1.

Record Nr.	UNINA9910586582803321
Autore	Rosenberg Arnold L. <1941->
Titolo	Understanding computation : pillars, paradigms, principles / / Arnold L. Rosenberg and Lenwood S. Heath
Pubbl/distr/stampa	Cham, Switzerland : , : Springer International Publishing, , [2022] ©2022
ISBN	3-031-10055-7
Descrizione fisica	1 online resource (577 pages)
Collana	Texts in Computer Science
Disciplina	511.3
Soggetti	Computational complexity Computational complexity - Data processing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Contents -- Preface -- Part I INTRODUCTION -- Chapter 1 Introducing Computation Theory -- 1.1 The Autobiographical (ALR) Seeds of Our Framework -- 1.1.1 Computation by "Shapeless" Agents and Devices -- 1.1.2 Computation Theory as a Study of Computation -- 1.2 The Highlights of Our Framework: How We Tell the Story -- 1.3 Why Is a New Computation Theory Text Needed? -- Chapter 2 Introducing the Book -- 2.1 Computation Theory as a Branch of Discrete Mathematics -- 2.1.1 Dynamism Within Traditional Mathematics -- 2.1.2 Discrete Mathematics with Computational Objects -- 2.2 The Four Pillars of Computation Theory -- 2.2.1 Pillar S: STATE -- 2.2.2 Pillar E: ENCODING -- 2.2.3 Pillar N: NONDETERMINISM -- 2.2.4 Pillar P: PRESENTATION/SPECIFICATION -- 2.2.5 Summing Up -- 2.3 A Map of the Book by Chapter -- 2.4 Ways of Using this Book -- 2.4.1 As a Text for a "Classical" Theory Course -- 2.4.2 As a Primary Text: "Big Ideas in Computation" -- 2.4.3 As a Supplemental Text: "Theoretical Aspects of-" -- 2.5 Tools for Using the Book -- Part II Pillar S: STATE -- Chapter 3 Pure State-Based Computational Models -- 3.1 Online Automata and Their Languages -- 3.1.1 Basics of the OA Model -- 3.1.2 Preparing to Understand the Notion State -- 3.1.3 A Myhill-Nerode-like Theorem for OAs -- 3.2 Finite Automata and Regular Languages -- 3.2.1 Overview and History -- 3.2.2 Perspectives on Finite Automata -- 3.2.3 Why FAs Get Confused: a Consequence of

Finiteness -- Chapter 4 The Myhill-Nerode Theorem: Implications and Applications -- 4.1 The Myhill-Nerode Theorem for FAs -- 4.1.1 The Theorem: States Are Equivalence Classes -- 4.1.2 What Do L-Equivalence Classes Look Like? -- 4.2 Sample Applications of the Myhill-Nerode Theorem -- 4.2.1 Proving that Languages Are Not Regular -- 4.2.2 On Minimizing Finite Automata -- 4.2.3 Two-Way (Offline) Finite Automata.

4.2.4 Finite Automata with Probabilistic Transitions -- 4.2.5 State as a Memory-Constraining Resource -- Chapter 5 Online Turing Machines and the Implications of Online Computing -- 5.1 Online Turing Machines: Realizations of Infinite OAs -- 5.1.1 OTMs with Abstract Storage Devices -- 5.1.2 OTMs with Linear Tapes for Storage -- 5.2 The Nature of Online Computing -- 5.2.1 Online TMs with Multiple Complex Tapes -- 5.2.2 An Information-Retrieval Problem as a Language -- 5.2.3 The Impact of Tape Structure on Memory Locality -- 5.2.4 Tape Dimensionality and the Time Complexity of LDB -- Chapter 6 Pumping: Computational Pigeonholes in Finitary Systems -- 6.1 Introduction and Synopsis -- 6.2 Pumping in Algebraic Settings -- 6.3 Pumping in Regular Languages -- 6.3.1 Pumping in General Regular Languages -- 6.3.2 Pumping in Regular Tally Languages -- 6.4 Pumping in a Robotic Setting: a Mobile FA on a Mesh -- 6.4.1 The Mesh Mn and Its Subdivisions -- 6.4.2 The Mobile Finite Automaton (MFA) Model -- 6.4.3 An Inherent Limitation on Solo Autonomous MFAs -- Chapter 7 Mobility in Computing: An FA Navigates a Mesh -- 7.1 Introduction and Synopsis -- 7.2 MFAs That Compute in Read-Only Mode -- 7.2.1 How an MFA Can Exploit the Structure of Its Host Mesh -- 7.2.2 Solo MFAs Scalably Recognize Non-Regular Languages -- 7.3 MFAs as Object-Transporting Robots -- 7.3.1 Reversing M's Top-Row Pattern to Its Bottom Row -- 7.3.2 Rotating M's Top-Row Pattern to Its Bottom Row -- 7.3.3 Algorithms that Circumnavigate M's "Walls" -- Chapter 8 The Power of Cooperation: Teams of MFAs on a Mesh -- 8.1 Basics of MFA Teams -- 8.2 Strictly Coupled MFAs: Parallelism with No Added Power -- 8.3 Synchronized Computing: A 2-MFA Recognizes {a^k b^k} -- 8.4 Queued Computing: MFAs Proceed Through a Pipeline -- 8.4.1 The "Standard" Form of Pipelining.

