Contents

Preface xi

1 Preliminaries 1
  1.1 Abstract Syntax Trees 1
  1.2 Grammars 3
  1.3 Pattern Matching 5
  1.4 Recursive Functions 6
  1.5 Interpreters 8
  1.6 Example Compiler: A Partial Evaluator 10

2 Integers and Variables 13
  2.1 The \( \mathcal{L}_\text{Var} \) Language 13
  2.2 The x86 Instruction Set Assembly Language 16
  2.3 Planning the Trip to x86 21
  2.4 Remove Complex Operands 23
  2.5 Select Instructions 25
  2.6 Assign Homes 26
  2.7 Patch Instructions 27
  2.8 Generate Prelude and Conclusion 27
  2.9 Challenge: Partial Evaluator for \( \mathcal{L}_\text{Var} \) 28

3 Parsing 29
  3.1 Lexical Analysis and Regular Expressions 29
  3.2 Grammars and Parse Trees 31
  3.3 Ambiguous Grammars 33
  3.4 From Parse Trees to Abstract Syntax Trees 34
  3.5 Earley’s Algorithm 36
  3.6 The LALR(1) Algorithm 40
  3.7 Further Reading 43

4 Register Allocation 45
  4.1 Registers and Calling Conventions 46
  4.2 Liveness Analysis 49
  4.3 Build the Interference Graph 51
4.4 Graph Coloring via Sudoku 52
4.5 Patch Instructions 58
4.6 Generate Prelude and Conclusion 58
4.7 Challenge: Move Biasing 59
4.8 Further Reading 62

5 Booleans and Conditionals 65
5.1 The $\mathcal{L}_if$ Language 66
5.2 Type Checking $\mathcal{L}_if$ Programs 66
5.3 The $\mathcal{C}_if$ Intermediate Language 72
5.4 The x86if Language 72
5.5 Shrink the $\mathcal{L}_if$ Language 75
5.6 Remove Complex Operands 75
5.7 Explicate Control 76
5.8 Select Instructions 82
5.9 Register Allocation 83
5.10 Patch Instructions 84
5.11 Generate Prelude and Conclusion 84
5.12 Challenge: Optimize Blocks and Remove Jumps 85
5.13 Further Reading 88

6 Loops and Dataflow Analysis 91
6.1 The $\mathcal{L}_{while}$ Language 91
6.2 Cyclic Control Flow and Dataflow Analysis 91
6.3 Remove Complex Operands 96
6.4 Explicate Control 96
6.5 Register Allocation 96

7 Tuples and Garbage Collection 99
7.1 The $\mathcal{L}_{tup}$ Language 99
7.2 Garbage Collection 102
7.3 Expose Allocation 109
7.4 Remove Complex Operands 110
7.5 Explicate Control and the $\mathcal{C}_{tup}$ Language 110
7.6 Select Instructions and the x86Global Language 111
7.7 Register Allocation 116
7.8 Generate Prelude and Conclusion 116
7.9 Challenge: Arrays 118
7.10 Further Reading 123

8 Functions 125
8.1 The $\mathcal{L}_{fun}$ Language 125
8.2 Functions in x86 130
8.3 Shrink $\mathcal{L}_{fun}$ 133
8.4 Reveal Functions and the $\mathcal{L}_{funref}$ Language 133