TOC Course Title: Theory of Computation **Credit: 3**
Course No: CSIT.226 **Number of period per week: 3+3**
Nature of the Course: Theory + Lab **Total hours: 45+45**
Year: Second, Semester: Fourth
Level: B. Sc. CSIT

1. Course Introduction

This course presents a study of Finite State Machines and their languages. It covers the details of finite state automata, regular expressions, context free grammars. More, the course includes design of the Push-down automata and Turing Machines. The course also includes basics of undecidabilty and intractability.

2. Objectives

The main objective of the course is to introduce concepts of the models of computation and formal language approach to computation. The general objectives are to,

- introduce concepts in automata theory and theory of computation
- design different finite state machines
- design grammars and recognizers for different formal languages
- identify different formal language classes and their relationships
- determine the decidability and intractability of computational problems

3. Specific Objectives and Contents

Specific Objectives	Contents
• Revision of mathematical foundations for computation. • Understand the basic notations of symbols and their closures • Understand basic operations on strings and to know about language	**Unit I: Basic Foundations (3 Hrs)** 1.1. Review of Set Theory, Logic, Functions, Proofs 2.1. Automata, Computability and Complexity: Complexity Theory, Computability Theory, Automata Theory 1.2. Basic concepts of Automata Theory: Alphabets, Power of Alphabet, Kleen Closure Alphabet, Positive Closure of Alphabet, Strings, Empty String, Suffix, Prefix and Substring of a string, Concatenation of strings, Languages, Empty Language, Membership in Language.
• Understand basics of automata theory • Design DFA, NFA and ε-NFA for various languages.	**Unit II: Introduction to Finite Automata (8 Hrs)** 2.2. Introduction to Finite Automata, Introduction of Finite State Machine 2.3. Deterministic Finite Automata (DFA), Notations for DFA, Language of DFA, Extended Transition Function

• Reduce NFA to DFA and ε - NFA to NFA & DFA in different ways • Understand the Moore and Mealy Machines	of DFA 2.4. Non-Deterministic Finite Automaton (NFA), Notations for NFA, Language of NFA, Extended Transition Function of NFA 2.5. Equivalence of DFA and NFA, Subset-Construction Method for reduction of NFA to DFA, Theorems for equivalence of Language accepted by DFA and NFA 2.6. Finite Automaton with Epsilon Transition (ε - NFA), Notations for ε - NFA, Epsilon Closure of a State, Extended Transition Function of ε – NFA, Removing Epsilon Transition using the concept of Epsilon Closure, Equivalence of NFA and ε –NFA, Equivalence of DFA and ε – NFA 2.7. Finite State Machines with output: Moore machine and Mealy Machines
• Understand concepts of Regular Expressions • Write regular expressions for regular languages over various alphabet set • Construct regular expressions from finite state machines and vice versa • Understand use of pumping lemma for proving regular languages • Minimize Finite State Machine	**Unit III: Regular Expressions (6)** 3.1. Regular Expressions, Regular Operators, Regular Languages and their applications, Algebraic Rules for Regular Expressions, 3.2. Equivalence of Regular Expression and Finite Automata, Reduction of Regular Expression to ε – NFA, Conversion of DFA to Regular Expression, 3.3. Properties of Regular Languages, Pumping Lemma, Application of Pumping Lemma, Closure Properties of Regular Languages over (Union, Intersection , Complement) 3.4. Minimization of Finite State Machines: Table Filling Algorithm
• Understand structure and components of grammars. • Design CFG for various languages • Simplify the CFG • Understand and use different normal forms of CFG • Understand concepts of Chomsky Hierarchy, Context Sensitive Grammars, Unrestricted Grammars • Understand use of pumping lemma for proving context free languages.	**Unit IV: Context Free Grammar (9)** 4.1. Introduction to Context Free Grammar (CFG), Components of CFG, Use of CFG, Context Free Language (CFL) 4.2. Types of derivations: Bottomup and Topdown approach, Leftmost and Rightmost, Language of a grammar 4.3. Parse tree and its construction, Ambiguous grammar, Use of parse tree to show ambiguity in grammar 4.4. Regular Grammars: Right Linear and Left Linear, Equivalence of regular grammar and finite automata 4.5. Simplification of CFG: Removal of Useless symbols, Nullable Symbols, and Unit Productions, Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Backus-Naur Form (BNF) 4.6. Context Sensitive Grammar, Chomsky Hierarchy 4.7. Pumping Lemma for CFL, Application of Pumping

