Incomputability after Alan Turing

S. Barry Cooper

The year 2012 marks the 100th anniversary of the birth of Alan Turing. The following two articles were inspired by the work of Alan Turing. For information about the centenary celebration of Turing's life and work, visit the website `http://www.mathcomp.leeds.ac.uk/turing2012/`.

Living in a Computable World

Those of us old enough may remember being fascinated by George Gamow's popular books on mathematics and science—with the most famous being *One Two Three…Infinity*. Gamow got us to imagine living on the surface of a two-dimensional balloon with only two-dimensional experience of the surface. And then he got us to understand how we might detect its three-dimensional curved character via purely two-dimensional observations. In Figure 1 is his picture from page 103 of the 1961 edition.

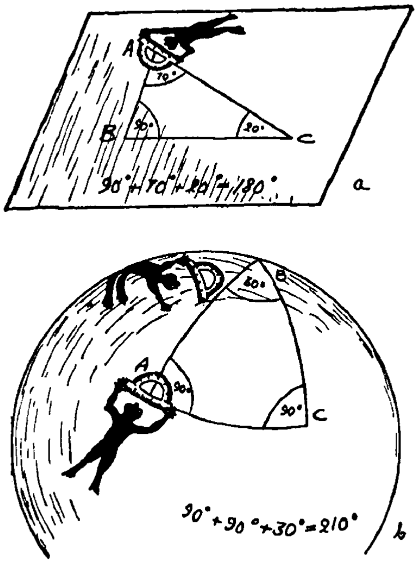

Figure 1. Two-dimensional scientists of the flat and curved "surface worlds" check the Euclidean theorem about the sum of the angles in a triangle. ***From [14].***

S. Barry Cooper is professor of mathematics at the University of Leeds and chair of the Turing Centenary Advisory Committee. His email address is `pmt6sbc@maths.leeds.ac.uk`.

Preparation of this article completed while the author was a visiting fellow at the Isaac Newton Institute for the Mathematical Sciences, in the program "Semantics and Syntax: A Legacy of Alan Turing".

DOI: http://dx.doi.org/10.1090/noti866

Algorithms, as a way of traversing our four dimensions, have been with us for literally thousands of years. They provide recipes for the control and understanding of every aspect of everyday life. Nowadays, they appear as computer programs. Algorithms, or computer programs, can be thought of as a kind of causal dimension all their own. Then the questions arise: Is there a causal dimension that is not algorithmic? Does it *matter* if there is?

Notice that Gamow's example showed that on one hand it was tricky to live in two dimensions and find evidence of a third. But it *did* matter precisely because we *could* find that evidence. Of course, if we took the mathematical model presented by the picture, the missing dimension becomes clear to us—we have an overview. But observe that while the mathematical overview gives us a better understanding of the nature of curved space, it does not tell us that the model is relevant to our world. We *still* need to look at the triangle from within the two-dimensional world to match up reality and mathematics and to be able to apply the full power of the model.

Back in the 1930s, people such as Kurt Gödel, Stephen Kleene, Alonzo Church, and Alan Turing did build mathematical models of the computable dimension of causal relations. This enabled Church and Turing to get outside this dimension and to use their models to explore the new dimension of incomputability.

A specially important part of what the twenty-four-year-old Alan Turing did was to base his investigation of the extent of the computable on a new *machine-like* model. The Turing machine was to make him famous in a way no one could have foreseen. He had the idea of using Gödel's coding trick to turn Turing machine programs into data the machine could compute from, and hence was born the *universal* Turing machine, able to take a code for *any* other given machine as input data and to compute exactly like it. The universal machine *stored* programs and so gave us an understanding of the modern stored-program computer before anyone had even built a real one.

This caused all sorts of problems. As with Gamow's example, it was easy to get the mathematical overview. The problem was to match it up with reality. And this was a problem with a practical aspect. Even a toy avatar of the abstract machine was hard to make. The engineers eventually came up with clever solutions to the problem: the EDVAC in Pennsylvania, the Manchester "Baby", Maurice Wilkes's EDSAC, and the Pilot ACE growing out of Turing's own attempts to build a computer at the National Physical Laboratory (NPL). But to this day, there are engineers who find it hard to excuse (or even understand) Turing's reputation as the "inventor" of the computer. The impact that John von Neumann's 1945 EDVAC report had on the history of computing has been better acknowledged than has the impact of Turing's work. Nevertheless, in his Hixon Symposium lecture in Pasadena in 1948 [37], von Neumann gave Turing his due.

More importantly, the computer changed every aspect of our lives and strengthened our experience of living in a computable world. Incomputability became a mathematical oddity, a playground for researchers who were not too concerned about real-world significance but liked doing hard mathematics with a distinct feel of reality. Of *course* it felt like reality. It was real numbers connected by computable relations—a bit like a well-behaved scientific world of information structured by computable causal relations.

