ϵ-moves (Eliminating ϵ-transitions):	Simpler Notations for DFA's	https://drive.google.com/file/d/1fy4EM7k-4XizPfizTkbbg_DB7KlJmTrO/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
9		1	*Conversion of NFA-ϵ (with epsilon) to DFA Ex-1	Extending the Transition Function to Strings	https://drive.google.com/file/d/1Ryd1fHX3Affm8zw2rLkhz_AC4GnV9y06/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
			*Conversion of NFA-ϵ (with epsilon) to DFA Ex-2	The Language of DFA	https://drive.google.com/file/d/1386wHNERcxQ2rO9CqIew7juv22juyf5M/view?usp=sharing				
			*Conversion of NFA-ϵ (with epsilon) to DFA Ex-3	Problems	https://drive.google.com/file/d/1CzTb51-h4YMlSswTVMYEw3q_xOSwwZ16/view?usp=s				
10	4	1	Introduction to Moore and Mealy machines; Conversion of Mealy to Moore machine; Conversion of Moore machine to Mealy machine	An informal view of NFA Definition Extended Transition Function Finding Strings in Text NFA for Text Search • DFA to recognize a Set Use of ε-transitions The formal Notation for an ε-NFA Epsilon-	https://drive.google.com/file/d/1WjA9Jb9ViHAhGxxMrSjPqfIdtWs_jRUa/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
MOCK TEST-1									
11	4	2	Introduction to Regular Expressions: Recursive definition of Regular	The operators of Regular Expressions Building Regular Expressions	https://drive.google.com/file/d/1J_1knNfNRA-sUq7y8OaKffjR-mAABi69/view?	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2

12		2	Conversion of Regular Expression to Finite Automata (FA):	Associativity and Commutativity Identities and Annihilators Distributive	https://drive.google.com/file/d/1gD2NxKsOch-JF5MJ07e9nbJleIaGIJGl/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
13	5	2	Conversion of Finite Automata (FA) to Regular Expression	From DFA's to Regular Expressions Converting DFA's to Regular	https://drive.google.com/file/d/15kCDXQinfSYO1qVga2fJm4faR_rVkRCA/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
14		2	Pumping Lemma for Regular Sets / Regular Languages: Introduction	Definition and Theorm of Pumping Lemma for Regular Languages	https://drive.google.com/file/d/1ZQekTPQVa-L1jj-2ZRd-Ap2UNdXJSj8w/view?usp=shari	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
15		2	Closure Properties & Decision Properties for Regular Sets / Regular	Closure of Regular Languages Under Boolean Operations Reversal	https://drive.google.com/file/d/1XssJpyQWRlbOHX6NuXlKPiJ8nTjXskrF/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
16	6	2	Equivalence between two Finite State Machines (FSMs) / Finite	Testing Equivalence of States Testing Equivalence of Regular	https://drive.google.com/file/d/1iQ_jm4g9FRtyeB6nHXZna3tYXAVotQyf/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
17		2	Minimization of Deterministic Finite Automata (DFA):	Minimization of DFA's Why the Minimization of DFA can't be beaten	https://drive.google.com/file/d/1XKAwk8zM4VjMZ-gKGyhBIxBxckpDYBom/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
		2	*Minimization of Deterministic Finite Automata (DFA) part 2:	Minimization of DFA's Why the Minimization of DFA can't be beaten		NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
	7	2	*Introduction to Regular Grammars	Regular grammar	https://drive.google.com/file/d/1H4ghAU2srIhfaWvPlR2rk-YFbazu8YOz/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2
		2	*Regular Grammars and Finite Automata (FA)	Testing Equivalence of States Testing Equivalence of Regular	https://drive.google.com/file/d/1mcT1FIETrNuGoUTURsBaPz2u1O8b1bjm/view?usp=sharing	NIL	CO1, CO2 & CO7	1, 2 & 5	T1, T2

18	7	3	Introduction to Context-free Grammars (CFG's): Definition of Parse Tree/Derivation Tree and Applications of CFGs Ambiguity	Minimization of DFA's Why the Minimization of DFA can't be beaten Problems	https://drive.google.com/file/d/1gIz85qyDrj-wBkYqcUol2kLZFqCI6tFi/view?usp=sharing https://drive.google.com/file/d/1GtOX9fxZ3fEHlojFMNbd-WGE2IGslNvI/view?usp=sharing https://drive.google.com/file/d/1_6nvQP93NiP218_I4fGcaM1ToQcVDym3/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
19		3	Context free Grammars (CFG's): Eliminating Ambiguity	Ambiguous Grammars Removing Ambiguity From Grammars	https://drive.google.com/file/d/1bIgg8cQnR5kMncXFy7US5UQOCMhSYmsw/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
Dussehra Recess (11/10/2021 to 16/10/2021): 1 WEEK									
I-MID EXAMS (08/11/2021 to 13/11/2021): 1 WEEK									
20	8	3	Introduction to Pushdown Automata (PDA): Introduction of PDA,	An Informal Example Definition of Context Free Grammars	https://drive.google.com/file/d/1Tj6s4j1QMbZHt97vUn9WWRMlRKbhKEpI/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
21		3	Pushdown Automata (PDA)- Acceptance of Languages by Final State &	Derivations Using a Grammar Leftmost and Rightmost Derivations	https://drive.google.com/file/d/1oWHfDftdUpLaUSQFJmctcop9q97jEVQE/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
22		3	Design of PDA for CFL: Example Problem-1	Constructing Parse Trees The Yield of a Parse Tree • Inference Derivations and	https://drive.google.com/file/d/1t9O3K5zU9fpXhF399qE_6GO8XIShVgc3/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
23	9	3	Design of PDA for CFL: Example Problem-2	Parsers The YACC Parser Generator Markup Languages	https://drive.google.com/file/d/1LxigB8Dk5TmqpUMdM_4QxjzI2x8F8XFD/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2

