RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

Credit Based Grading System

Information Technology, VI-Semester

IT-6004 Compiler Design

UNIT-I: Introduction to Compiler, analysis of source program, phases and passes, Bootstrapping, lexical analyzers, data structures in compilation – LEX: lexical analyzer generator, Input buffering, Specification and Recognition of tokens, YACC, The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

UNIT-II: Syntax Analysis: working of Parser, Top down parsing, Bottom-up parsing, Operator precedence parsing, predictive parsers, LR parsers (SLR, Canonical LR, LALR), constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator.

UNIT-III: Syntax Directed Translation: Definitions, Inherited Attributes, L-attributed definitions, S-attributed definitions, Dependency graph, Construction of syntax trees, Top down translation, postfix notation, bottom up evaluation.

UNIT-IV: Intermediate Code Generation: Three address code, quadruple & triples, translation of assignment statements, Boolean expression and control structures, Backpatching, Run Time Memory Management: Static and Dynamic storage allocation, stack based memory allocation schemes, Symbol Table management.

UNIT-V: Code Optimization and Generation: organization of code optimizer, basic blocks and flow graphs, DAG representation of basic blocks, loops in flow graph, peephole optimization, Basic of block optimization.

Reference Books:

1. A. V. Aho, R. Sethi & J. D. Ullman, Compilers: Principles, Techniques & Tools, Pearson Ed.
2. Alfred V. Aho, Jeffrey D. Ullman, Principles of Compiler Design, Narosa Publishing House.
3. Ronald Mak, Writing Compilers and Interpreters, Wiley India Edition.
4. Louden, Compiler Construction, Cengage learning.

List of Experiments:

1. Study of LEX tool and its installation in LINUX.
2. Write a program to identify string using lex tool.
3. Write a program to find keyword and identifiers using lex tool.
4. Write a program to count all character using lex tool.
5. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines.
6. Write a program to calculate First and Follow of a Grammar.
7. Write a program to construct LL(1) parser for given language.
8. Write a program to construct an operator precedence parser for a given language.
10. Write a program to construct a recursive descent parser for an expression. Study of YACC tool and installation in Linux operating system.
11. Write a program for implementing a calculator for computing the given expression using semantic rules of the YACC tool.
12. Write a program for implementing Shift Reduce parsing and LR parsing.
13. Write a program for generating various intermediate code forms i) Three address code ii) Polish notation.