

Giovanni Sommaruga • Thomas Strahm
Editors

Turing's Revolution

The Impact of His Ideas about Computability

 Birkhäuser

Contents

Part I Turing and the History of Computability Theory	
Conceptual Confluence in 1936: Post and Turing	3
Martin Davis and Wilfried Sieg	
1 Introduction	4
2 Substitution Puzzles: The Link	5
3 Effectively Calculable Functions	7
4 Canonical Systems: Post	10
5 Mechanical Procedures: Turing	14
6 Word Problems	17
7 Concluding Remarks	22
References	24
Algorithms: From Al-Khwarizmi to Turing and Beyond	29
Wolfgang Thomas	
1 Prologue	29
2 Some Prehistory: Al-Khwarizmi and Leibniz	30
3 Towards Hilbert's Entscheidungsproblem	33
4 Turing's Breakthrough	35
5 Moves Towards Computer Science	36
6 New Facets of "Algorithm"	37
7 Algorithms as Molecules in Large Organisms	39
8 Returning to Leibnizian Visions?	40
References	41
The Stored-Program Universal Computer: Did Zuse	
Anticipate Turing and von Neumann?	43
B. Jack Copeland and Giovanni Sommaruga	
1 Introduction	44
2 Zuse: A Brief Biography	49
3 Turing, von Neumann, and the Universal Electronic Computer	58

4	A Hierarchy of Programming Paradigms	75
4.1	Paradigm P1	78
4.2	Paradigm P2	78
4.3	Paradigm P3	79
4.4	Paradigm P4	80
4.5	Paradigm P5	81
4.6	Paradigm P6	81
5	Zuse and the Concept of the Universal Machine	85
6	Zuse and the Stored-Program Concept.....	89
7	Concluding Remarks.....	95

Part II Generalizing Turing Computability Theory

Theses for Computation and Recursion on Concrete and Abstract Structures	105
---	------------

Solomon Feferman

1	Introduction	105
2	Recursion and Computation on the Natural Numbers, and the Church-Turing Thesis.....	108
3	Computation on Concrete Structures and “Proofs” of CT	111
4	Proposed Generalizations of Theories of Computation and CT to Abstract Structures; Theses for Algorithms	114
5	Recursion on Abstract Structures	120
	References	124

Generalizing Computability Theory to Abstract Algebras	127
---	------------

J.V. Tucker and J.I. Zucker

1	Introduction	127
2	On Generalizing Computability Theory	129
3	While Computation on Standard Many-Sorted Total Algebras	131
3.1	Basic Concepts: Signatures and Partial Algebras	132
3.2	Syntax and Semantics of Σ -Terms	134
3.3	Adding Counters: N-Standard Signatures and Algebras	134
3.4	Adding Arrays: Algebras A^* of Signature Σ^*	135
4	The While Programming Language.....	135
4.1	While , While ^N and While [*] Computability	136
5	Representations of Semantic Functions; Universality.....	137
5.1	Numbering of Syntax	137
5.2	Representation of States	138
5.3	Representation of Term Evaluation; Term Evaluation Property ...	138
6	Concepts of Semicomputability	140
6.1	Merging Two Procedures; Closure Theorems	141
6.2	Projective While semicomputability and Computability	143
6.3	While [*] Semicomputability	143
6.4	Projective While [*] Semicomputability.....	144
6.5	Computation Trees; Engeler’s Lemma	144

6.6	Projective Equivalence Theorem for <i>While</i> *	145
6.7	Semicomputability and Projective Semicomputability on \mathcal{R}_1	146
7	Computation on Topological Partial Algebras	148
7.1	Abstract Versus Concrete Data Types of Reals; Continuity; Partiality	148
7.2	Examples of Nondeterminism and Many-Valuedness	149
7.3	Partial Algebra of Reals; Completeness for the Abstract Model	152
8	Comparing Models and Generalizing the Church-Turing Thesis	154
8.1	Abstract Models of Computation	154
8.2	Generalizing the Church-Turing Thesis	155
8.3	Concluding Remarks	157
	References	158
	Discrete Transfinite Computation	161
	P.D. Welch	
1	Introduction	161
1.1	Computation in the Limit	163
2	What ITTM's Can Achieve?	166
2.1	Comparisons with Kleene Recursion	169
2.2	Degree Theory and Complexity of ITTM Computations	173
2.3	Truth and Arithmetical Quasi-Inductive Sets	174
3	Variant ITTM Models	175
3.1	Longer Tapes	176
4	Other Transfinite Machines	178
4.1	Infinite Time Register Machines (ITRM's)	178
4.2	Ordinal Register Machines (ORM's)	180
5	Infinite Time Blum-Shub-Smale Machines (IBSSM's)	181
6	Conclusions	183
	References	184
	Semantics-to-Syntax Analyses of Algorithms	187
	Yuri Gurevich	
1	Introduction	188
1.1	Terminology	188
1.2	What's in the Paper?	190
2	Turing	192
2.1	Turing's Species of Algorithms	192
2.2	Turing's Fulcrum	194
2.3	On Turing's Results and Argumentation	194
2.4	Two Critical Quotes	195
3	Kolmogorov	196
3.1	Kolmogorov's Species of Algorithms	196
3.2	Kolmogorov's Fulcrum	196

