Book Review

The Essential Turing

Reviewed by Andrew Hodges

The Essential Turing
A. M. Turing, B. Jack Copeland, ed.
Oxford University Press, 2004
US$29.95, 662 pages
ISBN 0198250800

The Essential Turing is a selection of writings of the British mathematician Alan M. Turing (1912–1954). Extended introductions and annotations are added by the editor, the New Zealand philosopher B. Jack Copeland, and there are some supplementary papers and commentaries by other authors. Alan Turing was the founder of the theory of computability, with his paper "On Computable numbers, with an application to the *Entscheidungsproblem*" (Turing 1936). This, a classic breakthrough of twentieth century mathematics, was written when he was twenty-three. In the course of this work in mathematical logic, he defined the concept of the universal machine. As he himself put it, digital computers are practical versions of this concept; and he himself created an original detailed design for an electronic computer in 1945–46. His 1936 analysis of mental rule-based operations was the starting point for his later advocacy of what is now called Artificial Intelligence. His paper "Computing machinery and intelligence" (Turing 1950a) is one of the most cited in modern philosophical literature. His paper in mathematical biology (Turing 1952) then inaugurated a new field in nonlinear applied mathematics. But he was also the leading scientific figure in the British codebreaking effort of the Second World War, with particular responsibility for the German Enigma-enciphered naval communications, though this work remained secret until the 1970s and only in the 1990s were documents from the time made public.

Andrew Hodges is lecturer in mathematics at Wadham College, University of Oxford. His email address is `andrew.hodges@wadh.ox.ac.uk`.

Many mathematicians would see Turing as a hero for the 1936 work alone. But he also makes a striking exemplar of mathematical achievement in his breadth of attack. He made no distinction between "pure" and "applied" and tackled every kind of problem from group theory to biology, from arguing with Wittgenstein to analysing electronic component characteristics—a strategy diametrically opposite to today's narrow research training. The fact that few had ever heard of him when he died mysteriously in 1954, and that his work in defeating Nazi Germany remained unknown for so long, typifies the unsung creative power of mathematics which the public—indeed our own students and our colleagues in the sciences—should understand much better.

Many therefore will welcome this new edition and the increased availability of Turing's work. But the foregoing remarks should make it clear that defining the Turing *oeuvre* is not straightforward. There is no default option of reproducing published papers and compiling them under a new cover. There is a spectrum ranging from formal publication to reports, talks, unpublished papers, unfinished work, letters, and several areas where other people developed work that he had inspired. Choices here are not easy. Nor it is straightforward to define a genre or field in which to place his work, and the usual criteria of important papers in leading journals are of no use. Turing ignored conventional classifications, and created work which would now be described as at the foundations of computer science or the cognitive sciences, areas which in his time had no clear names of their own.

On top of this, there is the difficulty that his war work was never written for publication and exists only in operational reports which have been released in a chaotic fashion. Indeed, even since this collection was published, Turing's report on his advanced electronic speech scrambler, with the wonderful date 6 June 1944, has emerged (Turing 1944).

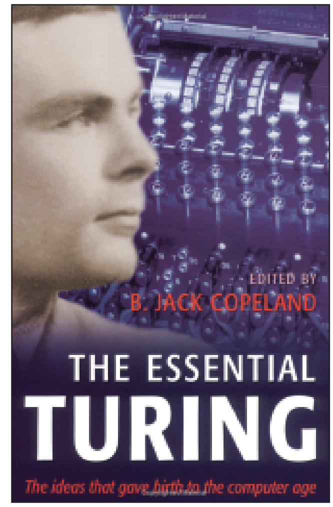
In the last decade the World Wide Web has transformed access to original Turing documents. In particular, the collection of Turing's papers at King's College, Cambridge University, is now accessible at `http://www.turingarchive.org`. Those interested in original material are now much less dependent on the work of editors and publishers. Even so, a printed source-book will be valued by many to whom Turing is a somewhat legendary figure, often cited but not easy to look up and quote. Such a book has enormous potential to educate and to inspire.

Copeland's Anthology

We now come to Copeland's own editorial choices. This review will consider points of interest in an order roughly corresponding to the order of Copeland's anthology, which in turn reflects Turing's chronology. But one point should be made clear at the outset. Oxford University Press bills this as "the first purchasable book by Turing", but Copeland's volume is not the first edition of Turing's papers and not the most complete. A four-volume *Collected Works of A. M. Turing* was published by Elsevier (Turing 1992, 2001). This work, totalling some 938 pages, resulted from the protracted collaboration of distinguished mathematicians and computer scientists. Particularly notable is the volume where John L. Britton, as editor, annotated Turing's pure-mathematical work in line-by-line detail.

Unfortunately, little effort was made to promote the *Collected Works*, and the high price guaranteed it few sales, even to university libraries. For this reason, Copeland's new collection, offered at a paperback price, makes a good part of Turing's work much more accessible in practice. It is still odd, however, that Copeland virtually ignores the *Collected Works*. Indeed, an inattentive reader, missing the small print on pages 409, 510, and 581, would remain ignorant of it.

This omission is compounded by another: Copeland does not list Turing's works, so the reader cannot even guess how complete his selection is. There are quotations from some of Turing's papers which have not been included in the work, but there is no overview of Turing's output, nor any explanation of what is considered "essential" and why. Yet it would have been simple for Copeland to explain the relationship of his collection to the existing edition, and to refer readers to it for other (presumably "inessential") Turing works. Indeed, Copeland could justly claim to have advanced upon the *Collected Works* in some areas—as he does when detailing transcription errors in (Turing 1948a). He has included more technical Enigma material than the *Collected Works* did, as well as Turing's late talks on machine intelligence, which Copeland finds particularly important. But the inclusions and exclusions are nowhere systematically listed.

