Texts and Monographs in Computer Science

Editor
David Gries

Advisory Board
F. L. Bauer
K. S. Fu
J. J. Horning
R. Reddy
D. C. Tsichritzis
W. M. Waite

The AKM Series in Theoretical Computer Science

A Subseries of Texts and Monographs in Computer Science

A Basis for Theoretical Computer Science
by M. A. Arbib, A. J. Kfoury, and R. N. Moll

A Programming Approach to Computability
by A. J. Kfoury, R. N. Moll, and M. A. Arbib

An Introduction to Formal Language Theory
by R. N. Moll, M. A. Arbib, and A. J. Kfoury

A Programming Approach to Computability

A. J. Kfoury
Robert N. Moll
Michael A. Arbib

Springer-Verlag
New York Heidelberg Berlin

A. J. Kfoury
Department of Mathematics
Boston University
Boston, MA 02215
U.S.A.

Robert N. Moll
Department of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003
U.S.A.

Michael A. Arbib
Department of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003
U.S.A.

Series Editor

David Gries
Department of Computer Science
Cornell University
Upson Hall
Ithaca, NY 14853
U.S.A.

On the front cover is a likeness (based on a postcard kindly supplied by Professor S. C. Kleene) of the ninth century mathematician Abu Jafar Muhammad Ibn Musa Al-Khowarizmi. The word "algorithm" is derived from the last part of his name.

With 36 Figures

Library of Congress Cataloging in Publication Data
Kfoury, A. J.
A programming approach to computability.
(The AKM series in theoretical computer science)
(Texts and monographs in computer science)
"February 1982."
Bibliography: p.
Includes index.
1. Computable functions. 2. Programming
(Electronic computers) I. Moll, Robert N.
II. Arbib, Michael A. III. Title. IV. Series.
V. Series: Texts and monographs in computer science.
QA9.59.K46 1982 519.4 82-10488

All rights reserved.
No part of this book may be translated or reproduced in any form without written permission from Springer-Verlag, 175 Fifth Avenue, New York, New York 10010, U.S.A.

© 1982 by Springer-Verlag New York Inc.
Softcover reprint of the hardcover 1st edition 1982

9 8 7 6 5 4 3 2 1

ISBN-13:978-1-4612-5751-6 e-ISBN-13:978-1-4612-5749-3
DOI:10.1007/978-1-4612-5749-3

Preface

Computability theory is at the heart of theoretical computer science. Yet, ironically, many of its basic results were discovered by mathematical logicians prior to the development of the first stored-program computer. As a result, many texts on computability theory strike today's computer science students as far removed from their concerns. To remedy this, we base our approach to computability on the language of **while**-programs, a lean subset of PASCAL, and postpone consideration of such classic models as Turing machines, string-rewriting systems, and μ-recursive functions till the final chapter. Moreover, we balance the presentation of unsolvability results such as the unsolvability of the Halting Problem with a presentation of the positive results of modern programming methodology, including the use of proof rules, and the denotational semantics of programs.

Computer science seeks to provide a scientific basis for the study of information processing, the solution of problems by algorithms, and the design and programming of computers. The last 40 years have seen increasing sophistication in the science, in the microelectronics which has made machines of staggering complexity economically feasible, in the advances in programming methodology which allow immense programs to be designed with increasing speed and reduced error, and in the development of mathematical techniques to allow the rigorous specification of program, process, and machine. The present volume is one of a series, the AKM Series in Theoretical Computer Science, designed to make key mathematical developments in computer science readily accessible to undergraduate and beginning graduate students. The book is essentially self-contained—what little background is required may be found in the AKM volume *A Basis for Theoretical Computer Science*.

The book is organized as follows. Chapter 1 provides a preview of the techniques to be used throughout the book in determining whether or not a function is computable by some algorithm.

Chapters 2 and 3 establish our framework for the study of computability, using the lean PASCAL subset we call "**while**-programs". In these two chapters we take a close look at the intuitive idea of an algorithm as formulated in this simple programming language. We also introduce the idea of a universal program or interpreter which can simulate any program P when given P's data together with an encoding of P as an additional input.

In Chapter 4 we develop basic techniques of computability theory, which we then use to examine the computational status of various natural problems.

Chapter 5 provides an introduction to program correctness and to alternative theories of semantics of programming languages.

In Chapter 6 we consider the role of self-reference in computability theory as embodied in a result known as the Recursion Theorem. We also study conditions under which the theory can be made model independent.

In Chapters 7 and 8 our emphasis changes from functions to sets. Recursive sets, whose characteristic functions are computable, are compared with recursively enumerable sets whose elements are listable by algorithms. As we shall see, these two concepts do not coincide.

Finally, in Chapter 9 we examine alternative formulations of computability theory, and we show how one of these formulations (the Turing machine model) can be used to embed the results of computability theory in a large variety of other symbol-processing systems, including formal languages.

More detailed outlines may be found at the beginning of each chapter.

In keeping with the style of the AKM series, we have resisted the temptation to include more material than could possibly be covered in a one-semester course. Nevertheless, in moderating the pace for a single term's work, the instructor may find it appropriate to omit material from one or more of Sections 5.3, 6.2, 7.3, 8.2, 9.2, and 9.3 (the starred sections in the table of contents).

The book grew out of our teaching classes at the University of Massachusetts at Amherst and at Boston University over several years. We owe special thanks to Ernest Manes for letting us use material from the forthcoming text *Formal Semantics of Programming Languages* by M. A. Arbib and E. Manes in the preparation of Section 5.1. We thank our students for helping us integrate computability theory into the "world view" of modern computer science, and we thank Susan Parker for all her efforts in typing the manuscript.

February 1982

A. J. Kfoury
R. N. Moll
M. A. Arbib

Contents

*These sections may be omitted with no loss of continuity.