

10B11CI612: Compiler Design

Course Credit: 4

Semester: VI

Introduction

This course covers the basics of writing a compiler to translate from a simple high-level language to machine code. Topics include lexical analysis, top-down and LR parsing, syntax-directed translation, and code generation and optimization. Students will write a small compiler. The format of the course will be lecture-discussions, assignments. Students are strongly encouraged to participate actively in class discussions.

Course Objectives (Post-conditions)

Knowledge objectives:

At the conclusion of the course, following learning objectives are expected to be achieved:

1. Fluency in describing the theory and practice of compilation, in particular, the lexical analysis, syntax, and semantic analysis, code generation and optimization phases of compilation.
2. Ability to create lexical rules and grammars for a programming language.
3. Ability to use Flex or similar tools to create a lexical analyzer and Yacc/Bison tools to create a parser
4. Ability to implement a various parser such as a bottom-up SLR parser without using any compiler-generation tools
5. Ability to implement a various intermediate code generation techniques without using any compiler-generation tools
6. Ability to implement a various code optimizing techniques without using any compiler-generation tools
7. Ability to implement semantic rules into a parser that performs attribution while parsing
8. Ability to design a compiler for a concise programming language

Application objectives:

The students shall acquire the generic skills to design and implement a compiler along with analysis of practical aspects.

Expected Student Background (Preconditions)

Students are expected to have a solid grasp of the fundamentals of computer system, including a basic understanding of the operation of the computer, especially CPU. In addition, students are expected to know application development environment and programming concepts. Assembly programming ability will be helpful, as we will be looking at implementations of microprocessor operations.

Topics Outline:

S NO	Topics	Hrs
1	Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.	3
2	Lexical analysis: interface with input, parser and symbol table, token, lexeme and patterns,	5

	difficulties in lexical analysis, error reporting, and implementation. Regular definition, Transition diagrams, LEX.	
3	Syntax analysis: context free grammars, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing,	4
4	Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.	4
5	Syntax directed definitions: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.	5
6	Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.	3
7	Run time system: storage organization, activation tree, activation record, parameter passing	4
8	symbol table, dynamic storage allocation	3
9	Intermediate code generation: intermediate representations, translation of declarations, assignments	5
10	Intermediate Code generation for control flow, boolean expressions and procedure calls, implementation issues.	3
11	Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code generation	3
12	Code optimization, source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations.	4
13	Code improving transformations, dealing with aliases, data flow analysis of structured flow graphs	3
	Total	49

References

1. A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques, and Tools, Pearson Education, 2007 (second ed.).
2. K.D. Cooper, and L. Torczon, Engineering a Compiler, Elsevier, 2004.

Evaluation Scheme:

S.No	Examination	Marks
1	T-1	15
2	T-2	25
3	T-3	35
4	*Internal Marks	25

*Internal Marks Breakdown:

Assignments 9 marks (3x3)

Quizzes 12 marks (3x4)

Regularity 4 Marks