Contents

Preface ix

I Precise Type Analyses 1

1 ***Substructural Type Systems*** **3**
David Walker

1.1 Structural Properties 4
1.2 A Linear Type System 6
1.3 Extensions and Variations 17
1.4 An Ordered Type System 30
1.5 Further Applications 36
1.6 Notes 40

2 ***Dependent Types*** **45**
David Aspinall and Martin Hofmann

2.1 Motivations 45
2.2 Pure First-Order Dependent Types 50
2.3 Properties 54
2.4 Algorithmic Typing and Equality 56
2.5 Dependent Sum Types 61
2.6 The Calculus of Constructions 64
2.7 Relating Abstractions: Pure Type Systems 71
2.8 Programming with Dependent Types 74
2.9 Implementation of Dependent Types 83
2.10 Further Reading 86

3 ***Effect Types and Region-Based Memory Management*** **87**
Fritz Henglein, Henning Makholm, and Henning Niss

3.1 Introduction and Overview 87
3.2 Value Flow by Typing with Labels 90
3.3 Effects 102
3.4 Region-Based Memory Management 106
3.5 The Tofte–Talpin Type System 114
3.6 Region Inference 123
3.7 More Powerful Models for Region-Based Memory Management 127
3.8 Practical Region-Based Memory Management Systems 133

II Types for Low-Level Languages 137

4 ***Typed Assembly Language*** **141**
Greg Morrisett

4.1 TAL-0: Control-Flow-Safety 142
4.2 The TAL-0 Type System 146
4.3 TAL-1: Simple Memory-Safety 155
4.4 TAL-1 Changes to the Type System 161
4.5 Compiling to TAL-1 164
4.6 Scaling to Other Language Features 167
4.7 Some Real World Issues 172
4.8 Conclusions 175

5 ***Proof-Carrying Code*** **177**
George Necula

5.1 Overview of Proof Carrying Code 177
5.2 Formalizing the Safety Policy 182
5.3 Verification-Condition Generation 187
5.4 Soundness Proof 199
5.5 The Representation and Checking of Proofs 204
5.6 Proof Generation 214
5.7 PCC beyond Types 216
5.8 Conclusion 219

III Types and Reasoning about Programs 221

6 *Logical Relations and a Case Study in Equivalence Checking* 223
Karl Crary

6.1 The Equivalence Problem 224
6.2 Non-Type-Directed Equivalence Checking 225
6.3 Type-Driven Equivalence 227
6.4 An Equivalence Algorithm 228
6.5 Completeness: A First Attempt 232
6.6 Logical Relations 233
6.7 A Monotone Logical Relation 236
6.8 The Main Lemma 237
6.9 The Fundamental Theorem 239
6.10 Notes 243

7 *Typed Operational Reasoning* 245
Andrew Pitts

7.1 Introduction 245
7.2 Overview 246
7.3 Motivating Examples 247
7.4 The Language 253
7.5 Contextual Equivalence 261
7.6 An Operationally Based Logical Relation 266
7.7 Operational Extensionality 279
7.8 Notes 288

IV Types for Programming in the Large 291

8 *Design Considerations for ML-Style Module Systems* 293
Robert Harper and Benjamin C. Pierce

8.1 Basic Modularity 294
8.2 Type Checking and Evaluation of Modules 298
8.3 Compilation and Linking 302
8.4 Phase Distinction 305
8.5 Abstract Type Components 307
8.6 Module Hierarchies 317
8.7 Signature Families 320
8.8 Module Families 324
8.9 Advanced Topics 338
8.10 Relation to Some Existing Languages 341
8.11 History and Further Reading 343

9 *Type Definitions* 347
Christopher A. Stone

9.1 Definitions in the Typing Context 351
9.2 Definitions in Module Interfaces 358
9.3 Singleton Kinds 367
9.4 Notes 384

V Type Inference 387

10 *The Essence of ML Type Inference* 389
François Pottier and Didier Rémy

10.1 What Is ML? 389
10.2 Constraints 407
10.3 HM(X) 422
10.4 Constraint Generation 429
10.5 Type Soundness 434
10.6 Constraint Solving 438
10.7 From ML-the-Calculus to ML-the-Language 451
10.8 Rows 460

A *Solutions to Selected Exercises* 491

References 535

Index 567