1. Record Nr. UNINA9910824075303321

Autore	Tourlakis George J
Titolo	Theory of computation / / George Tourlakis
Pubbl/distr/stampa	Hoboken, N.J., : Wiley, 2012
ISBN	1-280-59246-X 9786613622297 1-118-31535-9 1-118-31536-7 1-118-31533-2
Edizione	[1st ed.]
Descrizione fisica	1 online resource (410 p.)
Classificazione	MAT008000
Disciplina	511.3/52
Soggetti	Computable functions Functional programming languages
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Theory of Computation; CONTENTS; Preface; 1 Mathematical Foundations; 1.1 Sets and Logic; Naively; 1.1.1 A Detour via Logic; 1.1.2 Sets and their Operations; 1.1.3 Alphabets, Strings and Languages; 1.2 Relations and Functions; 1.3 Big and Small Infinite Sets; Diagonalization; 1.4 Induction from a User's Perspective; 1.4.1 Complete, or Course-of-Values, Induction; 1.4.2 Simple Induction; 1.4.3 The Least Principle; 1.4.4 The Equivalence of Induction and the Least Principle; 1.5 Why Induction Ticks; 1.6 Inductively Defined Sets; 1.7 Recursive Definitions of Functions; 1.8 Additional Exercises 2 Algorithms, Computable Functions and Computations 2.1 A Theory of Computability; 2.1.1 A Programming Framework for Computable Functions; 2.1.2 Primitive Recursive Functions; 2.1.3 Simultaneous Primitive Recursion; 2.1.4 Pairing Functions; 2.1.5 Iteration; 2.2 A Programming Formalism for the Primitive Recursive Functions; 2.2.1 PR vs. L; 2.2.2 Incompleteness of PR; 2.3 URM Computations and their Arithmetization; 2.4 A Double Recursion that Leads Outside the Primitive Recursive Function Class; 2.4.1 The Ackermann Function; 2.4.2 Properties of the Ackermann Function 2.4.3 The Ackermann Function Majorizes All the Functions of PR 2.4.4

The Graph of the Ackermann Function is in PR*; 2.5 Semi-computable Relations; Unsolvability; 2.6 The Iteration Theorem of Kleene; 2.7 Diagonalization Revisited; Unsolvability via Reductions; 2.7.1 More Diagonalization; 2.7.2 Reducibility via the S-m-n Theorem; 2.7.3 More Dovetailing; 2.7.4 Recursive Enumerations; 2.8 Productive and Creative Sets; 2.9 The Recursion Theorem; 2.9.1 Applications of the Recursion Theorem; 2.10 Completeness; 2.11 Unprovability from Unsolvability 3.5 Additional Exercises 4 Adding a Stack to a NFA: Pushdown Automata; 4.1 The PDA; 4.2 PDA Computations; 4.2.1 ES vs AS vs ES+AS; 4.3 The PDA-acceptable Languages are the Context Free Languages; 4.4 Non Context Free Languages; Another Pumping Lemma; 4.5 Additional Exercises; 5 Computational Complexity; 5.1 Adding a Second Stack; Turing Machines; 5.1.1 Turing Machines; 5.1.2 N P-Completeness; 5.1.3 Cook's Theorem; 5.2 Axt, Loop Program, and Grzegorczyk Hierarchies; 5.3 Additional Exercises; Bibliography; Index

Sommario/riassunto

"In the (meta)theory of computing, the fundamental questions of the limitations of computing are addressed. These limitations, which are intrinsic rather than technology dependent, may immediately rule out the existence of algorithmic solutions for some problems while for others they rule out efficient solutions. The author's approach is anchored on the concrete (and assumed) practical knowledge about general computer programming, attained readers in a first year programming course, as well as the knowledge of discrete mathematics at the same level. The book develops the meta-theory of general computing and builds on the reader's prior computing experience. Metatheory via the programming formalism known as Shepherdson-Sturgis Unbounded Register Machines (URM)--a straightforward abstraction of modern high level programming languages--is developed. Restrictions of the URM programming language are also discussed. The author has chosen to focus on the high level language approach of URMs as opposed to the Turing Machine since URMs relate more directly to programming learned in prior experiences. The author presents the topics of automata and languages only after readers become familiar, to some extent, with the (general) computability theory including the special computability theory of more "practical" functions, the primitive recursive functions. Automata are presented as a very restricted programming formalism, and their limitations (in expressivity) and their associated languages are studied. In addition, this book contains tools that, in principle, can search a set of algorithms to see whether a problem is solvable, or more specifically, if it can be solved by an algorithm whose computations are efficient. Chapter coverage includes: Mathematical Background; Algorithms, Computable Functions, and Computations; A Subset of the URM Language: FA and NFA; and Adding a Stack to an NFA: Pushdown Automata"--
"The book develops the meta-theory of general computing and builds on the reader's prior computing experience. Metatheory via the programming formalism known as Shepherdson-Sturgis Unbounded Register Machines (URM)--a straightforward abstraction of modern high-level programming languages--is developed. Restrictions of the URM programming language are also discussed. The author has chosen to focus on the high-level language approach of URMs as opposed to the Turing Machine since URMs relate more directly to programming learned in prior experiences"--