A Short History of Incomputability

Computability has always been with us. The universe is full of it: natural laws whose computability enables us to survive in the world, animal and human behavior guided by biological and learned algorithms, computable natural constants such as π and *e*. The algorithmic content gives the mathematics of nature its infinitary character and opens the door to incomputability. Richard Feynman, no less, may have decided [13] "It is really true, somehow, that the physical world is representable in a discretized way, and…we are going to have to change the laws of physics," but real numbers persist in the mathematics of the real world.

There have ever been doubts about our ability to make sense of causality. Adding deities may reassure but produces its own uncertainties. Questioning the scope of predictable causation certainly goes back to the eleventh century and Al-Ghazali's *The Incoherence of the Philosophers* and is traceable through Hume and Berkeley, arriving, for instance, at the modern interest in emergent phenomena. According to the *Oxford English Dictionary* (1971 edition), the first recorded use of the word "incomputable" goes back to 1606, some forty years even before "computable". The term acquired its precise meaning only in the 1930s, with the formulation of a number of different models of what it means for a function over the natural numbers to be computable. As we have already mentioned, it was these that enabled Church and Turing to get their examples of incomputable objects. The key observation, captured in what we now know as the *Church-Turing Thesis*, is that there is a robust intuitive notion of computability to which all our different formalisms converge. It was Turing's carefully argued 1936 paper, based on the Turing machine model (see Figure 2), that convinced Gödel of the validity of such a thesis.

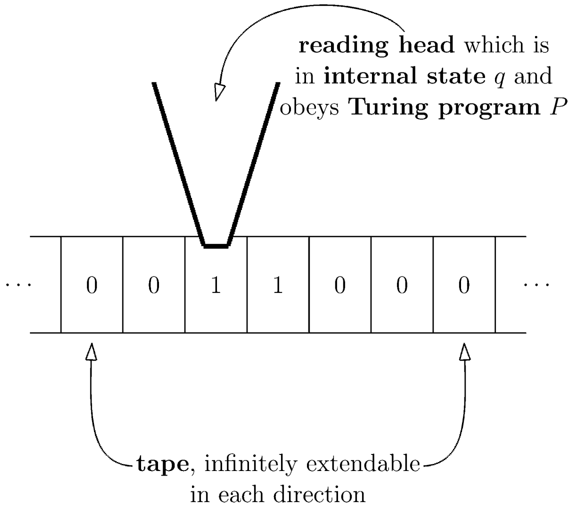

Figure 2. Turing machine from S. B. Cooper, *Computability Theory*, Chapman & Hall/CRC, 2004.

As Gödel's friend Hao Wang recounts [38, p. 96]:

> Over the years G habitually credited A. M. Turing's paper of 1936 as the definitive work in capturing the intuitive concept [of computability], and did not mention Church or E. Post in this connection. He must have felt that Turing was the only one who gave persuasive arguments to show the adequacy of the precise concept.…In particular, he had probably been aware of the arguments offered by Church for his 'thesis' and decided that they were inadequate. It is clear that G and Turing (1912–1954) had great admiration for each other,…

Within mathematics, Turing's paper was a blow to David Hilbert's view, famously expressed on September 8, 1931, in a Königsberg address (quoting from John Dawson's 1997 biography of Gödel) that:

> For the mathematician there is no Ignorabimus, and, in my opinion, not at all for natural science either… The true reason why [no one] has succeeded in finding an unsolvable problem is, in my opinion, that there is no unsolvable problem. In contrast to the foolish Ignorabimus, our credo avers:
>
> We must know, We shall know.

The universality of Turing's machine meant it had to implement a lot of faulty programs. Some programs would lead to computations that never stopped, and one could not tell in general which inputs led to a proper computation. The Halting Problem for the Universal Turing Machine (UTM) turns out to be unsolvable. The set of inputs that lead to a terminating computation of the UTM is *computably enumerable*: you can progressively set in motion an array of all possible computations and observe which ones output a result. This enables you to enumerate the inputs on which the machine halts, but this set is *not computable*, since you can *never be certain* that a computation in progress will not one day succeed.

More dramatically, all sorts of mathematical theories are capable of "talking about" our UTM. Turing used natural numbers to code the activities of the machine, using the trick Gödel had earlier used to enable Peano arithmetic to talk about itself. Turing's discovery was that *any* reasonably strong mathematical theory was *undecidable,* that is, had an incomputable set of theorems. In particular, Turing had a proof of what became known as *Church's Theorem*, telling us that there is no computer program for testing a statement in natural language for logical validity. Since then, a huge number of undecidable theories have been found.

The drawback to this powerful technique for proving the existence of many natural incomputable sets is that there are no other good techniques for proving incomputability. Moreover, John Myhill showed in 1955 that the known natural examples of incomputable objects tended to be all the same—computationally, just notational translations of each other. This would not have mattered if it were not for the fact that the so-called Turing universe of incomputable sets turned out to have a very rich and mathematically challenging structure, and if this were to be embodied in the real world to any extent, most of the computational character of its embodiment would be hidden from us. If we managed to solve a problem, all well and good. If we failed, we might never know whether that bit of the world had an incomputable character different from that of the halting problem or might be computable via some program that has eluded us so far.