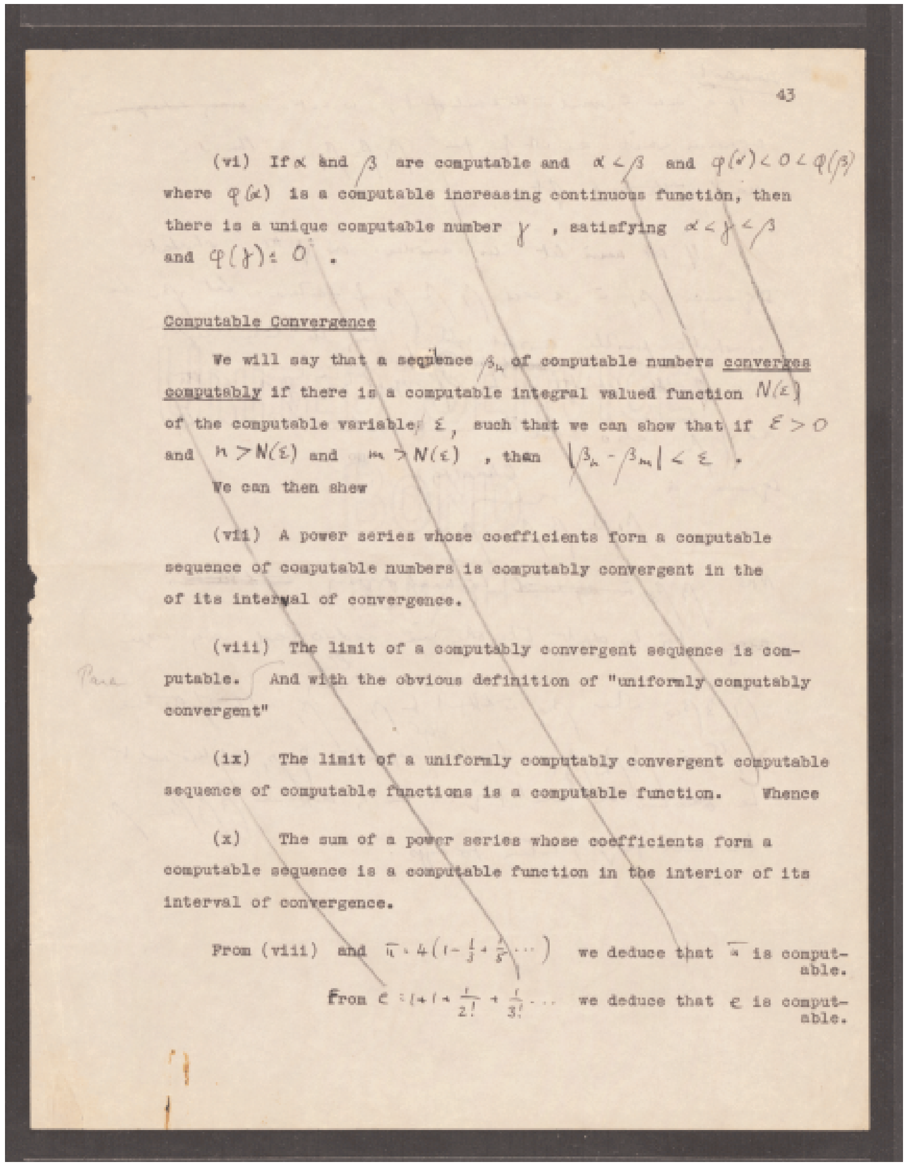


Copeland's edition has all the papers reset in a uniform typography: some readers will always prefer to see the original format and this decision means the loss of original page references. But Copeland is certainly no slouch when it comes to textual detail. For example, he devotes nearly a page to discussion of the spelling of the word "program".

More problematic is the central question of what is "essential". To illustrate how differently the "essential" Turing may appear in different eyes, it is worth recalling the survey of the topologist M. H. A. (Max) Newman, written for the Biographical Memoirs of the Royal Society after Turing's death (Newman 1955). In some ways Turing's mentor and father figure, Newman interestingly defined him as "at heart more of an applied than a pure mathematician" and devoted serious attention only to his mathematical papers. Of course, Turing's war work was then totally secret, but even so Newman's characterization of it as a cruel loss to science was somewhat severe. Computer design and Artificial Intelligence received the briefest of mentions. This was too narrow a mathematical viewpoint, but it did reflect, perhaps, that *sub specie aeternitatis* aspect of mathematics in which Turing shared: he threw himself into the war effort (as did Newman) but never, even in its darkest days, forgot that he was a serious mathematician. In contrast to Newman, Copeland highlights the Enigma cipher machine as the subject of his second main section, the third focal point being Artificial Intelligence and the Turing Test.

Computability and Logic

In one respect, however, Copeland is entirely in unison with Newman, and that is on the topic of computability, which forms the first main section of his volume. *The Essential Turing* includes not only "On Computable Numbers", but also part of a paper by Emil Post which gives some corrections, and another technical commentary by Donald Davies. (For



43

(vi) If α and β are computable and $\alpha < \beta$ and $\varphi(\alpha) < 0 < \varphi(\beta)$ where $\varphi(\alpha)$ is a computable increasing continuous function, then there is a unique computable number γ, satisfying $\alpha < \gamma < \beta$ and $\varphi(\gamma) = 0$.

<u>Computable Convergence</u>

We will say that a sequence β_n of computable numbers <u>converges computably</u> if there is a computable integral valued function $N(\varepsilon)$ of the computable variable ε, such that we can show that if $\varepsilon > 0$ and $n > N(\varepsilon)$ and $m > N(\varepsilon)$, then $|\beta_n - \beta_m| < \varepsilon$.

We can then shew

(vii) A power series whose coefficients form a computable sequence of computable numbers is computably convergent in the of its interval of convergence.

(viii) The limit of a computably convergent sequence is computable. And with the obvious definition of "uniformly computably convergent"

(ix) The limit of a uniformly computably convergent computable sequence of computable functions is a computable function. Whence

(x) The sum of a power series whose coefficients form a computable sequence is a computable function in the interior of its interval of convergence.

