The Book Review Column[1]
by Frederic Green

Department of Mathematics and Computer Science
Clark University
Worcester, MA 01610
email: fgreen@clarku.edu

From the ethereal heights of the theory of computability, to the developing eras of computer science, to the sometimes vexing complexities of the present day. We touch upon each of these in the following reviews:

1. **The Foundations of Computability Theory (Second Edition)**, by Borut Robič. Reviewed by Erick Galinkin.

2. **Ideas that Created the Future: Classic Papers of Computer Science**, Edited by Harry Lewis. Reviewed by William Gasarch.

3. **Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion**, by Hal Abelson, Ken Ledeen, Harry Lewis, and Wendy Seltzer. Reviewed by William Gasarch.

I hope you are all healthy and safe, are seeing the light at the end of the tunnel, and are maintaining your eagerness to read new books. Please contact me to write a review! Choose from among the books listed on the next page. Or choose one of your own. The latter is actually preferable in the current circumstances, as I can then ask the publisher to forward it directly to you.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. *Algorithms and Data Structures Foundations and Probabilistic Methods for Design and Analysis*, by Helmut Knebl
2. *The Algorithm Design Manual*, by Steven S. Skiena
3. *Algorithms and Data Structures*, by Helmut Knebl
4. *Beyond the Worst-Case Analysis of Algorithms*, by Tim Roughgarden

Computability, Complexity, Logic

1. *Applied Logic for Computer Scientists: Computational Deduction and Formal Proofs*, by Mauricio Ayala-Rincón and Flávio L.C. de Moura.
2. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, by Martin Grohe.
3. *Semigroups in Complete Lattices*, by P. Eklund, J. Gutiérrez García, U. Höhle, and J. Kortelainen.

Miscellaneous Computer Science

1. *Elements of Causal Inference: Foundations and Learning Algorithms*, by Jonas Peters, Dominik Janzing, and Bernhard Schölkopf.
2. *Partially Observed Markov Decision Processes,* by Vikram Krishnamurthy
3. *Statistical Modeling and Machine Learning for Molecular Biology*, by Alan Moses
4. *Language, Cognition, and Computational Models,* Theirry Poibeau and Aline Villavicencio, eds.
5. *Computational Bayesian Statistics, An Introduction,* by M. Antónia Amaral Turkman, Carlos Daniel Paulino, and Peter Müller.
6. *Variational Bayesian Learning Theory,* by Shinichi Nakajima, Kazuho Watanabe, and Masashi Sugiyama.
7. *Knowledge Engineering: Building Cognitive Assistants for Evidence-based Reasoning*, by Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, and David A. Schum.
8. *Quantum Computing: An Applied Approach*, by Jack D. Hidary

Discrete Mathematics and Computing

1. *Mathematics in Computing*, by Gerard O'Regan
2. *Understand Mathematics, Understand Computing – Discrete Mathematics That All Computing Students Should Know*, by Arnold L. Rosenberg and Denis Trystram

Cryptography and Security

1. *Computer Security and the Internet: Tools and Jewels*, by Paul C. van Oorschot

Combinatorics and Graph Theory

1. *The Zeroth Book of Graph Theory: An Annotated Translation of Les Réseaux (ou Graphes) – André Sainte-Laguë (1926)*, translated by Martin Charles Golumbic
2. *Finite Geometry and Combinatorial Applications*, by Simeon Ball
3. *Combinatorics, Words and Symbolic Dynamics,* Edited by Valérie Berthé and Michel Rigo

Programming etc.

1. *Formal Methods: An Appetizer*, by Flemming Nielson and Hanne Riis Nielson
2. *Programming for the Puzzled: Learn to Program While Solving Puzzles*, by Srini Devadas.
3. *Sequential and Parallel Algorithms and Data Structures*, by P. Sanders, K. Mehlhorn, M. Dietzfelbinger, R. Dementiev

Miscellaneous Mathematics

1. *Introduction to Probability*, by David F. Anderson, Timo Seppäläinen, and Benedek Valkó.
2. *Algebra and Geometry with Python*, by Sergei Kurgalin and Sergei Borzunov.

Review of[2]
The Foundations of Computability Theory (Second Edition)
by Borut Robič
Springer-Verlag, 2020
$109.99, Hardcover, 422 pages

Review by
Erick Galinkin (erick.galinkin@rapid7.com)
AI Research Team
Rapid7

1 Overview

Computability theory forms the foundation for much of theoretical computer science. Many of our great unsolved questions stem from the need to understand what problems can even be solved. The greatest question of computer science, P vs. NP, even sidesteps this entirely, asking instead how efficiently we can find solutions for the problems that we know are solvable. For many students both at the undergraduate and graduate level, a first exposure to computability theory follows a standard sequence on data structures and algorithms and students often marvel at the first results they see on undecidability – how could we possibly prove that we can never solve a problem?

This book, in contrast with other books that are often used as first exposures to computability, finite automata, Turing machines, and the like, focuses very specifically on the notion of what is computable and how computability theory, as a science unto itself, fits into the grander scheme. The book is appropriate for advanced undergraduates and beginning graduate students in computer science or mathematics who are interested in theoretical computer science. Robič sidesteps the standard theoretical computer science progression – understanding finite automata and pushdown automata before moving into Turing machines – by setting the stage with Hilbert's program and mathematical prerequisites before introducing the Turing machine absent the usual prerequisites, and then introducing advanced topics often absent in introductory texts. Most chapters are relatively short and contain problem sets, making it appropriate for both a classroom text or for self-study.

