RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Artificial Intelligence and Data Science, VI-Semester

Open Elective AD 604 (C) Compiler Design

Unit-I: Introduction to compiling & Lexical Analysis

Introduction of Compiler, Major data Structure in compiler, types of Compiler, Front-end and Back-endof compiler, Compiler structure: analysis-synthesis model of compilation, various phases of a compiler,Lexical analysis: Input buffering , Specification & Recognition of Tokens,Design of a Lexical AnalyzerGenerator, LEX.

Unit-II Syntax Analysis &Syntax Directed Translation

Syntax analysis: CFGs, Top down parsing, Brute force approach, recursive descent parsing,transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LRparsers (SLR, LALR, LR),Parser generation. Syntax directed definitions: Construction of Syntax trees,Bottom up evaluation of S-attributed definition, L-attribute definition, Top down translation, Bottom Upevaluation of inherited attributes Recursive Evaluation, Analysis of Syntax directed definition.

Unit-III Type Checking & Run Time Environment

Type checking: type system, specification of simple type checker, equivalence of expression, types, typeconversion, overloading of functions and operations, polymorphic functions. Run time Environment:storage organization, Storage allocation strategies, parameter passing, dynamic storage allocation, Symbol table, Error Detection & Recovery, Ad-Hoc and Systematic Methods.

Unit –IV Code Generation

Intermediate code generation: Declarations, Assignment statements, Boolean expressions, Casestatements, Back patching, Procedure calls Code Generation: Issues in the design of code generator, Basicblock and flow graphs, Register allocation and assignment, DAG representation of basic blocks, peepholeoptimization, generating code from DAG.

Unit –V Code Optimization

Introduction to Code optimization: sources of optimization of basic blocks, loops in flow graphs, deadcode elimination, loop optimization, Introduction to global data flow analysis, Code Improvingtransformations ,Data flow analysis of structure flow graph Symbolic debugging of optimized code.

References:s

1. A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools , Pearson Education
2 Raghavan, Compiler Design, TMH Pub.
3. Louden. Compiler Construction: Principles and Practice, Cengage Learning
4. A. C. Holub. Compiler Design in C , Prentice-Hall Inc., 1993.
5. Mak, writing compiler & Interpreters, Willey Pub.