

An Introduction to Kolmogorov Complexity and Its Applications



Ming Li
Paul Vitányi

SECOND EDITION



Springer

An Introduction to Kolmogorov Complexity and Its Applications



Ming Li
Paul Vitányi

SECOND EDITION



Springer

Ming Li Paul Vitányi

AN INTRODUCTION TO
KOLMOGOROV COMPLEXITY
AND ITS APPLICATIONS

Second Edition

With 41 Illustrations



Springer

Ming Li
Department of Computer Science
University of Waterloo
Waterloo, Ontario
N2L 3G1 Canada

Paul Vitányi
Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

Series Editors

David Gries
Fred B. Schneider

Department of Computer Science
Cornell University
Johnson Hall
Ithaca, NY 14853-7501, USA

Library of Congress Cataloging-in-Publication Data

Li, Ming.

An introduction to Kolmogorov complexity and its applications /
Ming Li, Paul Vitányi. — 2nd ed.

p. cm. — (Graduate texts in computer science)

Includes bibliographical references (p. -) and index.

ISBN 0-387-94868-6 (hardcover : alk. paper)

1. Kolmogorov complexity. I. Vitányi, P.M.B. II. Title.

III. Series: Graduate texts in computer science (Springer-Verlag New
York Inc.)

QA267.7.L5 1997

511.3—dc20

96-42357

Printed on acid-free paper.

© 1997, 1993 Springer-Verlag New York, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Production managed by Lesley Poliner; manufacturing supervised by Jacqui Ashri.

Photocomposed copy prepared from the authors' \LaTeX files.

Printed and bound by R.R. Donnelley and Sons, Harrisonburg, VA.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

ISBN 0-387-94868-6 Springer-Verlag New York Berlin Heidelberg SPIN 10550243

Preface to the First Edition

We are to admit no more causes of natural things (as we are told by Newton) than such as are both true and sufficient to explain their appearances. This central theme is basic to the pursuit of science, and goes back to the principle known as Occam's razor: "if presented with a choice between indifferent alternatives, then one ought to select the simplest one." Unconsciously or explicitly, informal applications of this principle in science and mathematics abound.

The conglomerate of different research threads drawing on an objective and absolute form of this approach appears to be part of a single emerging discipline, which will become a major applied science like information theory or probability theory. We aim at providing a unified and comprehensive introduction to the central ideas and applications of this discipline.

Intuitively, the amount of information in a finite string is the size (number of binary digits, or *bits*) of the shortest program that without additional data, computes the string and terminates. A similar definition can be given for infinite strings, but in this case the program produces element after element forever. Thus, a long sequence of 1's such as

$$\underbrace{1111\dots 1}_{10,000 \text{ times}}$$

contains little information because a program of size about $\log 10,000$ bits outputs it:

```
for i := 1 to 10,000
  print 1
```

Likewise, the transcendental number $\pi = 3.1415\dots$, an infinite sequence of seemingly "random" decimal digits, contains but a few bits of information. (There is a short program that produces the consecutive digits of π forever.) Such a definition would appear to make the amount of information in a string (or other object) depend on the particular programming language used.

Fortunately, it can be shown that all reasonable choices of programming languages lead to quantification of the amount of "absolute" information in individual objects that is invariant up to an additive constant. We call this quantity the "Kolmogorov complexity" of the object. If an object contains regularities, then it has a shorter description than itself. We call such an object "compressible."

The application of Kolmogorov complexity takes a variety of forms, for example, using the fact that some strings are extremely compressible; using the compressibility of strings as a selection criterion; using the fact that many strings are not compressible at all; and using the fact that

some strings may be compressed, but that it takes a lot of effort to do so.

The theory dealing with the quantity of information in individual objects goes by names such as “algorithmic information theory,” “Kolmogorov complexity,” “K-complexity,” “Kolmogorov-Chaitin randomness,” “algorithmic complexity,” “stochastic complexity,” “descriptive complexity,” “minimum description length,” “program-size complexity,” and others. Each such name may represent a variation of the basic underlying idea or a different point of departure. The mathematical formulation in each case tends to reflect the particular traditions of the field that gave birth to it, be it probability theory, information theory, theory of computing, statistics, or artificial intelligence.