2 Summary of Contents

The book is broken up into four parts. Throughout, Robič provides a "fast track" and "detours". The bulk of the content is contained in the "fast track", with many of the proofs and context reserved for the "detours". As one might expect, the early chapters are heavier on the detours, since technical results are difficult to set aside in the later chapters.

Part 1 consists of mathematical prerequisites and motivation for the topic. Chapter 1 sets the stage and provides historical context for the study of algorithms from Euclid's algorithm for finding the greatest common denominator, through Leibnitz and Babbage's mechanized computers. Chapter 2 moves into set theory, introducing Cantor and his notion of countability, cardinality, axiomatization. It then discusses the school of logicism led by Boole, Frege and Peano, and finally Russell and Whitehead. Chapter 2 concludes with David Hilbert and the idea of formalism – posing the questions that would eventually motivate Gödel:

the consistency and completeness of arithmetic. Chapter 3 delves deep into formalism, providing the reader with some symbolic logic, notions of truth in logical systems, propositional calculus, and models. The chapter then moves into formalization of set theory, introducing Ernst Zermelo, Abraham Fraenkel, and their ZF axiomatization of set theory. An alternate axiomatic system, NBG – von Neumann, Bernays, Gödel – is also discussed, and some gritty technical details of each are included by the author in detours. Chapter 4 finally gets to the problem which inspired Turing and his eponymous machine, the Entscheidungsproblem. Robič discusses formalism and Hilbert's program before introducing the problem of deciding whether any formula of a formal axiomatic system can be derived in that system. The chapter concludes with a very clear description of Gödel's incompleteness theorems, setting the stage for part 2.

Part 2 forms the bulk of the book and the main results of classical computability theory. Chapter 5 defines the terms *algorithm* and *computation*, considering what these words could mean to different people. The chapter introduces important notation and terminology, especially the term efficiently calculable, which is used throughout the remainder of the book. The chapter concludes with the Church-Turing thesis and the notions of total and partial functions. Chapter 6 introduces the idea of Turing machines for the first time. Robič addresses the practical consequences of Turing machines including universal Turing machines, operating systems, and RAM machines, ending with set generation and recognition by Turing machines. Chapter 7 introduces many of the most important tools and theorems for computability theory: the recursion theorem, the parameter theorem, and the padding lemma. Chapter 8 deals with incomputable problems. The chapter discusses the halting problem, and illustrates examples of other incomputable problems from a variety of disciplines including games, number theory, algebra, and topology. Chapter 9 provides techniques for proving incomputability – diagonalization, reduction, the recursion theorem, and Rice's theorem – concluding part 2 by allowing readers to use a variety of techniques to prove incomputability.

Part 3 deals with material that is not often included in standard introductory material. Specifically, part 3 discusses relative computability: the degrees of computability. Chapter 10 deals with Turing machines that have external help – the concept of oracular Turing machines. Robič also discusses the practical impact of these results: machines with databases or network connections. Chapter 11 introduces Turing reductions and Turing degrees – ideas that underlie the ideas of relative computability. The chapter establishes the base cases, that there are at least two Turing degrees: the Turing degree of the halting problem, and the Turing degree of problems which are computable. Chapter 12 identifies the Turing hierarchy building upon the idea of Turing degrees and establishing the Turing jump and the jump hierarchy, establishing a preorder on the degrees of unsolvability. Chapter 13 establishes the class of degrees of unsolvabilty. This ultimately establishes an infinite hierarchy, with each T-degree forming a countably infinite set between each degree of unsolvability, ultimately establishing the cardinality of the class as $2^{\aleph_0}$ and the presence of a minimum – though no maximum – degree of unsolvability. Chapter 14 introduces Post's problem and the priority method, providing the primary contemporary technique for establishing results about computably enumerable sets. Chapter 15 introduces prenex normal form and the arithmetical hierarchy, establishing a connection between the arithmetical hierarchy and the Turing hierarchy.

Part 4 consists of a two chapters: chapter 16, which is the longest chapter in the book, and chapter 17, which forms a sort of prologue. Chapter 16 walks through the historical context that led to the development of the Church-Turing thesis and discusses the provability of the Church-Turing thesis. The chapter then deals with other formulations of the Church-Turing thesis: algorithmic, complexity-theoretic, and physical – concluding with hypercomputing as a hypothetical way to disprove the physical Church-Turing thesis. Chapter 17 provides further reading on a variety of topics including computational complexity theory and more advanced computability theory.

3 Conclusion and Opinion

For first courses in theoretical computer science, this book could provide a great alternative or supplemental text to the standard texts in the field. Most of the chapters are fairly small and contain a number of relevant problems at the end to solidify the concepts. There are a small number of appendices to deal with mathematical prerequisites and notation. In addition, the book contains an extensive glossary and a bibliography with 281 references. This makes the text great for self-study.

Overall, the book rates as my personal favorite introductory text on the topic. The book is well written, provides a small number of well-thought out exercises, and offers the fast track and detour options giving it great opportunity to be used for both undergraduate and graduate courses. Although it does not deal with some of the standard introductory material on deterministic finite automata and pushdown automata, removing this content provides an opportunity to introduce additional mathematical rigor. This is a deliberate choice by Robič to allow for the more advanced topics, such as Turing degrees and the arithmetical hierarchy, to be introduced.

