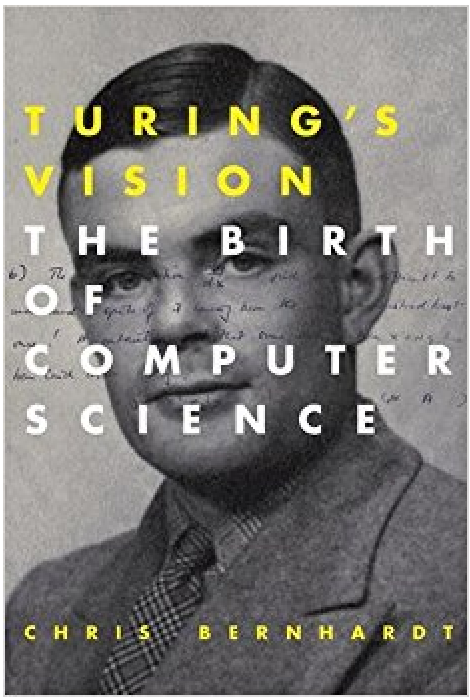
Turing's Vision: The Birth Of Computer Science (MIT Press)



Synopsis

In 1936, when he was just twenty-four years old, Alan Turing wrote a remarkable paper in which he outlined the theory of computation, laying out the ideas that underlie all modern computers. This groundbreaking and powerful theory now forms the basis of computer science. In Turing’s Vision, Chris Bernhardt explains the theory, Turing’s most important contribution, for the general reader. Bernhardt argues that the strength of Turing’s theory is its simplicity, and that, explained in a straightforward manner, it is eminently understandable by the nonspecialist. As Marvin Minsky writes, "The sheer simplicity of the theory’s foundation and extraordinary short path from this foundation to its logical and surprising conclusions give the theory a mathematical beauty that alone guarantees it a permanent place in computer theory." Bernhardt begins with the foundation and systematically builds to the surprising conclusions. He also views Turing’s theory in the context of mathematical history, other views of computation (including those of Alonzo Church), Turing’s later work, and the birth of the modern computer.In the paper, "On Computable Numbers, with an Application to the Entscheidungsproblem," Turing thinks carefully about how humans perform computation, breaking it down into a sequence of steps, and then constructs theoretical machines capable of performing each step. Turing wanted to show that there were problems that were beyond any computer’s ability to solve; in particular, he wanted to find a decision problem that he could prove was undecidable. To explain Turing’s ideas, Bernhardt examines three well-known decision problems to explore the concept of undecidability; investigates theoretical computing machines, including Turing machines; explains universal machines; and proves that certain problems are undecidable, including Turing’s problem concerning computable numbers.

Book Information

Series: MIT Press

Paperback: 208 pages

Publisher: The MIT Press; Reprint edition (April 21, 2017)

Language: English

ISBN-10: 0262533510

ISBN-13: 978-0262533515

Product Dimensions: 5.4 x 0.6 x 8 inches

Shipping Weight: 9.1 ounces (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars   6 customer reviews

Best Sellers Rank: #520,209 in Books (See Top 100 in Books)   #127 inÃ Â Books > Biographies &

Memoirs > Professionals & Academics > Computers & Technology   #167 inÃ Â Books > Computers & Technology > History & Culture > History   #1032 inÃ Â Books > Biographies & Memoirs > Professionals & Academics > Scientists

Customer Reviews

A fascinating account of Alan Turing’s epic research paper, which kicked off the entire computer revolution. I’m particularly impressed by the amount of detail the author includes while keeping everything simple, transparent, and a pleasure to read. (Ian Stewart, author of In Pursuit of the Unknown: 17 Equations That Changed the World)This is a delightful introduction for the lay reader to the ideas surrounding Alan Turing’s great paper of 1936. (Scott Aaronson, Associate Professor of Electrical Engineering and Computer Science, MIT)Over the past several decades, Alan Turing, known as the father of computer science, has become an intellectual and cultural icon. Chris Bernhardt has written a very clear and accessible book that explains Turing’s work, showing how his ideas have developed into some of the most important ideas in computer science today. (Noson S. Yanofsky, author of The Outer Limits of Reason: What Science, Mathematics, and Logic Cannot Tell Us)The dazzling array of computer applications, from desktop to cell phone, has obscured the play of ideas that first set our modern era in motion. In this account, Bernhardt reveals the crucial contribution to these developments made by Alan Turing and other early computer scientists. A marvelous book. (A. K. Dewdney, Professor Emeritus, Department of Computer Science, University of Western Ontario)

Chris Bernhardt is Professor of Mathematics at Fairfield University.

good story

What an outstanding, ground-up view of the technology and thinking that informs an increasing amount of how and what tools we use virtually every day to make decisions and, well, live.

I was hoping for more about the development of computer technology, sort of a timeline. The author tried to simplify the concepts and terms used in this field, but I still had trouble following it. But that’s because of my own limitations grasping abstract processes.

I absolutely love this book. Computer Science is my hobby, and a major part of my profession; thus making this book indispensable in terms of providing historical value to understanding CS roots. Reading this work has opened my mind to pioneers such as Charles Sanders Pierce, the father of pragmatism, and his numerous collections of philosophic, logical and mathematical writings. Furthermore, there is an endless array of notable reads that should be attempted; and not just by those who study computer science. For example, the writings of Euclid, George Boole, Russel Bertrand and much more. I highly recommend this to anyone who wants to "feel" what computer science truly is. Alan Turing's work is so influential and most people fail to realize its significance. Therefore do yourself the favor and pick this book up, you will not regret it, and likely it will help shape your mind as it did mine. Enjoy!