From (viii) and $\pi = 4(1 - \frac{1}{3} + \frac{1}{5} \cdots)$ we deduce that π is computable.

From $e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} \cdots$ we deduce that e is computable.

The image above is AMT/C/15/01c.2 at http://www.turingarchive.org/browse.php/C/15, on the website of the Turing Archive of King's College, Cambridge University. Permission to publish has been granted by P. N. Furbank, executor of Turing's will.

A page from the typescript of Alan Turing's classic 1936 paper, "On computable numbers, with an application to the Entscheidungsproblem".

This text appeared on page 256 of Turing's published paper, with some very minor textual changes. The typewriting is not his, but the inserted mathematical expressions are in his own hand. The lines drawn through the material may mean that it had been retyped, or merely that it was finished with and could be recycled as scrap paper (see below).

The results on this page show that "computable numbers" include all the real numbers that normally arise in mathematics through limit definitions. Although Turing's paper is usually thought of as concerned with the discrete world of mathematical logic, Turing wanted to connect computability with the mainstream of continuous analysis. In fact his opening remarks rather rashly asserted that he would soon give a theory of real functions based on the concept of computable numbers. Turing subsequently abandoned this ambition, leaving it to modern theorists of "computable analysis" to follow up. However, his later note (Turing 1937) made a first step in this direction.

Only six pages of this typescript survive in the Turing Archive at King's College, Cambridge. Their existence has been overlooked because they were used as scrap paper: the reverse sides contain Turing's manuscript for another paper, "A note on normal numbers" (Turing 1936?). This other paper was never published, but there is a modern transcription and detailed annotation in the *Collected Works*. It was probably stimulated by the work of his friend David Champernowne (1933). Champernowne noticed that the number .123456789101112131415... is normal in base 10, meaning that its digits and groups of digits are all uniformly distributed in the infinite limit. In attempting to generalize this result, Turing found himself giving constructive definitions of infinite decimals. It seems quite possible that Turing considered this question around 1933–34 and that it influenced the approach he took in 1935 when he formulated his definition of a computable number.

—*Andrew Hodges*

References:

D. G. Champernowne (1933), The construction of decimals normal in the scale of ten, *J. Lond. Math. Soc.* **8**, 254-260.

A. M. Turing (1936), On computable numbers, with an application to the Entscheidungsproblem, *Proc. Lond. Math. Soc.* **42**(2), 230-265.

A. M. Turing (1936?), A note on normal numbers, manuscript and typescript available at http://www.turingarchive.org, item C/15. Text in *The Collected Works of A. M. Turing: Pure mathematics*, J. L. Britton (ed.), North-Holland, 1992.

A. M. Turing (1937), On computable numbers, with an application to the Entscheidungsproblem. A correction, *Proc. Lond. Math. Soc.* **43**(2), 544-546.

comparison, Martin Davis's collection of classic papers *The Undecidable* (Davis 1965), recently republished, only offers Post's paper, but offers all of it.) Copeland also gives good didactic material on Hilbert's formalist program and the working of the universal machine. Turing's formidable 1939 paper on ordinal logics (his 1938 Princeton Ph.D. thesis, as supervised by Church) is also included. This issue of the *Notices* includes a review of this paper by Solomon Feferman and a further article by Martin Davis on "Turing Reducibility". In this case it is probably beyond the powers of any editor to make the technical apparatus accessible. Copeland does not try to explain the idea of the lambda-calculus, and does not offer the mathematical content that the authors in the *Notices* supply, but succeeds in giving a clear survey. His treatment is enhanced by the inclusion of some previously unpublished correspondence with Newman from the King's College archive.

Mathematics and Cryptography

After this first section, Turing's mathematical work is marginalised, and the message seems to be that mathematics is less than essential. An example comes in Turing's interest in probability theory. Turing's first substantial research work was an independent proof of the Central Limit Theorem—unmentioned by Copeland, but given an excellent review by Zabell (1995). This won his Fellowship of King's College in 1935. But perhaps more importantly, probability theory was the key to his advanced cryptanalytic methods, which made cryptography into a science. Turing developed new Bayesian inference methods for the Enigma decipherment problem, work in which he was assisted by I. J. (Jack) Good after 1941. Good became a distinguished mathematician and statistician, and his book *Probability and the Weighing of Evidence* (Good 1950) expounded and developed the material that Turing originated but never wrote in his own name. ("Weight of evidence" is essentially equivalent to Shannon's measure of information, which Turing formulated and used independently.) In the *Collected Works*, this work was well accounted for, thanks to Good's wealth of historical material (Good 1992, 1993, 2001). It has inspired modern developments (Orlitsky et al. 2003). Yet the entire subject of probability and statistics is virtually unmentioned in *The Essential Turing*. This is rather like telling the story of the atomic bomb without mentioning nuclear physics. Because of this omission, Copeland does not justify his claim (p. 2) that in *The Essential Turing* "the full story of Turing's involvement in the Enigma is told for the first time."

However, Copeland gives a full description of the Enigma machine and of the early Polish and British methods for deciphering it. The power of Turing's applied logic comes through in his beautiful "simultaneous scanning" method to defeat the plugboard complication of the Enigma. For this, Copeland supplies the pertinent excerpt from Turing's 1940 technical report, as eventually released in 1996, supplemented by a part of A. P. Mahon's more readable internal history of the work. He does not include Mahon's striking conclusion: this gave a mathematician's apology for war work which probably also reflects Turing's sentiments. "While we broke German Naval Cyphers because it was our job to do so and because we believed it to be worthwhile, we also broke them because the problem was an interesting and amusing one. The work of Hut 8 combined to a remarkable extent a sense of urgency and importance with the pleasure of playing an intellectual game."