8.4.2 Streaming a Pipeline/Queue of Agents -- 8.5 Ushered Computing: Two MFAs Sweep a Mesh-Quadrant -- 8.6 Sentry-Enabled Computing: MFAs Identify Home Wedges -- Part III Pillar E: ENCODING -- Chapter 9 Countability and Uncountability: The Precursors of ENCODING -- 9.1 Encoding Functions and Proofs of Countability -- 9.2 Diagonalization: Proofs of Uncountability -- 9.3 Where Has (Un) Countability Led Us? -- Chapter 10 Computability Theory -- 10.1 Introduction and History -- 10.1.1 Formalizing Mathematical Reasoning -- 10.1.2 Abstract Computational Models: the Church-Turing Thesis -- 10.1.3 Thinking About Thinking About Computation -- 10.2 Preliminaries -- 10.2.1 Computational Problems as Formal Languages -- 10.2.2 Functions and Partial Functions -- 10.2.3 Self-Referential Programs: Interpreters and Compilers -- 10.3 The Halting Problem: The "Oldest" Unsolvable Problem -- 10.3.1 The Halting Problem Is Semisolvable but Not Solvable -- 10.3.2 Why We Care About the Halting Problem-An Example -- 10.4 Mapping Reducibility -- 10.4.1 Basic Properties of m-Reducibility -- 10.4.2 The s-m-n Theorem: An Invaluable Source of Encodings -- 10.5 The Rice-Myhill-Shapiro Theorem -- 10.6 Complete-or "Hardest"-Semidecidable Problems -- 10.7 Some Important Limitations of Computability -- Chapter 11 A Church-Turing Zoo of Computational Models -- 11.1 The Church-Turing Thesis and Universal Models -- 11.2 Universality in Simplified OTMs -- 11.2.1 An OTM with a One-Ended Worktape -- 11.2.2 An

OTM with Two Stacks Instead of a Worktape -- 11.2.3 An OTM with a FIFO Queue Instead of a Worktape -- 11.2.4 An OTM with a "Paper" Worktape -- 11.2.5 An OTM with Registers Instead of a Worktape -- 11.2.6 A Mobile FA on a Semi-Infinite Mesh -- 11.3 Enhanced OTMs that Are No More Powerful than OTMs -- 11.3.1 An OTM Which Has Several Linear Worktapes.

11.3.2 An OTM Having Multidimensional Worktapes -- 11.3.3 An OTM Having a "Random-Access" Worktape -- 11.4 Models Outside the Classical Mold -- 11.4.1 Cellular Automata and Kindred Models -- 11.4.2 TMs as Volunteeers -- Simulation by Dovetailing -- 11.5 Learning from the Zoo -- Chapter 12 Pairing Functions as Encoding Mechanisms -- 12.1 PFs as Storage Mappings for Extendible Arrays/Tables -- 12.1.1 Insights on PFs via the Cauchy-Cantor Diagonal PF -- 12.1.2 PFs for Extendible Fixed-Aspect-Ratio Arrays -- 12.1.3 The Issue of PF Compactness -- 12.1.4 Extendible Storage Mappings via Hashing -- 12.2 Pairing Functions and Volunteer Computing -- 12.2.1 Basic Questions About Additive PFs -- 12.2.2 APFs that Have Desirable Computational Properties -- Part IV Pillar N: NONDETERMINISM -- Chapter 13 Nondeterminism as Unbounded Parallelism -- 13.1 Nondeterministic Online Automata -- 13.2 A Formal Use of Nondeterminism's Implied Parallelism -- 13.3 An Overview of Nondeterminism in Computation Theory -- Chapter 14 Nondeterministic Finite Automata -- 14.1 How Nondeterminism Impacts FA Behavior -- 14.1.1 NFAs Are No More Powerful than DFAs -- 14.1.2 Does the Subset Construction Waste DFA States? -- 14.2 An Application: the Kleene-Myhill Theorem -- 14.2.1 NFAs with Autonomous Moves -- 14.2.2 The Kleene-Myhill Theorem -- Chapter 15 Nondeterminism as Unbounded Search -- 15.1 Introduction -- 15.2 Nondeterministic Turing Machines -- 15.2.1 The NTM Model -- 15.2.2 An Unbounded Search: an OTM Simulates an NTM -- 15.3 Unbounded Search as Guess-then-Verify -- 15.4 Unbounded Search as Guess-plus-Verify -- 15.4.1 Homomorphisms on Formal Languages -- 15.4.2 Homomorphisms and Regular Languages -- Chapter 16 Complexity Theory -- 16.1 Introduction -- 16.2 Time and Space Complexity -- 16.2.1 On Measuring Time Complexity -- 16.2.2 On Measuring Space Complexity.