24		3	Equivalence of PDA's and CFG's: Conversion of CFG to PDA	Ambiguous Grammars Removing Ambiguity From Grammars	https://drive.google.com/file/d/1AZMn0MNeBfRCONmHF7ELwFqvLbjwJqvG/view?usp=sharin	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
25		3	Equivalence of PDA's and CFG's: Conversion of PDA to CFG	Informal Introduction The Formal Definition of Pushdown Automata	https://drive.google.com/file/d/1M8N0sSPSQ7eVD7jEM8_vEwT1-VGWEBb4/view	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
26	10	3	Deterministic Pushdown Automata (DPDA)	Acceptance by Final State Acceptance by Empty Stack From Empty Stack to Final	https://drive.google.com/file/d/1UTm5vPUCSquWIEDDKjyXpQBnTu9dypAA/view?usp=sharing	NIL	CO1, CO3, CO4 & CO7	1, 2 & 5	T1, T2
27		4	Normal Forms for CFGs: Removing useless symbols and	Eliminating Useless Symbols Computing the Generating and Reachable	https://drive.google.com/file/d/1x7C78AAf1oZeZ1p4O0DruiIHl1i0Gw2y/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
28		4	Normal Forms for CFGs: Removing Null productions	Greiback Normal Form Problems	https://drive.google.com/file/d/17lPt3ow2nhDwQslUuHUJeTGZ1vjVZXIX/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
29		4	Normal forms for CFG: Chomsky Normal Form (CNF)	The Size of Parse Trees Statement of the Pumping Lemma Applications of		NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
30	11	4	Normal forms for CFG: Reducing CFG to Chomsky Normal Form	Substitutions Applications of the Substitution Theorem Reversal Intersection	https://drive.google.com/file/d/1bUu_CiKl61I3o2gSwS5YOBdXeOnC1TIv/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
31		4	Normal forms for CFG: Chomsky Normal Form (CNF) with Example	Complexity of Converting Among CFG's and PDA's Running Time of Conversion		NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
32	12	4	Normal forms for CFG: Greibach Normal Form (GNF) with Example	Programs that Print Hello World The Hypothetical Hello World	https://drive.google.com/file/d/1oyYsY-I6Lw3FBdXvelN644CwFUiOnvPj/view?usp=sha	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2

33		4	Normal forms for CFG: Greibach Normal Form (GNF)-Solving	Storage in the State Multiple Tracks Subroutines Problems	ring	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
34		4	Pumping Lemma for Context Free Languages: Statement of pumping	Multitape Turing Machines Equivalence of One Tape and Multitape TM's	https://drive.google.com/file/d/1ESqbcsz_SygGZrAArf0q4WloWL7u1wmL/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
35		4	Closure Properties of Context-Free Languages (CFLs): Closure	Substitutions Applications of the Substitution Theorem Reversal Intersection	https://drive.google.com/file/d/15brMbfknP8srEWLlA1Qp2ksg94SQAk9B/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
36	13	4	Introduction to Turing Machine (TM): Introduction, Formal	Programs that Print Hello World The Hypothetical Hello World	https://drive.google.com/file/d/1LD0gYfAYYMLzzzvkHD2PG47_pd65MbPa/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
37		4	Turing Machine (TM)-Closure properties of Recursive and Recursively	Programs that Print Hello World The Hypothetical Hello World	https://drive.google.com/file/d/1y4zRkHz9giWoB9wMFLzB1yd3VackKujp/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
38	14	4	*Design of Turing Machine (TM) for Languages, Example	Storage in the State Multiple Tracks Subroutines Problems	https://drive.google.com/file/d/1r6e_bXzOaKyMW4bgkk8sJYvtWaA5klvL/view?usp=sharing	NIL	CO1, CO3, CO5 & CO7	1, 2 & 5	T1, T2
			Design of Turing Machine-Example-1	Storage in the State Multiple Tracks Subroutines Problems	https://drive.google.com/file/d/1U5P3v79Eoxo4yBGPnVu_7m64acif_IuL/view?usp=sharing				
			Design of Turing Machine - Example-2	Storage in the State Multiple Tracks Subroutines Problems	https://drive.google.com/file/d/143ZER6M0fr25Ki-P4Rm3GfdkFHnmWJBT/view?u				
MOCK TEST-2									
39	14	5	Types of Turing Machines, Turing Machines and Halting	Enumerating the Binary Strings Codes for Turing Machines	https://drive.google.com/file/d/10BON__9mnZaoLgJKpnXgmsWGKNVaD0e-/view?usp=shari	NIL	CO1, CO5, CO6 & CO7	1, 2 & 5	T1, T2