This omission of Bayesian inference methods also weakens Copeland's claims about the genesis of Artificial Intelligence in wartime Bletchley Park. Copeland argues that the serial trial of a million or so Enigma rotor positions lies behind the identification of "search" in (Turing 1948a) as a concept central to "intelligent machinery". This is an unnecessarily weak link on which to hang the claim. Such brute force "search" was the bluntest of instruments in codebreaking. A more substantial point lies in Turing's successful mechanization of judgment through his quantified "weight of evidence", prefiguring the sophisticated Bayesian inference programs used today in AI applications.

Mathematics and Computer Science

More generally, the hinterland of mathematical theory and practice, as the basis and motivation for advances in computing, is weakly represented. Turing not only worked on computable numbers in the abstract: he knew all about computing numbers in practice. As the *Notices* article by Andrew Booker describes, in 1937–39 Turing developed new methods for investigating the Riemann zeta-function, which led to a need to compute its zeros: for Newman, the abandonment of such work was the cruel blow dealt by the war. But Copeland never mentions complex analysis, nor the special machine Turing designed for computing the zeta-function, nor his 1950 computer program superseding it. It is striking that the first thing Turing did in 1950, when he was able to use one of the world's first computers, was to use it to investigate the zeta-function. (In contrast, he did no experimental work with computers on Artificial Intelligence).

The exclusion of mathematics gives a lopsided view of Turing's mind at work. Thus his pre-war connection with von Neumann through research in continuous groups (Turing 1938), and the development of computability within mathematics (Turing 1950b) go unmentioned. So does Turing's work in the numerical analysis of matrix inversion

(Turing 1948b), although it led the way in showing the viability of numerical methods for large-scale applied mathematical problems and thus made a start on the serious analysis of algorithms. (The work of Higham (1996) has drawn attention to the importance of this work.) The "program proof" of (Turing 1949), anticipating ideas of the 1960s, plays no role: for Copeland the only "essential" topic in computing is Artificial Intelligence.

In defence of this narrow focus it could be said that Turing's central motivation in 1945 did not lie in standard mathematics, nor in practical computer science, but in wanting to build a computer as "a brain". But Turing knew a great deal about the relation between mathematics and the physical world, discreteness and continuity. This knowledge was inseparable from his prospectus for computing and for Artificial Intelligence. Turing's central idea of modelling the brain brought him to consider the approximation to continuous systems by the discrete, including chaotic and thermodynamic effects (Turing 1948a, 1950a). Thus his background as an all-purpose mathematician, rather than as a verbal philosopher, is still important even if this narrow remit is accepted.

Origin of the Digital Computer

Turing's ambition to "build a brain" brings us to the question of his 1945–46 technical proposal for the Automatic Computing Engine (Turing 1946)—the first really detailed electronic computer design and prospectus for what a computer could do. This report was omitted in Newman's 1955 memoir and has had serious recognition only since the 1970s. In this neglect, it stands in complete contrast with the June 1945 "Draft report on the EDVAC" by von Neumann which has always been regarded as the *fons et origo* of the computer. Indeed histories of computers too often tell a story of engineered machines and American corporate history, from Hollerith to microprocessors, without any references to Turing at all. Copeland has previously done much to advance Turing's claim (and British-based work generally), and *The Essential Turing* is billed on its cover as giving "The ideas that gave birth to the computer age." It is therefore odd that the ACE report, the first detailed prospectus for an electronic computer, is omitted. Various bits are quoted, but they do not allow the reader to judge Turing's total vision.

Interdisciplinary culture clashes abound in the question of the origin of the computer. Some computer historians consider the use of electronic components to be the crucial innovation. Binary number representation is often held to be a breakthrough, and I have been surprised to hear Martin Campbell-Kelly, a leading figure in this field, suggest that Turing needed to learn this idea from von Neumann. Copeland, in contrast, focuses very clearly on the principle of the universal machine as the crucial factor. This is unlikely to change the minds of those who consider other issues to be paramount, but it is consistent with Turing's approach.

To argue that Turing's logical work was critical to the modern digital computer, Copeland discusses the concept of stored-program computers in his introduction to (Turing 1936). In fact he virtually *identifies* the concept with the universal machine, by giving Turing's own later account of the connection (Turing 1947). The danger here is that of being ahistorical: that is, forgetting that the 1945 now far in the past was in 1936 far in the future. But Copeland certainly makes a strong argument that von Neumann knew of Turing's ideas and used them, without citation, in the EDVAC report. This is a difficult topic; although von Neumann certainly spoke clearly after 1945 of the importance of Turing's universal machine, there is little to document its influence in the formative period. His strongest evidence remains the statement Brian Randell got from Stanley Frankel long ago (Randell 1972): that "in or about 1943 or '44 von Neumann was well aware of the fundamental importance of Turing's paper of 1936…" More recently Martin Davis has also tackled this problem in his book *The Universal Computer: The Road from Leibniz to Turing* (Davis 2000), giving a vivid discussion of von Neumann's debt to Turing. This is the judgment of a unique source who was immersed in the logic and computer worlds of that early period. Copeland weakens the case for Turing's influence by making no reference to Davis's analysis.

The Essential Turing also makes another argument about the origin of the digital computer, concerned with the question of assigning credit for the Manchester machine which, though tiny, was the world's first working stored program computer in June 1948. But a general problem in both discussions of origins is that Turing never gave a full analysis of his own contribution—in particular, the insight that a program is itself a form of data and can be treated by the computer as such. On the first page of his 1945–46 plan, Turing said that control of an entire calculation could be "looked after by the machine itself". Into this phrase we may read the future of subroutines, languages, compilers, and operating systems; but Turing himself did not spell out those implications systematically. The word "itself" is a Gödelian self-reference; it comes from thinking of programs as data for other programs, but Turing never enlarged his observations on this logical ancestry. Turing used program modification immediately—he described how one could "pretend that the instructions were really numbers"—but did not take the opportunity to explain that such "pretence" embodied an essential and novel principle. Even when making much of the potential of program modification for "learning", he never gave a serious analysis of the program-as-data aspect. Later, while at Manchester, Turing (1950a) wrote that Babbage had "all the essential ideas", thus undermining the appreciation of program-as-data as a crucial advance.

