Only Human: a book review of *The Turing Guide*

Bjørn Kjos-Hanssen

August 16, 2018

Introduction

A victorious yet tragic hero. A genius but ominously an augur of the end of human dominance. Such characterizations come to mind regarding Alan Mathison Turing (1912–1954) after reading the *The Turing Guide*.

It is a 500-page compilation of articles by many authors, written for "general readers", which strikes a balance between focusing on Turing himself, and on the collection of topics he was involved in. The driving force behind the book is philosopher Jack Copeland, who has written many books and articles about Turing and participates in sixteen of the fourty-two Chapters of the Guide. Officially the author list is Copeland, Bowen, Sprevak, Wilson, "and others"[1]. The Chapters are lightly cross-referenced, but are largely independent. The book is solidly proof-read: I got to page 55 before finding the first error ("during the did decades").

Turing's appeal in the popular imagination may stem from checking several boxes: he is viewed as a genius, a hero, and even a tragic hero. In support of the genius label, he defined a mathematical notion of computer that turned out to be the right one, proved some fundamental results (existence of the universal computer, unsolvability of its halting problem), and arguably founded mathematical biology (see Part VI below). As for heroism, he worked on cryptography during World War II, leading a large team. However, the claim in the Preface that

> It is no overstatement to say that, without Turing, the war [...] might even have been won by the Nazis.

is in Chapter 9 modified to indicate that perhaps his work helped ensure that the war ended in 1945 rather than 1946. As for tragedy, he was convicted of homosexuality and ordered into female hormone therapy. Moreover, he died from cyanide poisoning, in a mysterious event that has been called a suicide, but which may well have been an accident stemming from his home laboratory.

Turing earned the standing to present to us all his thoughts on human and machine intelligence, and as discussed below, those thoughts now seem prophetic.

The *Guide* is divided into eight Parts, each worth a section of this review.

Part I: Biography

In this Part we learn many interesting facts. Turing died in a manner that involved cyanide and a lab room next to his bedroom, but that the jury is still out on whether it was suicide or some kind of experiment gone awry. He thought that intellectual activity mainly consists of various kinds of search, and that we should expect the machines to take control. This is a possible counterpoint to the label of hero: perhaps he hastened the day of the "Singularity" when machines take over and render humans irrelevant.

It is argued that he took his court-ordered female hormone therapy with an impressively resilient attitude, treating it almost as a case of freshman hazing. If true, that tends to make him less a tragic hero and more a simply mysterious hero.

His work on morphogenesis [17] is described as "even deeper" than the discovery of DNA molecules.

[1] Amusingly, as of March 8, 2018, MathSciNet gives "and others" the author ID 1227985 and reports that this book is his or her only publication so far.

Part II: The universal machine and beyond

Copeland writes about the move from electromagnetically-controlled relays to blindingly fast digital electronics, which were first used for breaking German codes but would turn out to combine stunningly with the universal Turing machine. Turing's ACE (Automatic Computing Engine) machine ran at 1 MHz which outperformed the competition at the time; von Neumann's design for a computer was less focused on speed. Turing's was essentially a RISC (Reduced Instruction Set Computer). Copeland gives us the impression that Turing was an engineer(ing professor) as much as a mathematician (mathematics professor). It seems that Turing looked at Turing machines as idealized machines perhaps more than as purely combinatorial mathematical concepts.

Part III: Codebreaker

Copeland argues convincingly, to me, that Turing's contributions did not change the victor of World War Two from the Axis powers to the Allies. On the other hand, it may have saved on the order of 10 million lives by helping to shorten the war.

We get a very detailed description of how the Germans' secure communications machine "Enigma" worked. For a mathematician a more mathematical treatment would have been preferable; the given description of how some wheels are attached to others in certain ways and triples of letters associated with others was a bit bewildering.

Breaking the Germans' codes was not a matter of solving a well-defined math problem, but rather of thinking of lots of aspects of what the Germans were doing and finding a series of weak links, something to hack. Again we perhaps see Turing's engineering essence above his mathematical one.