40		5	*Turing Machine (TM) and Computable Functions	Recursive Languages Complements of Recursive and RE languages The Universal Language	https://drive.google.com/file/d/1ANwOP8l41kwOY-CxXfeEApWFiH4JXpRS/view?usp=sharing	NIL	CO1, CO5, CO6 & CO7	1, 2 & 5	T1, T2
41	15	5	Undecidable Problems about Recursively Enumerable Languages(R	Definition The Modified PCP Completion of the Proof of PCP	https://drive.google.com/file/d/1Tg9XrLOGv-PDF-REooXs2V7rkkAMSDT/view?	NIL	CO1, CO5, CO6 & CO7	1, 2 & 5	T1, T2
42		5	Undecidability-Post's Correspondence Problem (PCP) & Linear	Problems About Programs Undecidability of Ambiguity for CFG's	https://drive.google.com/file/d/1iY0SNQnRQU6zc8huIms6RLzHIyow0UEn/view?usp=sharing	NIL	CO1, CO5, CO6 & CO7	1, 2 & 5	T1, T2
II-MID EXAMS (10/01/2022 to 18/01/2022): 1 WEEK									

*** Topics beyond Syllabus**

NPTEL Web Course:

1. **NPTEL Web Course:**
   http://nptel.ac.in/courses/106103070/
2. **NPTEL Video Course:**
   http://nptel.ac.in/courses/111103016/
   https://nptel.ac.in/courses/106106049/

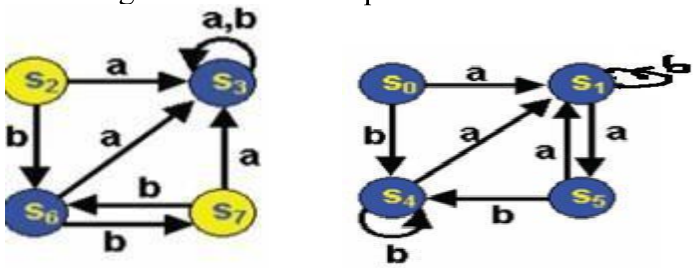
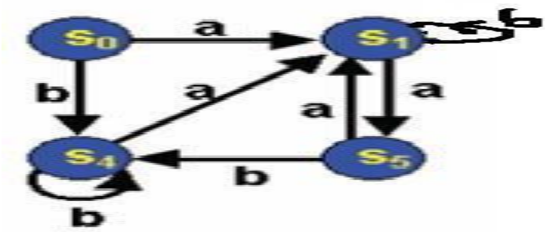
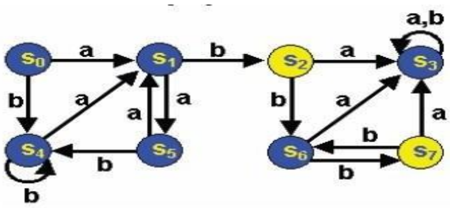
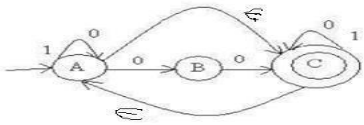
NPTEL Online Courses and Certification
https://swayam.gov.in/nd1_noc19_cs79/preview

IX. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcomes	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	1	-	-	-	-	-	-	1	-	1	2	2	2
CO2	3	3	2	2	1	-	-	-	-	1	-	2	3	2	2
CO3	3	3	3	1	1	-	-	-	-	1	-	2	3	1	1
CO4	3	3	2	2	1	-	-	-	-	1	-	2	3	2	2
CO5	3	3	2	2	1	-	-	-	-	1	-	2	3	2	2
AVG	**2.8**	**2.8**	**2**	**1.75**	**1**	**-**	**-**	**-**	**-**	**1**		**1.8**	**2.8**	**1.8**	**1.8**

X. QUESTION BANK: (JNTUH)

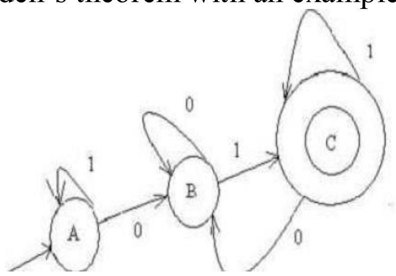
S. No.	Questions	Blooms Taxonomy Level
UNIT - I		
Short Answer Questions		
1.	**Explain** transition diagram, transition table with example.	Understand
2.	**Define** transition function of DFA.	Remember
3.	**Define** ε –transitions.	Remember
4.	**Construct** a DFA to accept even number of 0's.	Apply
5.	**Define** Kleene closure and positive closure.	Remember
6.	**Construct** a DFA to accept empty language.	Apply
7.	**Explain** power of an alphabet (∑*)?	Understand
8.	**Write** transition diagram for DFA accepting string ending with 00 defined over an alphabet ∑= {0,1}	Apply
9.	**Write** transition diagram for DFA to accept exactly one a defined over an alphabet ∑= {a,b}	Apply
10.	**Define** NFA with an example.	Remember
11.	**Explain** the different Operations on the languages.	Understand
13.	**Define** Moore Machines.	Remember
14.	**Define** Mealy Machines.	Remember
15.	**Write** DFA for odd number of 1's.	Apply
16.	**Write** NFA for (0+1)*101(0+1)*.	Apply
17.	**Write** DFA for (0+1)*10(0+1)*.	Apply
18.	**Define** ε - closure.	Remember
19.	**Write** NFA for (0+1)*001(0+1)*.	Apply
20.	**Write** DFA for (0+1)*00(0+1)*.	Apply
21	**Define** FSM and its structure with an example**.**	Remember
22	**Give** any two comparisions between NFA and DFA	Remember
Long Answer Questions		
1.	**Construct** a DFA to accept set of all strings ending with 010. Define language over an alphabet ∑ = { 0,1} and write for the above DFA .	Apply
2.	**Construct** a Moore machine to accept the following language. L = { w |w mod 3 = 0} on ∑ = { 0,1,2}	Apply
3.	**Write** any six differences between DFA and NFA	Apply
4.	**Write** NFA with Ɛ to NFA conversion with an example.	Understand
5.	**Construct** NFA for (0 + 1)*(00 + 11)(0 + 1)* and Convert to DFA.	Apply
6.	**Design** DFA for the following languages shown below ∑ = { a,b} a.L={w/ w does not contain the substring ab} b.L={w/ w contains neither the substring ab nor ba} c.L={w/ w is any string that doesn't contain exactly two a} d.L={w/ w is any string except a and b}	Apply