Unfortunately, Turing never gave more than a brief indication of how his 1936 theory led to his 1945–46 design. Secrecy would have inhibited a full account, though Good's example in publishing so much probability theory shows that there was no total bar on the exposition of general principles. However, such a contribution from Turing was hardly encouraged by the embryonic computer industry. In 1953 a semi-popular book *Faster than Thought* (Bowden 1953) surveyed British computing, with Babbage as the star. In contrast, the editor included a philistine "glossary" entry on Turing thus:

> **Türing machine:** In 1936 Dr Turing wrote a paper on the design and limitations of computing machines. For this reason they are sometimes known by his name. The umlaut is an unearned and undesirable addition, due, presumably to an impression that anything so incomprehensible must be Teutonic.

It is not difficult to decode the word "incomprehensible" as meaning "mathematical". A more common view is that Turing's contribution is comprehensible, but purely theoretical. Students often gain the impression that Turing was never connected with anything as vulgar as an actual computer. This is the reverse of the truth: Turing avidly desired the practical business of design and construction. In fact rather than let his claim depend solely upon the abstract principle of 1936, it would be better to emphasise that from 1943 onwards he was in effective command of every aspect of making that principle into a practical proposition—scientific, technological, organizational, motivational. It was in that grasp of its potential that he was the inventor of the computer: that was the essential Turing. In particular his vision for the future of software engineering, based on his deeper understanding of the universal machine being able to "look after itself", was ahead of von Neumann's. Without seeing Turing's ACE report, readers cannot judge his place in the history of the "practical" universal machine. A related point is that Turing's practical wartime experience with digital machinery was crucial, and it would have been worth including some documentation of this experience from his wartime reports. Copeland marginalises these questions, because his attention is concentrated on Artificial Intelligence.

Artificial Intelligence

Copeland is right to emphasise that well before 1956, when the Dartmouth conference inaugurated "Artificial Intelligence" as a research area, Turing had developed strong lines of research, both top down and bottom up, in modern parlance. (Nor was Turing alone in the British scene—he was one of a very lively group of "cybernetic" pioneers.) But as in other ways, Turing suffered the consequences of his own self-effacing reticence. Turing never published the neuron-inspired networks he sketched in (Turing 1948a), even though they played a role in motivating the arguments in his famous 1950 paper about the possibility of learning machines. Nor, oddly, did he try them out when the fully engineered Manchester computer became available. Here another surprising omission comes in Copeland's discussion. Copeland and Proudfoot (1996) drew fresh attention to Turing's 1948 neural architectures (which had been published in 1968 and 1969, and then in the *Collected Works* in 1992). A young computer scientist, Christof Teuscher, then did Ph.D. work in implementing and exploring them; this has won several awards, including one from the European Research Consortium for Informatics and Mathematics. His publications (Teuscher 2002, 2004) are not referenced by Copeland.

Mathematics and Biology

Of Turing's theory of morphogenesis, only the published work (Turing 1952) is included by Copeland. Turing left much more work unfinished at his death. The editor of the *Morphogenesis* volume of the *Collected Works*, P. T. Saunders, edited and included the most coherent parts of these manuscripts. Copeland does not include any of this. His footnote does make a rare concession to the existence of the *Collected Works*, and he does cite the much more extensive work by Jonathan Swinton (2004), which appeared in another volume of Turing-inspired studies, *Alan Turing: Life and Legacy of a Great Thinker*, edited by Christof Teuscher (2004). But Copeland gives no hint of the content of these other scholars' work, preferring to concentrate on his own interpretation of the material in (Turing 1952).

That interpretation is, unfortunately, skewed by Copeland's insistence on describing Turing's theory as "Artificial Life", a term coined in the 1980s, and his linking of it to genetic and evolutionary algorithms. But these "a-life" developments are much closer to von Neumann's ideas; Turing's work in mathematical biology was essentially complementary. Genetic algorithms explore the logic of evolution without the constraints of physical embodiment. Turing's work attacked the question of what paths could be physically available for evolution to exploit. It was rooted in physical chemistry and used techniques in nonlinear differential equations, entirely different from discrete logic. Swinton has called it "good old-fashioned applied mathematics". There is a slender connection with Turing's machine-intelligence ideas, through the question of brain growth, but his morphogenetic theory really forms quite a different field of enquiry.

Philosophy of Computation

The absence of Turing's unpublished work in mathematical biology may disappoint some readers, but applied mathematics is not Copeland's strength, and it is not surprising that it is somewhat sidelined. His emphasis is naturally on the philosophy of computation, and this is the area where one would expect the greatest expertise. This is, indeed, the topic emphasised by Copeland in his discussion of Turing's late writing on AI, which forms his main claim to new scholarship. In particular, a topic central to Copeland's approach is the discussion of *Church's thesis* concerning the definition of "effectively calculable". Turing entered into this subject in a somewhat awkward way: in 1935 Church proposed a definition of effective calculability in terms of the lambda-calculus. Turing's 1936 definition of computability turned out to be mathematically equivalent, and he had to write an appendix to his paper showing this, so delaying his publication. But Church in turn accepted that Turing's analysis of computation gave a far more direct and intuitive argument for why this definition should be made. It is common now to refer to this joint position as the *Church-Turing thesis*.

In many earlier articles, e.g., (Copeland 2000, 2002), Copeland has made very distinctive claims about the Church-Turing thesis. Surprisingly, he has not made these claims so prominently in *The Essential Turing*: it is more that they lurk behind the prefaces and annotations. But they deserve review here nevertheless: it is important for mathematicians to be aware of what philosophers are making of their work, and students of this volume should be aware that the "Further Reading" recommended by its editor may lead them to highly questionable statements.