	Lemma, Closure Properties of CFL
• Understand basics of PDA • Design PDA with empty stack or final state for various CFG • Understand difference between Deterministic and Non-deterministic PDA • Reduce CFG to PDA and vice-versa	**Unit V: Push Down Automata (7 Hrs)** 5.1. Introduction to Push Down Automata (PDA), Representation of PDA, Operations of PDA, Move of a PDA, Instantaneous Description for PDA, 5.2. Deterministic PDA, Non Deterministic PDA, Acceptance of strings by PDA, Language of PDA, 5.3. Construction of PDA by Final State , Construction of PDA by Empty Stack, Conversion of PDA by Final State to PDA accepting by Empty Stack and vice-versa, 5.4. Conversion of CFG to PDA, Conversion of PDA to CFG
• Understand basics of Turing Machine and its relationship to computers • Design and trace Turing Machine for various languages • Explore the use of Turing Machine in different roles • Encode a general Turing Machine using Universal Turing Machine and encoding Technique	**Unit VI: Turing Machines (10 Hrs)** 6.1. Introduction to Turing Machines (TM), Notations of Turing Machine, Language of a Turing Machine, Instantaneous Description for Turing Machine, Acceptance of a string by a Turing Machines 6.2. Turing Machine as a Language Recognizer, Turing Machine as a Computing Function, Turing Machine with Storage in its State, Turing Machine as a enumerator of stings of a language, Turing Machine as Subroutine 6.3. Turing Machine with Multiple Tracks, Turing Machine with Multiple Tapes, Equivalence of Multitape-TM and Multitrack-TM, Non-Deterministic Turing Machines, Restricted Turing Machines: With Semi-infinite Tape, Multistack Machines, Counter Machines 6.4. Curch Turing Thesis, Universal Turing Machine, Turing Machine and Computers, 6.5. Encoding of Turing Machine, Enumerating Binary Strings, Codes of Turing Machine, Universal Turing Machine for encoding of Turing Machine
• Understand computational complexity and it is classes • Understand concepts of Np-Complete Problems • Explore a family of undecidable problems	**Unit VII: Undecidability and Intractability (5 Hrs)** 7.1. Computational Complexity, Time and Space complexity of A Turing Machine, Intractability 7.2. Complexity Classes, Problem and its types: Absract, Decision, Optimization 7.3. Reducibility, Turing Reducible, Circuit Satisfiability, Cooks Theorem, 7.4. Undecidability, Undecidable Problems: Post's Correspondence Problem, Halting Problem and its proof, Undecidable Problem about Turing Machines

Evaluation System

Undergraduate Programs							
External Evaluation	**Marks**	**Internal Evaluation**	**Weightage**	**Marks**	**Practical**	**Weightage**	**Mark**
End semester examination	60	Assignments	20%	20	Practical Report copy	25%	20
(Details are given in the separate table at the end)		Quizzes	10%		Viva	25%	
		Attendance	20%		Practical Exam	50%	
		Internal Exams	50%				
Total External	60	Total Internal	100%	20		100%	20
Full Marks 60+20+20 = 100							

External evaluation

1. **End semester examination:**
   It is a written examination at the end of the semester. The questions will be asked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

2. **External Practical Evaluation:**
   After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the above mentioned evaluation. There will be an internal examiner to assist the external examiner. Three hours time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Full Marks: 100, Pass Marks: 45, Time: 3 Hrs

Nature of question	Total questions to be asked	Total questions to be answered	Total marks	Weightage
Group A: multiple choice*	20	20	20×1 = 20	60%
Group B: Short answer type questions	7	6	6×8 = 48	60%
Group C: Long answer type questions	3	2	2×16 =32	60%
			100	100%

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failed student will not be eligible to appear in the end semester examinations.

Internal evaluation

Assignment: Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

Quizzes: Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

Attendance in class: Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

Presentation: Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

Mid-term examination: It is a written examination and the questions will be asked covering all the topics in the session of the course.

Discussion and participation: Students will be evaluated on the basis of their active participation in the classroom discussions.

Instructional Techniques: All topics are discussed with emphasis on real-world application. List of instructional techniques is as follows:

- Lecture and Discussion
- Group work and Individual work
- Assignments
- Presentation by Students
- Quizzes
- Guest Lecture

Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during that period. If a student fails to attend a formal exam/quiz/test, there won't be any provision for re-exam. Unless and until the student clears one semester he/she will not be allowed to study in the following semesters.

Laboratory Work

Student should write programs and prepare lab sheet for most of the units in the syllabus. Majorly, students should practice design and implementation of Finite State Machines viz. DFA, NFA, PDA, and Turing Machine. Students are highly recommended to construct Tokenizers/

Lexers over/for some language. Students are advised to use regex, Perl, C++, Java for using regular expressions. However, nature of programming can be decided by the instructor. The lab work should be practiced for minimum of 3 lab hours per week.

Prescribed Text

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, **Introduction to Automata Theory, Languages, and Computation,** 3rd Edition, Pearson - Addison-Wesley.

References

1. Harry R. Lewis and Christos H. Papadimitriou, **Elements of the Theory of Computation**, 2nd Edition, Prentice Hall.

2. Michael Sipser, **Introduction to the Theory of Computation,** 3rd Edition, Thomson Course Technology

3. Efim Kinber, Carl Smith, **Theory of Computing: A Gentle introduction,** Prentice- Hall.

4. John Martin, **Introduction to Languages and the Theory of Computation,** 3rd Edition, Tata McGraw Hill.

5. Kenneth H. Rosen, **Discrete Mathematics and its Applications to Computers Science,** WCB/Mc-Graw Hill.