This simple difficulty with *recognizing* mathematical incomputability was explained by results that told us such recognition was itself a highly incomputable problem. Here were all the ingredients for a parting of the ways between the mathematics and real-world concerns. Turing himself made his last great contribution to the logic of computability theory in his amazing 1939 paper, based on his work with Alonzo Church in Princeton. Incomputability later played a very important role in mathematics and computer science, encountered in a range of undecidability results. Wilfried Sieg's *Normal forms for puzzles: A variant of Turing's Thesis*, in [7], is an excellent commentary on Turing's continuing interest in the decidability problems for games, groups, etc. Of course, the most celebrated undecidability result was the negative solution to Hilbert's Tenth Problem by Martin Davis, Yuri Matiyasevich, Julia Robinson, and Hilary Putnam. Who would have thought that just one existential quantifier added to everyday high school arithmetic would give rise to a problem unsolvable by any computer program?

After 1939 Turing's work was more obviously rooted in reality than the earlier overarching abstraction. The mathematicians had the halting problem and its variants, mathematically comprehensive and canonical, a little too grandiose for the everyday ad hoc world, while the endless complications of everyday existence could not be classified. The theory was useless.

For the recursion-theoretic period of mathematics, with its isolation and loss of sense of mission in the wider world, see recent papers by Robert I. Soare (for example [29]). Turing's Manchester work on artificial intelligence, connectionist models, and morphogenesis contained inspired anticipations of the shape of things to come.

Mathematical Steps towards an Incomputable Reality

The unsolvable problems of the 1930s may deliver examples of incomputable objects, but their mathematical abstraction appears far from the embodied mathematics of a Newton or an Einstein. At the same time, the deep and intractable problems from the real world are hard to subject to logical analysis. What makes Turing's work so important is the way it draws out the computability-theoretic core of very different real-world mysteries. He had a knack of getting inside structures and imaging their constructive cores in new ways. Typically, Turing does not *apply* mathematics; he *builds* it within the context he is exploring. For Einstein [10, p. 54]:

> When we say that we understand a group of natural phenomena, we mean

> that we have found a constructive theory which embraces them.

Turing takes this literally. Mathematics for Turing often comes mentally embodied. The, for many, repellent abstraction of Turing's 1939 paper is all about how Gödel's incompleteness theorem plays out in practice. We notice that the true sentence unprovable in Gödel's theory for arithmetic is easily described, while enlarging our theory by adding the sentence gives us a larger theory which has a similarly described unprovable sentence. For Turing, this was the seed for a computably iterated process of enlargement. So, using Kleene's computable ordinals, one could extend the process transfinitely. Later, logicians such as Solomon Feferman and Michael Rathjen would enhance Turing's ladder into the incomputable and proof-theoretically reach levels far beyond those Turing had achieved. But in a sense, Turing had found out what he wanted to know. Tucked away amongst the mountains of abstraction is his characteristically candid take on what he had done (Turing 1939, pp.134–5):

> Mathematical reasoning may be regarded…as the exercise of a combination of…intuition and ingenuity.… In pre-Gödel times it was thought by some that all the intuitive judgements of mathematics could be replaced by a finite number of…rules. The necessity for intuition would then be entirely eliminated. In our discussions, however, we have gone to the opposite extreme and eliminated not intuition but ingenuity, and this in spite of the fact that our aim has been in much the same direction.

The outcome is that mathematically we have a brilliant analysis of how we may constructively navigate our way through a phase transition but storing up a level of incomputability arising from the ad hoc nature of the route. There is a degree of arbitrariness in a climber's choice of hand and foot holds, which gives the climb more than algorithmic interest. This ties in with an irreversibility of computation noted by people such as Prigogine in quite different contexts.

There is a special interest for the mathematician in Turing's analysis of an incomputable route to computable outcomes. It is an analysis that fits nicely with our experience of creative derivation of theorems, followed by the uncovering of algorithmic approaches to them. We rely on the memetic character of our proofs, fit to circulate the community like a virus. A basic rule for lifting small truths to bigger ones, a logical counterpart of the more visceral causality of nature, is mathematical induction. For proof theorists, induction plays a key role. They categorize theorems according to the level of complexity of the induction used in the proof. Most theorems turn out to be proof-theoretically very simple. Now that we have a proof of Fermat's Last Theorem, the logician Angus MacIntyre has been able to outline it within first-order arithmetic. This view of the proof involves simple incremental accretions of truth. The discovery of the proof was something very different, as is our understanding of it.

We are beginning to see a pattern—literally: simple rules, unbounded iteration, emergent forms—defined at the edge of computability. This is just what Turing later observed in nature and mathematically tried to capture.

Another hugely important mathematical tool to take on our explorations of the incomputable is the oracle Turing machine, also tucked away on one page of Turing's 1939 paper. The idea was to allow the machine to compute relative to a given real which may or may not be computable. If one looked at the function computed *using* this oracle, one could frame the function as being computed *from* the oracular real as argument. It is then a small step to summarize what the machine does as computing one real from another. It delivers us a computational model within which to fit basic computable laws of nature, namely, most of what underlies our knowledge of how the world works. Harking back to the computable numbers of 1936, the oracle machine computes a real number but *from another* real number. We can allow the machine to compute relative to different oracles. Then we can acknowledge the higher type nature of the computational process by calling the mappings computed by oracle machines *Turing functionals*. These, acting over the reals, give us the *Turing universe*.

