Alan Turing: His Work and Impact
Edited by S. Barry Cooper and Jan van Leeuwen
Elsevier Science, 944 pages
List price: $74.95
Reviewed by Jeremy Avigad

The year 2012 marked the centennial of Alan Turing's birth, with conferences, articles, broadcasts, and celebrations in his honor. The popular media has embraced Turing, and even in staid academic circles his status borders on sainthood. In an essay in the collection under review, Jonathan Swinton sums up the state of affairs nicely:

> Propelled by a brilliant biography (Hodges, 1992), the awareness of Turing as an iconic, field-defining figure passed from logicians and philosophers, through gay rights activists, applied mathematicians, cryptographers, artificial lifers, mainstream computer scientists, and into popular scientific culture, where he is now firmly established as a boffin of fiction, stage, and screen.

Some may wonder whether the level of praise that has been lavished on Turing is justified. Yes, the 1936 paper in which he defined what are now known as Turing machines is a landmark in the history of science, but other logicians at the time were involved in the development of formal models of computation, including Herbrand, Gödel, Post, and Church. And, although Turing played an important part in the development of the modern electronic computer, his role should not be overstated: a number of early computing devices were built from the late 1930's to mid-1940's with little or no influence of Turing's early foundational work, and the real history involves a complex mix of theoretical and practical insights, with contributions from a substantial cast of researchers and engineers alongside Turing. Finally, although the hideous abuses that Turing was subjected to as a homosexual in postwar Britain are made even more ironic by the role he played in helping the allies win the war, those abuses are no more morally offensive than those suffered by countless others whose lives are left uncelebrated. With all this in mind, it is reasonable to ask: was all the hoopla really called for?

One need only sit down with Turing's work for an hour or two, however, to find such doubts pushed aside. His papers are full of technical detail, but are surprisingly easy to read. His writing is careful and methodical, but even after all these years the work feels creatively fresh and original. He dealt with weighty topics and raised important questions, but relied on technical mastery and a good deal of hard work to back up his answers. When we read his original papers, we get a sense of Turing as an accomplished mathematician and scientist, one whose philosophical disposition, talent, and drive are well worth celebrating.

One of many publications designed to mark the centennial is *Alan Turing: His Work and Impact*, a collection that surrounds some of Turing's most important works with a mixture of scholarly commentary, introductory background, historical and scientific surveys, and personal reflections. These supporting essays are written by notable researchers in mathematics, computer science, logic, biology, cognitive science, and philosophy, as well as a few who knew Turing personally.

It may be helpful to compare *Work and Impact* to the *Collected Works of A. M. Turing*, the four volumes of which appeared between 1992 and 2001. Whereas the *Collected Works* was more than forty years in the making, *Work and Impact* was completed in less than three. And whereas the *Collected Works* is regimented and carefully edited, *Work and Impact* is a much more heterogeneous sprawl, ranging from expositions of Turing's work itself to personal accounts of the effects that Turing's contributions have had on some of the essayists. According to the introduction, *Work and Impact* encourages "a visceral engagement" with Turing's work, and reflects "a personal organic involvement" on the part of many of the authors. Reading the work of a seminal and visionary thinker inevitably encourages one to try one's own hand at speculation, and there is no shortage of that, either. For all it offers, the price is quite reasonable: you can have all 944 pages of *Work and Impact* adorning your coffee table, in hardcover, for under $75. (It is worth mentioning that there is also a smaller, more focused collection published by Clarendon Press: *The Essential Turing* features a selection of Turing's most important works, edited by B. Jack Copeland with helpful introductory notes.)