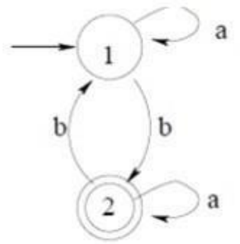
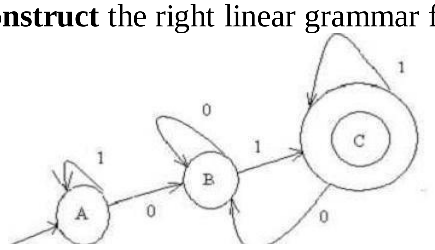
7.	**Illustrate** given 2 FA's are equivalent or not with an example. 	Apply
8.	**Construct** Mealy machine for (0 + 1)*(00 + 11) and convert to Moore machine.	Apply
9.	**Convert** NFA with Ɛ – a*b* to NFA.	Understand
10.	**Construct** NFA for (0 + 1)*101 and Convert to DFA.	Apply
11.	**Construct** a mealy machine that takes binary number as input and produces 2's complement of that number as output.Assume the string is read LSB to MSB and end carry is discarded.	Understand
12.	**Explain** with the following example the Minimize the DFA . 	Understand
13.	**Construct** a DFA, the language recognized by the Automaton being *L* □ {a^n*b/n* □ 0}. Draw the transition table.	Apply
14.	**Construct** the Minimized DFA 	Apply
15.	**Construct** the DFA that accepts/recognizes the language *L*(*M*) = | *w*□ {*a*, *b*, *c*}* and *w* contains the pattern *abac* }. Draw the transition table.	Apply
16.	**Construct** NFA for given NFA with Є-moves 	Apply
17.	**Differentiate** between DFA and NFA with an example.	Understand
18.	**Construct** a finite automaton accepting all strings over {0, 1} having even number of 0's and even number of 1's.	Apply
19.	**Construct** a Moore Machine to determine the residue mod 5 for each binary string treated as integer. Sketch the transition table.	Apply
20.	**Construct** the Moore Machine for the given Mealy machine	Understand

UNIT – II

Short Answer Questions

1.	**Define** Regular Languages.	Remember
2.	**Define** Pumping Lemma for Regular Languages.	Remember

<table>
<tr><td>3.</td><td>Write the applications of pumping lemma for regular languages.</td><td>Apply</td></tr>
<tr><td>4.</td><td>List any two applications of regular expression.</td><td>Remember</td></tr>
<tr><td>5.</td><td>Define Context Free Grammars.</td><td>Remember</td></tr>
<tr><td>6.</td><td>Define Left linear derivation.</td><td>Remember</td></tr>
<tr><td>7.</td><td>Write regular expression for denoting language containing empty string.</td><td>Apply</td></tr>
<tr><td>8.</td><td>Differentiate left linear and right linear derivations.</td><td>Understand</td></tr>
<tr><td>9.</td><td>Write the Context free grammar for palindrome.</td><td>Remember</td></tr>
<tr><td>10.</td><td>Define right linear grammars.</td><td>Remember</td></tr>
<tr><td>11.</td><td>Define Regular grammars.</td><td>Remember</td></tr>
<tr><td>12.</td><td>Write regular expressions for the Set of strings over {0, 1} whose last two symbols are the same.</td><td>Apply</td></tr>
<tr><td>13.</td><td>Define right linear derivation.</td><td>Remember</td></tr>
<tr><td>14.</td><td>Define left linear grammars.</td><td>Remember</td></tr>
<tr><td>15.</td><td>Write the regular language generated by regular expression (0+1)*001(0+1)*.</td><td>Apply</td></tr>
<tr><td>16.</td><td>Write the Regular Expression for the set of binary strings.</td><td>Apply</td></tr>
<tr><td>17.</td><td>Write the derivation of the string aaaa from CFG –
<table>
<tr><th>STATE/I</th><th>a</th><th>b</th><th>output</th></tr>
<tr><td>q0</td><td>q1</td><td>q2</td><td>1</td></tr>
<tr><td>q1</td><td>q1</td><td>q1</td><td>0</td></tr>
<tr><td>q2</td><td>q1</td><td>q0</td><td>1</td></tr>
</table>
S →a S/A   A →a</td><td>Apply</td></tr>
<tr><td>18.</td><td>Write the derivation of the string 110 from CFG –
S→A0/B A→0/12/B B →A/11</td><td>Apply</td></tr>
<tr><td>19.</td><td>Write the Regular Expression to generate atleast one b over
Σ ={a,b}</td><td>Apply</td></tr>
<tr><td>20.</td><td>Write the Context free grammar for equal number of a’s and b’s.</td><td>Apply</td></tr>
<tr><td colspan="2">Long Answer Questions</td><td></td></tr>
<tr><td>1.</td><td>Convert Regular Expression 01* + 1 to Finite Automata.</td><td>Understand</td></tr>
<tr><td>2.</td><td>Convert given Finite Automata to Regular Expression using Arden’s theorem with an example.

