Computation and Reasoning

A Type Theory for Computer Science

ZHAOHUI LUO
University of Edinburgh

CLARENDON PRESS • OXFORD
1994

Contents

1 Introduction 1

1.1 Motivations in computer science 2

1.2 Basic concepts in type theory 3

1.2.1 Objects and types 3

1.2.2 Propositions as types 5

1.2.3 Meaning and use 6

1.3 The conceptual universe of types 8

1.3.1 Paradoxical type structures 9

1.3.2 Martin-Löf's type theory 10

1.3.3 Impredicative type theories 11

1.3.4 Data types vs. logical propositions 12

1.4 Towards a unifying theory of dependent types 15

2 The Extended Calculus of Constructions 21

2.1 The language of ECC 21

2.1.1 Terms and computation 21

2.1.2 Judgements and inference rules 24

2.2 Informal explanations 26

2.2.1 Computation and computational equality 26

2.2.2 Judgements and context validity 28

2.2.3 Types and their meaning explanation 29

2.2.4 Non-propositional types 29

2.2.5 Predicative universes and the reflection principle 32

2.2.6 Propositions and the impredicative universe 34

2.2.7 Type equality and the cumulativity relation 36

2.3 Further remarks and discussion 38

2.3.1 Data types vs. logical propositions 38

2.3.2 Σ-types and existential types 41

2.3.3 Equalities: intensionality vs. extensionality 43

2.3.4 Decidability issues 46

2.3.5 On the use of the type theory 46

3 Basic meta-theoretic properties 49

3.1 Church-Rosser theorem and cumulativity 49

3.2 Derivable judgements and derivability 54

3.3 Principal types 61

4 Strong normalisation 65

4.1 The Girard–Tait reducibility method 66

4.1.1 The reducibility method and the notion of predicativity 66

4.1.2 Environments 69

4.1.3 Saturated sets and candidates of reducibility 71

4.2 Quasi-normalisation 74

4.2.1 Levels of types 74

4.2.2 Quasi-normalisation and degrees of types 76

4.2.3 The complexity measure of types 79

4.2.4 An inductive proof of the quasi-normalisation theorem 81

4.3 Strong normalisation 88

4.3.1 Possible denotations of objects 88

4.3.2 Assignments and valuations 90

4.3.3 The interpretation 92

4.3.4 Soundness of the interpretation 96

4.3.5 The strong normalisation theorem 100

5 The internal logic and decidability 103

5.1 The internal higher-order logic 103

5.1.1 The internal logic and its consistency 103

5.1.2 Understanding the logical operators 106

5.1.3 The Leibniz equality and equality reflection 109

5.2 Decidability of the type theory 112

5.2.1 Decidability of conversion and cumulativity 112

5.2.2 Decidability of type inference and type checking 112

6 A set-theoretic model 117

6.1 Understanding ECC in the ω-**Set** framework 118

6.2 Valid contexts and objects 120

6.3 Predicative universes and non-propositional types 121

6.4 The impredicative universe and propositions 123

6.5 Remarks 125

7 Computational and logical theories 127

7.1 Computational theories and inductive data types 128

7.1.1 The type of natural numbers 128

7.1.2 The type of lists of natural numbers 131

7.2 Abstract theories and abstract reasoning 132

7.2.1 A notion of abstract theory 133

7.2.2 Abstract reasoning 136

7.2.3 Theory morphisms and proof inheritance 136

7.3 Discussion 137

8 Specification and development of programs 139

8.1 A brief summary 140

8.2 Specifications and data refinement 141
8.2.1 Program specifications and their realisations 141
8.2.2 Specifications of abstract data types 144
8.2.3 Data refinement and implementation 146
8.3 Modular design and structured specification 149
8.3.1 Decomposition and sharing 150
8.3.2 Constructors and selectors 153
8.3.3 Constructor/selector implementation 156
8.4 Parameterised specification 156
8.4.1 Parameterised specifications 157
8.4.2 Implementation of parameterised specifications 159
8.5 Discussion 162

9 Towards a unifying theory of dependent types 165
9.1 A logical framework with inductive schemata 166
9.1.1 Martin-Löf's logical framework 166
9.1.2 Specifying type theories in LF 170
9.1.3 Inductive schemata 173
9.2 The formulation of UTT 175
9.2.1 SOL: the internal logical mechanism 175
9.2.2 Inductive data types 177
9.2.3 Predicative universes 182
9.2.4 A summary 185
9.3 Discussion 186
9.3.1 The internal logic and pure logical truths 186
9.3.2 Further separation of propositions and data types 195
9.3.3 Intensionality and η-equality rules 198
9.3.4 Understanding of the type theory 202
9.3.5 Inductive families of types 204
9.3.6 Subtyping and other implementation issues 208
9.4 Final remarks 209

Bibliography 211

Notation and Symbols 221

Index 225