The text may not be ideal for students who are not heavily mathematically inclined, however. Readers who are interested in computational complexity theory in particular will also likely be frustrated by how the book touches right up against the subject before changing course several times. Additionally, the omission of familiar beginner material such as regular languages, the pumping lemmas, context-free grammars, and Chomsky normal form may make the book a tough sell for a stand-alone first text.

With my bias towards mathematics in mind, I strongly recommend this book to anyone with even a passing interest. The material will be new to many and the writing is outstanding.

Review of[3]
Ideas that Created the Future:
Classic Papers of Computer Science
Edited by Harry Lewis
MIT Press, 2021
$60.00 paperback, $42.00 Kindle, 517 pages

Review by
William Gasarch (gasarch@umd.edu)
Computer Science Department
University of Maryland, College Park

Disclosure: Harry Lewis, the editor of this book, was my PhD adviser.

1 Introduction

What are the most important papers in computer science? What are the most important 46 papers in computer science? Yikes! Far more than 46 seem to qualify. Picking out the top 46 papers in computer science is the task that befallen Harry Lewis (henceforth Harry). I suspect it was both a burden (gee, which ones to omit?) and a joy (Wow, these papers are insightful!). He used two (probably more) criteria that helped him cut it down (1) he prefers short readable papers to long unreadable ones (don't we all!), and (2) no paper past 1980 (arbitrary but firm cutoff). There were also absurd financial constraints based on how much the rights to republish a paper costs. In some cases a paper that is available online for free cost too much to put into a book. See my comment on Turing's 1936 paper.

This book is a collection of the 46 most important papers in computer science, in the opinion of Harry, relative to some constraints. The complete list is at the end of the review. While I am sure many readers will think *why isn't X in the list*, or *why is X in the list*, or *I never heard of X*, I suspect that 2/3 of the people reading this review will agree with 2/3 of the papers on the list. I would urge people to read the book AND THEN have an intelligent debate about which papers should or should not be in it.

What does this book add that you could not get by just finding the papers online and reading them?

1. Harry carefully picked out which papers are important, short, and readable.

2. Each paper has an introduction (written by Harry) which tells you about the author(s), about the paper, and why the paper is important.

3. The papers are not there in their entirety. Some parts have been edited out which makes them more readable. In essence, he edited out the boring parts.

4. There is something about having a book in front of you, or even a kindle-version, that makes you want to read it, rather than going to the web and reading them one at a time. I used to think this was my inner-Luddite talking, but young people tell me that they also are more inclined to read solid books than ephemeral screens.

5. By reading these papers you can pick up certain threads: some people thought computers could eventually think or do all kinds of miraculous things, others were more modest. Some were interested in

[3] ©2021 William Gasarch

scientists using computers, others with business people. Later there was issues about ordinary people using them, an issue that just had not come up earlier. There are other threads to pick up on.

6. One of the proofreaders for the review is dyslexic and uses text-to-speech software. After she gave me (much appreciated) corrections and suggestions I inquired if she thinks she would like the book. She responded: *A book of past papers in it is great for someone like me since older papers are usually not in a format for text-to-speech software. But books published recently are in such a format. I had accepted that I was just never going to be able to read papers like these, but since they are in this book, I can!*

Some of the papers are from theory, so the reader of this review (in SIGACT News) probably knows what's in them, but probably *has not read them*. I was surprised to realize how many classic papers I had not actually read. Its great to see what the original authors thought of their work. Some of the papers were quite prescient. Others were . . . less so.

2 Comments on a Few of the Papers

It would be madness to comment on all 46 papers in the space of a review. So I comment on a selection, hoping to hit several types of papers. I won't comment on Harry's comments for every article; however, suffice it to say that he sets up what you are about to read very well.

2.1 Prior Analytics, by Aristotle

Computer Science goes back to Aristotle. Really! Aristotle was concerned with being able to make inferences based solely on the *form* of a set of sentences rather than their *content*, what we would call deductions. The excerpt we get is 3 pages which is plenty. It is surprisingly readable, but wordy given today's understanding. Harry's introduction is particularly helpful here. For example, Harry points out that the paper gives rise to the modern notion of a set.

Boole learned logic out of Aristotle's text. (An excerpt from Boole's book, *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*, is in this book). Boole took Aristotle's ideas further and Shannon applied them to real computers (Shannon paper, *A symbolic analysis of switching circuits* is in this book). The interplay of logic and computers is an example of a thread one can follow from this book that would be hard to follow if one looked for papers on the web.

2.2 The True Method by Gottfried Wilhelm Leibniz

Leibniz might be the first computer scientist (questions of *who was the first X* are always complicated). He build a nested-loop calculator that could multiply and divide. One competitor for the honor is Pascal who built an adding machine. Both feats are impressive.

The paper by Leibniz is quite readable but a bit long for its content (by modern standards). I was struck that he thought computing devices could settle all questions, both in Physics (where experiments are expensive) and Metaphysics (where experiments are impossible). He was an optimist. He was right in that computers can often do simulations and hence solve problems in Physics. He was wrong in that the problems of metaphysics are still unsolved and probably always will be.

