1 Basic set theory
1.1 Logical notation 1
1.2 Sets
1.2.1 Sets and properties 3
1.2.2 Some important sets 3
1.2.3 Constructions on sets 4
1.2.4 The axiom of foundation 6
1.3 Relations and functions 6
1.3.1 Lambda notation 7
1.3.2 Composing relations and functions 7
1.3.3 Direct and inverse image of a relation 9
1.3.4 Equivalence relations 9
1.4 Further reading 10

2 Introduction to operational semantics
2.1 IMP—a simple imperative language 11
2.2 The evaluation of arithmetic expressions 13
2.3 The evaluation of boolean expressions 17
2.4 The execution of commands 19
2.5 A simple proof 20
2.6 Alternative semantics 24
2.7 Further reading 26

3 Some principles of induction
3.1 Mathematical induction 27
3.2 Structural induction 28
3.3 Well-founded induction 31
3.4 Induction on derivations 35
3.5 Definitions by induction 39
3.6 Further reading

4 Inductive definitions

4.1 Rule induction
4.2 Special rule induction
4.3 Proof rules for operational semantics
  4.3.1 Rule induction for arithmetic expressions
  4.3.2 Rule induction for boolean expressions
  4.3.3 Rule induction for commands
4.4 Operators and their least fixed points
4.5 Further reading

5 The denotational semantics of IMP

5.1 Motivation
5.2 Denotational semantics
5.3 Equivalence of the semantics
5.4 Complete partial orders and continuous functions
5.5 The Knaster-Tarski Theorem
5.6 Further reading

6 The axiomatic semantics of IMP

6.1 The idea
6.2 The assertion language Assn
  6.2.1 Free and bound variables
  6.2.2 Substitution
6.3 Semantics of assertions
6.4 Proof rules for partial correctness
6.5 Soundness
6.6 Using the Hoare rules—an example
6.7 Further reading

7 Completeness of the Hoare rules
7.1 Gödel's Incompleteness Theorem 99
7.2 Weakest preconditions and expressiveness 100
7.3 Proof of Gödel's Theorem 110
7.4 Verification conditions 112
7.5 Predicate transformers 115
7.6 Further reading 117

8 Introduction to domain theory 119
8.1 Basic definitions 119
8.2 Streams—an example 121
8.3 Constructions on cpo's 123
  8.3.1 Discrete cpo's 124
  8.3.2 Finite products 125
  8.3.3 Function space 128
  8.3.4 Lifting 131
  8.3.5 Sums 133
8.4 A metalanguage 135
8.5 Further reading 139

9 Recursion equations 141
9.1 The language REC 141
9.2 Operational semantics of call-by-value 143
9.3 Denotational semantics of call-by-value 144
9.4 Equivalence of semantics for call-by-value 149
9.5 Operational semantics of call-by-name 153
9.6 Denotational semantics of call-by-name 154
9.7 Equivalence of semantics for call-by-name 157
9.8 Local declarations 161
9.9 Further reading 162

10 Techniques for recursion 163
10.1 Bekić's Theorem 163
13 Recursive types 251
13.1 An eager language 251
13.2 Eager operational semantics 255
13.3 Eager denotational semantics 257
13.4 Adequacy of eager semantics 262
13.5 The eager $\lambda$-calculus 267
   13.5.1 Equational theory 269
   13.5.2 A fixed-point operator 272
13.6 A lazy language 278
13.7 Lazy operational semantics 278
13.8 Lazy denotational semantics 281
13.9 Adequacy of lazy semantics 288
13.10 The lazy $\lambda$-calculus 290
   13.10.1 Equational theory 291
   13.10.2 A fixed-point operator 292
13.11 Further reading 295

14 Nondeterminism and parallelism 297
14.1 Introduction 297
14.2 Guarded commands 298
14.3 Communicating processes 303
14.4 Milner's CCS 308
14.5 Pure CCS 311
14.6 A specification language 316
14.7 The modal $\nu$-calculus 321
14.8 Local model checking 327
14.9 Further reading 335

A Incompleteness and undecidability 337
   Bibliography 353
   Index 357