

The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine

Reviewed by Richard J. Lipton

**The Annotated Turing: A Guided Tour through
Alan Turing's Historic Paper on Computability
and the Turing Machine**

Charles Petzold

Wiley Publishing, Inc., 2008

US\$29.99, 384 pages

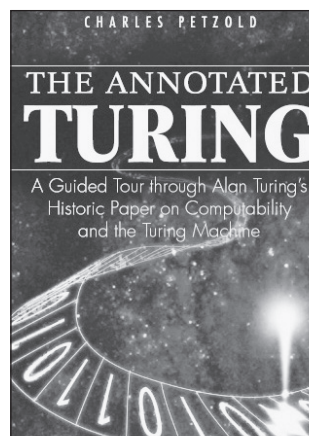
ISBN: 978-0-470-22905-7

This is a review of a “review” of the famous 1936 paper of Alan Turing. The “review” is the book by Charles Petzold entitled *The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*. I believe that Turing, with his use of diagonalization, might have liked the notion that this is a review of a review—but that is just speculation.

Turing's brilliant paper “On Computable Numbers, with Applications to the Entscheidungsproblem” created the modern notion of what it means to be computable: a number, a function—any object is considered computable if there is a machine that is able to compute it.

Turing was not the first to try to formally define the notion of computability, nor was he the last. But he undoubtedly found *the definition from the book*, to paraphrase Paul Erdős. Earlier definitions were given by Kurt Gödel, based on a suggestion of Jacques Herbrand, and another by Alonzo Church; later definitions were given by Andrey Markov and

Richard Lipton is professor and Frederick G. Storey Chair in Computing at Georgia Institute of Technology. His email address is rjl@cc.gatech.edu.



Emil Post, to name just two.

But Turing's definition, based on what we call Turing Machines (TMs) in his honor, is the right definition, the definition that is central to modern complexity. I cannot imagine the creation of the famous P=NP problem, for example, by Stephen Cook and Richard Karp with-

out TMs. Turing's paper is truly a historic paper, as Petzold calls it.

Turing also introduced a beautiful problem, the Halting Problem. He proved that this natural problem had no solution. None. Without his definition of what is computable, it would have been impossible to prove that something cannot be computed. This happens throughout mathematics: for instance, without the notion of what it means to solve a polynomial equation by radicals, Niels Abel would have been unable to prove that the general equation of degree five is unsolvable. In a similar manner, Turing shows that there is no machine that can tell if an arbitrary other machine will halt or not, hence the name “Halting Problem”.

Enter Charles Petzold, who is a famous technical writer and expert programmer. He has written an

entire book of over three hundred pages on Turing's paper. Petzold's goal is nicely stated:

Although the concept of the Turing Machine is well known, Turing's original 1936 paper is only rarely read. That's too bad, because the paper is not only a fascinating read but a milestone in the history of computing and 20th century intellectual thought in general.

I totally agree with his comment, and I would go further and argue that "rarely" is probably too weak. Nowadays the paper is almost never read.

Petzold's book is a clever mixture of background material that is needed to understand the notions used by Turing in the original paper. The obvious way to do this might have been to have an appendix with the original paper and commentary in the body of the book. But Petzold correctly surmises that few would turn to the appendix, thus defeating his whole purpose; he really wants us to see the Turing paper. He wants us to see Turing's paper with its old style; with its use of obsolete terminology, such as the notion "circle-free"—machines that keep printing output forever—with some long German phrases; and with typos too.

He achieves this by an interesting device: Turing's paper is "cut" into many small snippets, usually at most a few lines. These are typeset in the original fonts in gray boxes. Essentially the book is a snippet, comments by Petzold, a snippet, and so on. It reminds me a bit of Literate Programming due to Donald Knuth, only applied to a mathematical paper rather than to a program. Here is one snippet from the beginning of Turing's paper:

According to my definition, a number is computable if its decimal can be written down by a machine.

Turing's writing is a combination. His general comments and discussion are beautiful and a pleasure to read. His discussion and explanation of the details are of necessity more concrete and sometimes a bit hard to follow. This is where Petzold's remarks are the most helpful.

One question is, Does Petzold's book solve the problem it set out to solve, getting people to read Turing? Does it really get people to read the original paper or do they tend to skip the snippets and read Petzold's commentary? I do not know. I found the snippets often to be an interruption in the flow, but others may have a different opinion.

Another question is, Does Petzold's book present the basic material of TMs and the Halting Problem in a readable manner? I believe the answer here is a resounding "yes". The background material for each snippet not only explains the next point but also supplies interesting context and history. This

material is well written, as are his other books—his writing is crystal clear.

My last point concerns teaching and understanding the Halting Problem. To those who know the proof it is almost trivial. Assume there is a way to solve the Halting Problem: invoke the existence of a universal machine, add the diagonalization method of Georg Cantor, and get a contradiction. The proof is complete.

However, my experience in teaching this material over many years is that students follow the proof of the Halting Problem step by step, but many do not "get it". It could be me, but I have talked to many colleagues, and we agree that many students have trouble with this famous result. I think there are two major reasons the proof is hard for them to grasp: the proof uses "proof by contradiction" and there is something deep about the diagonalization method.

The reason I raise this point with respect to the book under review is that, for all its details, I still think the Halting Problem proof will be hard for many to follow. It has nothing to do with Petzold's writing, although perhaps the goals of seeing Turing's original paper and understanding the Halting Problem are not compatible. I wonder whether students will find the proof here easier or harder than the proofs in conventional textbooks. I just do not know.

There are several audiences for this book. For beginners this book might not be the best way to learn the basics of TMs and the Halting Problem, but such readers would nevertheless enjoy the careful and detailed commentary. Those who know this basic material will enjoy seeing Turing's paper flow by with the many comments and background material.