In fact, Turing was not notably interested in the mathematical development of his oracle machines despite his preoccupation with computers that interacted. It was left to another seminal figure, Emil Post, to gather together equivalence classes of reals—or *degrees of unsolvability*—that were computable from each other. Then, using an ordering induced by the ordering of reals via Turing functionals, Post obtained a structure that has become known simply as the *Turing degrees*.

Three key things we observe about this structure are: firstly, that it is very complex; secondly, that if we take some scientific domain described in terms of real numbers and computable laws over them, then it is embeddable in the Turing universe, so that the structure of the corresponding restriction of the Turing universe tells us something about the causal structure of the real world (the causal third dimension of the Gamow-like two-dimensional person we met earlier); thirdly, we can view this model as a terrain in which computation can be hosted but in which information takes a leading role, structuring the form of the computation—a *reembodiment* of computation.

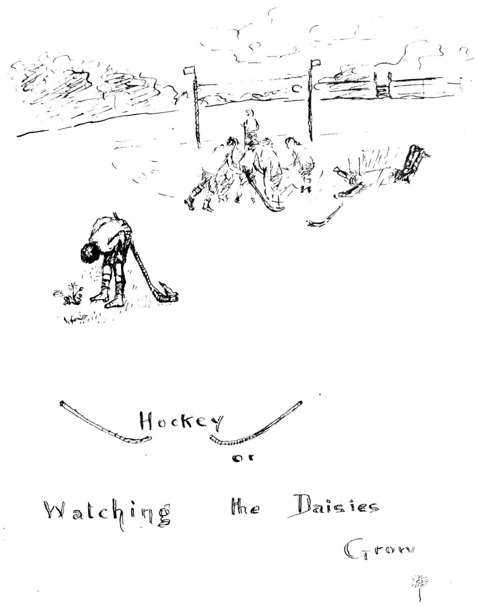

Figure 3. Drawing of Alan Turing by his mother, at his preparatory school, Hazelhurst, Sussex, 1923. ***Image courtesy of Sherborne School.***

Messages from the Real World

Turing's final years in Manchester saw both personal eclipse and the sowing of the seeds of a later renaissance—a growing renown and increasing scientific impact. Today, his fame appears as a mixed blessing: too many webpages and articles in periodicals with misleading information about what Turing did and did not do. On the other hand, having mathematicians such as Turing and Gödel in the public eye is good for basic science. As far as the science itself goes, Turing's work has been powerfully influential in a piecemeal way, with a number of different fields laying claim to particular bits of Turing. In the October 2004 *Notices*, Lenore Blum wrote a nice description of the dichotomy between "two traditions of computation" (in "Computing over the reals: Where Turing meets Newton", pp. 1024–1034):

> The two major traditions of the theory of computation have, for the most part, run a parallel nonintersecting course. On the one hand, we have numerical analysis and scientific computation; on the other hand, we have the tradition of computation theory arising from logic and computer science.

Turing's 1948 "Rounding-off errors in matrix processes" was influential in the former and the 1936 Turing machine paper in the latter.

Now there is a growing appreciation of the coherence of approach represented by these different contributions. On the one hand, we have a computational world over which we have control; on the other, we must live with approximations and errors. As we move up the informational-type structure from discrete to continuous data, we lose the sure footholds but identify emergent controls at higher levels. Turing had earlier introduced Bayesian code breaking methods at Bletchley Park in another adjustment to the realities of extracting form from a complex world.

Turing's late great contributions traced the computational content of phase transitions in the real world from two different vantage points. Turing had been interested in the emergence of form in nature from his school days; see his mother's sketch (Figure 3) of Alan "Watching the Daisies Grow". A short piece by Peter Saunders in the forthcoming *Alan Turing—His Work and Impact* ([7], edited by Cooper and van Leeuwen) discusses Turing's motivation and background reading in getting interested in morphogenesis. Saunders writes:

> The obvious question to ask about "The Chemical Basis of Morphogenesis" is why Turing took up the problem at all. Pattern formation, interesting though it may be to biologists, does not look like the sort of fundamental problem that Turing characteristically chose to devote his time and effort to. The answer is simply that he saw it not as a mere puzzle but as a way of addressing what he considered to be a crucial issue in biology. As he said to his student Robin Gandy, his aim was to "defeat the argument from design."

