Brooklyn College
Department of Computer and Information Sciences

CISC 3390 [29] Compiler Construction

3 hours; 3 credits

Review of programming language structures, translation, loading, execution, and storage allocation. Compilation of simple expressions and statements. Organization of a compiler including compile-time and run-time symbol tables, lexical scan, object code generation, error diagnostics, object code optimization techniques, and overall design. Use of compiler-writing languages and bootstrapping.

Textbook:

Compilers: Principles, Techniques, and Tools. By: A. V. Aho, M. S. Lam, R. Sethi, and J. Ullman.

Syllabus:

I. Introduction (3 Hrs.)
  A. Evolution of Compilation Process
  B. Anatomy of a Compiler

II. Lexical Analysis (6 hrs.)
  A. Pragmatic Aspects of Lexical Analysis
    1. Transition Diagrams
    2. General Code structure of a Scanner
  B. Theoretical Aspects of Lexical Analysis
    1. Introduction to Finite Automaton, Grammars and Languages
    2. Transition Diagrams as semantically enhanced finite automaton.
    3. Transformation of regular expressions into their corresponding finite automaton.
    4. Regular expressions.
    5. Transformation of regular grammars into their corresponding finite automaton.
    6. Brief summary of basic concepts of automatic scanner generation.
  Transformation of nondeterministic finite automaton into deterministic
  finite automaton and then into actual code.

III. Parsing (12 hrs.)
  A. Basic Concepts
    1. Context-free grammars
    2. Derivations of strings
    3. Parse trees
    4. Top-down and Bottom-up parsing
  B. Top-down Parsing
    1. Top-down recursive with backtracking
    2. Recursive Descent

3. LL parsing (top-down table-driven parsing).
C. Bottom-up
1. Operator Precedence
2. LR Parsing

IV. Intermediate Code (3.5 hrs.)
A. Formats

V. Semantic Processing (3.5 hrs.)
A. Expressions
B. Conditionals
C. Looping Constructs
D. Array Indexing
E. Subroutine Invocation
F. Temporary Generation

VI. Run-Time Considerations (4.5 hrs.)
1. Stack manipulation
2. Static chains and displays
3. Parameter transmission

VII. Optimization (3 hrs.)
A. Basic Concepts
1. Optimization criteria
2. Basic blocks and flow graphs
B. Brief overview of some standard optimizations
1. Code Motion
2. Subexpression elimination
3. Loop jamming and unrolling
4. Live/Dead variable analysis

Project Discussion: 4.5