This raises the question about the proper name for the area. Although there is a good case to be made for each of the alternatives listed above, and a name like “Solomonoff-Kolmogorov-Chaitin complexity” would give proper credit to the inventors, we regard “Kolmogorov complexity” as well entrenched and commonly understood, and we shall use it hereafter.

The mathematical theory of Kolmogorov complexity contains deep and sophisticated mathematics. Yet one needs to know only a small amount of this mathematics to apply the notions fruitfully in widely divergent areas, from sorting algorithms to combinatorial theory, and from inductive reasoning and machine learning to dissipationless computing.

Formal knowledge of basic principles does not necessarily imply the wherewithal to apply it, perhaps especially so in the case of Kolmogorov complexity. It is our purpose to develop the theory in detail and outline a wide range of illustrative applications. In fact, while the pure theory of the subject will have its appeal to the select few, the surprisingly large field of its applications will, we hope, delight the multitude.

The mathematical theory of Kolmogorov complexity is treated in Chapters 2, 3, and 4; the applications are treated in Chapters 5 through 8. Chapter 1 can be skipped by the reader who wants to proceed immediately to the technicalities. Section 1.1 is meant as a leisurely, informal introduction and peek at the contents of the book. The remainder of Chapter 1 is a compilation of material on diverse notations and disciplines drawn upon.

We define mathematical notions and establish uniform notation to be used throughout. In some cases we choose nonstandard notation since the standard one is homonymous. For instance, the notions “absolute value,” “cardinality of a set,” and “length of a string,” are commonly denoted in the same way as $|\cdot|$. We choose distinguishing notations $|\cdot|$, $d(\cdot)$, and $l(\cdot)$, respectively.

Briefly, we review the basic elements of computability theory and probability theory that are required. Finally, in order to place the subject in the appropriate historical and conceptual context we trace the main roots of Kolmogorov complexity.

This way the stage is set for Chapters 2 and 3, where we introduce the notion of optimal effective descriptions of objects. The length of such a description (or the number of bits of information in it) is its Kolmogorov complexity. We treat all aspects of the elementary mathematical theory of Kolmogorov complexity. This body of knowledge may be called *algorithmic complexity theory*. The theory of Martin-Löf tests for randomness of finite objects and infinite sequences is inextricably intertwined with the theory of Kolmogorov complexity and is completely treated. We also investigate the statistical properties of finite strings with high Kolmogorov complexity. Both of these topics are eminently useful in the applications part of the book. We also investigate the recursion-theoretic properties of Kolmogorov complexity (relations with Gödel's incompleteness result), and the Kolmogorov complexity version of information theory, which we may call "algorithmic information theory" or "absolute information theory."

The treatment of *algorithmic probability theory* in Chapter 4 presupposes Sections 1.6, 1.11.2, and Chapter 3 (at least Sections 3.1 through 3.4). Just as Chapters 2 and 3 deal with the optimal effective description length of objects, we now turn to optimal (greatest) effective probability of objects. We treat the elementary mathematical theory in detail. Subsequently, we develop the theory of effective randomness tests under arbitrary recursive distributions for both finite and infinite sequences. This leads to several classes of randomness tests, each of which has a universal randomness test. This is the basis for the treatment of a mathematical theory of inductive reasoning in Chapter 5 and the theory of algorithmic entropy in Chapter 8.

Chapter 5 develops a general theory of inductive reasoning and applies the developed notions to particular problems of inductive inference, prediction, mistake bounds, computational learning theory, and minimum description length induction in statistics. This development can be viewed both as a resolution of certain problems in philosophy about the concept and feasibility of induction (and the ambiguous notion of "Occam's razor"), as well as a mathematical theory underlying computational machine learning and statistical reasoning.