It seems Turing wanted to add a bit more computational convergence to Darwin's theory and not leave an opportunity for God to tidy up. In the same volume, Philip Maini outlines the way in which Turing produced complex outcomes from simple algorithmic ingredients:

> Alan Turing's paper, "The chemical basis of morphogenesis" [35] has been hugely influential in a number of areas. In this paper, Turing proposed that biological pattern formation arises in response to a chemical pre-pattern which, in turn, is set up by a process which is now known as *diffusion-driven instability* [see Figure 4]. The genius of this work was that he considered a system which was stable in the absence of diffusion and then showed that the addition of diffusion, which is naturally stabilizing, actually caused an instability. Thus it was the integration of the parts that was as crucial to the understanding of embryological development as the parts themselves—patterns emerged or self-organized as a result of the individual parts interacting. To see how far ahead of his time he was, one has to note that it is only now in the post-genomic era of systems biology that the majority of the scientific community has arrived at the conclusion he came to some 60 years ago.

Things have moved on since 1952, but the basic approach retains a powerful influence on the field, hardly noticed by logicians and computer scientists as a group.

More importantly for us, Turing's examples point to general principles underlying emergent phenomena. Turing would have been a teenager at the time the "British emergentists" such as C. D. Broad, Samuel Alexander, and C. Lloyd Morgan were at their height. Broad and Turing overlapped at Cambridge. As a group, they were very prescient in seeking out examples of the complex arising from simple rules in highly connected environments. Unfortunately, some of their examples of emergence taken from chemistry turned out to be explainable in terms of quantum mechanics. But Turing's differential equations gave us a new sense both of the character and the origins of emergence and indicated how emergent form might be captured mathematically.

Turing's equations might well have computable solutions, but they pointed to the principle of emergent patterns on animal coats, etc., corresponding to definable relations over basic mathematical structure. If one could define a relation in nature, it had a robustness, a tangible presence that one might expect to find observable, just as Turing had brought us to expect observed emergent phenomena to have descriptions. These descriptions, if at least as complicated as that giving us the halting problem, could be expected to lead to incomputable sets. Somewhere between the tidy abstraction of the universal Turing machine and the mysteries of emergence in nature we have the fractal family. Well known as analogues of emergent phenomena, they also have their own well-defined mathematics. For various reasons, the most informative of these is the Mandelbrot set (detail Figure 5). As Roger Penrose puts it in his 1994 book *The Emperor's New Mind* [26]:

> Now we witnessed…a certain extraordinarily complicated looking set, namely the Mandelbrot set. Although the rules which provide its definition are surprisingly simple, the set itself exhibits an endless variety of highly elaborate structures.

With its definition based on a well-known simple equation over the complex numbers, the quantifier form of the complement of the Mandelbrot set can be reduced to something similar to that of the halting set of the UTM. Using this, one can simulate it on a computer screen, getting the fascinating range of images we recognize so easily. It has what the halting problem does not: the visual embodiment of a natural phenomenon. This enables us to appreciate the higher-order intricacy we encounter as we travel deeper and deeper into this endlessly surprising mathematical object. It is harder to be sure of one's notion of computability in this context. But for the computable analysts, the computability of the Mandelbrot set is still a challenging open problem.

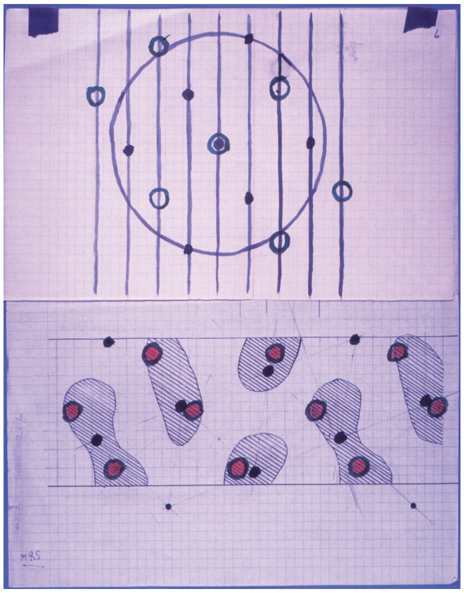

Figure 4. Colored diagrams showing patterns of dappling and calculations, made by Turing in connection with work on morphogenesis. ***Courtesy of P. N. Furbank.***

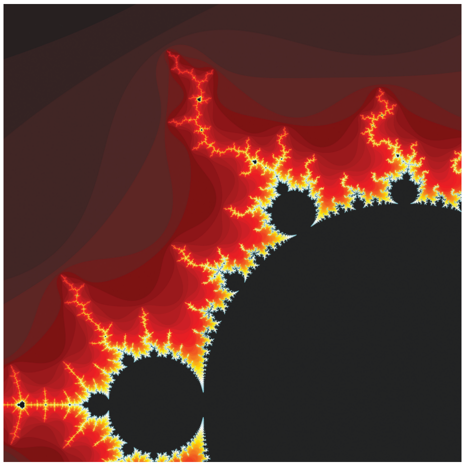

Figure 5. Mandelbrot set. ***Courtesy of Niall Douglas.***

Compared to other levels of the real world that Turing was drawn to, emergence seems relatively straightforwardly mathematical. One has an objective view of the whole picture, basic rules, and emergence of a surprising character—the surprise is one of the criteria for emergence, though there is no proper definition. Mathematically, we tend to look for quantifiers or nonlinearity in the description based on the basic operations, a kind of association of true emergence with halting problem-like incomputability. At the quantum level, which Turing was always very interested in but did not live long enough to get a grip on, the problem of pinning down the basic causality is not so easy. We are looking down from above and are never sure we have the whole picture. And the phase transition from quantum ambiguity to the familiar classical world seems to involve not just definability but a transition from a structure in which there are some prohibitions on simultaneous definition of entities—something familiar to model theorists—which disappear as one passes from particle physics to other scientific areas. There is experimental evidence that such breakdowns of definability occur in human mentality. The problem we as observers have in this context is not one of viewing from above but of being trapped inside, although modern neuroscience is augmenting this inner view with a huge amount of useful information.