4	Gandy	197
4.1	Gandy's Species of Algorithms	197
4.2	Gandy's Fulcrum	199
4.3	Comments	199
5	Sequential Algorithms	200
5.1	Motivation	200
5.2	The Species	201
5.3	The Fulcrum	203
6	Final Remarks	204
	References	205
	The Information Content of Typical Reals	207
	George Barmpalias and Andy Lewis-Pye	
1	Introduction	207
1.1	The Algorithmic View of the Continuum	208
1.2	Properties of Degrees	209
1.3	A History of Measure and Category Arguments in the Turing Degrees	210
1.4	Overview	212
2	Typical Degrees and Calibration of Typicality	213
2.1	Large Sets and Typical Reals	213
2.2	Properties of Degrees and Definability	215
2.3	Very Basic Questions Remain Open	216
3	Properties of the Typical Degrees and Their Predecessors	217
3.1	Some Properties of the Typical Degrees	217
3.2	Properties of Typical Degrees are Inherited by the Non-zero Degrees They Compute	219
4	Genericity and Randomness	221
	References	223
	Proof Theoretic Analysis by Iterated Reflection	225
	L.D. Beklemishev	
1	Preliminary Notes	226
2	Introduction	229
3	Constructing Iterated Reflection Principles	235
4	Iterated Π_2 -Reflection and the Fast Growing Hierarchy	241
5	Uniform Reflection Is Not Much Stronger Than Local Reflection	244
6	Extending Conservation Results to Iterated Reflection Principles	249
7	Schmerl's Formula	253
8	Ordinal Analysis of Fragments	255
9	Conclusion and Further Work	259
	Appendix 1	260
	Appendix 2	262
	Appendix 3	265
	References	269

Part III Philosophical Reflections

Alan Turing and the Foundation of Computer Science 273

Juraj Hromkovič

1	“What is Mathematics?” or The Dream of Leibniz	273
2	“Is Mathematics Perfect?” or The Dream of Hilbert	275
3	The Incompleteness Theorem of Gödel as the End of the Dreams of Leibniz and Hilbert	276
4	Alan Turing and the Foundation of Informatics	278
5	Conclusion	281
	References	281

Proving Things About the Informal..... 283

Stewart Shapiro

1	The Received View: No Proof.....	283
2	Vagueness: Does Church’s Thesis Even Have a Truth Value?.....	284
3	Theses	286
4	The Opposition: It Is Possible to Prove These Things	286
5	So What Is It to Prove Something?	288
6	Epistemology.....	291
	References	295

Why Turing’s Thesis Is Not a Thesis 297

Robert Irving Soare

1	Introduction	297
	1.1 Precision of Thought and Terminology	297
	1.2 The Main Points of this Paper	298
2	What is a Thesis?	299
3	The Concept of Effectively Calculable	300
4	Turing’s Paper in 1936.....	301
	4.1 What is a Procedure?.....	301
	4.2 Turing’s Definition of Effectively Calculable Functions	301
	4.3 Gödel’s Reaction to Turing’s Paper	302
	4.4 Church’s Reaction to Turing’s Paper	302
	4.5 Kleene’s Reaction to Turing’s Paper	302
5	Church Rejects Post’s “Working Hypothesis”	303
6	Remarks by Other Authors	304
7	Gandy and Sieg: Proving Turing’s Thesis 1.1	304
	7.1 Gandy	304
	7.2 Sieg	304
8	Feferman: Concrete and Abstract Structures	305
	8.1 Feferman’s Review of the History of the Subject.....	305
	8.2 Feferman on Concrete Structures.....	305
	8.3 Feferman on Kleene’s Naming of CTT	306
9	Gurevich: What is an Algorithm?	306

10 Kripke: On Proving Turing’s Thesis 306

10.1 Kripke on Computable Functions in Soare 306

10.2 Kripke’s Suggested Proof of the Machine Thesis 1.2 307

11 Turing on Definition Versus Theorem 307

References 308

Incomputability Emergent, and Higher Type Computation 311

S. Barry Cooper

1 Introduction 312

2 Incomputability: Unruly Guest at the Computer’s Reception 314

3 Incomputability, Randomness and Higher Type Computation..... 316

4 Embodiment Beyond the Classical Model of Computation..... 320

5 Living in a World of Higher Type Computation..... 323

6 Turing’s Mathematical Host..... 327

References 329