Markus Müller-Olm

Modular Compiler Verification

A Refinement-Algebraic Approach Advocating Stepwise Abstraction

Technische Universität Darmstadt
FACHBEREICH INFORMATIK
BIBLIOTHEK
Inventar-Nr.: M96 - 03898
Sachgebiete:
Standort:

Springer

Table of Contents

1. **Introduction** 1
   1.1 Organization of This Book 6
   1.2 Related Literature on Code Generator Verification 7

2. **Complete Boolean Lattices** 9
   2.1 Basic Definitions 9
   2.2 Functions 11
      2.2.1 Functions from a Set to an Ordered Set or Lattice 11
      2.2.2 Functions Between Lattices 12
   2.3 Fixpoints 14

3. **Galois Connections** 15
   3.1 Definition and Basic Facts 15
   3.2 Distributive Galois Connections 17
   3.3 Galois Connections Between Power Sets 19
   3.4 Composing Galois Connections 20

4. **States, Valuation Functions and Predicates** 23
   4.1 Variables 26
   4.2 States 27
      4.2.1 Variation 27
   4.3 Valuation Functions 28
      4.3.1 Projection and Lifting 28
      4.3.2 Substitution 30
      4.3.3 Independence 32
      4.3.4 Extending and Restricting the Alphabet 34
   4.4 Predicates 35
      4.4.1 Quantification 35

5. **The Algebra of Commands** 39
   5.1 Definitions 41
   5.2 Dualization 43
   5.3 Simulations 44
   5.4 Basic Homogeneous Commands 45

5.4.1 Multiple Assignments 46
5.4.2 Assertions and Assumptions 48
5.4.3 Derived Laws for Basic Homogeneous Commands 53
5.5 Basic Heterogeneous Commands 57
5.6 Distribution Properties 67
5.6.1 Basic Commands 67
5.6.2 Preservation of Distribution Properties 67
5.6.3 Distribution Properties of Composition Operators 68
5.7 Derived Commands 69
5.7.1 Choice Assignments 70
5.7.2 Conditionals 78
5.7.3 Arrays 80
5.8 Discussion 81

6. Communication and Time 85
6.1 An Operational Framework 87
6.1.1 Communication Commands 89
6.1.2 Loops 91
6.1.3 Remarks 94
6.2 Definitions 95
6.3 Input and Output Processes 96
6.4 Stop 98
6.5 Delays 98
6.6 Time Bounds 101
6.7 Discussion 103

7. Data Refinement 105

8. Transputer Base Model 113
8.1 Basic Data-Types and Constants 114
8.2 Modeling the Transputer's Components 116
8.2.1 Registers 116
8.2.2 Memory 116
8.2.3 Links 118
8.3 Instructions and Operations 118
8.4 Capturing the Dynamic Behavior 119
8.5 Discussion 124

9. A Small Hard Real-Time Programming Language 127
9.1 Abstract Syntax 129
9.2 Static Semantics 130
9.3 Dynamic Semantics 133

10. A Hierarchy of Views 135
10.1 Symbolic Representation of Control Point 137
10.2 Large Operands 147
10.2.1 Some Properties of *prefix* 148
10.2.2 Performing the Abstraction 151
10.2.3 Operations 154
10.3 Workspace 156
10.4 Symbolic Addressing 160
10.4.1 Instructions and Operations 163
10.4.2 Extending and Restricting Dictionaries 169
10.5 Forgetting About Registers 175
10.6 Discussion 177

11. Compiling-Correctness Relations 179
11.1 Programs 181
11.2 Statements 183
11.3 Expressions 185
11.4 Monadic and Binary Operators 186

12. Translation Theorems 189
12.1 Preliminaries 189
12.2 Programs 193
12.3 Statements 194
12.4 Expressions 204
12.5 Monadic Operators 207
12.6 Binary Operators 209
12.6.1 Arithmetic Operators 209
12.6.2 Relational Operators 212
12.6.3 Boolean Operators 213

13. A Functional Implementation 215
13.1 Syntactic Dictionaries 216
13.2 Refined Correctness Relations 217
13.3 Refined Translation Theorems 218
13.4 Compiling Functions 220
13.4.1 Constructive Translation of Conditionals 225
13.4.2 Constructive Translation of Loops 226
13.5 Discussion 230

14. Conclusion 233
14.1 Topics for Future Research 236
14.1.1 Extensions 236
14.1.2 Improvements 237
14.1.3 Modifications 237

References 239

Index 245