In his final years Turing approached brain functionality from two directions: mathematically modelling the physical connectivity of the brain

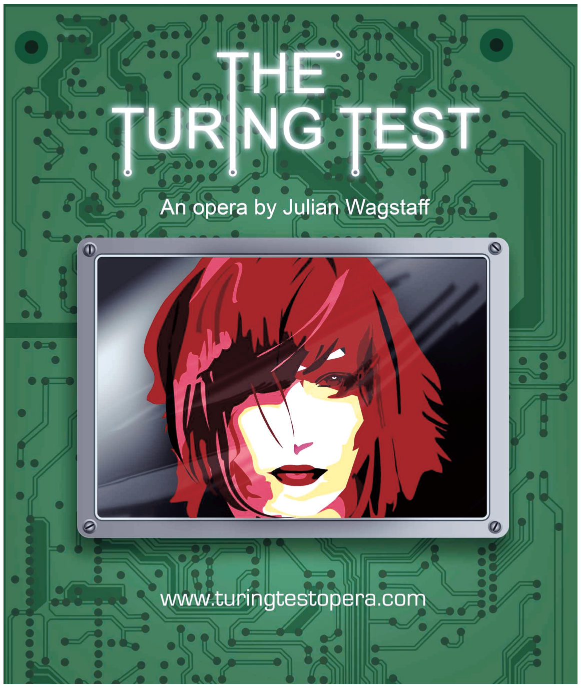

Figure 6. Turing Test image by Joe Smith and Pete Rix. ***Image courtesy of the composer, Julian Wagstaff.***

and via his much better known discussion of intelligent thought. The latter is of special interest for mathematicians, and not just for its relevance to incomputability. Around the same time as Turing was approaching "intuition and ingenuity" via his hierarchical analysis of the limits of Gödel's theorem, Jacques Hadamard was covering very similar ground from a more sociological viewpoint. Given that the mathematical product is presented algorithmically via a proof, the associated mathematical thinking becomes a good case study for clarifying the "intuition versus ingenuity" dichotomy. A principal source for Hadamard's 1945 book *The Psychology of Invention in the Mathematical Field* [16] were lectures of Henri Poincaré to the Société de Psychologie in Paris in the early part of the twentieth century. Here is Hadamard's account of an example of apparent nonalgorithmic thinking:

> At first Poincaré attacked [a problem] vainly for a fortnight, attempting to prove there could not be any such function…[quoting Poincaré]: "Having reached Coutances, we entered an omnibus to go some place or other. At the moment when I put my foot on the step, the idea came to me, without anything in my former thoughts seeming to have paved the way for it…I did not verify the idea…I went on with a conversation already commenced, but I felt a perfect certainty. On my return to Caen, for conscience sake, I verified the result at my leisure."

Many writers focus on the surprise and dissociation from consciously rational thought. What is also striking is the connection with Turing's 1939 paper—the "perfect certainty" that Poincaré experienced. Did Poincaré have all the details of the proof immediately sorted in his mind? Unlikely. What we understand from Turing's analysis is that there is a process of definition, with a range of proofs emerging. Poincaré extracted one of these proofs on his return to Caen.

As well as the connection back to Turing's 1939 paper, there is the connection forward to his 1952 paper and the final work defining emergence of form in nature. Since 1954 neuroscience is one of the areas in which emergence has taken centerstage for many researchers. Turing's connection between definability in terms of both logical structure and physical context is a remarkable anticipation of current thinking.

At the bottom of the process is the basic physical functionality, and Turing made his own groundbreaking contribution to connectionist models of the brain in his unpublished 1948 National Physical Laboratory report *Intelligent Machinery* [36]. Called unorganized machines by Turing, they were preempted by the more famous neural nets of McCulloch and Pitts. See Christof Teuscher's book *Turing's Connectionism* [30]. These are Teuscher's comments on the history in his commentary on *Intelligent Machinery* in *Alan Turing—His Work and Impact* [31]:

> In his work, Turing makes no reference to McCulloch and Pitts' 1943 paper, nor do they mention Turing's work [on unorganised machines]. It is an open question how much their work influenced each other, yet, we have to assume that they were at least aware of each other's ideas. We hypothesize that both bad timing and the fact that Turing's neurons are simpler and more abstract contributed to his work being largely ignored.

