Course Title: Theory of Computation **Full Marks: 60+20+20**
Course No: CSC 257 **Pass Marks: 24+8+8**
Nature of the Course: Theory + Lab **Credit Hours: 3**
Year: Second, Semester: Fourth

Course Description: 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.

Course 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 and grammars and recognizers for different formal languages, identify different formal language classes and their relationships, and determine the decidability and intractability of computational problems.

Detail Syllabus

Chapters / Units	Teaching Methodology	Teaching Hours
Unit I: Basic Foundations 1.1. Review of Set Theory, Logic, Functions, Proofs 1.2. Automata, Computability and Complexity: Complexity Theory, Computability Theory, Automata Theory 1.3. 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	**Class Lecture**	**3 Hours**
Unit II: Introduction to Finite Automata 2.1. Introduction to Finite Automata, Introduction of Finite State Machine 2.2. Deterministic Finite Automata (DFA), Notations for DFA, Language of DFA, Extended Transition Function of DFA Non-Deterministic Finite Automaton (NFA), Notations for NFA, Language of NFA, Extended Transition 2.3. Equivalence of DFA and NFA, Subset-Construction	**Class Lecture** + **Lab Session**	**8 Hours**

2.4. Method for reduction of NFA to DFA, Theorems for equivalence of Language accepted by DFA and NFA: For any NFA, N = (Q_N, $\sum$, δ_N, q_0, F_N) accepting language L $\subseteq$ $\sum$* there is a DFA D = (Q_D, $\sum$, δ_D, q_0',F_D) that also accepts L i.e. L (N) = L (D), A language L is accepted by some NFA if L is accepted by some DFA. 2.5. 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 2.6. Finite State Machines with output: Moore Machine and Mealy Machines, Illustration of the Moore and Mealy Machines		
Unit III: Regular Expressions 3.1. Regular Expressions, Operators of Regular Expressions (Union, Concatenation, Kleen), 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, Arden's Theorem 3.3. Properties of Regular Languages, Pumping Lemma for regular expression, Application of Pumping Lemma, Closure Properties of Regular Languages over (Union, Intersection , Complement), Minimization of Finite State Machines: Table Filling Algorithm	**Class Lecture** **+** **Lab Session**	**6 Hours**
Unit IV: Context Free Grammar 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, Sentential Form (Left, Right), Language of a grammar 4.3. Parse tree and its construction, Ambiguous	**Class Lecture** **+** **Lab Session**	**9 hours**

grammar, Use of parse tree to show ambiguity in grammar, Inherent Ambiguity 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(Type 0, 1, 2, 3) , Pumping Lemma for CFL, Application of Pumping Lemma, Closure Properties of CFL		
Unit V: Push Down Automata 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, Conversion of CFG to PDA, Conversion of PDA to CFG	**Class Lecture** + **Lab Session**	**7 Hours**
Unit VI: Turing Machines 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	**Class Lecture** + **Lab Session**	**10 Hours**

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, Encoding of Turing Machine, Enumerating Binary Strings, Codes of Turing Machine, Universal Turing Machine for encoding of Turing Machine		
Unit VII: Undecidability and Intractability 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	**Class Lecture** **+** **Lab Session**	**5 Hours**

Text Books

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

Reference Books

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.

Laboratory Work Manual

Student should write programs and prepare lab sheets 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/ Lexical analyzer over/for some language. The nature of programming can be decided by the instructor and students as per their comfort. The instructors have to prepare lab sheets for individual unit covering the concept of all units as per the requirement. The sample lab sessions can be as following descriptions;

Unit I: Basic Foundations (5 Hrs)

- Write programs for illustrating the concepts of Strings, Prefix, Suffix and Substring of a String.

Unit II & III: Introduction to Finite Automata and Regular Expressions (14 Hrs)

- Write programs for illustrating the concepts of
  - Determinstic Finite Automata
  - Non-Deterministic Finite Automata
- Write programs for implementing Tokenizers like for valid C-identifiers, Keywords, e-mail validators, phone number etc.
- Write programs that implement NFA for text search.
- Write programs for implementing regular expressions.

Unit IV & V: Context Free Grammar and Push Down Automata (14 Hrs)

- Write Program for simulation of Leftmost/Rightmost Derivations.
- Write Program for Parse Tree Contruction.
- Write programs for illustrating the concepts of context free grammar and its accptance using the concepts of Push Down Automata
  - Acceptance by Final State
  - Acceptance by Empty Stack

Unit VI: Turing Machines (12 Hrs)

- Write programs for illustrating the concepts of Turing Machine as a Language Recognizer.

Model Question
Tribhuvan University
Institute of Science and Technology

Course Title: Theory of Computation | **Full Marks:** 60
Course No: CSC257 | **Pass Marks:** 24
Level: B. Sc CSIT Second Year/ Fourth Semester | **Time: 3 Hrs**

Section A
Long Answer Questions

Attempt any Two questions. *[2*10=20]*

1. Define the extended transition function of DFA. Draw a DFA accepting language L= $\{1^n \mid n=2,3,4\ldots\ldots\}$. Show acceptance of strings 1110011 and 1110 using extended transition function. [2+4+4]
2. What is deterministic pushdown automaton? Configure a pushdown automaton accepting the language, L= $\{wCw^R \mid w \in (a,b)^*\}$. Show instantaneous description of strings abbCbba and baCba. [2+4+4]
3. How a Turing Machine works? Construct a Turing Machine accepting the language, L= $\{ (^n)^n \}$. Also show the transition diagram of the machine. Illustrate whether a string (( )) is accepted by the Turing Machine or not. [2+6+2]

Section B
Short Answer Questions

Attempt any Eight questions. *[8*5=40]*

4. When a grammar is said to be in CNF? Convert following grammar to CNF; [1+4]
   $S \rightarrow 1A \mid 0B \mid \epsilon$
   $A \rightarrow 1AA \mid 0S \mid 0$
   $B \rightarrow 0BB \mid 1 \mid A$
   $C \rightarrow CA \mid CS$
5. Define epsilon NFA. Configure equivalent epsilon NFA for the regular expression (ab U a)*. [1+4]
6. Differentiate Kleen Closure from Positive Closure. For $\sum = \{0,1\}$, compute $\sum^*$ and $\sum^2$. [3+2]
7. Write the regular expression over {0, 1} for strings [2.5+2.5]
   a. not ending with 0.
   b. of length at least 3 that ends with 00.
8. What is undecidable problem? Define Post's Correspondence Problem with an example. [1+4]
9. How pumping lemma can be used to prove that any language is not a regular language? Show that language, $L=\{0^r 1^r | n \geq 0\}$ is not a regular language. [4+1]
10. Discuss how Turing Machine with multiple tracks differs from a Turing Machine with multiple tapes. [5]
11. How context free grammars are defined? Write a context free grammar over {0,1}, where the strings start and end with the same symbol. [2+3]
12. What is halting problem? How can you argue that halting problem is undecidable? [1+4]