This paper can be viewed as a research proposal. The following line in it would likely not be in an NSF grant proposal today:

It is one of my ambitions to finish this project if God grants me the time.

2.3 On Computable Numbers, With an Application to the Entscheidungsproblem, by Alan Turing

This was the most expensive paper to get permission to reprint here, even though its free on line in the following places:
https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf
https://academic.oup.com/plms/article/s2-42/1/230/1491926?login=true
http://www.turingarchive.org/browse.php/b/12
https://www.wolframscience.com/prizes/tm23/images/Turing.pdf

As you may have guessed, this is the paper where Turing machines (not called that in the paper) are defined and HALT is shown to be undecidable. Interesting historically since he has to also convince the reader that Turing-computable is computable (which we, with 20-20 hindsight, find obvious) and that computable is Turing-computable (which we call Church's thesis or the Church-Turing thesis).

2.4 A Logical Calculus of the Ideas Immanent in Nervous Activity, by Warren McCulloh and Walter Pitt

The goal of this paper was to model how the brain works. That sounds hard. And indeed it is. This paper both fails and succeeds.

1. The model of the brain they come up with is not accurate for the real brain.
2. They lay the groundwork for (1) finite automata, which Kleene and others picked up on, and (2) Neural Nets, which Rosenblatt picked up on (Rosenblatt's paper *The Perceptron* is in the book).

This paper is also the first one that talks of having computers mimic what people do by trying to do it the way people do it. In later years some people in AI wanted to have computers just do X that people do well, while others wanted computers to do X the way people do it. This difference is an interesting thread to follow.

2.5 As We May Think, by Vannevar Bush

This article appeared in *The Atlantic*, a popular magazine. That goes to show that you can say scientifically important things in popular media if (1) you're smart enough, and (2) the science isn't to advanced.

When I read this article my first reaction was:

Wow! This article, written in 1945, predicts Google, Kindle, Browsing the web, Data Structures, Data Base, and P vs NP.

Upon reading it again I realized that I was projecting the future into the past. But not much. Truly a visionary article.

2.6 Some Moral and Technical Consequences of Automation, by Norbert Wiener

This paper warns of the the dangers of computers. The dangers fall into two categories.

1. Computers can be trained for some scenarios but then another scenario happens and the machines reaction could be dangerous. Wiener was probably thinking of automated weapons systems and accidental nuclear war. Today the same problem might plague self-driving cars.

2. In the story *The Sorcerer's Apprentice*, within the movie *Fantasia*, Mickey Mouse asks a broom to get him some water. The broom keeps doing this and Mickey almost drowns. Computers may take us to literally and that could be dangerous. I wonder if computer-trading on Wall Street may have this problem, or computers doing bidding at auctions.

Has anything like *The Sorcerer's Apprentice* really occurred? As it happens, the second edition of *Blown to Bits* by Hal Abelson, Ken Ledeen, Harry Lewis (Same Harry Lewis!), and Wendy Seltzer, which is reviewed in this column by William Gasarch (Same William Gasarch!) has the following story which I paraphrase: On Amazon a used book *The Making of a Fly*, list price $70.00, was listed at around $23,000,000. It does get 4.1 stars (out of 5) so it is probably a very good book. But this was not the reason why it was so expensive. There were two sellers: Bordeebook and Profnath. They had different programs to price their books:

- Bordeebook was programmed to raise the price when others did. In particular, it charged approximately 1.23 times what Profnath was charging.
- Profnath was programmed to undercut its competitors by just a little. In particular, it charged approximately 0.998 times what Bordeebook charged.

I leave it to the reader to work out how this causes prices to go to infinity.
This story is more amusing than dangerous. But it is still a cautionary tale and a proof-of-concept.

2.7 Cramming More Components onto Integrated Circuits, by Gordon Moore

One sign that computer science is a new field is that some of the writers of classic papers are still alive. As of April 2021, when I am writing this review, Gordon Moore is alive at the age of 92. In fact, for the last 20 (or so) papers, most of the authors are still alive.

This article appeared in *Electronics*, a popular electrical engineering magazine. As with Bush's article, this shows that you can say scientifically important things in popular media if (1) you're smart enough, and (2) the science isn't to advanced.

The paper states, almost in passing, that the number of components on a chip will double every two year. He later revised that down to once-a-year and then to once-every-1.5 years. He also predicted it would stop after about 40 years, which was about right. (The article appeared in 1965.)

Moore never stated that speed doubles every 1.5 years; however, I suspect he was happy with this law being named after him. That version of Moore's law held for about 40 years also.

The paper is a good read. Moore tells us why he thinks component density will increase. His arguments are intelligent and he is correct.

2.8 Solution of a Problem in Concurrent Program Control, by Edsger Dijkstra

In this paper Dijkstra gives an elegant proof that a concurrent program is correct.

There are several other papers in the volume on proving programs correct:

1. *The Structure of the "THE"–Mulitprogramming System* by Edsger Dijkstra. This paper introduces semaphores, which are a method for reasoning about atomic actions in a program. They also apply the methods to an actual system.
2. *Go To Considered Harmful* by Edsger Dijkstra. This paper argues why we should get rid of the Goto statement so that it would be easier to reason about programs.