</td><td>Understand</td></tr>
<tr><td>3.</td><td>Construct Right linear , Left linear Regular Grammars for 01*+1.</td><td>Apply</td></tr>
<tr><td>4.</td><td>Explain Identity rules . Simplify the Regular Expression -
Є + 1*(011)*(1*(011)*)*</td><td>Understand</td></tr>
<tr><td>5.</td><td>Construct Regular grammar for the given Finite Automata.
(a+b)*ab*.</td><td>Apply</td></tr>
</table>

6.	**Construct** Leftmost Derivation. , Rightmost Derivation, Derivation Tree for the following grammar S→ aB/bA A→ a /aS /bAA B→ b / bS /aBB For the string aaabbabbba .	Apply
7.	**Explain** the properties, applications of Context Free Languages	Understand
8.	**Construct** right linear and left linear grammars for given Regular Expression.	Apply
9.	**Construct** a Transition System M accepting L(G) for a given Regular Grammar G.	Apply
10.	**Discuss** the properties of Context free Language. Explain the pumping lemma with an example.	Understand
11.	**Write** regular expressions for the given Finite Automata  	Apply
12.	**Construct** a NFA with Є equivalent to the regular expression 10 + (0+ 11)0*1	Apply
13.	**Construct** Leftmost Derivation. , Rightmost Derivation, Derivation Tree for the following grammar *G* = (*V*, *T*, *P*, *S*) with *N* = {*E*}, *S* = *E*, *T* = {*id*, +,*(,)} E→ E+E E→E* E E→ (E) E→ id Obtain id+id*id in right most derivation, left most derivation	Apply
14.	**Write** a CFG that generates equal number of a's and b's.	Apply
15.	**Convert** G = ({S},{a},{ S → aS /a},{S}) into FA	Understand
16.	**Construct** a Regular expression for the set all strings of 0's and 1's with at least two consecutive 0's	Apply
17.	**Construct** context free grammar which generates palindrome strings ∑={a,b}	Apply
18.	**Construct** equivalent NFA with ϵ for the given regular expression 0*(1(0+1))*.	Apply
19.	**Construct** the right linear grammar for the following  	Apply
20.	**Write** 12 identity rules for regular expressions	Apply

UNIT – III		
Short Answer Questions		
1.	**Define** Greibach normal form.	Remember
2.	**Define** nullable Variable.	Remember
3.	**Write** the minimized CFG for the following grammar S→ABCa | bD A→BC |b B→b | ε C→Ð| ε D→d	Remember
4.	**Convert** the grammar to CNF - S →bA/aB A→aS/a B→bS/b.	Understand
5.	**Explain** the elimination of UNIT production.	Understand
6.	**Explain** the elimination of useless symbols in productions.	Understand
7.	**Define** CNF.	Remember
8.	**Write** the minimization of CFG – A →a B → aa S → a S/A	Understand
9.	**Define** the ambiguity in CFG.	Remember
10.	**What** is the use of CNF and GNF.	
11.	**Write** the minimization of CFG - S → aS1b S1→aS1b/ε.	Understand
12.	**Write** the minimization of CFG - S → A A →aA/ ε.	Understand
13.	**Write** the minimization of CFG - A →a. S →AB / a	Understand
14.	**Write** the minimization of CFG - S→aS/A/C A →a B →aa C→aCb.	Understand
15.	**Write** the minimization of CFG - S→AbA A→Aa/ ε.	Understand
16.	**Write** the minimization of CFG - S→aSa S →bSb S→a/b/ ε.	Understand
17.	**Write** the minimization of CFG - S→A0/B A→0/12/B B →A/11.	Understand
18.	**Convert** the grammar to CNF - S→aSa/aa S→bSb/bb S →a/b.	Understand
19.	**Convert** the grammar to CNF - S→aAbB A→aA/a B→bB/a.	Understand
20.	**Define** PDA.	Remember
21.	**Define** NPDA.	Remember
22.	**Differentiate** between deterministic and nondeterministic PDA.	Understand
23.	**Define** the language of DPDA.	Remember
24.	**List** the steps to convert CFG to PDA.	Remember
25.	**Explain** – acceptance of PDF by final state.	Understand
26.	**Explain** – acceptance of PDF by empty stack.	Understand
27.	**Convert** the following PDA to CFG δ(q0,b,z0)={q0,zz0)	Apply
28.	**Convert** the following PDA to CFG (q0, b, z)=(q0,zz)	Apply
29.	**Convert** the following PDA to CFG δ(q0, ϵ ,z0)=(q0,ϵ)	Apply
30.	**Convert** the following PDA to CFG δ(q0,a,z) = (q1,z)	Apply
31.	**Convert** the following PDA to CFG δ(q1,b,z)=(q1,ϵ)	Apply
32.	**Convert** the following PDA to CFG δ(q1,a,z0)=(q0,z0)	Apply
33.	**Convert** the following PDA to CFG δ(q0,0,z0)={q0,xz0)	Apply
34.	**Convert** the following PDA to CFG δ(q0,0,x)=(q0,xx)	Apply