Chapter 6 introduces the incompressibility method. Its utility is demonstrated in a plethora of examples of proving mathematical and computational results. Examples include combinatorial properties, the time complexity of computations, the average-case analysis of algorithms such as Heapsort, language recognition, string matching, "pumping lemmas"

in formal language theory, lower bounds in parallel computation, and Turing machine complexity. Chapter 6 assumes only the most basic notions and facts of Sections 2.1, 2.2, 3.1, 3.3.

Some parts of the treatment of resource-bounded Kolmogorov complexity and its many applications in computational complexity theory in Chapter 7 presuppose familiarity with a first-year graduate theory course in computer science or basic understanding of the material in Section 1.7.4. Sections 7.5 and 7.7 on “universal optimal search” and “logical depth” only require material covered in this book. The section on “logical depth” is technical and can be viewed as a mathematical basis with which to study the emergence of life-like phenomena—thus forming a bridge to Chapter 8, which deals with applications of Kolmogorov complexity to relations between physics and computation.

Chapter 8 presupposes parts of Chapters 2, 3, 4, the basics of information theory as given in Section 1.11, and some familiarity with college physics. It treats physical theories like dissipationless reversible computing, information distance and picture similarity, thermodynamics of computation, statistical thermodynamics, entropy, and chaos from a Kolmogorov complexity point of view. At the end of the book there is a comprehensive listing of the literature on theory and applications of Kolmogorov complexity and a detailed index.

How to Use This Book

The technical content of this book consists of four layers. The main text is the first layer. The second layer consists of examples in the main text. These elaborate the theory developed from the main theorems. The third layer consists of nonindented, smaller-font paragraphs interspersed with the main text. The purpose of such paragraphs is to have an explanatory aside, to raise some technical issues that are important but would distract attention from the main narrative, or to point to alternative or related technical issues. Much of the technical content of the literature on Kolmogorov complexity and related issues appears in the fourth layer, the exercises. When the idea behind a nontrivial exercise is not our own, we have tried to give credit to the person who originated the idea. Corresponding references to the literature are usually given in comments to an exercise or in the historical section of that chapter.

Starred sections are not really required for the understanding of the sequel and should be omitted at first reading. The application sections are not starred. The exercises are grouped together at the end of main sections. Each group relates to the material in between it and the previous group. Each chapter is concluded by an extensive historical section with full references. For convenience, all references in the text to the Kolmogorov complexity literature and other relevant literature are given in full where they occur. The book concludes with a References section intended as a separate exhaustive listing of the literature restricted to the

theory and the direct applications of Kolmogorov complexity. There are reference items that do not occur in the text and text references that do not occur in the References. We added a very detailed index combining the index to notation, the name index, and the concept index. The page number where a notion is defined first is printed in boldface. The initial part of the Index is an index to notation. Names as “J. von Neumann” are indexed European style “Neumann, J. von.”

The exercises are sometimes trivial, sometimes genuine exercises, but more often compilations of entire research papers or even well-known open problems. There are good arguments to include both: the easy and real exercises to let the student exercise his comprehension of the material in the main text; the contents of research papers to have a comprehensive coverage of the field in a small number of pages; and research problems to show where the field is (or could be) heading. To save the reader the problem of having to determine which is which: “I found this simple exercise in number theory that looked like Pythagoras’s Theorem. Seems difficult.” “Oh, that is Fermat’s Last Theorem; it was unsolved for three hundred and fifty years...,” we have adopted the system of *rating numbers* used by D.E. Knuth [*The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley, 1973 (2nd Edition), pp. xvii–xix]. The interpretation is as follows:

- 00 A very easy exercise that can be answered immediately, from the top of your head, if the material in the text is understood.
- 10 A simple problem to exercise understanding of the text. Use fifteen minutes to think, and possibly pencil and paper.
- 20 An average problem to test basic understanding of the text and may take one or two hours to answer completely.
- 30 A moderately difficult or complex problem taking perhaps several hours to a day to solve satisfactorily.
- 40 A quite difficult or lengthy problem, suitable for a term project, often a significant result in the research literature. We would expect a very bright student or researcher to be able to solve the problem in a reasonable amount of time, but the solution is not trivial.
- 50 A research problem that, to the authors’ knowledge, is open at the time of writing. If the reader has found a solution, he is urged to write it up for publication; furthermore, the authors of this book would appreciate hearing about the solution as soon as possible (provided it is correct).