The best thing about this book is it will leave you hungering for more details from e.g. Martin Davis' Computability and Unsolvability.A Turing Machine in this book is described differently from what you'd find in Charles Petzold's The Annotated Turing. Instead of being in tabular form it's described by a "state diagram." These diagrams are introduced to describe finite automata, and a Turing Machine is presented as a finite automaton with additional capabilities. So in addition to accepting and non-accepting states that halt, a TM can also diverge (not halt). It's an appealing way to learn about TMs.The author is a mathematician, but you don't need experience reading definition-theorem-proof textbooks. In fact there's a section toward the end of the book titled Proof by Contradiction containing a thorough proof that sqrt(2) is not a rational number.I had a few "that's cool as s**t" moments while reading the book. One example is the tag system for the modified Collatz function.There were a couple places in the book I found unconvincing due to the introduction of an underspecified new type of TM. The multi-tape TM in the section RAMs Can Be Emulated by Turing Machines, Chap 6. And the TM that depends on the results of two other TMs running simultaneously in parallel in the proof on page 117.Professor Bernhardt is an excellent writer and Turing's Vision is a fine introduction to computation.

Turing's Vision gives the reader an overview of the logic required to understand the foundations of computability and computer science. Most works on Turing for the popular audience focus on his remarkable life but in this book the author tries to convey the core ideas of what computability is how computation can take place and the limits on what is computable. The author gives a history of where mathematical logic stood at the turn of the century and takes the reader through how Turing solved a major outstanding problem posed by Hilbert.The book is accessible to the general audience but this is a mathematical text so patience is often required. The book starts out by describing the world in the early 20th century from a mathematicians perspective. In particular the author covers topics like the goal of formalizing all of mathematics so that any statement could be proven or disproven within that system as well as the entscheidungsproblem, which was effectively the halting problem. The author goes through the foundations of arithmetic and gives a quick overview of things like GÃƒÂ¶del's results on the incompleteness of any mathematical system to describe itself fully using its axioms. The author covers the history as well as people working on these logic questions and gives a rough overview of who was working on what to the reader. The author then goes on to describe finite automata, a computational system which can answer a wide array of questions and the author also shows what cannot be computed by finite automata. The author describes Turing machines and how they cover and extend the boundaries of what can be computed by finite automata as well as what things like polynomial time and NP problems are. Though not comprehensive the book remains accessible so its definitely quite refreshing for the non-mathematician reader. The author then spends time on some more formal mathematics, in particular lambda calculus, a system of computing that is equivalent to Turing's more intuitive machine. Parts of the book like this add to completeness but can be skipped over by the reader less interested in all the logical structure behind the ideas. The author spends time on how to correspond Turing machines to finite automata. It is quite remarkable how certain problems are easily solved in one system but incredibly complicated in others. Modern computer architecture is covered briefly so theory becomes practical but these details are covered quite casually. The author then goes back to logic to cover the initial question posed at the beginning and goes through logical paradoxes when one looks at sets which refer to themselves. This concept is shown to be relevant for programs which are trying to decide on whether programs will halt or not and that impossibility is why the halting problem cannot be solved. The mathematical side of the book is furthered toward the end as the author spends time on understanding diagonalization arguments that are required to follow Turing's original argument. The author then goes on to discuss what can be computed as a consequence of these results from counting and cardinality. The author ends with a brief history of Turing and his work and life, it is very brief so just ties actual events to the book.Turing's Vision is a introduction to the foundational logic required to understand what a turing machine can and cannot do and what the nature of computation is. The reader gets a sense of the limits of what can be answered by a computer and why. There are other books which have a similar reader level like The Annotated Turing for example which are very good but this book focuses more on the logic side rather than deciphering the paper that Turing wrote. For the more technically inclined reader this is recommended but it is approachable for those who are willing to go through some more formal arguments

Turing's Vision: The Birth of Computer Science (MIT Press) The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine Computer Vision: Algorithms and Applications (Texts in Computer Science) The Man Who Knew Too Much: Alan Turing and the Invention of the Computer The Universal Computer: The Road from Leibniz to Turing Eye Exercises to Improve Vision: Recover Your Vision Naturally with Simple Exercises (Vision Training) 1st Grade Computer Basics : The Computer and Its Parts: Computers for Kids First Grade (Children's Computer Hardware Books) The Computer Music Tutorial (MIT Press) The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation (MIT Press) Extremal Combinatorics: With Applications in Computer Science (Texts in Theoretical Computer Science. An EATCS Series) Computer Science for the Curious: Why Study Computer Science? (The Stuck Student's Guide to Picking the Best College Major and Career) Fundamentals of Discrete Math for Computer Science: A Problem-Solving Primer (Undergraduate Topics in Computer Science) Mathematics and Computer Science in Medical Imaging (Nato a S I Series Series III, Computer and Systems Sciences) Alan Turing: Unlocking the Enigma (Kindle Single) Alan Turing: The Enigma: The Book That Inspired the Film "The Imitation Game" Turing's Cathedral: The Origins of the Digital Universe Alan Turing: The Enigma The Imitation Game: Alan Turing Decoded Alan Turing (Profiles in Mathematics) Turing: Pioneer of the Information Age