Turing is best known for his work on computability, the "Turing test," and his contributions to the British war effort in decrypting messages encoded by the German *Enigma*. But his collected works contain contributions to a much broader array of topics, both pure and applied. The *Collected Works* is divided into four volumes:

1. Mathematical logic
2. Pure mathematics
3. Mechanical intelligence
4. Morphogenesis

The four parts of *Work and Impact* roughly mirror these divisions, but with more colorful and descriptive labels:

1. How do we compute? What can we prove?
2. Hiding and unhiding information: cryptography, complexity, and number theory
3. Building a brain: intelligent machines, practice, and theory
4. The mathematics of emergence: the mysteries of morphogenesis

In the collection, more than 80 surveys and commentaries supplement two dozen or so of Turing's original works. Space does not allow me to discuss all of the supporting essays, but I will try to convey a sense of the range of topics they address, and pass along some of the observations and comments that caught my attention. Some of the more speculative pieces stand in stark contrast to Turing's deliberate, measured style of writing, and I generally preferred the more conservative analyses, as well as the essays that help situate Turing's work within our current scientific understanding. But these preferences should not be interpreted as a value judgment: like the seven-season run of *Mad Men*, everyone spending time with *Work and Impact* will have their favorite parts, and it is often more interesting to talk about the general plot lines rather than the individual scenes.

Part I contains Turing's work on logic and computability. It begins with the 1936 classic, "On computable numbers, with an application to the *Entscheidungsproblem*," which introduced what are now known as Turing machines. This is followed by a 1937 paper, "Computability and λ-definability," which demonstrates the equivalence of Turing computability with Church's notion of computability by λ-terms, and the Herbrand-Gödel-Kleene notion of a general recursive function. In the years after World War II, these papers played a key role in laying the foundations for a theory of computation, which was needed to make sense of and support the technology that has transformed our lives so dramatically. These early works are so iconic that providing informative commentary is no easy to task, but a number of essays in the collection rise to the occasion.

It is interesting to note, however, that Turing's early work dealt with the foundations of mathematics as much as with the foundation of computing. In the wake of Gödel's discovery of the incompleteness phenomena, Turing wrote papers on truth and ordinal logics, which provide ways of extending a formal system with stronger axioms. These are helpfully explained in articles by Michael Rathjen, Solomon Feferman, and Philip Welch. Turing was interested not only in logical strength but in the *expressivity* of formal languages, as is made clear in his essays "Practical forms of type theory," and "The reform of mathematical notation and phraseology." The latter begins amusingly as follows:

> It has long been recognised that mathematics and logic are virtually the same and that they may be expected to merge imperceptably into one another. Actually this merging process has not gone all that far, and mathematics has profited very little from researches in symbolic logic. The chief reasons for this seem to be a lack of liaison between the logician and the mathematician-in-the-street. Symbolic logic is a very alarming mouthful for most mathematicians, and the logicians are not very much interested in making it more palatable. It seems however that symbolic logic has a number of small lessons for the mathematician which may be taught without it being necessary for him to learn very much of symbolic logic.

The particular small lessons Turing discusses in this essay involve ways that Church's simple type theory provides a good framework to encode ordinary mathematical assertions and make them precise. In his associated commentary, Stephen Wolfram underscores the importance and difficulty of making sense of informal notation, and relates the challenge to the design of computer algebra systems.

Part II, which surveys the range of Turing's mathematical contributions outside of logic and the theory of computability, offers more surprises to those only casually familiar with Turing's work. For example, few may be aware that, as an undergraduate, Turing did substantial work in probability. After attending a series of lectures in 1933 in which the astrophysicist Sir Arthur Stanley Eddington noted that repeated experimental measurements are often normally distributed, Turing set himself the task of providing a rigorous explanation. Within a few months, he had proved a general form of the central limit theorem for independent but not necessarily identically distributed variables. What he had discovered was a version of a theorem that the Finnish mathematician Lindeberg has proved 12 years earlier, as well as part of a converse, due to Feller and Lévy, which had not yet been published at the time. These results are now textbook standards, and fundamental to modern probability. And Turing discovered them as an *undergraduate*.

Others will be pleased to learn that Turing had a longstanding interest in the Riemann zeta hypothesis, an interest he shared with Littlewood's student, Stanley Skewes, a regular rowing partner of his in Cambridge. Turing introduced a number of important ideas, including algorithmic methods for computing the zeros of the zeta function that are now standard in computational branches of analytic number theory. Denis Hejhal and Andrew Odlyzko do a fine job of explaining the intricacies of Turing's methods, but the take-away message is that Turing was no mere dabbler: when he picked up a problem, he threw himself into it, with ingenuity and hard work.