The main point is that nowadays two different versions of the Church-Turing thesis can be stated, concerning what could be done by (1) a *human being* carrying out a process mechanically, or (2) *any physical process*. It is certainly of interest to study Turing's texts in the light of this modern framework. But it should be borne in mind that even the word "Thesis" was not used until 1952, and that the "physical" Church-Turing thesis was not clearly distinguished and examined until about 1980.

Copeland's distinctive contribution has been his insistence that Turing and Church were always crystal clear that their ideas were absolutely restricted to the model of the human being working to rule. Moreover according to Copeland (2002), the reason for this restriction was specifically that there might be more general machines capable of computing functions which the human worker could not compute—that is to say, functions which according to Turing's definition, we call uncomputable.

In the same article, Copeland asserts Church's agreement with Turing on this question. But in the relevant text of (Church 1937), one finds that Church did *not* actually characterise computable functions as those which can be produced by the human worker; instead he defined computability in terms of "a computing machine, occupying a finite space and with working parts of finite size", describing the human worker as a particular case. In a recent article Copeland (2006) has argued that this is because Church's term "computing machine" means, by definition, a machine designed to imitate human work—whereas the term "machine", *tout court*, would imply something *not* thus restricted. But a glance shows that these writers were unaware of this verbal distinction. Thus Church, in his review of Post's work, restated the definition in terms of an "arbitrary machine". Turing, in the opening statement of his 1936 paper, said that "a number is computable if its decimal can be written down by a machine." In the formal statement in (Turing 1939), Turing characterized effective calculability in terms of what "could be carried out by a machine"—without mentioning the human model at all.

Turing later spoke of human rule-followers, mechanical processes, and physical machines without drawing any attention to the distinction Copeland insists upon as essential. Turing did indeed often explain the scope of a computer in terms of replacing the work of a human calculator, but he also said that a universal machine could replace the "engineering" of special-purpose machines. Turing's post-war lecture to mathematicians (Turing 1947) opened by saying he had been led to the universal machine by analysing "digital computing machines". (He continued by comparing the digital computer *favourably* with differential analysers, showing that he did not see its digital character as a real restriction on its scope.) Turing's post-war focus was in what he called "man as a machine", and he was naturally drawn to the picture of the brain as a physical machine.

This is where this question starts to become interesting, because it is bound up closely with arguments for and against the possibility of Artificial Intelligence. Turing's famous paper (Turing 1950a), appealed to the idea that the brain, as a physical machine, could be simulated by a computer. It was implicit in his estimate of the number of bits of storage in the brain, and it was addressed directly in what Turing called the Argument from Continuity of the Nervous System. In the later radio talk (Turing 1951), he explained this idea even more explicitly, stating the idea that a universal machine could do the work "of any machine into which one can feed data and which will later print out results." So in *The Essential Turing*, Copeland steps in (p. 479) to inform the reader that this was *not the Church-Turing thesis* but a *different* thesis. In modern terms, one would indeed make a distinction, as explained above. But in 1951 there was no well-defined "thesis" at all, and this distinction did not exist.

This would be little more than a quibble over words and definitions, if it were not for the fact that Copeland claims to have made a discovery in Turing's texts, overlooked by everyone else, which presages a revolution in science and technology. As already mentioned, Copeland holds that Turing always had clearly in mind that there could be physical machines ("hypermachines") with the ability to compute uncomputable functions. In fact, Copeland specifically identifies the "oracle-machines" of (Turing 1939) as being just such entities.

What are these "oracle-machines"? The following comments should be read in conjunction with the articles by Solomon Feferman and Martin Davis. Turing machines typically solve infinitely many cases of a problem. For instance, there is a Turing machine which given any integer n, correctly decides whether n is prime. But Turing (1936) showed the existence of well-defined problems where no Turing machine can solve all the cases. Nowadays perhaps the best known such problem is Hilbert's Tenth Problem, that of deciding whether a Diophantine equation has a solution. A Diophantine oracle would have the property that given any Diophantine equation (e.g., the Fermat-Wiles equation) it would supply the truth about its solubility. Mathematicians would naturally see oracles as a purely mathematical definition, useful for defining *relative* computability: if you could solve Hilbert's Tenth Problem, what else could you do? This became a standard idea in the text of (Davis 1958). In fact Turing (1939) did indeed use the oracle in this way, but he also had an extra-mathematical interpretation for it: he saw the oracle as related to what he called "intuition", the nonmechanical step involved in seeing the truth of a formally unprovable Gödel sentence. However he made no suggestion of engineering any such object, and emphasised that an oracle, by its nature, could not be a machine.

In contrast, Copeland claimed in a *Scientific American* article (Copeland and Proudfoot 1999) that "Turing did imagine" an oracle which would physically "work", e.g., by measuring "a quantity of electricity" to infinite precision, and that now "the search is under way" for such oracles. These, if found, would bring about a new revolution in computing. In another paper (Copeland 1998) he asserted that the oracles were only theoretical for Turing in the same sense as the atomic Turing machine component operations were theoretical. This is also a far-fetched claim: the primitive operations of a Turing machine could be implemented by simple switches such as 1936 automatic telephone exchanges already used. Oracles need to store an infinite database.

Indeed Copeland is determined to detect references to physical oracles in Turing's later work. He has two main arguments, both fallacious. In the first (Copeland 2000, 2006), he identifies the "infinite store" appearing in the semi-popular account of computability in (Turing 1950) as a reference to the infinite database of an oracle. It is not: Turing's analysis makes it obvious that this is simply a description of the unlimited tape available to a Turing machine. This passage explains computability as a theoretical bound on what actual finite computers can do; in fact it emphasises the *finiteness* of the means Turing believed necessary for the simulation of human intelligence—not much more than a trillion bits of storage. It is surprising that Copeland should insist on this quite elementary misreading, given that he has—for well over a decade—devoted so much scholarship to Turing's work.