This scale is “logarithmic”: a problem of rating 17 is a bit simpler than average. Problems with rating 50, subsequently solved, will appear in

a next edition of this book with rating 45. Rates are sometimes based on the use of solutions to earlier problems. The rating of an exercise is based on that of its most difficult item, but not on the number of items. Assigning accurate rating numbers is impossible—one man’s meat is another man’s poison—and our rating will differ from ratings by others.

An orthogonal rating “M” implies that the problem involves more mathematical concepts and motivation than is necessary for someone who is primarily interested in Kolmogorov complexity and applications. Exercises marked “HM” require the use of calculus or other higher mathematics not developed in this book. Some exercises are marked with “•”; and these are especially instructive or useful. Exercises marked “O” are problems that are, to our knowledge, unsolved at the time of writing. The rating of such exercises is based on our estimate of the difficulty of solving them. Obviously, such an estimate may be totally wrong.

Solutions to exercises, or references to the literature where such solutions can be found, appear in the “Comments” paragraph at the end of each exercise. Nobody is expected to be able to solve all exercises.

The material presented in this book draws on work that until now was available only in the form of advanced research publications, possibly not translated into English, or was unpublished. A large portion of the material is new. The book is appropriate for either a one- or a two-semester introductory course in departments of mathematics, computer science, physics, probability theory and statistics, artificial intelligence, cognitive science, and philosophy. Outlines of possible one-semester courses that can be taught using this book are presented below.

Fortunately, the field of descriptive complexity is fairly young and the basics can still be comprehensively covered. We have tried to the best of our abilities to read, digest, and verify the literature on the topics covered in this book. We have taken pains to establish correctly the history of the main ideas involved. We apologize to those who have been unintentionally slighted in the historical sections. Many people have generously and selflessly contributed to verify and correct drafts of this book. We thank them below and apologize to those we forgot. In a work of this scope and size there are bound to remain factual errors and incorrect attributions. We greatly appreciate notification of errors or any other comments the reader may have, preferably by email to kolmogorov@cwi.nl, in order that future editions may be corrected.

Acknowledgments

We thank Greg Chaitin, Péter Gács, Leonid Levin, and Ray Solomonoff for taking the time to tell us about the early history of our subject and for introducing us to many of its applications. Juris Hartmanis and Joel Seiferas initiated us into Kolmogorov complexity in various ways.

Many people gave substantial suggestions for examples and exercises, or pointed out errors in a draft version. Apart from the people already mentioned, these are, in alphabetical order, Eric Allender, Charles Bennett, Piotr Berman, Robert Black, Ron Book, Dany Breslauer, Harry Buhrman, Peter van Emde Boas, William Gasarch, Joe Halpern, Jan Heering, G. Hotz, Tao Jiang, Max Kanovich, Danny Krizanc, Evangelos Kranakis, Michiel van Lambalgen, Luc Longpré, Donald Loveland, Albert Meyer, Lambert Meertens, Ian Munro, Pekka Orponen, Ramamohan Paturi, Jorma Rissanen, Jeff Shallit, A.Kh. Shen', J. Laurie Snell, Th. Tsantilas, John Tromp, Vladimir Uspensky, N.K. Vereshchagin, Osamu Watanabe, and Yaacov Yesha. Apart from them, we thank the many students and colleagues who contributed to this book.

We especially thank Péter Gács for the extraordinary kindness of reading and commenting in detail on the entire manuscript, including the exercises. His expert advice and deep insight saved us from many pitfalls and misunderstandings. Piergiorgio Odifreddi carefully checked and commented on the first three chapters. Parts of the book have been tested in one-semester courses and seminars at the University of Amsterdam in 1988 and 1989, the University of Waterloo in 1989, Dartmouth College in 1990, the Universitat Politècnica de Catalunya in Barcelona in 1991/1992, the University of California at Santa Barbara, Johns Hopkins University, and Boston University in 1992/1993.