3. *An Axiomatic Basis for Computer Programming* by Tony Hoare. This paper lays out a framework for proving programs correct; however, it also points out obstacles to the endeavor.

4. *Social Processes and Proofs of Theorems and Programs* by Richard DeMillo, Richard Lipton, and Alan Perlis. This paper argues that the field of proving programs correct is doomed to failure. This may have been a self-full-filling prophecy since this paper led to funding being cut.

All of these papers were influential. It is interesting to see them all in the same volume.

2.9 Managing the Development of Large Software Systems, by Winston Royce

Having a team of programmers write a large piece of software is hard! This paper provides, to quote Harry's introduction to it: *useful wisdom that has passed the test of time and remains good advice today*. The paper is also quite readable, particularly since it has lots of nice diagrams that build on each other.

Another excellent paper (actually a book excerpt) on this topic that appears in this book is *The Mythical Man-Month* by Fred Brooks. This paper makes the obvious-in-hindsight point that putting more people on a project that is falling behind will only make it fall further behind.

2.10 The Complexity of Theorem-Proving Procedures, by Stephen Cook

This is the paper that proves SAT is NP-complete (actually, this paper deals with TAUT). Karp's paper *Reducibility Among Combinatorial Problems* (which is also in this book) showed 21 problems are NP-complete. Cook's paper and Karp's paper together launched modern complexity theory.

2.11 A Statistical Interpretation of Term Specificity and its Application to Retrieval, by Karen Sparck Jones

This is an early paper on search. I quote from the introduction by Harry:

The more often a term occurs in a document, the more relevant it is to the document's content. For example, a paper that uses the term `zebra` *repeatedly is probably at least somewhat about zebras. But of course, the same paper will use the term* `the` *repeatedly, so mere frequency within a document is an unreliable indicator of a word's significance.*

The introduction then goes on to say that you must compare the frequency of a word in a document with its frequency in *other documents*. This paper shows how to really do this.

2.12 A Protocol for Packet Network Intercommunication, by Vinton Cerf and Robert Kahn

No, Al Gore did not invent the internet; however, there is a good argument that Cerf and Kahn did. In this paper, written in 1974, they designed standards and protocols for what we now call the internet. It was very important that the protocols be standardized to make the world wide web truly world wide.

2.13 New Directions in Cryptography, by Whitfield Diffie and Martin Hellman

This paper started a revolution in cryptography in two ways: (1) they showed that there was a way Alice and Bob can establish a shared secret key without having a secret channel (by making some reasonable hardness assumptions), (2) making cryptography into a rigorous field of study.

It's interesting to read it now to see that *they knew* it would start a revolution. This paper lead to *A Method For Obtaining Digital Signatures and Public-Key Cryptosystems*, by Rivest-Shamir-Adleman (which is also

in the book). Both the Diffie-Helman paper and the RSA-paper are quite readable. This is not surprising since they were early and use what are now well known concepts.

3 Who Should Read This Book?

You should read it. You should also give it to your CS-inclined niece or great niece for a graduation present. Note that the price, $60.00 softcover, $42.00 Kindle, for a book that is over 500 pages is, as the kids say, jawsome (jaw dropping awesome).

It is very interesting to read these papers and see the roots of computer science. It's interesting to (1) follow some threads, (2) see where people were correct in their predictions, (3) see where people were wrong in their predictions (Turing thought that ESP was real and might distinguish humans from machines in the paper *Computing Machinery and Intelligence*). And it's interesting to read papers that you thought you knew about and find out they didn't quite say what you thought.

4 The 46 Papers

1. Prior Analytics ($\sim$ 350 BCE), Aristotle
2. The True Method (1677), Gottfried Wilhelm Leibniz
3. Sketch of the Analytical Engine (1843), L. F. Menabrea, with Notes by the Translator, Ada Augusta, Countess of Lovelace
4. An Investigation of the Laws of Thought on Which Are Founded the Mathematical Theories of Logic and Probabilities (1854), George Boole
5. Mathematical Problems (1900), David Hilbert
6. On Computable Numbers, with an Application to the Entscheidungsproblem (1936), Alan Mathison Turing
7. A Proposed Automatic Calculating Machine (1937), Howard Hathaway Aiken
8. A Symbolic Analysis of Relay and Switching Circuits (1938), Claude Shannon
9. A Logical Calculus of the Ideas Immanent in Nervous Activity (1943) Warren McCulloch and Walter Pitts
10. First Draft of a Report on the EDVAC (1945), John von Neumann
11. As We May Think (1945), Vannevar Bush
12. A Mathematical Theory of Communication (1948), Claude Shannon
13. Error Detecting and Error Correcting Codes (1950), R. W. Hamming
14. Computing Machinery and Intelligence (1950), Alan Mathison Turing
15. The Best Way to Design an Automatic Calculating Machine (1951), Maurice Wilkes