The second argument, more subtle and complex, involves computability, randomness, and learning. It is most clearly stated in (Copeland 2006), which holds that the pre-war oracles reappear as a necessary feature in Turing's post-war theory of machine-based learning. Turing's 1948 work had a picture of neural nets which could be trained into functionality by "reward" and "punishment" operations—the same fundamental scenario as in modern bottom-up Artificial Intelligence techniques. Copeland holds that this model of learning is a *development* of the idea of intuition. To this it may be objected that Turing's post-war behaviorist model is pretty well the *antithesis* of his prewar picture of "intuitive" knowledge. But leaving this general question aside, Copeland's proposal has the more concrete problem that a program modified in accordance with some finite learning or training process, is only modified into a different program. It does not go beyond the scope of computable functions. Nor is there any reason to suppose, from Turing's writing, that the process of finding better and better algorithms requires access to an uncomputable source. Yet Copeland (2006) concludes by describing a specific procedure in which an oracle, as defined in (Turing 1939), would supply the training sequence. This is quite foreign to Turing's exposition: not only is such an oracle essentially infinite, but it holds exact data and is nothing like the random trial-and-error process Turing suggested, using human infancy as a model.

One might add that this terminology of "hypercomputing", with its spurious connotation of technological feasibility, has been too readily allowed to pass into the currency of computer science. The generally valuable Turing-inspired volume edited by Teuscher (2004), contains not only Copeland's views on this subject but another article (by M. Stannett) advocating absurd propositions such as that the Fourier decomposition of a function implies the possibility of infinite data storage in a finite piece of wire.

Turing did not use the word "machine" with perfect clarity, and it is impossible to read past minds. What we can see, however, is the *mathematical and scientific use* to which he put his words. And Copeland's insistence on detecting implemented oracles between the lines of Turing's post-war texts renders Turing's advocacy of Artificial Intelligence incoherent. Why should Turing have devoted so much time and trouble to promulgating his "heretical" theory that intelligence could be simulated by a computer program if, all the time, he envisaged the engineering of physical processes beyond the scope of digital computers, or considered it vital to have access to an uncomputable oracle? And, if these issues were as crucial as Copeland believes, why did Turing not make them plain rather than leave them in an obscure form requiring decryption by philosophers?

Nevertheless, *The Essential Turing* has the great virtue of making Turing's own texts accessible, so that readers can assess such arguments for themselves. A particular case is that 1951 radio talk, which was unfortunately omitted from the *Collected Works*. Indeed Copeland has usefully drawn attention to an important feature of that talk. When Turing then discussed the computer simulation of the brain, and the idea that a universal machine could simulate any machine, he touched on the possibility that this might actually be impossible, even in principle, because of the uncertainty in quantum mechanics. This clearly departed from what he had said in (Turing 1950a), which made no mention of quantum mechanics when discussing the mechanical simulation of the brain. Turing attributed this view to Eddington, rather than assert it as his own view, but we can see that he did take it seriously as an objection to his Artificial Intelligence thesis.

So Turing did indeed contribute to the long process of distinguishing and discussing a "physical" Church-Turing thesis. With further experience and thought, Turing naturally developed a clearer idea of what he considered involved in "mind" and "machine". He did not, as Copeland implicitly assumes, adopt a philosophical position and hold it unchangingly from start to end. Evolution is in the nature of mathematics and science, and it continues vigorously now. This 1951 development, stating a new objection from quantum mechanics, is especially interesting because it is just the objection to AI which in the 1980s Roger Penrose developed into a full-scale critique by combining it with what Turing called the Mathematical Argument. Copeland does not make this connection, and seems not to notice the significance of Eddington who had played an energising part in Turing's early thought. Elsewhere (p. 477) he quotes from Turing's juvenile but striking essay from about 1932, based on Eddington's ideas about the mind, yet does not point out its reference to quantum mechanical indeterminacy.

Logic and Physics

The trouble here seems to be Copeland's lack of interest in physics, as profound as his lack of concern with statistics and number theory. The *Essential Turing* does not mention Turing's notes and letters describing his last year of interest in fundamental physics. It does not include Robin Gandy's letter to Newman describing Turing's ideas at the time of his death and the only hint we have about what Turing might have done if he had lived (Gandy 1954). This refers to Turing's intent to find a "new quantum mechanics", definitely suggesting he was trying to *defeat* the Eddington (and later Penrose) objection along with the others. He noted with interest the surprising feature of standard quantum mechanics that in the limit of continuous observation a system cannot evolve. This, nowadays known as the "Quantum Zeno effect", was not deep or new but gave a vivid pointer to an area where Turing might have been led had he lived.

John Britton, editing the *Pure Mathematics* volume of the *Collected Works*, contributed a story from personal recollection. Turing gave a talk at Manchester about the number N, which he defined in terms of the probability that a piece of chalk would jump from his hand and write a line of Shakespeare. Probabilities and physical prediction were natural starting points for his mathematical thought. From an early age, Turing was aware of the importance of physical materialism, the magical power of mathematics to encode the laws of physical matter, and the puzzle of the apparent conflict of physical determinism with human will and consciousness. Turing's mathematical life started with Einstein and Eddington, and it ended in the same physical world. Eddington asked how could "this collection of ordinary atoms be a thinking machine?" and Turing found a new answer. The "imitation game" is at heart the drama of materialist scientific explanation for the phenomenon of Mind, with the mathematical discovery of computability as its new leading actor.

The Essential Turing does not present this background, and it shifts the emphasis from mathematics to philosophy, but it makes a good part of Turing accessible to readers of all kinds: his vivid and direct writing will now reach a new audience and encourage new thoughts. We may regret that the self-effacing Turing did not write more on the genesis and development of his theory of minds and machines. We may likewise regret that he did not write more about his extraordinary life and experiences. Commentators will, necessarily, have to interpret those silences, and these interpretations will arouse controversy. But the controversies are always modern, challenging, and as wide-ranging as Alan Turing himself.

