DIE GRUNDLEHREN DER

MATHEMATISCHEN WISSENSCHAFTEN

IN EINZELDARSTELLUNGEN MIT BESONDERER BERÜCKSICHTIGUNG DER ANWENDUNGSGEBIETE

BAND 127

SPRINGER-VERLAG BERLIN HEIDELBERG GMBH 1965

ENUMERABILITY · DECIDABILITY COMPUTABILITY

AN INTRODUCTION TO THE THEORY OF RECURSIVE FUNCTIONS

BY

HANS HERMES
PROFESSOR OF MATHEMATICAL LOGIC
AND FOUNDATIONS OF MATHEMATICS, UNIVERSITY OF MÜNSTER

TRANSLATED BY

G. T. HERMAN AND O. PLASSMANN

1965

SPRINGER-VERLAG BERLIN HEIDELBERG GMBH

Library of Congress Catalog Card Number 65-12556

ISBN 978-3-662-11688-3 ISBN 978-3-662-11686-9 (eBook)
DOI 10.1007/978-3-662-11686-9

ORIGINALLY PUBLISHED BY SPRINGER-VERLAG · BERLIN · HEIDELBERG · NEW YORK IN 1965
SOFTCOVER REPRINT OF THE HARDCOVER 1ST EDITION 1965

PREFACE TO THE ORIGINAL EDITION

The task of developing algorithms to solve problems has always been considered by mathematicians to be an especially interesting and important one. Normally an algorithm is applicable only to a narrowly limited group of problems. Such is for instance the Euclidean algorithm, which determines the greatest common divisor of two numbers, or the well-known procedure which is used to obtain the square root of a natural number in decimal notation. The more important these *special* algorithms are, all the more desirable it seems to have algorithms of a greater range of applicability at one's disposal. Throughout the centuries, attempts to provide algorithms applicable as widely as possible were rather unsuccessful. It was only in the second half of the last century that the first appreciable advance took place. Namely, an important group of the inferences of the logic of predicates was given in the form of a calculus. (Here the Boolean algebra played an essential pioneer role.) One could now perhaps have conjectured that *all* mathematical problems are solvable by algorithms. However, well-known, yet unsolved problems (problems like the word problem of group theory or Hilbert's tenth problem, which considers the question of solvability of Diophantine equations) were warnings to be careful. Nevertheless, the impulse had been given to search for the essence of algorithms. *Leibniz* already had inquired into this problem, but without success. The mathematicians of our century however, experienced in dealing with abstract problems and especially in operating with formal languages, were successful. About 1936 several suggestions to make precise the concept of algorithm and related concepts were made at almost the same time (Church's thesis). Although these suggestions (the number of which has been increased since) often originated from widely different initial considerations, they have been proved to be equivalent. The motivations for these precise replacements, the fact of their equivalence, and the experimental fact that all algorithms which have occurred in mathematics so far are cases of these precise concepts (at least if we concentrate on their essential nucleus) have convinced nearly all research workers of this field that these precise replacements are adequate interpretations of the at first intuitively given concept of algorithm.

Once we have accepted a precise replacement of the concept of algorithm, it becomes possible to attempt the problem whether there exist well-defined collections of problems which cannot be handled by algorithms, and if that is the case, to give concrete cases of this kind. Many such investigations were carried out during the last few decades. The undecidability of arithmetic and other mathematical theories was shown, further the unsolvability of the word problem of group theory. Many mathematicians consider these results and the theory on which they are based to be the most characteristic achievements of mathematics in the first half of the twentieth century.

If we grant the legitimacy of the suggested precise replacements of the concept of algorithm and related concepts, then we can say that the mathematicians have shown by strictly mathematical methods that there exist mathematical problems which cannot be dealt with by the methods of calculating mathematics. In view of the important role which mathematics plays today in our conception of the world this fact is of great philosophical interest. *Post* speaks of a natural law about the "limitations of the mathematicizing power of Homo Sapiens". Here we also find a starting point for the discussion of the question, what the actual creative activity of the mathematician consists in.

In this book we shall give an introduction to the theory of algorithms. First of all we shall try to convince the reader that the given precise replacements represent the intuitive concepts adequately. The best way to do this is to use one of these precise replacements, namely that of the Turing machine, as a starting point. We shall deal with the most important constructive concepts, like the concepts of computable function, of decidable property, and of set generated by a system of rules, using Turing machines as a basis. We shall discuss several other precise replacements of the concept of algorithm (e.g. μ-recursiveness, recursiveness) and prove their equivalence. As applications we shall, among others, prove the undecidability of the predicate calculus and the incompleteness of arithmetic; further, the most important preliminary step for the proof of the unsolvability of the word problem of group theory, i.e. the proof of the unsolvability for the word problem for Thue-systems, will be shown.

The theory will be developed from the point of view of the classical logic. This will be especially noticeable by the application of the classical existence operator, e.g. in the definition of computability. We call a function computable if *there exists* an algorithm to find the values for arbitrarily given arguments. — However, this will be made especially clear in all cases where proofs are carried out constructively.

In contrast to many publications on the subject we shall be careful to distinguish between the formulae of a symbolic language and the things denoted by them, in any case in the basic definitions.

At the end of several paragraphs there are a few, mainly easy, exercises, which the reader should attempt to solve.