16.3 Reducibility, Hardness, and Completeness -- 16.3.1 A General Look at Resource-Bounded Computation -- 16.3.2 Efficient Mapping Reducibility -- 16.3.3 Hard Problems -- Complete Problems -- 16.3.4 An NP-Complete Version of the Halting Problem -- 16.3.5 The Cook-Levin Theorem: The NP-Completeness of SAT -- 16.4 Nondeterminism and Space Complexity -- 16.4.1 Simulating Nondeterminism Space-Efficiently -- 16.4.2 Looking Beyond Savitch's Theorem -- Part V Pillar P: PRESENTATION/SPECIFICATION -- Chapter 17 The Elements of Formal Language Theory -- 17.1 Early History of Computational Language Research -- 17.1.1 The Birth of Sophisticated Programming Languages -- 17.2 Elaborating on the Elements of Formal Languages -- 17.2.1 Language Generation via Formal Grammars -- 17.2.2 Closure Properties of Interest -- 17.2.3 Decision Properties of Interest -- 17.3 Finite Automata and Regular Languages -- 17.3.1 Mechanisms for Generating Regular Languages -- 17.3.2 Closure Properties of the Regular Languages -- 17.3.3 Decision Properties Concerning Regular Languages -- 17.4 Context-Free Languages: Their Grammars and Automata -- 17.4.1 Mechanisms for Generating the Context-Free Languages -- 17.4.2 Closure Properties of the Context-Free Languages -- 17.4.3 Decision Properties of the Context-Free Languages -- Appendix A A Chapter-Long Text on Discrete Mathematics -- A.1 Sets and Their Operations -- A.2 Binary Relations -- A.2.1 The Formal

Notion of Binary Relation -- A.2.2 Equivalence Relations -- A.3 Functions -- A.3.1 What Is a Function? -- A.3.2 Special Categories of Functions -- A.3.3 Finite Functions and Pigeonholes -- A.4 Formal Languages -- A.4.1 The Notion of Language in Computation Theory -- A.4.2 Languages as Metaphors for Computational Problems -- A.5 Graphs and Trees -- A.5.1 Basic Definitions -- A.5.2 Two Recurring Families of Graphs.
A.6 Useful Quantitative Notions.

2.

Record Nr.	UNINA9910150619803321
Autore	Pimsleur
Titolo	Pimsleur Turkish Level 1 Lessons 1-5 : Learn to Speak and Understand Turkish with Pimsleur Language Programs
Pubbl/distr/stampa	: Pimsleur (Simon & Schuster)
ISBN	1-4423-2383-3
Lingua di pubblicazione	Inglese
Formato	Musica
Livello bibliografico	Monografia
Sommario/riassunto	The Pimsleur® Method: the easiest, fastest way to learn a new language. Completely portable, easily downloadable, and lots of fun. Youll be speaking and understanding in no time flat!Each lesson in Turkish Phase 1, Units 1-5 provides 30 minutes of spoken language practice, with an introductory conversation, and new vocabulary and structures. Detailed instructions enable you to understand and participate in the conversation. Each lesson contains practice for vocabulary introduced in previous lessons. The emphasis is on pronunciation and comprehension, and on learning tospeakTurkish.

3.	Record Nr.	UNINA9910794658703321
	Titolo	Redesigning AI : work, democracy, and justice in the age of automation / / contributed by Daron Acemoglu
	Pubbl/distr/stampa	Cambridge, Massachusetts : , : Boston Review, , [2021] ©2021
	ISBN	1-946511-63-3
	Descrizione fisica	1 online resource (96 pages)
	Collana	Boston Review / Forum
	Disciplina	338.064
	Soggetti	Computers and civilization Artificial intelligence Ethics
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Nota di bibliografia	Includes bibliographical references.
	Nota di contenuto	Intro -- CONTENTS -- EDITORS' NOTE -- FORUM -- REDESIGNING AI -- FORUM RESPONSES -- A WORLD WITH LESS WORK -- CENTERING WORKERS' POWER AND RIGHTS -- THE PANDEMIC BOLSTERED SUPPORT FOR NECESSARY REFORMS -- TECHNOLOGY-FOCUSED SOLUTIONS WON'T WORK -- DECOLONIZING AI -- THE PROBLEM IS WAGES, NOT JOBS -- BEYOND THE AUTOMATION-ONLY APPROACH -- BETWEEN DYSTOPIA AND UTOPIA -- THE FRONTIER OF AI SCIENCE SHOULD BE IN UNIVERSITIES -- THE MEANS OF PREDICTION -- IT IS NOT TOO LATE -- ESSAYS -- STOP BUILDING BAD AI -- WORKPLACE TRAINING IN THE AGE OF AI -- MEDICINE'S MACHINE LEARNING PROBLEM -- THE PAST AND FUTURE OF AI -- CODING CARE -- CONTRIBUTORS.
	Sommario/riassunto	"A look at how new technologies can be put to use in the creation of a more just society"--The MIT Press webpage for this title (viewed May 19, 2021).