
Contents

| | | |
|----------|---|-----------|
| 1 | The Structure of Compilers | 1 |
| 1.1 | Compilation Subtasks | 2 |
| 1.2 | Lexical Analysis | 3 |
| 1.3 | The Screener | 4 |
| 1.4 | Syntactic Analysis | 5 |
| 1.5 | Semantic Analysis | 6 |
| 1.6 | Machine-Independent Optimization | 7 |
| 1.7 | Generation of the Target Program | 8 |
| 1.8 | Specification and Generation of Compiler Components | 9 |
| 2 | Lexical Analysis | 11 |
| 2.1 | The Task of Lexical Analysis | 11 |
| 2.2 | Regular Expressions and Finite Automata | 12 |
| 2.2.1 | Words and Languages | 12 |
| 2.3 | A Language for Specifying Lexical Analyzers | 27 |
| 2.3.1 | Character Classes | 28 |
| 2.3.2 | Nonrecursive Parentheses | 28 |
| 2.4 | Scanner Generation | 29 |
| 2.4.1 | Character Classes | 29 |
| 2.4.2 | An Implementation of the <i>until</i> -Construct | 30 |
| 2.4.3 | Sequences of Regular Expressions | 32 |
| 2.4.4 | The Implementation of a Scanner | 34 |
| 2.5 | The Screener | 36 |
| 2.5.1 | Scanner States | 37 |
| 2.5.2 | Recognizing Reserved Words | 38 |
| 2.6 | Exercises | 39 |
| 2.7 | Bibliographic Notes | 41 |
| 3 | Syntactic Analysis | 43 |
| 3.1 | The Task of Syntactic Analysis | 43 |
| 3.2 | Foundations | 46 |
| 3.2.1 | Context-Free Grammars | 46 |
| 3.2.2 | Productivity and Reachability of Nonterminals | 52 |
| 3.2.3 | Pushdown Automata | 57 |

| | | |
|----------|--|-----|
| 3.2.4 | The Item-Pushdown Automaton to a Context-Free Grammar | 59 |
| 3.2.5 | first- and follow-Sets | 64 |
| 3.2.6 | The Special Case first_1 and follow_1 | 71 |
| 3.2.7 | Pure Union Problems | 74 |
| 3.3 | <i>Top-Down</i> Syntax Analysis | 77 |
| 3.3.1 | Introduction | 77 |
| 3.3.2 | $LL(k)$: Definition, Examples, and Properties | 79 |
| 3.3.3 | Left Recursion | 85 |
| 3.3.4 | Strong $LL(k)$ -Parsers | 87 |
| 3.3.5 | Right-Regular Context-Free Grammars | 89 |
| 3.3.6 | <i>Recursive-Descent</i> Parsers for $RLL(1)$ -Grammars | 92 |
| 3.3.7 | Tree Generation by Means of <i>Recursive-Descent</i> Parsers | 96 |
| 3.3.8 | Error Handling within $RLL(1)$ -Parsers | 97 |
| 3.4 | <i>Bottom-up</i> Syntax Analysis | 101 |
| 3.4.1 | Introduction | 101 |
| 3.4.2 | $LR(k)$ -Parsers | 102 |
| 3.4.3 | $LR(k)$: Definition, Properties, and Examples | 112 |
| 3.4.4 | Error Handling in LR -Parsers | 124 |
| 3.5 | Exercises | 131 |
| 3.6 | Bibliographic Notes | 136 |
| 4 | Semantic Analysis | 139 |
| 4.1 | The Task of Semantic Analysis | 139 |
| 4.1.1 | Rules for Validity and Visibility | 144 |
| 4.1.2 | Checking the Context-Conditions | 147 |
| 4.1.3 | Overloading of Identifiers | 154 |
| 4.2 | Type Inference | 157 |
| 4.2.1 | Syntax-Directed Type Inference | 165 |
| 4.2.2 | Polymorphism | 168 |
| 4.2.3 | Side Effects | 171 |
| 4.2.4 | Type Classes | 173 |
| 4.3 | Attribute Grammars | 180 |
| 4.3.1 | The Semantics of an Attribute Grammar | 184 |
| 4.3.2 | Some Attribute Grammars | 185 |
| 4.4 | The Generation of Attribute Evaluators | 191 |
| 4.4.1 | Demand-Driven Attribute Evaluation | 191 |
| 4.4.2 | Static Precomputations of Attribute Evaluators | 193 |
| 4.4.3 | Visit-Oriented Attribute Evaluation | 200 |
| 4.4.4 | Parser-Directed Attribute Evaluation | 206 |
| 4.5 | Exercises | 213 |
| 4.6 | Bibliographic Notes | 215 |
| | References | 217 |
| | Index | 221 |



<http://www.springer.com/978-3-642-17539-8>

Compiler Design

Syntactic and Semantic Analysis

Wilhelm, R.; Seidl, H.; Hack, S.

2013, X, 225 p., Hardcover

ISBN: 978-3-642-17539-8