Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:
www.springer.com/series/7592

Gilles Dowek

Proofs and Algorithms

An Introduction to Logic and Computability

Springer

Gilles Dowek
École Polytechnique
Palaiseau
France
gilles.dowek@polytechnique.edu

Based on course notes by Gilles Dowek, published simultaneously in French by École Polytechnique with the following title: “Les démonstrations et les algorithmes”. The translator of the work is Maribel Fernandez.

ISSN 1863-7310
ISBN 978-0-85729-120-2 e-ISBN 978-0-85729-121-9
DOI 10.1007/978-0-85729-121-9
Springer London Dordrecht Heidelberg New York

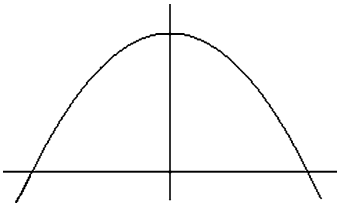
British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Acknowledgements

The author would like to thank René Cori, René David, Maribel Fernández, Jean-Baptiste Joinet, Claude Kirchner, Jean-Louis Krivine, Daniel Lascar, Stéphane Lengrand, Michel Parigot, Laurence Rideau and Paul Rozière.

Contents

Introduction



There are several ways to find the area of the segment of parabola depicted above. One method consists of covering the area with an infinite number of small triangles, proving that each of them has a specific area, then adding together all the areas of the triangles. This is *grosso modo* the method that Archimedes used to show that this area is equal to $4/3$. Another method, which gives the same result, has been known since the 17th century: the area can be obtained by computing $\int_{-1}^{1}(1-x^2)dx$. To integrate this polynomial function we do not need to build a proof, we can simply use an algorithm.

Building a proof and applying an algorithm are two well-known mathematical techniques; they have co-existed for a long time. With the advent of computers, which allow us to implement algorithms at a scale that was unimaginable in the past, there has been a renewed interest in algorithmic methods.

The co-existence of these two problem-solving techniques leads us to question their relationship. To what extent the construction of a proof can be replaced by the application of an algorithm? This book describes a set of results, some positive and some negative, that provide a partial answer to this question. We start by giving a precise definition of the notion of a proof, in the first part of the book, and of the notion of an algorithm, in the second part of the book. A precise definition of the notion of proof will allow us to understand how to prove independence theorems, which state that there are certain problems for which no proof can provide a solution. A precise definition of the notion of an algorithm will allow us to understand how to prove undecidability theorems, which state that certain problems cannot be

solved in an algorithmic way. It will also lead us to a better understanding of algorithms, which can be written in different ways (for instance, as a set of rewriting rules, as terms in the lambda-calculus, or as Turing machines), and to the discovery that behind this apparent diversity there is a deep unifying notion: the idea that a computation is a sequence of small steps.

The third part of the book focuses on the links between the notions of proof and algorithm. The main result in this part is Church's theorem, establishing that provability is an undecidable problem in predicate logic; Gödel's famous theorem is a corollary of this result. This negative result will be counterbalanced by two positive results. First, although undecidable, this problem is semi-decidable, and this will lead us to the development of algorithms that search for proofs. Second, by adding axioms to predicate logic we can, in certain cases, make the problem decidable. This will lead us to the development of decision algorithms for specific theories.

The final chapter of the book will describe a different link between proofs and algorithms: some proofs, those that are said to be *constructive*, can be used as algorithms.

Over the next chapters we will explore the deep connections that exist between the concepts of proof and algorithm, and unveil the complexity that hides behind the apparently obvious notion of truth.