We learn about Turing's cryptologic work. To decode German messages one had to basically search through a huge space for some input whose output would be a recognizable German language message. Various heuristics and methods to reduce this search space were considered. Turing made extensive use of probability and used phrases like "cross" and "direct" where other less mathematically serious colleagues used "starfish" and "beetle".

The *Bombes* were electromagnetic devices created to carry out the search that remained to be done after all heuristics and mathematical simplifications had been applied. Turing played a leading role in adapting these from Polish cryptanalysts' *Bombas* (see page 6).

Enigma is an elaboration of Vigenère ciphers which are elaborations of the simple Cæsar ciphers. Turing wrote a manuscript on the deciphering of such ciphers using Bayes' Theorem, which has recently been released to the ArXiv [13].

Banburismus was a mechanical (not even electromechanical) means of reducing the search space before starting a Bombe run. It involved punch cards inspired by the loom industry (as also Lovelace and Babbage had been). It is explained that people of intermediate skill were not needed for the endeavor: there were the manual card-punchers and measurers, there were the cryptanalysts, who had a much more enjoyable job, and then there were Turing and his ilk who designed the algorithms the cryptanalysts carried out. Sometimes the attempts at explaining mathematical ideas in plain language arguably become too vague (p. 139):

> "A two-letter sequence such as 'en' occurs more frequently in English than the combination of 'e' and 'n' counted separately".

A more advanced machine, Tunny, took over from Enigma, and we learn about the methods and computers (Colossus) used to decode Tunny messages. Encryption by vector addition mod 2 is nicely explained. Doing it twice recovers the original message since

$$(A + B) + B = A,$$

and it satisfies the associative law. Special tricks included waiting for Germans to send the same message again, but with some minor variation, because the Germans thought the first message did not go through. Two similar messages could be more easily broken and this is explained in some detail. The

Colossus computer used electronic valves. These had at the time a status similar to that of quantum bits now: believed to be too flaky to be used en masse, i.e., have many of them in one computer. It is claimed that had many Colossi not been destroyed after the war, things like the Internet and social networking might have happened a decade earlier than they did. (The idea of Facebook starting already in 1994 may not be universally viewed as a positive, however.)

A Chapter by Eleanor Ireland details the secrecy and tedium of working on the Colossus machines. Global WWII events and their relation to Bletchley Park are detailed. We hear what a large industrial-scale cyberwarfare operation it was. Turing however was "flicking the walls with his fingers as he walked", an image that may feel familiar to mathematicians and children alike. We learn that when two messages are "in depth", meaning encrypted by adding the same vector, we can add the encrypted versions

$$(A+B)+(C+B)=A+C$$

and cancel out the encrypting vector B. Next, we think of a piece of German phrase we think might be used and add that to some consecutive entries of the vector $A+C$. If our German phrase was in A or C we would be left with a fragment of C or A, respectively. Intimate knowledge of the German language, as it was used by the Tunny operators, was key.

Brian Randell writes about the revelation of some classified information about Ultra, the codename for the British efforts against German cryptography, in the 1970s. It reminds me that Turing via his Bletchley Park work becomes an almost unbelievable incarnation of the "nerd's superhero": someone who through mathematical work becomes a leader among thousands of regular people engaged in the largest war of all time.

Turing visited the U.S. to help with their own Bombe-making, and worked on a speech encryption device. Much work has been done on the preservation of Bletchley Park's historical WWII buildings via increasing the public and funders' interest with books, TV reports, special events and publications.

Part IV: Computers after the war

Baby, the first stored-program computer, was built in Manchester, England but with inspiration from Princeton. Interestingly von Neumann (at Princeton) pushed the idea of a CPU with an accumulator (familiar to those who have studied machine/assembly language) whereas Turing liked a more decentralized design.

Turing developed the ACE computer rivaling Baby. It was fast, but ultimately obsolete compared to rival designs. At the time random-access memory had not been developed. Rather than scanning through memory until the desired memory location arrived, Turing's design used something called optimum programming to lay out instructions in memory so that the desired info in memory tended to arrive quickly, or rather, at the exactly right time. Such programming suited Turing rather well, as the architecture was similar to that of his own Turing machines.

