BOOK REVIEW

Title? Alan Turing and the Theoretical Foundation of the Information Age

Chris Bernhardt, *Turing's Vision: the Birth of Computer Science.* Cambridge, MA: MIT Press 2016. xvii + 189 pp. $26.95 HB. .

Paul E. Ceruzzi

Space History Department, Smithsonian Institution's National Air and Space Museum,Washington, DC.

ceruzzip@si.edu

In recent years, especially since the centenary of his birth in 2012, Alan Turing has been the subject of several books, plays, movies, and television programs. Most readers of this journal are familiar with the basic outline of his life and work: his publication in 1936 of a paper that has been heralded as describing the theoretical basis for digital computing, his World War II posting at Bletchley Park in the UK, where he helped break German codes and arguably shortened the war, his later prosecution for engaging in homosexual acts, and his tragic death at the age of 41in 1954. *Turing's Vision*, by Chris Bernhardt, a professor of mathematics at Fairfield University in the United States, concentrates on the first of those events, namely the publication of "On Computable Numbers, with an Application to the Entscheidungsproblem" in the *Proceedings* of the London Mathematical Society, series 2, vol. 42 (1936-37), pp. 230-265.

Turing did not intend to describe the design of a digital computer when he wrote that paper. As Bernhardt notes, Turing's paper is a beautiful, well-crafted argument of theoretical mathematics, which addressed one of the central mathematical questions posed at the beginning of the twentieth century. In addressing—and answering—that question, Turing proposed an abstract "machine," which he employed to advance his argument. There was little physical mechanism in the machine he described. It resembled what a human being would do to solve a problem with pencil and paper. But it did adhere to the classic definition of a "machine" in that the steps he described to mark and to move from one place to another on a recording medium (a "tape") were strictly determined, not subject to human intuition or choice. Turing created this theoretical, but inherently simple, device to advance his argument. In the course of doing that he demonstrated that a properly constructed machine of this type was in theory "universal": it need not be specially constructed to solve a specific problem. That insight turned out to be the theoretic basis for almost everything that has happened in digital computing since the 1940s. As mentioned above, a classic definition of a machine is that it is deterministic—it does what it was designed to do, and no more. But the computer extends this definition. We all know this every day, as we use personal computers (or smartphones) that can do sophisticated mathematics, manipulate text, images, sound, can communicate—all by calling up software, not by physically modifying the machine.

The implications of this insight, and Turing's role in the invention of the digital computer, are covered in the later chapters of this book. The bulk of the book is about what Turing proved in his paper, using this theoretical machine. The author argues that the consequences of that proof were as profound as the construction of the Turing Machine. The consequences of that paper are well-known, but the context of the paper, its argument, and its impact on mathematics are less so. This book addresses that shortcoming.

The author remarks that Turing's work is taught and studied today because it is considered to have laid the foundations for the theory of digital computation. It marked the beginning of an era in mathematics. Turing's 1936 paper also marked the *end* of an era, an era shaped by a number of mathematicians concerned with the foundations of mathematics. They included Bertrand Russell, Gottlob Frege, Emil Post, Kurt Gödel, Wilhem Ackermann, and above all, David Hilbert, who taught mathematics at Göttingen University in Germany. In 1900 Hilbert gave an address to the Second International Congress of Mathematicians at their meeting in Paris. In the talk he listed 23 unsolved problems that he hoped mathematicians would address—and solve—in the coming century. It was a bold address, and those problems did indeed become the focus of mathematical thought in the coming decades. Several of those problems dealt with the foundations of mathematics. Hilbert wanted mathematicians to show that the discipline was consistent: that one could not use the axioms of mathematics to derive a theorem and its opposite at the same time. Another was to show that mathematics was complete: that it would be possible to derive all true formulas from fundamental axioms. The tenth problem concerned the "Entscheidung der Lösbarkeit einer diophantischen Gleichung." In my translation, the "ability to decide whether there is a solution to a Diophantine equation." Not to find the solution, but only to determine if there was a solution or not. By the 1920s, mathematicians were addressing Hilbert's problems on all fronts. The question of "decidability" had gone far beyond the problem posed in the tenth problem, to a question of whether there was an effective procedure, what today we would call an *algorithm*, which would determine whether a problem was provable. This became known as the "Entscheidungsproblem." In the words of Hilbert and his assistant Wilhelm Ackermann, writing in 1928 in their book *Fundamentals of Mathematical Logic*, it was “the central problem of mathematical logic.”