This document has been prepared using the \LaTeX system. We thank Donald Knuth for \TeX , Leslie Lamport for \LaTeX , and Jan van der Steen at CWI for online help. Some figures were prepared by John Tromp using the `xpic` program.

The London Mathematical Society kindly gave permission to reproduce a long extract by A.M. Turing. The Indian Statistical Institute, through the editor of *Sankhyā*, kindly gave permission to quote A.N. Kolmogorov.

We gratefully acknowledge the financial support by NSF Grant DCR-8606366, ONR Grant N00014-85-k-0445, ARO Grant DAAL03-86-K-0171, the Natural Sciences and Engineering Research Council of Canada through operating grants OGP-0036747, OGP-046506, and International Scientific Exchange Awards ISE0046203, ISE0125663, and NWO Grant NF 62-376. The book was conceived in late Spring 1986 in the Valley of the Moon in Sonoma County, California. The actual writing lasted on and off from autumn 1987 until summer 1993.

One of us [PV] gives very special thanks to his lovely wife Pauline for insisting from the outset on the significance of this enterprise. The Aiken Computation Laboratory of Harvard University, Cambridge, Massachusetts, USA; the Computer Science Department of York University, Ontario, Canada; the Computer Science Department of the University

of Waterloo, Ontario, Canada; and CWI, Amsterdam, the Netherlands provided the working environments in which this book could be written.

Preface to the Second Edition

When this book was conceived ten years ago, few scientists realized the width of scope and the power for applicability of the central ideas. Partially because of the enthusiastic reception of the first edition, open problems have been solved and new applications have been developed. We have added new material on the relation between data compression and minimum description length induction, computational learning, and universal prediction; circuit theory; distributed algorithmics; instance complexity; CD compression; computational complexity; Kolmogorov random graphs; shortest encoding of routing tables in communication networks; computable universal distributions; average case properties; the equality of statistical entropy and expected Kolmogorov complexity; and so on. Apart from being used by researchers and as reference work, the book is now commonly used for graduate courses and seminars. In recognition of this fact, the second edition has been produced in text-book style. We have preserved as much as possible the ordering of the material as it was in the first edition. The many exercises bunched together at the ends of some chapters have been moved to the appropriate sections. The comprehensive bibliography on Kolmogorov complexity at the end of the book has been updated, as have the “History and References” sections of the chapters. Many readers were kind enough to express their appreciation for the first edition and to send notification of typos, errors, and comments. Their number is too large to thank them individually, so we thank them all collectively.

Outlines of One-Semester Courses

We have mapped out a number of one-semester courses on a variety of topics. These topics range from basic courses in theory and applications to special interest courses in learning theory, randomness, or information theory using the Kolmogorov complexity approach.

PREREQUISITES: Sections 1.1, 1.2, 1.7 (except Section 1.7.4).

I. Course on Basic Algorithmic Complexity and Applications

TYPE OF COMPLEXITY	THEORY	APPLICATIONS
plain complexity	2.1, 2.2, 2.3	4.4, Chapter 6
prefix complexity	1.11.2, 3.1 3.3, 3.4	5.1, 5.1.3, 5.2, 5.5 8.2, 8.3 8
resource-bounded complexity	7.1, 7.5, 7.7	7.2, 7.3, 7.6, 7.7

II. Course on Algorithmic Complexity

TYPE OF COMPLEXITY	BASICS	RANDOMNESS	ALGORITHMIC PROPERTIES
state \times symbol	1.12		
plain complexity	2.1, 2.2, 2.3	2.4	2.7
prefix complexity	1.11.2, 3.1 3.3, 3.4	3.5	3.7, 3.8
monotone complexity	4.5 (intro)	4.5.4	

III. Course on Algorithmic Randomness

RANDOMNESS TESTS ACCORDING TO	COMPLEXITY USED	FINITE STRINGS	INFINITE SEQUENCES
von Mises			1.9
Martin-Löf	2.1, 2.2	2.4	2.5
prefix complexity	1.11.2, 3.1, 3.3, 3.4	3.5	3.6, 4.5.6
general discrete	1.6 (intro), 4.3.1	4.3	
general continuous	1.6 (intro), 4.5 (intro), 4.5.1		4.5

