Course 27. Compiler Design (Web Course)

Faculty Coordinator(s):

1. Prof. Sanjeev K. Aggarwal
   Department of Computer Science and Engineering
   Indian Institute of Technology, Kanpur
   Kanpur
   E-Mail: ska@cse.iitk.ac.in
   Telephone: (91-512) Off: 259 7614

Detailed Syllabus:

Compiler Design (total 50 lectures)

Module 1: (3 lectures) Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

Module 2: (5 lectures) Lexical analysis: interface with input, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, and implementation. Regular definition, Transition diagrams, LEX.

Module 3: (4 lectures) Syntax analysis: context free grammars, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing.

Module 4: (4 lectures) Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Module 5: (5 lectures) Syntax directed definitions: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.

Module 6: (3 lectures) Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Module 7: (4 lectures) Run time system: storage organization, activation tree, activation record, parameter passing

Module 8: (3 lectures) symbol table, dynamic storage allocation.

Module 9: (3 lectures) Intermediate code generation: intermediate representations, translation of declarations, assignments
Module 10: (3 lectures) Intermediate Code generation for control flow, boolean expressions and procedure calls, implementation issues.

Module 11: (3 lectures) Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code generation

Module 12: (3 lectures) DAG representation of programs, code generation from dags, peep hole optimization, code generator generators, specifications of machine.

Module 13: (4 lectures) Code optimization, source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations.

Module 14: (3 lectures) Code improving transformations, dealing with aliases, data flow analysis of structured flow graphs

The course will have project where students will have to develop compiler for a subset of C language using tools like Lex and Yacc. The target environment will be SPIM simulator.