Part II contains excerpts from, and commentaries on, Turing's 1940 paper on the Enigma. It also contains Turing's 1944 notes on the design of a system, "Delilah," for voice encryption. A few years later Turing turned his attention to the burgeoning field of electronic computers, whose early days are colorfully described in an article by Toby Howard. The first digital stored-program computer was designed and built at the University of Manchester, and known as the "Baby." Its first program ran on June 21, 1948, and was designed to take an input, N, and compute the largest proper divisor. Within a few weeks, the program was able to compute the largest factor of $N = 218$. It took the Baby, however, 52 minutes to complete the task.

Turing had accepted a post in Manchester in May 1948, but he did not arrive until October 1948, having written programs for the Baby in the interim. *Work and Impact* contains a remarkable report written by Turing, "Checking a large routine," which discusses strategies for verifying the correctness of a computer program. As Cliff B. Jones correctly points out in his essay, this foreshadows the study of methods of program verification, an ongoing and active research area in computer science. Part II also devotes 7 pages to excerpts from Turing's 90-page programmer's handbook for a successor to the Baby, known as the Ferranti Mark I. Though the machine code was rudimentary, the excerpt reads much like any contemporary introduction to programming, with advice on how to break a problem down to components and modularize the task through the use of subroutines.

Part II also contains Turing's 1949 paper that shows that the word problem in semi-groups with cancellation is undecidable. Having shown in 1936 that the halting problem is algorithmically unsolvable, here he was concerned with exploring the extent to which unsolvability arises in traditional mathematical questions. The work improves on the prior result, due to Post and Markov independently, that the word problem for semi-groups is unsolvable. The result was later extended to groups by Novikov and Boone independently, in the mid-1950's. There is now, of course, a well-developed theory of undecidability phenomena in mathematics. One of the crowning achievements is the result of Matiyasevich, Robinson, Davis, and Putnam, who showed that there is no algorithm to determine whether a given diophantine equation has a solution in the integers, thereby providing a negative answer to Hilbert's 17th problem.

The last work by Turing presented in this section is a paper called "Rounding off errors in matrix processes." As Lenore Blum explains, this paper introduces the notion of the *condition number* of a matrix, a quantity which measures the extent to which calculations with linear systems of equations are subject to rounding errors. Both the problem of managing rounding errors and the notion of condition number itself are now central to numeric computation.

Part III turns to Turing's work on mechanized intelligence, with such landmark papers as "Intelligent machinery," "Computing machinery and intelligence," and "Digital computers applied to games." Discussions of the Turing test are now commonplace in undergraduate philosophy classes, but the essays make clear that the idea was not just a passing fancy for Turing. These writings are among his most speculative and most prescient. "Intelligent machinery" in and of itself inaugurated the field of artificial intelligence, distinguished clearly between discrete and continuous models of computation, and even foreshadowed the study of neural networks and machine learning.

The essays are also among his most provocative. "Computing machinery and intelligence," which proposes the Turing test, addresses a wide range of objections to the thesis that machines can exhibit intelligent behavior, including theological, philosophical, and mathematical arguments. Responding to a sceptical 1947 lecture, "The Mind of Mechanical Man," by the British neurologist and neurosurgeon Geoffrey Jefferson, Turing challenges us to imagine how we might react to a machine that could not only compose a sonnet, but also respond to an interrogator's queries about the choice of words and the sentiments expressed. But even when Turing was being provocative, he was making a clear scientific point. Recognizing that questions as to whether machines can think are not only vague but also loaded with social and religious implications, Turing deftly shifted focus to the much more scientifically fruitful question as to whether machines can exhibit intelligent behavior. (The Turing test was designed to frame that question in even more pointed observational terms.) The essays moreover provide a thorough and thoughtful exploration of what it *means* to exhibit intelligent behavior, ranging from carrying out mathematical calculations and playing games to composing sonnets and even engaging in frivolous banter. The first step in developing a new science is to gain clarity as to the phenomena one is trying to model and explain, and Turing did an admirable job in mapping out the landscape.