Turing had a great deal of foresight with regard to the design of machine language. Brian E. Carpenter and Robert W. Doran give a beautifully simple description of recursion: a computer must keep track of where it is, so a stack is needed.

Copeland and composer Jason Long describe how Turing and colleagues made computer music. For someone growing up with Commodore machines in the 1980s the similarity is striking and appealing.

We are also taken on a trip back to the time of Charles Babbage. Babbage was focused on arithmetic and algebra. He acknowledged that Ada Lovelace saw further and envisioned a machine that could make music and replicate the brain. The situation is summarized by saying that Babbage was focused on hardware (and algebra), Lovelace on applications, and Turing on theory (as he developed mathematical theory of what was needed to achieve Lovelace's vision).

Part V: Artificial intelligence and the mind

Perhaps the most famous idea named after Turing is the Turing test. Turing proposed that to test whether a machine had achieved intelligence, it should be asked to try to fool a human into thinking it was human. More precisely, a human judge gets to interrogate both another human and the machine (via a neutral interface such as computer chat window), and is asked to guess which is the human.

Turing hypothesized that around our time machines would be able to fool some people some of the time, and that in another 50 years or so machines would be able to fully pass the Turing test. So far, so good, for these predictions: for instance Google's artificial intelligence is able to play the role of someone booking an appointment with a hair dresser, in such a way as to not be detected as a machine.

In this Part we learn that Turing wanted to define intelligence subjectively, as behavior that we find mysterious and admirable but do not fully understand [16]. This way the judge in the Turing test becomes an important participant. Diane Proudfoot gives a delightful discussion of some of my favorite topics like consciousness zombies and solipsism. Turing imagined child machines that learned, a precursor to today's machine learning. The Chapter by Proudfoot includes an unnerving observation: robots must look like humans in order to build rapport (make a connection) with humans, in order to learn from humans.

The Chapter on computer chess discusses heuristic search. Rather than searching through possible moves, one uses guiding rules such as "a rook is worth 5 points". With machine learning one could even discover that it is better to value a rook at, say, 4.9 points. As in some other Chapters, however, there is a bit of historic trivia of little interest such as, who first lost a game of chess to a computer, who first won etc. There is also some material that perhaps is of interest to lay-persons, such as a complete transcript of the first game of chess between a human and Turing acting as a computer. And some fascinating tidbits such as Mozart's *Musikalische Würfelspiel* (randomly generated Mozart music).

The book does have a smattering of strange matters to a mathematician.

- The standard normal distribution is described as having mean 0, standard deviation 1, and also height 1 at the mean.
- The proof of the undecidability of the halting problem (p. 410–411) seems to make no use of the crucial negation step whereby a computation halts if and only if it does *not*.
- The distinction between countable sets and computably enumerable sets is missing in the same Chapter 37. (Very nice though is that Chapter's display of an explicit polynomial over $\mathbb{Z}$ that produces the primes, and no other positive integers.)

A Chapter on WWII coding methods reads a bit tedious at times (imagine going through a detailed computation with repeated Bayes theorem usage, in prose rather than equations) but there are some interesting things for me such as the use of the *score* $\log p$ of a probability p to simplify hand calculations so that the clerks at Bletchley Park could use addition rather than multiplication.

Extra-sensory perception (ESP) was credible to many scientists at Turing's time. He apparently worried that ESP used by the judge in the Turing test would lead the judge to falsely fail to attribute intelligence to the machine. Thus in a sense Turing seems to truly sympathize, in theory, with intelligent machines.

In Sprevak's Chapter about cognitive science, a discussion of finite automata as inspired by McCulloch and Pitts [6] would have been appropriate. There is a sense in which the language

$$(01)^* = \{\emptyset, 01, 0101, 010101, \dots\}$$

is understandable by humans and

$$\{0^n1^n : n \geq 0\} = \{\emptyset, 01, 0011, 000111, 00001111, \dots\}$$

is not. For the former, we just have to scan the whole input, rejecting if we see 00 or 11. Only our lifespan or fatigue limits us in this regard. For the latter, we have to keep a counter, and for large n that is beyond our memory capabilities whether in our brain or in hardware or paper.

