

Contents

1	Introduction	1
1.1	Minds, Method and Machines	3
1.2	Language and Science	4
1.3	The Age of Machinery	7
1.4	The Mechanization of Mathematical Language	8
2	Babbage's Engines	17
2.1	The Division of Mental Labour	18
2.2	The Difference Engine	21
2.3	The Meanings of the Difference Engine	25
2.4	The Mechanical Notation	28
2.5	The Analytical Engine	31
2.6	The Science of Operations	41
2.7	The Meanings of the Analytical Engine	44
2.8	Conclusions	48
3	Semi-Automatic Computing	53
3.1	The Census Problem	53
3.2	The Hollerith Tabulating System of 1890	55
3.3	Further Developments in Punched Card Machines	57
3.4	Comrie and the Mechanization of Scientific Calculation	60
3.5	Semi-Automatic Programming	65
4	Logic, Computability and Formal Systems	67
4.1	Gödel's Construction	69
4.2	Recursive Functions	72
4.3	λ -definability	74
4.4	Direct Approaches to Defining Effective Computability	75
4.5	Turing's Machine Table Notation	77
4.6	Universal Machines	89
4.7	The Concept of a Formal Language	92
4.8	The Relationship Between Turing's Work and Logic	96

5	Automating Control	99
5.1	Konrad Zuse's Early Machines	100
5.2	Mark I: The Automatic Sequence Controlled Calculator	102
5.3	The ENIAC	107
5.4	The Bell Labs Relay Machines	115
5.5	The Significance of the Automatic Calculators	118
6	Logic and the Invention of the Computer	123
6.1	The Origins of the Stored-Program Computer	126
6.2	The Early Development of Cybernetics	130
6.3	Von Neumann's Design for the EDVAC	133
6.4	Logic and the Stored-Program Concept	136
6.5	The EDVAC Code and Address Modification	139
6.6	Turing and the ACE	142
6.7	Giant Brains	145
6.8	Universal Machines	147
6.9	General-Purpose Machines	153
6.10	Conclusions	154
7	Machine Code Programming and Logic	157
7.1	Sequencing of Operations	158
7.2	Transfer of Control	162
7.3	Condition Testing	164
7.4	Instruction Modification	167
7.5	Subroutines	170
7.6	Machine Code and Program Structures	172
7.7	Machine Code and Logic	174
7.8	Syntax	176
7.9	Flow Diagrams and Program Semantics	179
7.10	Programs as Metalinguistic Expressions	182
7.11	Conclusions	183
8	The Invention of Programming Languages	185
8.1	Automatic Coding	186
8.2	The Semantics of Pseudocodes	188
8.3	Formula Translation	193
8.4	Fortran and Increasing Linguistic Complexity	197
8.5	Universal Languages	204
8.6	Algol 60 as a Formal Language	209
8.7	The Influence of Logic on Algol	217
8.8	Lisp and Recursive Function Theory	220
8.9	Conclusions	224
9	The Algol Research Programme	225
9.1	Algol 60 as a Concrete Paradigm	226
9.2	Normal Science in the Algol Research Programme	229

9.3	The Description of Programming Languages	230
9.4	Different Philosophies of Programming Language Design	237
9.5	Logic and the Design of Control Structures	239
9.6	Logic and Data Structures	244
9.7	Modelling Data for Information Retrieval	247
9.8	Conclusions	252
10	The Logic of Correctness in Software Engineering	253
10.1	Checking Computations	253
10.2	Debugging and Testing	255
10.3	Correctness Proofs	257
10.4	Constructive Methods	261
10.5	Specifications and Correctness	263
10.6	Structured Programming	265
10.7	Proof and Testing	269
10.8	Conclusions	275
11	The Unification of Data and Algorithms	277
11.1	Simulation Languages	278
11.2	Modelling the Real World	281
11.3	Simula 67	282
11.4	Data Abstraction	283
11.5	Smalltalk	288
11.6	The Relationship Between Smalltalk and Logic	293
11.7	Conclusions	296
12	Conclusions	297
12.1	Paradigms and Revolutions	298
12.2	Relating Theory and Practice	301
12.3	Methodological Conclusions	303
Appendix	Turing's Universal Machine	307
A.1	General Purpose <i>m</i> -functions	307
A.2	The Contents of the Tape	310
A.3	The Main Table	312
References	317
Index	335



<http://www.springer.com/978-1-84882-554-3>

A Science of Operations
Machines, Logic and the Invention of Programming
Priestley, M.
2011, X, 342 p., Hardcover
ISBN: 978-1-84882-554-3