16. The Education of a Computer (1952), Grace Murray Hopper
17. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem (1956), Joseph B. Kruskal, Jr.
18. The Perceptron: A Probabilistic Model for Information Storage and Organization (1958), Frank Rosenblatt
19. Some Moral and Technical Consequences of Automation (1960), Norbert Wiener
20. Man-Computer Symbiosis (1960), J. C. R. Licklider
21. Recursive Functions of Symbolic Expressions and Their Computation by Machine (1960), John McCarthy
22. Augmenting Human Intellect: A Conceptual Framework (1962), Douglas C. Engelbart
23. An Experimental Time-Sharing System (1962), Fernando Corbato, Marjorie Merwin Daggett, and Robert C. Daley
24. Sketchpad (1963), Ivan E. Sutherland
25. Cramming More Components onto Integrated Circuits (1965), Gordon Moore
26. Solution of a Problem in Concurrent Program Control (1965), Edsger Dijkstra
27. ELIZA – A Computer Program for the Study of Natural Language Communication between Man and Machine (1966), Joseph Weizenbaum
28. The Structure of the THE-Multiprogramming System (1968), Edsger Dijkstra
29. Go To Statement Considered Harmful (1968), Edsger Dijkstra
30. Gaussian Elimination is Not Optimal (1969), Volker Strassen
31. An Axiomatic Basis for Computer Programming (1969), C. A. R. Hoare
32. A Relational Model of Large Shared Data Banks (1970), Edgar F. Codd
33. Managing the Development of Large Software Systems (1970), Winston W. Royce
34. The Complexity of Theorem-Proving Procedures (1971), Stephen A. Cook
35. A Statistical Interpretation of Term Specificity and Its Application in Retrieval (1972), Karen Spärck Jones
36. Reducibility among Combinatorial Problems (1972), Richard Karp
37. The Unix Time-Sharing System (1974), Dennis Ritchie and Kenneth Thompson
38. A Protocol for Packet Network Intercommunication (1974), Vinton Cerf and Robert Kahn
39. Programming with Abstract Data Types (1974), Barbara Liskov and Stephen Zilles

40. The Mythical Man-Month (1975), Frederick C. Brooks

41. Ethernet: Distributed Packet Switching for Local Computer Networks (1976), Robert Metcalfe and David R. Boggs

42. New Directions in Cryptography (1976), Whitfield Diffie and Martin Hellman

43. Big Omicron and Big Omega and Big Theta (1976), Donald E. Knuth

44. Social Processes and Proofs of Theorems and Programs (1977), Richard DeMillo, Richard Lipton, and Alan Perlis

45. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems (1978), Ronald Rivest, Adi Shamir, and Len Adleman

46. How to Share a Secret (1979), Adi Shamir

Review of[4]
Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion
by Hal Abelson, Ken Ledeen, Harry Lewis, and Wendy Seltzer
Addison-Wesley, 2021
$25.95, Softcover, 300 pages

Review by
William Gasarch (gasarch@umd.edu)
Computer Science Department
University of Maryland, College Park

Disclosure: Harry Lewis, one of the authors of the book under review, was my adviser.

1 Overview

I reviewed the first edition of this book in 2009. Here is a link to the entire column which includes reviews of other books as well: https://www.cs.umd.edu/users/gasarch/bookrev/40-1.pdf

If you are reading this as an online pdf you can probably click on he above line and get to my old review. (Is that illegal on your part? On my part? I have no idea.) Some of this review will overlap with some of that review.

I begin with the first paragraph of my old review just to show how one issue seems to have been solved.

BEGIN EXCERPT

The Music industry has the following valid complaint: **People who download music illegally are ripping off the artists! That's an outrage! That's our job!**

END EXCERPT

In their attempt to rein in Napster and other file sharing services, the record companies acted very badly— suing people who had nothing to do with downloading music. Weird Al has a great song about it, available for free on YouTube, with a neat video, here:

https://www.youtube.com/watch?v=zGM8PT1eAvY

At one time downloading music was a big issue. This issue has been solved (somewhat) by changing the business model in several ways:

1. iTunes is cheap and easy to use, so its easier and perhaps cheaper than pirating.
2. There are subscription services which give you unlimited access to LOTS of music.
3. Some groups have their music available for free and ask you to give a donation to them.
4. Some creators (not many) make money off YouTube by posting there and having advertisements in their videos. This may explain how I can listen to the entire score of *Hamilton* on YouTube for free. I'm still waiting for my song, *Muffin Math*, which is here:

   https://www.youtube.com/watch?v=4xQFlsK7jKg

   to make the big bucks. It has 16 likes and 0 dislikes. (My proofreader Emily tells me that the only way *Muffin Math* will make money is if it is put into a cringe compilation. She may be right.)

[4] © William Gasarch, 2021

5. View your free-to-all songs on YouTube as advertisements for you concerts or show. This is an alternative explanation as to why I can listen to the score of *Hamilton* for free.

The movie business and the book business have the same problem (pirating) but does not seem to have found a new business model for a solution yet. Some of the above might work.

The book under review is about the changes to society caused by the electronic age. My impression is that they started out wanting to do an intelligent and unbiased view, and that they succeeded. However business look immoral and the government looks incompetent. As Stephan Colbert might say *the truth has an anti-business agenda*. My impression is that while writing the second edition they wanted to say what got better, but, alas, most things got worse.

The book is not just about business and government. It's about virtually all aspects of the electronic age. Much of what the book says is obvious *once you see it written down* but not obvious before that point. This is quite valuable.

2 Summary of Contents

2.1 Chapter 1: Digital Explosion—Why is it happening and what is at Stake?

The first chapter drives home the point that the world really has changed. It states 7 koans about the modern world which are obvious once they are stated, but worth stating. I give one example:

Koan 7: Bits move faster than thought.

