Computable Models

Raymond Turner

Computable Models

Springer

Raymond Turner
University of Essex
UK

ISBN 978-1-84882-051-7 e-ISBN 978-1-84882-052-4
DOI 10.1007/978-1-84882-052-4

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009920532

Printed on acid-free paper

Springer Science+Business Media
springer.com

Preface

I take the central task of *theoretical computing science (*TCS*)* to be the construction of mathematical models of computational phenomena. Such models provide us with a deeper understanding of the nature of *computation* and *representation*. For example, the early work on computability theory provided a mathematical model of computation. Later work on the semantics of programming languages enabled a precise articulation of the underlying differences among programming languages and led to a clearer understanding of the distinction between semantic representation and implementation. Early work in complexity theory supplied us with abstract notions that formally articulated informal ideas about the resources used during computation. Such mathematical modeling provides the means of exploring the properties and limitations of languages and tools that would otherwise be unavailable.

The aim of this book is to contribute to this fundamental activity. Here we have two interrelated goals. One is to provide a logical framework and foundation for the process of specification and the design of specification languages. The second is to employ this framework to introduce and study *computable models*. These extend the notion of specification to the more general arena of mathematical modeling where our aim is to build mathematical models that are constructed from specifications.

During the preparation of this book, every proper computer scientist at the University of Essex provided valuable feedback. Some provided quite detailed comments. I will not single out any of you; you know who you are. But to all who contributed, thank you. Referees on the various journal papers that led to the book also provided valuable advice and criticism. But my greatest debt is to my wife, Rosana. Over the years, she has read draft after draft and made innumerable (not literally) suggestions for change and improvement. Without her, the size of the set of errors that remains would be much greater than it is.

Contents

1 What is a Computable Model? 1
1.1 Mathematical Models 1
1.2 Specifications, Programs, and Models 2
1.3 Data Types and Programming Languages 3
1.4 Theories of Data 4
1.5 Recursive Models 5
1.6 Intensional Models 6
1.7 A Logical Foundation for Specification 7
1.8 Implementable Models 7
1.9 The Logical Setting 8
References 9

2 Typed Predicate Logic 11
2.1 Judgments and Contexts 11
2.2 Structural Rules 13
2.3 Types 13
2.4 Relations and Functions 15
2.5 Equality 15
2.6 Propositional Rules 16
2.7 Quantifier Rules 17
2.8 TPL Derivations 18
2.9 Type Inference 20
References 23

3 Data Types 25
3.1 Booleans 25
3.2 Products 27
3.3 Stacks 29
3.4 Terms 30
3.5 Numbers 30
3.6 Lists 32
3.7 A Type of Types 33

3.8 Theories of Data Types 33
References 35

4 Definability 37
4.1 Semidecidable Relations 38
4.2 Decidable Relations 39
References 41

5 Specification 43
5.1 A Logical Perspective 44
5.2 Some Specifications 45
5.3 Operations on Schema 48
5.4 Conservative Extensions 51
References 52

6 Functions 53
6.1 Totality and Functionality 53
6.2 Functional Application 54
6.3 Explicit Functions 57
6.4 The Elimination of Application 59
References 62

7 Preconditions 63
7.1 Specifications with Preconditions 63
7.2 Totality and Functionality 65
7.3 Functional Application 67
7.4 Application Elimination 69
7.5 Partial Functions 69
References 70

8 Natural Numbers 71
8.1 A Theory of Numbers 71
8.2 Numerical Specification 74
8.3 Recursive Specifications 79
8.4 Enriched Arithmetic 82
8.5 Arithmetic Interpretation 83
References 84

9 Typed Set Theory 85
9.1 CST 85
9.2 Elementary Properties 88
9.3 Subsets and Extensionality 89
9.4 New Sets from Old 90
9.5 Set-Theoretic Relations 97

9.6 Arithmetic Interpretation . . . 100
References . . . 102

10 Systems Modeling . . . 103
10.1 The Requirements . . . 103
10.2 The State . . . 104
10.3 Operations . . . 107
10.4 A Mathematical Model . . . 108
References . . . 111

11 A Type of Types . . . 113
11.1 The Type type . . . 114
11.2 Dependent Types . . . 115
11.3 Dependent Specifications . . . 116
11.4 Polymorphic Specifications . . . 117
11.5 Polymorphic Set Theory . . . 120
11.6 Specifications and Types . . . 122
11.7 Arithmetic Interpretation . . . 124
References . . . 125

12 Schemata . . . 127
12.1 A Theory of Relations . . . 127
12.2 A Minimal Theory . . . 130
12.3 Operations on Schemata . . . 132
12.4 Arithmetic Interpretation . . . 139
References . . . 140

13 Separation Types . . . 143
13.1 Theories with Separation . . . 143
13.2 Subtypes in Specification . . . 145
13.3 Preconditions and Functions . . . 146
13.4 Polymorphism and Subtypes . . . 148
13.5 The Elimination of Subtypes . . . 149
References . . . 153

14 Recursive Schemata . . . 155
14.1 Closure and Induction . . . 155
14.2 Simultaneous Recursion . . . 158
14.3 Arithmetic Interpretation . . . 162
14.4 Sets and Schemata . . . 162
References . . . 166

15 Inductive Types . . . 167
15.1 The General Form . . . 167

15.2 Some Inductive Types . . . 168
15.3 Conservative Extensions . . . 170
15.4 Finite Schemata . . . 171
References . . . 175

16 Recursive Functions . . . 177
16.1 General Form . . . 177
16.2 Numerical Recursion . . . 180
16.3 Recursive Functions and Inductive Types . . . 181
References . . . 183

17 Schema Definitions . . . 185
17.1 Schema Definitions . . . 185
17.2 Refinement . . . 188
17.3 Implementable Definitions . . . 191
17.4 The Limits of Refinement . . . 192
17.5 Properties of Schemata . . . 193
References . . . 194

18 Computable Ontology . . . 195
18.1 Implementable Models . . . 195
18.2 A Type of Events . . . 196
18.3 Arithmetic Interpretation . . . 197
18.4 Instants . . . 198
18.5 Implementation . . . 199
References . . . 199

19 Classes . . . 201
19.1 Classes and Judgments . . . 201
19.2 Class Elimination . . . 204
References . . . 205

20 Classes of Functions . . . 207
20.1 Function Application . . . 207
20.2 Specifications and Function Classes . . . 209
20.3 Partial Functions . . . 212
20.4 Polymorphism . . . 213
References . . . 215

21 Computable Analysis . . . 217
21.1 Cauchy Sequences . . . 217
21.2 Operations on the Real Numbers . . . 219
21.3 Implementation . . . 220
References . . . 222

22 Programming Language Specification 223
22.1 The Abstract Machine 223
22.2 A Programming Language and Its Specification 226
22.3 Implementation 228
References 229

23 Abstract Types 231
23.1 Axiomatic Specifications 231
23.2 Polymorphism and Data Abstraction 234
References 236

24 Conclusion 237

Index 239