2/4 B.Tech - SECOND SEMESTER

IT4T4 **AUTOMATA AND COMPILER DESIGN** **Credits: 3**
Lecture: 3 Periods/week **Internal assessment: 30 marks**
Practice/Interaction: 1Period/week **Semester end examination: 70 marks**

Objectives:

- To get familiar with regular expressions to describe a language using automata.
- Usage of context free grammars to describe the syntax of a language.
- To learn different parsing techniques.
- To provide techniques for syntactic, semantic language analysis, intermediate code Generation and optimization.

Outcomes:

Students will be able to

- Read and write finite automata and grammars for programming language constructs.
- Understand the functionality of parsing mechanisms.
- Construct syntax trees and generate intermediate code.
- Understand the concepts of storage administration for different programming environments.
- Understand the concepts of optimization and generate the machine code.

SYLLABUS

UNIT - I

Formal Language And Regular Expressions : Languages, Operations On Languages, Regular Expressions, Identity Rules For Regular Expressions, Finite Automata – DFA, NFA, Conversion Of Regular Expression to NFA, NFA To DFA. Introduction to Compilers: Phases of the Compiler.

UNIT- II

Syntax Analysis: Context Free Grammars, Top-Down Parsing, Recursive Descent Parsers: LL (K) Parsers. Bottom-Up Parsing: Shift Reduces Parser, LR Parsers: SLR, CLR, LALR.

UNIT- III

Syntax Directed Translation: Syntax Directed Definition, Construction of Syntax Trees, L-Attributed Definitions. Intermediate Code Generation: Intermediate Languages, Translation of Assignment Statements and Boolean Expressions.

UNIT- IV

Type Checking: Specification of Simple Type Checker, Equivalence of Type Expressions, Type Conversions Runtime Environments: Storage Organization, Storage Allocation Strategies, Access to Non Local Names, Parameter Passing, Symbol Table, Dynamics Storage Allocation Techniques.

UNIT- V

Code Optimization: Principal Sources Of Optimization, Optimization Of Basic Blocks, Loops In Flow Graphs, Global Data Flow Analysis, Peephole Optimization.

Code Generation: Issues in Design of Code Generator, Simple Code Generator, Register Allocation and Assignment, DAG Representation of Basic Block, Generating Code from DAGs.

Text Books:

1. Compilers Principles, Techniques and Tools, Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, PEA.
2. Introduction to Automata Theory Languages & Computation, 3rd Edition, Hopcroft, Ullman, PEA

Reference Books:

1. Theory of Computer Science, Automata Languages and Computation, 2nd Edition, Mishra, Chandra Shekaran, PHI.
2. Elements of Compiler Design, A.Meduna, Auerbach Publications, Taylor and Francis Group.

e-Learning Resources:

1. http://www.Practice/Interactionspoint.com/compiler_design/compiler_design_finite_automata.htm
2. nptel.ac.in/courses/106108113/
3. nptel.ac.in/courses/106108113/11