35.	**Convert** the following PDA to CFG δ(q0,1,x)=(q1,ϵ)	Apply
36.	**Convert** the following PDA to CFG δ(q1,1,x) = (q1,ϵ)	Apply
37.	**Convert** the following PDA to CFG δ(q1,ϵ,x)=(q1,ϵ)	Apply
38.	**Convert** the following PDA to CFG δ(q1,ϵ,z0)=(q1,ϵ)	Apply
39.	**Convert** the following PDA to CFG δ(q1,ϵ,z)=(q0,ϵ)	Apply
40.	**Convert** the following CFG to PDA S ABC | BbB	Apply
41.	**Convert** the following CFG to PDAA→aA | BaC|aaa	Apply
42.	**Convert** the following CFG to PDA B→bBb| a|D	Apply
43.	**Convert** the following CFG to PDA C→ CA|AC	Apply
44.	**Convert** the following CFG to PDA S→a S/A	Apply
Long Answer Questions		
1.	**Write** a short notes on Chomsky Normal Form and Griebach Normal Form.	Apply
2.	**Show** that the following grammar is ambiguous with respect to the string aaabbabbba. S→ aB | bA A→ aS| bAA|a B→ bS | aBB | b	Understand
3.	**Use** the following grammar : S→ ABC | BbB A→ aA |BaC|aaa B→bBb| a|D C→CA|AC D→ **ε** Eliminate **ε**-productions. Eliminate any unit productions in the resulting grammar. Eliminate any useless symbols in the resulting grammar. Convert the resulting grammar into Chomsky Normal Form	Apply
4.	**Illustrate** the construction of Griebach normal form with an example.	Apply
5.	**Show** that the following CFG ambiguous. S→ iCtS | iCtSeS | a C→b	Apply
6.	**Discuss** the Pumping lemma for Context Free Languages concept with example {$a^n b^n c^n$ where n>=0}	Understand
7.	**Write** the simplified CFG productions in S → a S1b S1 → a S1b/ Є	Apply
8.	**Convert** the following CFG into GNF. S→AA/a A → SS/b	Understand
9.	**Explain** unit production? Explain the procedure to eliminate unit production.	Understand
10.	**Explain** the procedure to eliminate ϵ-productions in grammar.	Understand
11.	**Convert** the following grammar into GNF G=({A1,A2,A3},{a,b},P,A) A1->A2A3 A2->A3A1/b A3->A1A2/a	Understand

12.	**Write** simplified CFG productions from the following grammar A->aBb/bBa B->aB/bB/ϵ	Apply
13.	**Convert** the following grammar into GNF S->ABA/AB/BA/AA/B A->aA/a B->bB/b	Understand
UNIT – IV		
Short Answer Questions		
1.	**Define** Turing Machine	Apply
2.	**Explain** the moves in Turing Machine.	Understand
3.	**Define** an Instantaneous Description of a Turing Machine.	Remember
4.	**Define** the Language of Turing Machine.	Remember
5.	**List** types of TM.	Remember
6.	**Define** Computable Functions by Turing Machines .	Remember
7.	**Write** the difference between Pushdown Automata and Turing Machine.	Apply
8.	**Explain** Church's Hypothesis.	Understand
9.	**Define** Context sensitive language.	Remember
10.	**Define** multi head Turing Machine.	Remember
11.	**Define** multi dimensional Turing Machine.	Remember
12.	**Define** multiple tapes Turing Machine.	Remember
13.	**Define** Recursive languages.	Remember
14.	**Define** Recursively enumerable languages.	Remember
15.	**Define** Two way infinite Turing Machine.	Remember
16.	**Define** Non deterministic Turing Machine.	Remember
17.	**Define** Counter machine.	Remember
18.	**Explain** the model of Turing machine.	Remember
19.	**Construct** Turing Machine for 1's complement for binary numbers.	Remember
20.	**Differentiate** Recursive languages and Recursively enumberable languages.	Remember
Long Answer Questions		
1.	**Define** a Turing Machine. With a neat diagram explain the working of a Turing Machine.	Remember
2.	**Differentiate** Turing Machine with other automata.	Apply
3.	**Construct** a Transition diagram for Turing Machine to accept the following language. L = { $0^n1^n0^n$ | n ≥1}	Apply
4.	**Construct** Transition diagram for Turing Machine that accepts the language L = {0^n1^n | n ≥1}. Give the transition diagram for the Turing Machine obtained and also show the moves made by the Turing machine for the string 000111.	Apply
5.	**Construct** a Transition diagram for Turing Machine to accept the language L= { $w\#w^R$ | w ϵ (a + b) *}	Apply
6.	**Write** short notes on Recursive and Recursively Enumerable languages.	Apply
7.	**Write** the properties of recursive and recursively enumerable languages.	Apply
8.	**Construct** a Turing Machine to accept strings formed with 0 and 1 and having substring 000.	Apply