IV. Course on Algorithmic Information Theory and Applications

TYPE OF COMPLEXITY USED	BASICS	ENTROPY	SYMMETRY OF INFORMATION
classical information theory	1.11	1.11	1.11
plain complexity	2.1, 2.2	2.8	2.8
prefix complexity	3.1, 3.3, 3.4		3.8, 3.9.1
resource-bounded	7.1		Exercises 7.1.11 7.1.12
applications		8.1, 8.4, 8.5	Theorem 7.2.6 Exercise 6.10.15

V. Course on Algorithmic Probability Theory, Learning, Inference and Prediction

THEORY	BASICS	UNIVERSAL DISTRIBUTION	APPLICATIONS TO INFERENCE
classical probability	1.6, 1.11.2		1.6
algorithmic complexity	2.1, 2.2, 2.3 3.1, 3.3, 3.4		8
algorithmic discrete probability	4.2, 4.1 4.3 (intro)	4.3.1, 4.3.2 4.3.3, 4.3.4, 4.3.6	
algorithmic contin. probability	4.5 (intro)	4.5.1, 4.5.2 4.5.4, 4.5.8	5.2
Solomonoff's inductive inference	5.1, 5.1.3, 5.2 5.4, 8	5.3, 5.4.3, 5.5 5.5.8	5.1.3

- VI. Course on the Incompressibility Method** Chapter 2 (Sections 2.1, 2.2, 2.4, 2.6, 2.8), Chapter 3 (mainly Sections 3.1, 3.3), Section 4.4, and Chapters 6 and 7. The course covers the basics of the theory with many applications in proving upper and lower bounds on the running time and space use of algorithms.
- VII. Course on Randomness, Information, and Physics** Course III and Chapter 8. In physics the applications of Kolmogorov complexity include theoretical illuminations of foundational issues. For example, the approximate equality of statistical entropy and expected Kolmogorov complexity, the nature of “entropy,” a fundamental resolution of the “Maxwell’s Demon” paradox. However, also more concrete applications like “information distance” and “thermodynamics of computation” are covered.

Contents

Preface to the First Edition	v
How to Use This Book	viii
Acknowledgments	x
Preface to the Second Edition	xii
Outlines of One-Semester Courses	xii
List of Figures	xix
1 Preliminaries	1
1.1 A Brief Introduction	1
1.2 Prerequisites and Notation	6
1.3 Numbers and Combinatorics	8
1.4 Binary Strings	12
1.5 Asymptotic Notation	15
1.6 Basics of Probability Theory	18
1.7 Basics of Computability Theory	24
1.8 The Roots of Kolmogorov Complexity	47
1.9 Randomness	49
1.10 Prediction and Probability	59
1.11 Information Theory and Coding	65
1.12 State \times Symbol Complexity	84
1.13 History and References	86

2	Algorithmic Complexity	93
2.1	The Invariance Theorem	96
2.2	Incompressibility	108
2.3	C as an Integer Function	119
2.4	Random Finite Sequences	127
2.5	*Random Infinite Sequences	136
2.6	Statistical Properties of Finite Sequences	158
2.7	Algorithmic Properties of C	167
2.8	Algorithmic Information Theory	179
2.9	History and References	185
3	Algorithmic Prefix Complexity	189
3.1	The Invariance Theorem	192
3.2	*Sizes of the Constants	197
3.3	Incompressibility	202
3.4	K as an Integer Function	206
3.5	Random Finite Sequences	208
3.6	*Random Infinite Sequences	211
3.7	Algorithmic Properties of K	224
3.8	*Complexity of Complexity	226
3.9	*Symmetry of Algorithmic Information	229
3.10	History and References	237
4	Algorithmic Probability	239
4.1	Enumerable Functions Revisited	240
4.2	Nonclassical Notation of Measures	242
4.3	Discrete Sample Space	245
4.4	Universal Average-Case Complexity	268
4.5	Continuous Sample Space	272
4.6	Universal Average-Case Complexity, Continued	307
4.7	History and References	307