References

[1] A. M. Turing (1992, 2001), *The Collected Works of A. M. Turing: Mechanical Intelligence*, (D. C. Ince, ed.), North-Holland, 1992; *Pure Mathematics*, (J. L. Britton, ed.), North-Holland, 1992; *Morphogenesis*, (P. T. Saunders, ed.), North-Holland, 1992; *Mathematical Logic*, (R. O. Gandy and C. E. M. Yates, eds.), North-Holland, 2001. These volumes are referred to below as CW1, 2, 3, 4 respectively.

[2] V. Bowden (ed.) (1953), *Faster than Thought*, Pitman, London, 1953.

[3] A. Church (1937), Review of Turing (1936), *J. Symbolic Logic* **2**, 42–3.

[4] B. J. Copeland and D. Proudfoot (1996), On Alan Turing's anticipation of connectionism, *Synthese* **108**, 361–377.

[5] B. J. Copeland (1998), Turing's o-machines, Penrose, Searle and the brain, *Analysis* **58**, 128–138.

[6] B. J. Copeland and D. Proudfoot (1999), Alan Turing's forgotten ideas in computer science, *Scientific American* **253**:4, 98–103.

[7] B. J. Copeland (2000), Narrow versus wide mechanism: including a re-examination of Turing's views on the mind-machine issue, *J. of Philosophy* **96**, 5–32.

[8] ____ (2002), The Church-Turing thesis, in E. N. Zalta (ed.), *Stanford Encyclopedia of Philosophy*, `http://plato.stanford.edu`.

[9] ____ (2006), Turing's thesis, in *Church's thesis after 70 years*, (A. Olszewski et al., eds.), Ontos Verlag, 2006.

[10] M. Davis (1958), *Computability and Unsolvability,* McGraw-Hill, 1958; new edition Dover, 1982.

[11] ____ (1965), *The Undecidable*, Raven Press, 1965; Dover 2004.

[12] ____ (2000), *The Universal Computer: The Road from Leibniz to Turing*, Norton; new edition as *Engines of Logic*, Norton, 2001.

[13] R. O. Gandy (1954), letter to M. H. A. Newman, available at `http://www.turingarchive.org`, item D/4. Text in CW4.

[14] I. J. Good (1950), *Probability and the Weighing of Evidence*, Griffin, London, 1950.

[15] ____ (1992), Introductory remarks for the article in *Biometrika* **66** (1979), "A. M. Turing's statistical work in World War II", in CW2.

[16] ____ (1993), Enigma and fish, in *Codebreakers*, (F. H. Hinsley and A. Stripp, eds.), Oxford University Press, 1993.

[17] ____ (2001), Commentary on Turing's manuscript "Minimum cost sequential analysis", in CW4.

[18] N. J. Higham (1996), *Accuracy and Stability of Numerical Algorithms*, SIAM, 1996.

[19] M. H. A. Newman (1955), Alan Mathison Turing, *Biographical Memoirs of Fellows of the Royal Society* **1**, 253–263, and in CW4.

[20] A. Orlitsky, N. P. Santhanam, and J. Zhang (2003), Always good Turing: Asymptotically optimal probability estimation, *Science* **302**, 427–431.

[21] B. Randell (1972), On Alan Turing and the origins of digital computers, *Machine Intelligence* **7**.

[22] J. Swinton (2004), Watching the daisies grow: Turing and Fibonacci phyllotaxis, in (Teuscher 2004). See also `http://www.swintons.net/jonathan/turing.htm`.

[23] C. Teuscher (2002), *Turing's Connectionism, an Investigation of Neural Network Architectures*, London: Springer, 2002. See also his article "Turing's Connectionism" in (Teuscher 2004).

[24] ____ (ed.) (2004), *Alan Turing: Life and Legacy of a Great Thinker*, Berlin: Springer, 2004.

[25] A. M. Turing (1936), On computable numbers, with an application to the Entscheidungsproblem, *Proc. Lond. Math. Soc.* **42**(2), 230–265. In CW4 and ET.

[26] ____ (1938), Finite approximations to Lie groups, *Ann. of Math.* **39**(1), 105–111. In CW2.

[27] ____ 1939), Systems of logic based on ordinals, *Proc. Lond. Math. Soc.* **45**(2), 161–228. In CW4 and ET.

[28] ____ (1944), Speech system "Delilah"—report on progress, typescript dated 6 June 1944. National Archives (London), box HW 62/6.

[29] ____ (1946), Proposed electronic calculator, copy of typescript at `http://www.turingarchive.org`, item C/32. Printed version in CW1.

[30] ____ (1947), Lecture to the London Mathematical Society, 20 February 1947, typescript at `http://www.turingarchive.org`, item B/1. Text in CW1 and in ET.

[31] ____ (1948a), Intelligent machinery, National Physical Laboratory report, typescript available at `http://www.turingarchive.org`, item C/11. Text in CW1 and (better) in ET.

[32] ____ (1948b), Rounding-off errors in matrix processes, *Quart. J. Mech. Appl. Math.* **1**, 287–308. In CW1.

[33] ____ (1949), Checking a large routine, paper given at EDSAC inaugural conference, in CW1.

[34] ____ (1950a), Computing machinery and intelligence, *Mind* **59**, 433–460. In CW1 and ET.

[35] ____ (1950b), The word problem in semi-groups with cancellation, *Ann. of Math.* **52**(2) 491–505. In CW2.

[36] ____ (1951), Can digital computers think? BBC radio talk, typescript available at `http://www.turingarchive.org`, item B/5. Text in ET.

[37] ____ (1952), The chemical basis of morphogenesis, *Phil. Trans. R. Soc. Lond.,* **B 237**, 37–72. In CW3 and ET.

[38] S. Zabell, (1995), Alan Turing and the central limit theorem, *American Mathematical Monthly* **102**, 483–494.