Turing’s 1936 paper answered that question. There are problems that are fundamentally undecidable. Bernhardt devotes considerable attention in the first two chapters of this book to this historical context, to show how much those questions were central to mathematical research in the early twentieth century, and to show how significant Turing’s paper was to bring that era to a close. Mathematicians absorbed the implications of Turing’s conclusion, if reluctantly. Bernhardt argues that as the digital computer entered our lives, Turing’s paper is studied not so much for what numbers a computer can or cannot compute, but what programs a computer can execute, and more importantly, what programs a real, not theoretical, computer can execute in a reasonable run of time.

In Chapter Three Bernhardt introduces the concept of Finite Automata, which he describes as simpler versions of Turing machines. From that description he goes on to Turing Machines, in Chapter Four. While these chapters take the lay reader step by step through the development of these concepts, they do require the reader’s careful concentration. The importance of this discussion is that to understand Finite Automata and Turing Machines one must comprehend and appreciate the fundamental Church-Turing thesis, which he states on page 62: “*Anything that can be computed can be computed by a Turing machine*” (Italics in the original. The thesis has that name because another mathematician, Alonzo Church, published a similar conclusion shortly before Turing’s, although Church did not introduce the elegant notion of a machine to advance his argument). This thesis, simply stated, is the basis for so much of the computer and information age in which we now live. It was not fully understood at first, even by some of the pioneers whom designed and built the first practical computers. A failure to appreciate the practical implications of the thesis led to statements, common in the 1950s, that only a few computers would satisfy the world's needs—of course, if these expensive and fragile machines were suited for only one or two narrow applications. But computers are not machines in the classical sense; they can do whatever one can program them to do.

Current interest among computer scientists concerns the practical implications of that thesis. Turing's machine had a "tape" that was as long as needed to do a computation; real computers have a finite amount of memory. Modern computers run at very high speeds, but there are always problems such as long-range climate modeling, where no machine is fast enough. Computer programmers have to be careful not to fall into the "Turing tar-pit": a place where it is theoretically possible to solve a problem, but if programmed into a real computer, would take hundreds of years to come up with an answer (the term comes from the late computer scientist Alan J. Perlis).

Subsequent chapters delve deeper into the logical underpinnings of the Church-Turing thesis. That leads to examples of real-world problems that are, in fact undecidable. The final two chapters bring the reader into the world of electronic digital computers, invented in response to the urgent demands for computation in World War II. As we now know, and which was kept secret for many years, Turing himself worked on some of these devices, at Bletchley Park in the U.K. Many of the early computers were designed on an *ad-hoc* basis with little theoretical understanding. A collaboration between Turing and John von Neumann at the Institute for Advanced Study in Princeton in the late 1930s may have provided a bridge to the modern theory of computer design, which is often erroneously attributed to von Neumann alone (he had several collaborators).

This last chapter, on “Turing’s Legacy,” skims through a lot of material in a few pages. Readers who wish to know more about the later stages of Turing’s career and the origins of the electronic computer have a wealth of scholarship on which to draw. Nevertheless, the final chapter is lacking the rigor and conciseness of the earlier chapters. The author provides a useful essay on “Further Reading,” which lists other publications that help one understand the nature of Turing’s work and its context. Each chapter also includes a set of endnotes, which elaborate on the finer points of the mathematics covered in the body of the book. As mentioned above, the book is accessible to a general reader, but be forewarned; it will require some effort to follow the argument. That effort, however, is well worth it.