5	Inductive Reasoning	315
5.1	Introduction	315
5.2	Solomonoff's Theory of Prediction	324
5.3	Universal Recursion Induction	335
5.4	Simple Pac-Learning	339
5.5	Hypothesis Identification by Minimum Description Length	351
5.6	History and References	372
6	The Incompressibility Method	379
6.1	Three Examples	380
6.2	High-Probability Properties	385
6.3	Combinatorics	389
6.4	Kolmogorov Random Graphs	396
6.5	Compact Routing	404
6.6	Average-Case Complexity of Heapsort	412
6.7	Longest Common Subsequence	417
6.8	Formal Language Theory	420
6.9	Online CFL Recognition	427
6.10	Turing Machine Time Complexity	432
6.11	Parallel Computation	445
6.12	Switching Lemma	449
6.13	History and References	452
7	Resource-Bounded Complexity	459
7.1	Mathematical Theory	460
7.2	Language Compression	476
7.3	Computational Complexity	488
7.4	Instance Complexity	495
7.5	Kt Complexity and Universal Optimal Search	502
7.6	Time-Limited Universal Distributions	506
7.7	Logical Depth	510
7.8	History and References	516

8	Physics, Information, and Computation	521
8.1	Algorithmic Complexity and Shannon's Entropy	522
8.2	Reversible Computation	528
8.3	Information Distance	537
8.4	Thermodynamics	554
8.5	Entropy Revisited	565
8.6	Compression in Nature	583
8.7	History and References	586
	References	591
	Index	618

List of Figures

1.1	Turing machine	28
1.2	Inferred probability for increasing n	60
1.3	Binary tree for $E(1) = 0$, $E(2) = 10$, $E(3) = 110$, $E(4) = 111$	72
1.4	Binary tree for $E(1) = 0$, $E(2) = 01$, $E(3) = 011$, $E(4) = 0111$	73
2.1	The graph of the integer function $C(x)$	121
2.2	The graph of the integer function $C(x l(x))$	123
2.3	Test of Example 2.4.1	128
2.4	Complexity oscillations of initial segments of infinite high-complexity sequences	139
2.5	Three notions of “chaotic” infinite sequences	148
3.1	The 425-bit universal combinator U' in pixels	201
3.2	The graphs of $K(x)$ and $K(x l(x))$	207
3.3	Complexity oscillations of a typical random sequence ω	215
3.4	K -complexity criteria for randomness of infinite sequences	215
3.5	Complexity oscillations of Ω	216
4.1	Graph of $\mathbf{m}(x)$ with lower bound $1/x \cdot \log x \cdot \log \log x \cdots$	249

4.2	Relations between five complexities	285
5.1	Trivial consistent automaton	317
5.2	Smallest consistent automaton	317
5.3	Sample data set	365
5.4	Imperfect decision tree	366
5.5	Perfect decision tree	367
6.1	Single-tape Turing machine	381
6.2	The two possible nni's on (u, v) : swap $B \leftrightarrow C$ or $B \leftrightarrow D$	416
6.3	The nni distance between (i) and (ii) is 2	416
6.4	Multitape Turing machine	428
8.1	Reversible Boolean gates	530
8.2	Implementing reversible AND gate and NOT gate	531
8.3	Controlling billiard ball movements	532
8.4	A billiard ball computer	533
8.5	Combining irreversible computations of y from x and x from y to achieve a reversible computation of y from x	543
8.6	Reversible execution of concatenated programs for $(y x)$ and $(z y)$ to transform x into z	545
8.7	Carnot cycle	555
8.8	Heat engine	556
8.9	State space	559
8.10	Atomic spin in CuO_2 at low temperature	562
8.11	Regular "up" and "down" spins	563
8.12	Algorithmic entropy: left a random micro state, right a regular micro state	567
8.13	Szilard engine	568
8.14	Adiabatic demagnetization to achieve low temperature	583
8.15	The maze: a binary tree constructed from matches	584
8.16	Time required for <i>Formica sanguinea</i> scouts to transmit information about the direction to the syrup to the for- ager ants	585