

1. Introduction

Objective: To teach basics of compiler principles and design.

Credits: 3-0-1

2. Course Outline

UNIT - I: Compiler Structure

Analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

UNIT - II: Lexical Analysis

Interface with input program, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, implementation, regular definition, transition diagrams.

UNIT - III: Syntax Analysis

CFGs, ambiguity, error detection and recovery, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LR parsers (SLR, LALR, LR).

UNIT - IV: Syntax Directed Translation

Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions

UNIT - V : Type Checking

Type systems, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

UNIT - VI : Run time Environments

Source language issues, storage organization, activation tree, activation record, stack allocation of activation records, parameter passing mechanisms, symbol tables, dynamic storage allocation techniques.

UNIT - VII : Intermediate Code Generation

Intermediate representations, translation of declarations, assignments, control flow, boolean expressions and procedure calls. Implementation issues.

UNIT - VIII : Code Generation and Instruction Selection

Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGs, peep hole optimization.

UNIT - IX : Code Optimizations

Principal Sources of Optimization, Optimization of basic blocks, Introduction to Dataflow Analysis (Reaching Definitions and Live Variable Analysis).

3. Reading Material

Text Books

1. AV Aho, MS Lam, R Sethi, JD Ullman: Compiler Design: Principles, Techniques and Tools, Pearson Education

Reference Books

1. AW Appel, M Ginsburg: Modern Compiler Implementation in C, Cambridge University Press
2. K Cooper, L Torzon: Engineering a Compiler, Morgan Kaufmann
3. J.P. Tremblay, P.G. Sorenson: Theory and Practice of Compiler Writing, McGraw Hill

Suggested Assignments

Programming assignments based on lexical analysis, construction of predictive/operator precedence/SLR parsing table and parser, symbol table, dynamic storage allocation strategies, syntax directed translation, data flow analysis