But Turing went further than that, with concrete proposals as to how various types of intelligent machine behavior might be obtained. For example, “Digital computers applied to games” presents an explicit two-move lookahead algorithm for playing chess, and a “run” of this algorithm against a novice player, with Turing playing the role of the computer and simulating the algorithm by hand. In this way, Turing’s essays on machine behavior are as scientifically substantive as they are entertaining. As a result, it is fitting that almost two dozen supplementary essays in Section III are devoted to exploring the significance of this work.

The historical data presented in Part II makes the essays in Part III even more poignant. It is one thing for us, today, to take a break from shuttling between Google and Siri to speculate on the nature of machine intelligence. It is yet another thing entirely to spend as much time as Turing did writing machine code to compute basic numeric operations, and come away from the experience with the conviction that computers are capable of exhibiting intelligent behavior.

Part IV deals with Turing’s work in biology, centered on the single long paper, “The chemical basis of morphogenesis.” The paper was published in 1952, just after Turing was tried and convicted for homosexual activities, and is reproduced in full. It addresses a fundamental question: how do structured and complex organisms, like plants, animals, and humans, develop from a single cell? Somehow, the information and mechanisms stored in the original cell have to govern a reproductive process in which future cells are created and assume radically different roles.

One might speculate about the path that led Turing to this work. Although Crick and Watson did not discover the structure of DNA until 1953, the paradigm of classical genetics — based on the Mendelian notion of a gene — was firmly in place in the early 1950’s, when Turing took up the project. What was called for, it would seem, is an analysis of genes as a method of encoding computer programs powerful enough to orchestrate the growth process.

But this is emphatically *not* the approach the Turing took. He was not one to use a tool or method just because it was near to hand. He approached every problem afresh, distilled it down to what he took to be the essentials, and adopted the methods that he felt were most appropriate. In the case at hand, Turing was more interested in the physical and chemical processes that could give rise to such complex behavior, and he found the language of dynamical systems and partial differential equations more appropriate to the analysis.

Turing's approach to the problem is intricate, but his exposition is, as usual, clear and methodical. The supplementary essays in Part IV, especially the surveys by Henri Berestycki and Hans Meinhardt, James D. Murray, Peter Saunders, and Jonathan Swinton do a good job of explaining the scientific substance of Turing's work and its contemporary relevance. We do not often think of Turing as an influential biologist, but, as Saunders points out, the paper has had a tremendous impact, with well over 3,000 citations listed on the ISI Web of Science. And although the general consensus is that the processes that Turing described cannot be the entire story, the mechanisms that underly morphogenesis are still poorly understood today. Which is to say, more than 60 years later, it is still too early to fully assess the importance of Turing's contributions to biology.

The editors of *Work and Impact*, S. Barry Cooper and Jan van Leeuwen, are right to emphasize the rewards of a personal engagement with Turing's work. The sheer magnitude of Turing's accomplishments is impressive; as Rodney A. Brooks notes in his essay:

> It is humbling to read Alan Turing's papers. He thought of it all. First.

But reading Turing is as encouraging as it is humbling. When we spend time with his work, we become keenly aware that Turing was not a comic-book superhero or a demigod; he was a pure and applied mathematician, practicing his craft, like the rest of us. He was exceedingly clever and talented, but he was also careful, methodical, thoughtful, and hard-working. He sought out questions and problems that he felt were interesting and important, and pursued answers with energy, focus, creativity, and élan. He was not afraid to entertain big ideas, but big ideas by themselves are light and airy things, and Turing did the work that was needed to make them substantial. There is a lot to be learned from his outlook and temperament, and *Alan Turing: His Work and Impact* can serve as both an instructional and an inspirational text.

Acknowledgments. I am grateful to Thomas Hales and Robert Lewis for very helpful comments.