Connectionist models have come a long way since Turing's time. Their physical emulation of the brain does bring dividends. Paul Smolensky, for instance, talks in his 1988 paper "On the proper treatment of connectionism" [28] of a possible challenge to "the strong construal of Church's Thesis as the claim that the class of well-defined computations is exhausted by those of Turing machines."

Of course, it was the celebrated 1950 *Mind* paper "Computing machinery and intelligence" [34] that became one of Turing's three most cited papers. The Turing Test for intelligence has entered the popular culture, with an opera (see Figure 6) and students walking around with T-shirts proclaiming "I failed the Turing Test".

So what has all this to do with the mathematics of incomputability? Turing's focus on the logical structure of computation has had huge influence on modern thinking. Nevertheless, since his time at Bletchley Park, Turing was involved in many ways with *embodied* computation. Implicit in this respect for embodiment is a recognition of a difference in the way humans (and intelligent machines in general) interact with—are *embedded* in—*information.* This interaction is seen as a necessary attribute of intelligence. People involved with taking artificial intelligence forward have had to take on board this extended, embodied, physical, information-respecting model of the future. We give some brief excerpts from Rodney Brooks's contribution ("The Case for Embodied Intelligence") [4] to *Alan Turing—His Work and Impact*:

> For me Alan Turing's 1948 paper Intelligent Machinery was more important than his 1950 paper Computing Machinery and Intelligence.…For me, the critical, and new, insights in Intelligent Machinery were two fold. First, Turing made the distinction between embodied and disembodied intelligence.… Modern researchers are now seriously investigating the embodied approach to intelligence and have rediscovered the importance of interaction with people as the basis for intelligence. My own work for the last twenty-five years has been based on these two ideas.

Turing Points the Way Past the Turing Barrier

Alan Turing's work was incomplete. For Turing the end came too early via an uneasy confluence of algorithm (the UK law of the time) and incomputability, a bizarre piece of history that nobody could have invented for a man who had served mathematics and science—and his country—so well. Some have questioned the description "computability theory" for the subject Turing cofounded with other luminaries of the period—Gödel, Post, Church, Kleene—because it deals primarily with the incomputable.

Turing (Figure 7) was a mathematician of his time who worked from within the world, trying to give mathematical substance to physical and mental processes. He gave us a basic model of what we understand to be computation. He observed computation as an organic whole, discovering incomputability as a definability-theoretic extension of Gödel's incompleteness theorem. He encouraged us to see the universe as something that does compute and to engage with its features, tracing embodied analogues of his halting problem within biology and neuroscience. He loved the truth and was open to doubts about his doubts. He thought a machine could not be intelligent if it was expected to be infallible. He saw something different about an embedded computer and was drawn to the computational mysteries of quantum theory.

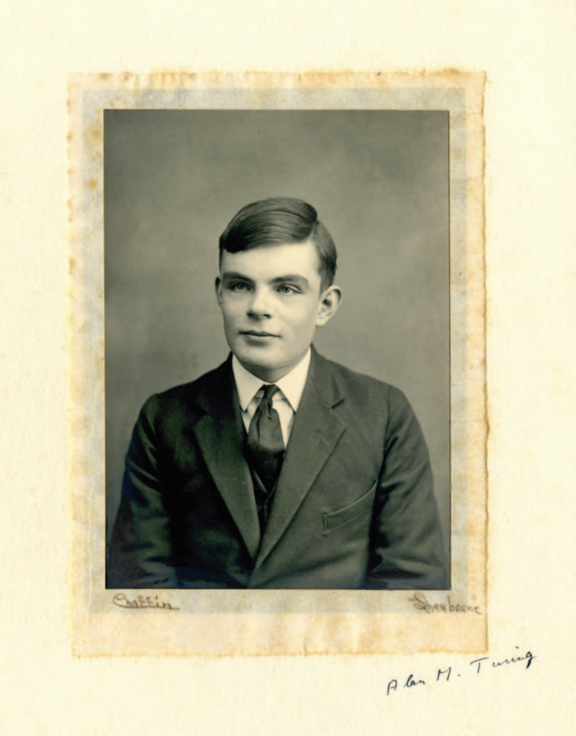

Figure 7. Alan Turing in 1928, age 16. ***Image courtesy of Sherborne School.***

Turing did not live long enough to appreciate Stephen Kleene's investigations of higher-type computability but would surely have made the connection between the mathematics of incomputability, definability, and the computations that arise from them. He did not see the mathematical theory of randomness flourish or the yearly award of a prize in his name for a subject he played a founding role in. He never saw the employment of thousands who talked about "Turing" machines and a "Turing" test. The "incomputable reality" (as *Nature* [6] recently described it) is still dangerous to inhabit. But we do have a lot to celebrate in 2012.

Bibliography

1. S. Alexander, *Space, Time, and Deity*, Vol. 2, 1927.
2. L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and Real Computation,* Springer, 1997.
3. C. D. Broad, *The Mind and Its Place in Nature*, Kegan-Paul, London, 1923.
4. R. Brooks, The case for embodied intelligence, in *Alan Turing—His Work and Impact* (S. B. Cooper and J. van Leeuwen, eds.), Elsevier, 2012.