A father gets a condolence card that his daughter is dead—before he knows she died.

A family gets coupons for maternity products—before they know their teenage daughter is pregnant.

Protests spring up via the internet in countries by people demanding freedom very quickly. Yeah!

Authorities shut down the internet just as fast. Boo!

Within 2 weeks of the first story saying how serious COVID was there were 45 funny songs about it on YouTube:

`http://www.cs.umd.edu/~gasarch/FUN//funnysongs.html`

(As a collector of novelty songs I found this both thrilling and exhausting.)

Within two days of the Jan 6 insurrection there was already a funny song on YouTube about it:

`https://www.youtube.com/watch?v=wT5kafhG3Qw`

You get the idea.

2.2 Chapter 2: Naked in the Sunlight—Privacy Lost, Privacy Abandoned

and

2.3 Chapter 3: Who Owns Your Privacy?

In my review of the first edition I noted that they used the phrase **little brother is watching**. That is, the government was no longer the biggest threat to privacy, your neighbor's cell phone was. While your neighbor's cell phone is still *a* threat to privacy, it is not clear what is the biggest threat: your Facebook friends, your enemies, strangers-with-cell-phones, government, corporations, or some combination as in this link:

`https://www.youtube.com/watch?v=cqggW08BWO0`

The Snowden files (sneaked out of the NSA in his pocket) revealed how much the government is tracking you. This may be a needed component for the war on terrorism but one wonders what else they are using the information for.

Corporations are also tracking you, though they claim they just want to know your habits so they can market to you better. Does that make it harmless?

1. They may be subpoenaed to give their information about you to the government. In writing that last sentence notice I used the phrase *their information about you*. That is really odd. Isn't it your information?

2. General privacy. I do not want others to know that I have a taste for European Kit Kat bars as they will make the leap to thinking I am a Europhile who likes the French.

3. They can use this information to manipulate you into (a) buying their product, or (b) voting for a political candidate. This is not a speculation. The book discusses what Cambridge Analytical did to help Trump win the presidency.

4. They may be breached and hence criminals get your credit card and other information. This is not hypothetical.

2.4 Chapter 4: Gatekeeper—Who's in Charge Here?

If a tree falls in a forest and there is no video of it on the web, did it make a sound?

There are gatekeepers on what you can find, and what you can say, on the web. The issues raised are so pervasive that they appear in this chapter and other chapters. Here are a few they bring up, and a personal one.

1. If you do not want something you did x years ago to still be on the web, can you force Google to no longer find it? Can Google do this? Europe has a *Right to be Forgotten* law. America does not. Google is trying to abide by it but it may be difficult in some cases.

2. A hypothetical conundrum:

   (a) If someone on Twitter urges Americans to commit violent acts to change an election, that tweet will be blocked.

   (b) If the President tweets something, then it is newsworthy, so it won't be blocked.

   (c) If the president goes on Twitter and urges Americans to commit violent acts to change an election, that tweet . . .. We have here a contradiction. Which rule applies?

   (d) Fortunately this could never happen in America, but it's a good thought experiment and points to how hard it is for Facebook or Twitter to get things just right.

3. If people post false information about COVID (the book is up to date enough to include material on COVID) that could cause a public health hazard, is Facebook forced to take it down? Free Speech versus Public Health concerns versus Facebook's fear of losing users.

4. There was a proof I needed for a blog post. I could not find it anywhere on the web, so I wrote it up myself and posted it, along with two point: (1) is this proof well known? (2) it was not on the web, and now it is. One of the comments claimed that it was well known since it was in the Arora-Barak

complexity textbook, which is widely used. So here is the question: if a result is in a standard textbook but not on the web, is it well known? I honestly do not know.

Here is the post:

https://blog.computationalcomplexity.org/2020/09/a-well-known-theorem-that-has-not-been.html

2.5 Chapter 5: Secret Bits—How Codes become Unbreakable

The good news: with modern crypto we can have privacy. The bad news: few people use it, and there are ways around it. The other bad news is that terrorists use it. Should the government force companies to have secret keys (backdoors)? Would that even work? No and no. This issue was unresolved in the first edition and still is. The chapter is a good read of where we have been and where we are now.

2.6 Chapter 6: Balance Toppled—Who Owns the Bits?

Copyright is a very odd concept since (Koan 2) Perfection is normal. That is, we can make perfect copies. But even when copies were imperfect there were issues. Here is an excerpt from the pre-internet copyright law:

You can't make a public performance of a musical work unless you're an agricultural society at an agricultural fair.

Glad they cleared that up.

This chapter discusses copyright in the modern era as it relates to the internet. The story goes back to lawsuits about VCRs that foreshadow lawsuits about Napster and other services (I wonder if painters sued photographers who took pictures of their paintings and sold them). Even though the book is written factually (unbiased), the companies look awful and the government looks incompetent.

I give an example of just how broken copyright law is:

The book *Ideas that created the future: Classic Papers in Computer Science* is a collection of 46 great papers in computer science with introductions to all of them. It was edited by Harry Lewis (same Harry Lewis!) which is reviewed in this column by William Gasarch (same William Gasarch!). Their publisher had to pay to reprint some of the papers. One of them was Alan Turing's 1936 paper *On Computable Numbers, with an Application to the Entscheidungsproblem*.