9.	**Construct** a Turing Machine that accepts the language $L = \{1^n2^n3^n \mid n \geq 1\}$. Give the transition diagram for the Turing Machine obtained and also show the moves made by the Turing machine for the string 111222333.	Apply
10.	**Define** Linear bounded automata and explain its model?	Apply
11.	Explain the power and limitations of Turing machine.	Create
12.	Construct Transition diagram for Turing Machine - $L=\{a^nb^nc^n/n>=1\}$	Apply
13.	Construct a Transition diagram for Turing Machine to implement addition of two unary numbers(X+Y).	Apply
14.	Construct a Linear Bounded automata for a language where $L=\{a^nb^n/n>=1\}$	Apply
15.	Explain the types of Turing machines.	Apply
16.	Write briefly about the following a)Church's Hypothesis b)Counter machine	Apply
17.	Construct a Transition table for Turing Machine to accept the following language. $L = \{ 0^n1^n0^n \mid n \geq 1\}$	Apply
UNIT – V		
Short Answer Questions		
1.	**Define** Chomsky hierarchy of languages.	Knowledge
2.	**Define** Universal Turing Machine	Knowledge
3.	**Define** Context sensitive language.	Knowledge
4.	**Define** decidability.	Knowledge
5.	**Define** P problems.	Knowledge
6.	**Define** Universal Turing Machines	Knowledge
7.	**Give** examples for Undecidable Problems	Understand
8.	**Define** Turing Machine halting problem.	Knowledge
9.	**Define** Turing Reducibility	Knowledge
10.	**Define** Post's Correspondence Problem.	Knowledge
11.	**Define** Type 0 grammars .	Knowledge
12.	**Define** Type 1 grammars .	Knowledge
13.	**Define** Type 2 grammars .	Knowledge
14.	**Define** Type 3 grammars .	Knowledge
15.	**Define** NP problems.	Knowledge
16.	**Define** NP complete problems	Knowledge
17.	**Define** NP Hard problems	Knowledge
18.	**Define** undecidability problem.	Knowledge
19.	**Define** turing Reducibility.	Knowledge
20.	**List** the types of grammars.	Knowledge
Long Answer Questions		
1.	**Explain** the concept of decidable and undecidability problems about Turing Machines.	Understand
2.	**Write** briefly about Chomsky hierarchy of languages..	Apply
3.	**Explain** individually classes P and NP	Understand

<table>
<tr><td>4.</td><td>Write a shot notes on post's correspondence problemand check the following is PCP or not.
<table><tr><td>I</td><td>A</td><td>B</td></tr><tr><td>1</td><td>11</td><td>111</td></tr><tr><td>2</td><td>100</td><td>001</td></tr><tr><td>3</td><td>111</td><td>11</td></tr></table></td><td>Apply</td></tr>
<tr><td>5.</td><td>Explain the Halting problem and Turing Reducibility.</td><td>Understand</td></tr>
<tr><td>6.</td><td>Write a short notes on universal Turing machine.</td><td>Apply</td></tr>
<tr><td>7.</td><td>Write a short notes on Chomsky hierarchy.</td><td>Apply</td></tr>
<tr><td>8.</td><td>Write a short notes on Context sensitive language and linear bounded
automata.</td><td>Apply</td></tr>
<tr><td>9.</td><td>Write a short note on NP complete</td><td>Apply</td></tr>
<tr><td>10.</td><td>Write a short note on NP hard problems.</td><td>Apply</td></tr>
<tr><td>11.</td><td>Write a shot notes on post's correspondence problem
and check the following is PCP or not.
<table><tr><td>I</td><td>A</td><td>B</td></tr><tr><td>1</td><td>100</td><td>1</td></tr><tr><td>2</td><td>0</td><td>100</td></tr><tr><td>3</td><td>1</td><td>0</td></tr></table></td><td>Apply</td></tr>
<tr><td>12.</td><td>Write a shot notes on post's correspondence problem
and check the following is PCP or not.
<table><tr><td>I</td><td>A</td><td>B</td></tr><tr><td>1</td><td>00</td><td>0</td></tr><tr><td>2</td><td>001</td><td>11</td></tr><tr><td>3</td><td>1000</td><td>011</td></tr></table></td><td>Apply</td></tr>
</table>

XI. OBJECTIVE QUESTIONS:

UNIT –I

Multiple Choice Questions

1. The prefix of abc is _ _ _ _ _

a. c b. b c. bc d.**a**

2.Which of the following is not a prefix of abc?

a.e b. a c. ab d. **bc**

3.Which of the following is not a suffix of abc ?

a.e b.c c.bc d.**ab**

4.Which of the following is not a proper prefix of doghouse ?

a.dog b.d c.do d.**doghouse**

5.If then the number of possible strings of length 'n' is

a.n b.n * n c.n n d.**2 n**

Fill in the Blanks

1. **<u>Language</u>** is a set of strings.
2. **<u>String</u>** is a finite sequence of symbols.
3. The basic limitation of FSM is that **<u>it can't remember arbitrary large amount of information</u>**
4. Application of Finite automata is **<u>Lexical analyzer</u>**

5. An FSM can be used to add two given integers .This is **false**

UNIT –II

Muiltile Choice Questions

1. In case of regular sets the question ' is the intersection of two languages a language of the same type ?' is _ _ _ _
   a. Decidable b. Un decidable c. **trivially decidable** d. Can't say
2. In case of regular sets the question ' is L1 n L2 = F ? ' is _ _ _
   a.**Decidable** b.Undecidable c.trivially decidable d.Can't say
3. Let r and s are regular expressions denoting the languages R and S. Then (r + s) denotes _ _
   a.RS b.R* c.**RUS** d.R+
4. Let r, s, t are regular expressions. (r*)* = _ _
   a.r b.**r*** c.F d.can't say
5. Let r, s, t are regular expressions. r(s+ t) = _ _ _ _
   a.r s b.r t c.rs - r t d.**rs +r t**

