

3/4 B.Tech. SECOND SEMESTER

IT6T5

AUTOMATA AND COMPILER DESIGN

Credits: 4

Lecture: 4 periods/week

Internal assessment: 30 marks

Tutorial: 1 period /week

Semester end examination: 70 marks

Objectives:

To get familiar with

- Regular expressions to describe a language using automata.
- The usage of context free grammars to describe the syntax of a language.
- Learning different parsing techniques
- The basic principles of language interpretation.
- How to provide techniques for syntactic, semantic language analysis, intermediate code generation and optimization.

Outcomes:

Student will be able to

- Develop non deterministic and deterministic finite-state automata
- Understand the phases of programming language compilers
- Perform lexical analysis using regular grammars
- Construct parse trees during syntax analysis.
- Interpret top-down and bottom-up parsing techniques
- Understand the process of syntax-directed translation schemes.
- Generate and optimize the machine code.
- Analyze the results from language compilation.

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.

UNIT III

BOTTOM-UP PARSING:

Shift Reduces Parser, LR Parsers: SLR, CLR, LALR, Using Ambiguous Parsers.

UNIT IV

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 V

TYPE CHECKING:

Specification of Simple Type Checker, Equivalence of Type Expressions, Type Conversions, Overloading Of Functions and Operators.

UNIT VI

RUNTIME ENVIRONMENTS:

Storage Organization, Storage Allocation Strategies, Access to Non Local Names, Parameter Passing, Symbol Table, Dynamics Storage Allocation Techniques.

UNIT VII

CODE OPTIMIZATION:

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

UNIT VIII

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

Reference Books:

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