

II Year – II SEMESTER

T P
3+1 0

LANGUAGE PROCESSORS (50%FLAT+50%CD)

Objectives: Describes how a programming language works, how input is converted into output from the machine hardware level and various phases of compiler

UNIT I:

Objectives: Delineation of various components of formal languages and grammars, regular expressions and equivalence of finite automata and regular expressions.

Formal Language and Regular Expressions:

Languages, operations on languages, regular expressions (re), languages associated with (re), operations on (re), Identity rules for (re), Finite Automata: DFA, NFA, Conversion of regular expression to NFA, NFA to DFA. Applications of Finite Automata to lexical analysis

UNIT II:

Objectives: Illustration of grammars and their role in compilers and various parsing techniques

Context Free grammars and parsing:

Context free Grammars, Leftmost Derivations, Rightmost Derivations, Parse Trees, Ambiguity Grammars, Top-Down Parsing, Recursive Descent Parsers: LL(1) Parsers.

Rightmost Parsers: Shift Reduce Parser, LR (0) Parser, SLR (1) Parser, LR (1) & LALR (1) Parsers, Ambiguous Grammars

UNIT III:

Objectives: Description of Syntax trees, its variants, language classifications

Syntax Directed Translation:

Definitions, construction of Syntax Trees, S-attributed and L-attributed grammars, Intermediate code generation, abstract syntax tree, translation of simple statements and control flow statements.

Semantic Analysis:

Semantic Errors, Chomsky hierarchy of languages and recognizers, Type checking, type conversions, equivalence of type expressions.

UNIT IV:

Objectives: Focus on various storage allocation schemes

Storage Organization:

Storage language Issues, Storage Allocation, Storage Allocation Strategies, Scope, Access to Nonlocal Names, Parameter Passing, Dynamics Storage Allocation Techniques.

UNIT V:

Objectives: Enforces various schemes for optimizing code

Code Optimization:

Issues in the design of code optimization, Principal sources of optimization, optimization of basic blocks, Loop optimization, peephole optimization

UNIT VI:

Objectives: Describes the role of code generator and its design issues

Code Generation:

Issues in the design of code Generation, Machine Dependent Code Generation, object code forms, Register allocation and assignment, DAG representation of basic Blocks, Generating code from DAGs.

Text Books:

1. A Text Book on Automata Theory, Nasir S.F.B, P.K. Srimani, Cambridge university Press
2. Introduction to Automata Theory, Formal languages and computation, Shamalendu kandar,

3. Compilers Principles, Techniques and Tools, Aho, Ullman, Ravi Sethi, PEA
4. Introduction to theory of computation, 2nd ed, Michel sipser, CENGAGE
5. Principles of Compiler Design, A.V. Aho . J.D.Ullman; PEA

Reference Books:

1. Theory of Computer Science, Automata languages and computation , 2/e, Mishra, Chandra Shekaran, PHI

2. Theory of Computation , aproblem solving approach, kavi
Mahesh, Wiley