Fill in the Blanks

1. Let r, s, t are regular expressions. (r + s) t = **r t +st**
2. In NFA for r=e the minimum number of states are**1**
3. (e + 00)* =**(00)***
4. 1 + 01 =**(e + 0) 1**
5. 'The regular sets are closed under union' is **true**

UNIT –III

Multiple Choice Questions

1. Regular grammars also known as _ _ _ _ _ _ _ _ _ _ _ _ _ grammar
   a.Type 0 b.Type 1 c.Type 2 d.**Type3**
2. _ _ _ _ _ grammar is also known as Type 3 grammar.
   a.un restricted b.context free c.context sensitive d.**regular grammar**
3. Which of the following is related to regular grammar ?
   a.right linear b.left linear c.**Right linear & left linear** d.CFG
4. Regular grammar is a subset of _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ grammar.
   a.Type 0 . b.Type 1 c.Type 2 d.Type 0,1 &2
5. Let L1 =(a+b) * a L2 =b*(a+b), L1 intersection L2 = _ _ _ _ _ _ _ _ _ _ _
   a.(a+b) * ab b.ab (a+b) * c.a (a+b) * b d.**b(a+b)*a**

Fill in the Blanks

1. Let A = {0,1 } L= A * Let R = { 0 n1n , n >0 } then LUR **regular**
2. Pumping lemma is generally used for **proving a given grammar is not regular**
3. The logic of pumping lemma is a good example of **the pigeon hole principle**
4. In CFG each production is of the form Where A is a variable and is string of Symbols from ***(VUT)** (V, T are variables and terminals)
5. CFG is not closed under **complementation**

UNIT –IV

Multiple Choice Questions

1. Turing machine can be used to
   a.Accept languages b.Compute functions c.**a & b** d.none
2. Any turing machine is more powerful than FSM because____
   a.Tape movement is confined to one direction
   b.It has no finite state control

c.**It has the capability to remember arbitrary long input symbols**
d.TM is not powerful than FSM

3. In which of the following the head movement is in both directions
a.TM b.FSM c.LBA d.**a& c**

4.A turing machine is
a.**Recursively enumerable language** b.RL c.CFL d.CSL

5. Any Turning machine with m symbols and n states can be simulated by another TM with just 2 s symbols and less than
a.8mn states b.4mn+8states c.8mn+ 4 states d.**mn states**

Fill in the Blanks

1. The format: A->aB refers to **Greibach Normal Form**
2. **Greibach Normal Form** does not have left recursions.
3. Every grammar in Chomsky Normal Form is **context free**
4. Let G be a grammar. When the production in G satisfy certain restrictions, then G is said to be in **normal form**
5. Let G be a grammar: S->AB|e, A->a, B->b, Is the given grammar in CNF(True/False) **True.**

UNIT –V

Multiple Choice Questions

1.PCP having no solution is called
a. undecidability of PCP b.**decidability of PCP** c.Semi-decidability of PCP d None

2. Which of the following is type- 2 grammar?
a.A→ α where A is terminal b.**A→ α where A is Variable**c.Both d.None

3. A recursive language is also called
a) **Decidable** b) Undecidable c) Both (a) and (b) d) None of these

4. The complement of recursive language is
a) **Also recursive** b) Regular c) Both (a) and (b) d) None of these

5. Recursively enumerable language are closed under
a) Concatenation b) Intersection c) Union d) **All of these**

Fill in the Blanks

1. Recursive languages are **Accepted by turing machine**
2. **Halting problem & Boolean Satisfiability** problem are unsolvable?
3. The value of n if turing machine is defined using n-tuples: **7**
4. If d is not defined on the current state and the current tape symbol, then the machine **halts**
5. A language L is said to be **decidable** if there is a turing machine M such that L(M)=L and M halts at every point.

XII WEBSITES:

1. www.ieee.org
2. www.acm.org/dl
3. www.cs.vu.nl
4. www.cs.unm.edu
5. www.people.westminstercolleg.edu
6. http://nptel.ac.in/courses/106103070/(webcourse)
7. http://nptel.ac.in/courses/106106049/(VideoLectures)
8. http://nptel.ac.in/courses/106104028/(VideoLectures)

XIII EXPERT DETAILS:

1. Dr.Dr. DigantaGoswami, IIT Guwahati
2. Prof.S omenathBiswas, IIT Kanpur

XIV JOURNALS:

1. IEEE transactions on Computer Science
2. IEEE transactions on Fuzzy Systems
3. IEEE transactions on Neural Networks
4. IEEE Computer magazine
5. IEEE transaction in software engineering

XV LIST OF TOPICS FOR STUDENT SEMINARS:

1. Languages of context free grammars
2. Finite automata over free groups
3. OntheRegularityoflanguagesgeneratedbycontextfreeevolutionarygrammars
4. Computer studies of Turing machine problems

XVI CASE STUDIES / SMALL PROJECTS

1. Church's Hypothesis
2. P and NP problems
3. NP complete and NP hard problems
4. Universal Turing machine
5. Counter machines