

G52CMP — Compilers

2002–2003

Examinable Material

Examination (May)

- Choice of FIVE questions
- Marks will be awarded for the best THREE answers

Source Material

- Watt and Brown, “Programming Language Processors in Java”, Chaps. 1,3,4,5,6
- Slides used in lectures (hand-written and electronic)

Topics

- Programming languages and compilers — basic components.
- Grammars. Extended BNF.
- Recursive Descent Syntax Analysis.
- Abstract Syntax Trees. Visitor Pattern.
- Contextual Analysis.
- Run-time storage allocation.
- Compiler-compilers. Grammar analysis.

Compilers and Languages.

Basic Components

- Language specification:
 - syntax
 - contextual constraints
 - semantics

Phases of a compiler

- Syntax analysis
- Contextual analysis
- Code Generation
- Subphases (lexical analysis, code optimisation)

Source: Watt and Brown, chapters 1 and 3.

(Extended) BNF

- elements: (non)terminal, production, alternation, sequencing, iteration
- (definition of) operator precedence, order of evaluation
- left/right recursion, ambiguity
- transformation (factoring, introduction/removal of recursion)

Source: Watt and Brown, chapter 4, lecture notes.

Recursive Descent Syntax Analysis

- Specification and construction of parsers.
- Difficulties (choice of production, left recursion etc.).

Abstract Syntax Trees

- Construction (by hand) of MiniTriangle ASTs (reference material on grammar and AST structure supplied)
- Construction (Java implementation) of ASTs during syntax analysis
- Use in structuring phases of compiler.
- Visitor Pattern. Java implementation. Applications.

Source: Watt and Brown, chapters 4,5 and 6. Slides.

Visitor Pattern

- Visitor interface (structure).
- Structure of implementations (eg Checker). Double dispatching.
- Applications.

Source: lectures, handouts, Watt and Brown, chapter 4, 5 and 6.

Contextual Analysis

- Identification table, attributes.
- Block structure, scope.
- Type checking.
 - function
 - implementation
- Implementation using visitor pattern.

Source: lectures, handouts, Watt and Brown, chapter 5.

Run-time storage allocation

- Expression evaluation (stack machine).
- Storage allocation.
 - Stack frames.
 - Static and dynamic links.

Source: Watt and Brown, sections 6.1–6.4.

Grammar Analysis

- Compiler-compilers.
- Consistency check (derivable symbols, non-emptiness of language generated).
- Nullable, FIRST, FOLLOW, LOOKAHEAD
 - Use in recursive-descent parsing.
 - Constructing defining equations.
 - Solution of equations using successive approximation.

Source: lectures, handouts.