Part VI: Biological growth

This interesting Part introduces morphogenesis via the tale of the sweating grasshoppers and the fire. The basic idea is pretty clear even in the absence of any differential equations. While it is not mentioned in the *Guide*, Turing's work is related (see [2, 3]) to Schelling's [11] work on segregation. If individuals tend to prefer to live close to similar individuals, how do segregated neighborhoods form? In terms of a tolerance parameter, higher tolerance may lead individuals to be less likely to move, which can actually lead to more segregation: once individuals land in a rather homogeneous area they are likely to stay. Here, the neighborhoods (in economics) are analogous to the stripes on a zebra (in biology).

The Chapter about radiolaria is amazing: suffice it to say that it concerns single-cell organisms shaped like Platonic solids with spikes!

Part VII: Mathematics

Here we learn that Turing worked on the Central Limit Theorem and on the Riemann ζ-function [15, 18]. Conveniently for this book, Turing worked on a lot of very central topics.

Turing's work [14] on the *Entscheidungsproblem* (the decision problem for validity in first-order logic) is discussed in several Chapters in the book. One Chapter makes it seem like Turing did the most and Gödel a relatively minor amount, but Rod Downey's Chapter gives the view that the *Entscheidungsproblem* had arguably already been solved before Turing. In any case, Gödel showed that any computable axiom system gives an incomplete set of theorems, thus absent an algorithm for which new axioms to add, it is clear that there can be no algorithm to decide which results are true and which are false in arithmetic. Downey discussed randomness and Turing's work on absolutely normal numbers and how they correspond to finite-state random sequences. He adds that it is not clear whether one can physically generate true randomness. One might add that it is not clear what that would even mean. Cornout argued that we need a principle, namely

> events of very low probability simply *do not happen*,

in order to give a non-circular explanation of what probability is [12].

Part VIII: Finale

To me this was the most interesting Part of the book. It deals with various arguments for how the time evolution of our physical Universe may not be computable. Of course, if the universe is finite and discrete then in some sense it is computable. However, even in that scenario, as the universe is very large, it is conceivable that the universe is not well modeled as being computable, but is better thought of as containing some random or noncomputable aspects.

Early thinking on this topic may have been motivated by the idea that, surely, human minds can do things that Turing machines cannot. In "Renewing Philosophy" [9], Hilary Putnam argues that artificial intelligence is impossible on the basis of the work of Pour-El and Richards [8] on non-computability in classical physics. Next, there was the thought that physics, with its marvelous use of higher mathematics, may contain undecidability [4]. That argument has lost some of its shimmer [5]. Finally, at present it seems a technological solution for achieving non-computability is all we are left to imagine. Now it seems that while perhaps *technology based on physical systems* can carry out non-Turing machine computations, that seems unlikely to mean *human minds* can do the same (see Figure 1).

Andréka, Németi and Székely [1] work on using time travel (closed timelike curves) to compensate for the lack of space (and time). In the so-called Malament–Hogarth spacetimes one can compute forever and thus solve the halting problem. Hogarth worked on this in the 90s and Welch [21] showed that even a larger class of problems than those solvable

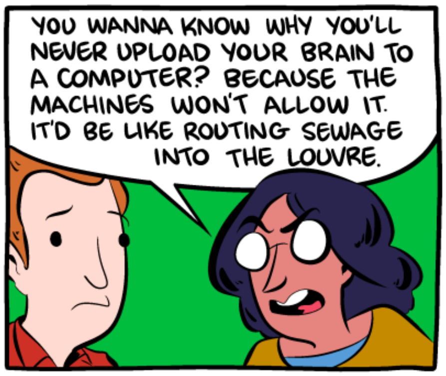

Figure 1: The impact of Turing's work on popular culture [20].

using the halting problem (all hyperarithmetic problems) can be solved in MH-spacetimes.

Interestingly for our times, hypercomputation using CTCs is a technological solution. Thus researchers are no longer claiming nature itself, and certainly not humans, are super-Turing-machine. For another example, consider Christina Perri's song "Human" [7] with the eerie lyrics:

> But I'm only human
> And I bleed when I fall down
> I'm only human
> And I crash and I break down

Could these words have been sung a century ago?