1. The publisher had to pay over $3000.
2. The paper can be found at the following places for free:

   https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf

   https://academic.oup.com/plms/article/s2-42/1/230/1491926?login=true

   http://www.turingarchive.org/browse.php/b/12

   https://www.wolframscience.com/prizes/tm23/images/Turing.pdf
3. Did I find these versions of the paper in some dark corner of the dark web? No. Did I buy the web addresses in a dark alley from suspicious people? No. I Googled

   Alan Turing's 1936 paper
4. Why did I give you all four links instead of just one? Because link-rot is a big problem. For the people reading this review in 2031 (the review is on the web so this is plausible) some of those links will no longer work.

5. It is quite likely that Alan Turing, if he were alive, would be quite happy to have his paper free online. (He would be 109 years old. I found that on Wikipedia. Imagine how hard that would have been to find 30 years ago.)

6. I suspect all of the papers in the book are online for free. (The book is still worth getting. Read the review to see why.)

Clearly copyright law is not working here.

2.7 Chapter 7: You Can't Say That on the Internet: Guarding the Frontiers of Digital Expression

What can and can't one say on radio? TV? the Internet? This chapter gives a nice history of what has happened here, Much like the last chapter, some of the laws being debated here have been looked at before in other contexts. It is good to know their origins.

If Alice made a child porn movie and Bob's company distributed it, they would both be breaking the law. If Carol's trucking company takes the DVDs of the movie from point A to point B, then Carol is probably not breaking the law[5]. Is Facebook a publisher or a trucking company? Are they liable for what people post? These are hard questions to sort out. The law has not caught up to the reality; however, in this case there may be no easy answers.

Child porn and sedition are easy calls in that most reasonable people would agree they are bad and have to be dealt with. Long past are the days when an isolated utterance of *shit* or *fuck* caused a problem

That last paragraph was bullshit. The following happened to the first edition of this book. The chapter *Bits in the Air* had the following quote which they talked about in terms of the FCC possibly imposing a fine over the use of the word *shit*:

They (Bush (W) and Blair) were discussing what the UN might do to quell the conflict between Israel and militant forces in Lebanon. "See the irony is," said Bush, "what they need to do is get Syria to get Hezbollah to stop doing this shit and it's over.

The first edition was heavily used in high schools. The Texas school authorities threatened to stop using it the authors did not remove the word *shit*. The authors ended up making a version available that used *s**t*.

1. I am surprised (but pleased) that they accepted *s**t*. Everyone will know what it means.

2. Do the Texas authorities know the irony of censoring the chapter on censorship? Do they know that George Carlin's routine *7 words you can't say on TV* is available, uncensored, on YouTube: `https://www.youtube.com/watch?v=kyBH5oNQOS0`

3. The story is absolutely true: you can't make this shit up.

So we are trying to grapple with the complex issues of Free Speech vs Hate Speech vs Dangerous-to-society-speech (e.g., anti-vaxers) in the electronic age, when we have not even solved more mundane issues.

The opening part of *Bits in the Air* in the second edition is about President Trump's fondness for... hmmm, oh shit, there are some words I do not want to write down. Just Google *Donald Trump Access Hollywood Tape*. The book uses dashes to avoid spelling out the bad words.

[5]This may depend on if Carol knows that she is delivering child porn DVDs. IANAL.

2.8 Chapter 8: Bits in the Air—Old Metaphors, New Technologies, and Free Speech

At one time the radio spectrum needed the government to referee it so that stations would not interfere with each other. This is no longer true. Yet our laws still operate as though it is true and entrenched interests are resistant to change. This chapter tells that story. And more.

2.9 The Next Frontier

There is a lot in this chapter, but I will just talk about an issue that is not in any other chapters and was not in the first edition: AI decision making.

1. There are Machine Learning programs that are making decisions about college admissions, sentencing recommendations, and hiring. Since no human is involved there is no bias. Yeah!
2. There are Machine Learning programs that are making decisions about college admissions, sentencing recommendations, and hiring. Since they use past data, they reinforce past bias. Boo!

Which one of these is true? There have been many cases of clearly unfair and racist decisions made by an ML program. Why?

1. The programmers were overt racists who coded it in.
2. The programmers used past data which reflected racism of the time.

To figure out which one is true would be a simple matter of looking at the code. That last line was bullshit for two reasons:

1. The companies won't let you see the code.
2. An ML program learns on its own. It is quite likely that nobody, not even the people who write the code, know what it's doing or why.

3 Opinion

The one word that describes this book is *intelligent*. For every issue they give history, context, relevancy, and current status. If this ends up making certain people or organizations look bad, that's fine. The second edition is depressing in that things have not gotten better, and in some ways have gotten worse, since the first edition.

Who should read this book? People affected by technology should read this book. Who should not read this book: $\emptyset$.

So now the elephant in the room: Should you buy the second edition if you already have the first? I give two answers, though both are yes.

1. F**k yes! Some of the chapters are new, and some of the chapters are updated versions with new examples.

2. The first edition is available for free download at `www.bitsbook.com`. The second edition will be available for free Creative Commons download at some point. These are legal downloads (if you care). Do with that information what you will.

   I asked Harry Lewis what their business model was. He says that printing it out is more expensive than buying it, and the fact that its online will create buzz. The first edition did sell pretty well, so this alternative business model seems to be working.