5. S. B. COOPER, *Computability Theory*, Chapman & Hall/CRC Press, Boca Raton, FL, New York, London, 2004.

6. ______, Turing centenary: The incomputable reality, *Nature*, **482**:465, 2012.

7. S. B. COOPER and J. VAN LEEUWEN (eds.), *Alan Turing—His Work and Impact*, Elsevier, 2012.

8. A. R. DAMASIO, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Brace, 1999.

9. M. DAVIS, *The Universal Computer: The Road from Leibniz to Turing*, A K Peters/CRC Press, 2011.

10. A. EINSTEIN, *Out of My Later Years*, volume 48, Philosophical Library, 1950.

11. S. FEFERMAN, Transfinite recursive progressions of axiomatic theories, *J. Symbolic Logic,* **27**:259–316, 1962.

12. S. FEFERMAN, Turing in the Land of O(z), in *The Universal Turing Machine: A Half-Century Survey* (R. Herken, ed.), Oxford University Press, New York, 1988, pp. 113–147.

13. R. P. FEYNMAN, Simulating physics with computers, *Int. J. Theoretical Physics*, **21**:467–488, 1981/82.

14. G. GAMOW, *One, Two, Three…Infinity* (1947, revised 1961), Viking Press (copyright renewed by Barbara Gamow, 1974), Dover Publications.

15. R. O. GANDY, The confluence of ideas in 1936, in *The Universal Turing Machine: A Half-Century Survey* (R. Herken, ed.), Oxford University Press, New York, 1988, pp. 51–102.

16. J. HADAMARD, *The Psychology of Invention in the Mathematical Field*, Princeton Univ. Press, Princeton, 1945.

17. W. HASKER, *The Emergent Self,* Cornell University Press, Ithaca, London, 1999.

18. A. HODGES, *Alan Turing: The Enigma*, Vintage, London, Melbourne, Johannesburg, 1992.

19. J. KIM, *Physicalism, or Something Near Enough*, Princeton University Press, Princeton, Oxford, 2005.

20. S. C. KLEENE, Recursive functionals and quantifiers of finite types. I, *Trans. Amer. Math. Soc.*, **91**:1–52, 1959.

21. S. C. KLEENE, Recursive functionals and quantifiers of finite types. II, *Trans. Amer. Math. Soc.*, **108**:106–142, 1963.

22. W. MCCULLOCH and W. PITTS, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, **5**:115–133, 1943.

23 B. P. MCLAUGHLIN, The rise and fall of British emergentism, in *Emergence or Reduction?—Essays on the Prospects of Nonreductive Physicalism* (A. Beckermann, H. Flohr, J. Kim, eds.), de Gruyter, Berlin, 1992, pp. 49–93.

24. J. MYHILL, Creative sets, *Z. Math. Logik Grundlagen Math.*, **1**:97–108, 1955.

25. P. ODIFREDDI, *Classical Recursion Theory*, North-Holland, Amsterdam, New York, Oxford, 1989.

26. R. PENROSE, *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*, Oxford University Press, Oxford, New York, Melbourne, 2002.

27. E. L. POST, Degrees of recursive unsolvability: Preliminary report (abstract), *Bull. Amer. Math. Soc.*, **54**:641–642, 1948.

28. P. SMOLENSKY, On the proper treatment of connectionism, *Behavioral and Brain Sciences*, **11**:1–74, 1988.

29. R. I. SOARE, Turing computability and information content*, Philos. Trans. Royal Soc. London*, *Series A*, to appear.

30. C. TEUSCHER, *Turing's Connectionism, An Investigation of Neural Network Architectures*, Springer-Verlag, London, 2002.

31. ______, A modern perspective on Turing's unorganized machines, in *Alan Turing—His Work and Impact* (S. B. Cooper and J. van Leeuwen, eds.), Elsevier, 2012.

32. A. M. TURING, On computable numbers with an application to the Entscheidungsproblem, *Proc. London Math. Soc.* (3), 42:230–265, 1936. A correction, **43**:544–546, 1937.

33. ______, Systems of logic based on ordinals, *Proc. London Math. Soc.* (3), **45**:161–228, 1939.

34 ______, Computing machinery and intelligence, *Mind*, **59**:433–460, 1950.

35. ______, The chemical basis of morphogenesis, *Philos. Trans. Royal Soc. London. Series B*, *Biological Sciences*, **237**(641):37–72, 1952.

36. ______, Intelligent machinery, in D. C. Ince, (ed.), *Collected Works of A. M. Turing—Mechanical Intelligence*, Elsevier Science Publishers, 1992.

37. J. VON NEUMANN, The general and logical theory of automata, in: L. A. Jeffress (ed.), *Cerebral Mechanisms in Behaviour: The Hixon Symposium*, September 1948, Pasadena, Wiley & Sons, New York, 1951.

38. H. WANG, *Reflections on Kurt Gödel*, MIT Press, Cambridge, MA, 1987.