Polish contribution

A good test of a biographical and historical book is how it holds up in light of new information. Sir Dermot Turing, Turing's nephew and author of a Chapter of *The Turing Guide*, in 2018 published a book *X, Y & Z* [19] in which he argues that Polish mathematicians like Marian Rejewski and Henryk Zygalski should get more credit and they have not gotten it because of an exaggerated "Turing cult" [10]. To *The Turing Guide*'s credit, it is indeed mentioned in the book that the Polish mathematicians had addressed the Enigma problem more as a pure math problem than the British, by the time the two groups compared notes. In particular, the Polish had the idea of using machinery to decrypt machine-produced codes, using what they called *bomba* (as opposed to Turing's *Bombe*). In my draft of this review, written before the article [10] appeared, I had already noted that *The Polish were ahead of the British for a while, as the former realized right away that the code breaking was fundamentally a mathematical problem.*

Conclusion

Overall I found this to be delightful book. It was even inspiring, for instance with the mentions of seminar topics which turned into new research directions. Mathematicians should find a mixture of things they already knew, things they are glad to learn, and a couple of things they disagree with. I imagine a general well-educated audience, especially scientists and engineers who do not specialize in mathematics, may enjoy the book the most.

References

[1] Hajnal Andréka, István Németi, and Gergely Székely. Closed timelike curves in relativistic computation. *Parallel Process. Lett.*, 22(3):1240010, 15, 2012.

[2] George Barmpalias, Richard Elwes, and Andrew Lewis-Pye. Digital morphogenesis via Schelling segregation. *Nonlinearity*, 31(4):1593–1638, 2018.

[3] George Barmpalias, Richard Elwes, and Andy Lewis-Pye. Digital morphogenesis via Schelling segregation. In *55th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2014*, pages 156–165. IEEE Computer Soc., Los Alamitos, CA, 2014.

[4] Robert Geroch and James B. Hartle. Computability and physical theories. *Found. Phys.*, 16(6):533–550, 1986.

[5] S. Hossenfelder. *Lost in Math: How Beauty Leads Physics Astray.* Basic Books, 2018.

[6] Warren McCulloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.

[7] Christina Perri and Martin Johnson. Human, November 2014. From the album "Head or Heart".

[8] Marian B. Pour-El and I. Richards. *Computability in Analysis and Physics.* Perspectives in Mathematical Logic. 1989.

[9] H. Putnam. *Renewing Philosophy.* Harvard University Press, 1992. Chapter 1: The project of Artificial Intelligence.

[10] David Sanderson. 'Turing cult' has obscured role of Polish codebreakers. *The Times of London*, 8 2018.

[11] Thomas C. Schelling. Models of segregation. *The American Economic Review*, 59(2):488–493, 1969.

[12] Glenn Shafer. Why did Cornout's Principle disappear?, 2006. Slides.

[13] I. Taylor. Alan M. Turing: The Applications of Probability to Cryptography. *ArXiv e-prints*, May 2015.

[14] A. M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. London Math. Soc. (2)*, 42(3):230–265, 1936.

[15] A. M. Turing. A method for the calculation of the zeta-function. *Proc. London Math. Soc. (2)*, 48:180–197, 1943.

[16] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[17] A. M. Turing. The chemical basis of morphogenesis. *Philos. Trans. Roy. Soc. London Ser. B*, 237(641):37–72, 1952.

[18] A. M. Turing. Some calculations of the Riemann zeta-function. *Proc. London Math. Soc. (3)*, 3:99–117, 1953.

[19] D. Turing. *X, Y & Z: The Real Story of How Enigma Was Broken.* History Press, 2018.

[20] Zack Weinersmith. The human mind. *Saturday Morning Breakfast Cereal*, 4131, June 2016. Reprinted with permission from the author.

[21] P. D. Welch. The Extent of Computation in Malament–Hogarth Spacetimes. *The British Journal for the Philosophy of Science*, 59(4):659–674, 2008.

Bjørn Kjos-Hanssen is a professor of mathematics at University of Hawaii at Manoa. His email address is `bjoern.kjos-hanssen@hawaii.edu`