ZMATH 2014f.00461

Witt, Kurt-Ulrich

Mathematical foundations for computer science. Sets, logic, recursion. (Mathematische Grundlagen für die Informatik. Mengen, Logik, Rekursion.)

Springer-Lehrbuch. Wiesbaden: Springer Vieweg (ISBN 978-3-658-03078-0/pbk; 978-3-658-03079-7/ebook). x, 222 p. (2013).

This textbook gives a thorough introduction to some basic mathematics every computer science student should understand before getting deeper into the particular questions of his or her profession. The main themes are logic, naive set theory, number systems and recursion and computability theory. It is meant as a one-semester course for first-year students. The first of its four parts is by far the longest one and gives mainly an extensive exposition of classical propositional logic. Its language, syntax and semantics are developed in great detail, and logical calculi are introduced together with their soundness and completeness. There is a whole section on the resolution calculus and another one on Horn logic. Then, classical predicate logic gets a short but sufficient treatment. A couple of nonclassical logics are merely mentioned together with their importance for computer science. The rest of the first part consists of sections on proof methods, set operations and Boolean algebra. The second part of the book is comparably short and deals with relations and functions on sets, while the third part is devoted to sets of numbers. Successively, the natural numbers, the integers, the rationals as well as the real and complex numbers are defined and constructed. On the way, cardinality of sets and the algebraic structures of groups, rings and fields are treated together with some of their elementary properties. Moreover, the generalised recursion scheme is developed from mathematical induction, and the recursive definition of the Fibonacci numbers as well as the Ackermann function and variants thereof are given as examples. The fourth part is dedicated to computability theory, using an approach employing primitive recursive functions and Kleene's μ-recursion. The equivalence of this approach with other models of computation (such as Turing machines or the λ-calculus) is merely mentioned in the context of Church's thesis without dealing with them any further. In the following, the chapter comes back to the Ackermann function as the standard example of a total computable function that is not primitive recursive, and then treats numerations of computable functions, the recursion theorem, computably enumerable and decidable versus undecidable sets, the halting problem, Rice's theorem and, finally, the undecidability of the correctness and the equivalence problem for programs. As a special feature, recursive functions are frequently also represented as (loop) programs in pseudo-code, thus illustrating how they "work" in an easily accessible way. Nearly all of the thirty sections of the book start with a short description of what is to be learned in it and ends with a summary of its contents highlighting the significance of the treated stuff and connections with particular questions of computer science. The sections contain a lot of motivations, examples and exercises, where solutions to the more difficult ones are given in a seperate section at the end of the book. Throughout, side notes are given where important notions, theorems or techniques are introduced, thus offering, together with the index, quick reference to them. All these features make this well-made book easily accessible to the intended audience and at the same time extremely well suited for self-study. It is an excellent addition to the impressive series of textbooks by the author.

Klaus D. Kiermeier (Berlin)

Classification: E35 E65 P25

Keywords: mathematical foundations of computer science; naive set theory; Boolean algebra; classical propositional logic; logical calculi; resolution calculus; Horn logic; classical predicate calculus; proof methods; number systems; algebraic structures; generalised recursion scheme; Fibonacci numbers; Ackermann function; computability theory; recursive functions; Church's thesis; numerations; recursion theorem; computably enumerable sets; halting problem; Rice's theorem; undecidability; programs in pseudo-code

doi:10.1007/978-3-658-03079-7