It corresponds to the introductory character of this book that not all results of the subject will be discussed. However, the references given at the end of most paragraphs will inform the reader of the latest developments in this theory. Besides, we would like, once and for all, to refer the reader to a basic work, namely the "Introduction to Metamathematics" by *S. C. Kleene* (Amsterdam 1952), and also to the papers published in *The Journal of Symbolic Logic* (1936—). Another book in which the theory is based on Turing machines is *M. Davis*: "Computability and Unsolvability" (New York 1958).

The present book is based upon the lectures which the author has given on this subject regularly since 1949. In 1955 a manuscript of the lecture course was published by Verlag Aschendorff (Münster) under the title "Entscheidungsprobleme in Mathematik und Logik".

I would like to express my gratitude to Dr. H. Kiesow and Dr. W. Oberschelp for the valuable assistance in preparing the manuscript. In this respect I am also grateful to Miss T. Hessling, Miss E. Herting, Mr. D. Titgemeyer and Mr. K. Hornung.

Münster i. W., Spring 1960 H. HERMES

PREFACE TO THE ENGLISH TRANSLATION

In the translated edition the text follows the original very closely. A few alterations were made and some errors corrected.

The translation was taken care of by Messrs. Gabor T. Herman, B. Sc., and O. Plassmann. I would like to express my gratitude to the translators, especially to Mr. Herman, who critically examined the whole text and suggested corrections and improvements at several places.

Münster i. W., Spring 1964 H. HERMES

CONTENTS

Page

Chapter 1. *Introductory Reflections on Algorithms* 1

§ 1. The Concept of Algorithm 1
§ 2. The Fundamental Concepts of the Theory of Constructivity 9
§ 3. The Concept of Turing Machine as an Exact Mathematical Substitute for the Concept of Algorithm 17
§ 4. Historical Remarks 26

Chapter 2. *Turing Machines* 31

§ 5. Definition of Turing Machines 31
§ 6. Precise Definition of Constructive Concepts by means of Turing Machines 35
§ 7. Combination of Turing Machines 44
§ 8. Special Turing Machines 48
§ 9. Examples of Turing-Computability and Turing-Decidability 55

Chapter 3. *μ-Recursive Functions* 59

§ 10. Primitive Recursive Functions 59
§ 11. Primitive Recursive Predicates 66
§ 12. The μ-Operator 74
§ 13. Example of a Computable Function which is not Primitive Recursive 82
§ 14. μ-Recursive Functions and Predicates 88

Chapter 4. *The Equivalence of Turing-Computability and μ-Recursiveness* . . 93

§ 15. Survey. Standard Turing-Computability 94
§ 16. The Turing-Computability of μ-Recursive Functions 98
§ 17. Gödel Numbering of Turing Machines 103
§ 18. The μ-Recursiveness of Turing-Computable Functions. Kleene's Normal Form 108

Chapter 5. *Recursive Functions* 113

§ 19. Definition of Recursive Functions 113
§ 20. The Recursiveness of μ-Recursive Functions 118
§ 21. The μ-Recursiveness of Recursive Functions 130

Chapter 6. *Undecidable Predicates* 141

§ 22. Simple Undecidable Predicates 141
§ 23. The Unsolvability of the Word Problem for Semi-Thue Systems and Thue Systems 145
§ 24. The Predicate Calculus 155
§ 25. The Undecidability of the Predicate Calculus 163
§ 26. The Incompleteness of the Predicate Calculus of the Second Order . 171
§ 27. The Undecidability and Incompleteness of Arithmetic 175

Page

Chapter 7. *Miscellaneous* 187

§ 28. Enumerable Predicates 187
§ 29. Arithmetical Predicates 192
§ 30. Universal Turing Machines 203
§ 31. λ-K-Definability 206
§ 32. The Minimal Logic of Fitch 219
§ 33. Further Precise Mathematical Replacements of the Concept of Algorithm 231
§ 34. Recursive Analysis 234

Author and Subject Index 241

LIST OF SYMBOLS

Symbols of the propositional calculus $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ 67

Symbols of the predicate calculus $\bigwedge$, $\bigvee$, $\bigwedge\limits_{x=0}^{n}$, $\bigvee\limits_{x=0}^{n}$ 67

Operator $\boldsymbol{\mu}$ 74; $\boldsymbol{\mu}\limits_{x=0}^{n}$ 75

Empty word $\square$ 146

Turing machines r 39; l 39; a_j 40; $*$, I 42;
R, L 49; $\boldsymbol{\rho}, \boldsymbol{\lambda}$ 49; S 49; $\mathfrak{R}, \mathfrak{L}$ 49;
T 52; $\boldsymbol{\sigma}$ 52; C 52; K 53; K_n 54;
$\overset{n}{\rightarrow}$, $\overset{\neq n}{\rightarrow}$, $\rightarrow$ 45

Tape inscriptions m, $\sim$, $*$, $*\cdots*$, $*\cdots$, W, X 48

Special functions χ_M 14; $S(x)$, x', U_n^i, C_0^0 60; C_n^k 63;
$V(x)$, $x \dot{-} y$, $|x-y|$, $sg(x)$, $\overline{sg}(x)$ 64;
$\varepsilon(x, y)$, $\delta(x, y)$ 65;
$p(n)$, p_n, $\exp(n, x)$, $l(x)$ 77;
$\sigma_n(x_1, \ldots, x_n)$, $\sigma_{nj}(x)$ 78;
$U(u)$ 111

Special predicates O_n, A_n 69; E, D, Pr 72; T_n 110