Pushing Back the Incomputable – Alan Turing's Ten Big Ideas

S Barry Cooper

There is nothing remarkable about mathematicians' achievements going unrecognised in a wider world changed by their discoveries. The hidden history of Alan Turing is just a particularly bizarre example — one we can expect to become much better known during the 2012 centenary of his birth. The mental check-list of things that make Turing remembered includes:

- Being just 24 years old when he came up with the idea of the "stored program" computer, basically the blue-print for every computer in existence today;
- His leading role at the secret decoding centre at Bletchley Park, helping shorten the 2nd World War by two years with his ground-breaking involvement in building and fully exploiting decoding machines;
- His seminal role in the actual designing and programming the early computers after the war, and his still important influence on how computer scientists see artificial intelligence;
- The innovative and original work in bringing mathematics to bear on important problems in biology and medicine;
- And his disgraceful neglect, and prosecution for being gay, in 1950s Manchester.

But for many of us the peculiar resonance between the personal and the scientific makes Turing specially iconic. And the visionary form this interaction took gives Turing's writings a relevance and impact which continues to this day. There are many facets to Turing's life that echo through mathematics, computing, physics and biology, through philosophy, and through economics, the humanities and the creative arts.

Why is that? Turing's life and science inhabited that mysterious region between the computable and the incomputable. Both in his research and in his life he persistently tried to make sense of things in what we can only describe as a computational sense. His well-known eccentricities such as chaining his mug to the radiator at Bletchley Park or riding his bicycle in the summer wearing a gas-mask, might be seen as the product of constructive thinking; and in his mathematics he was always wanting to bring the real world within the purview of computational mathematics. But he was ever meeting and dealing in creative ways with the inevitable challenges to computability, sometimes succeeding, as with his universal computing machine, or his war-time cryptography, or his mathematical modelling of cows spots, or zebra's stripes or the moving patterns of tropical fish — or mapping out the limits of computation as with his unsolvable halting problem, or his hierarchy of theories and his oracle Turing machines in his insufficiently understood 1939 paper. Of course, the interface between the computable and the incomputable is a hazardous area, as he found in the closing years of his life, bringing cruel uncertainties and a quite unpredictable end.

Alan Turing, aged 16, 1928 (photo in courtesy of Sherborne School archive)

1. Computation Disembodied

The 17th century saw a dramatic change in the balance between computational and descriptive sway in science. Robert Hooke may have toyed with the inverse square law in physics, but it is Isaac Newton's mathematics which delivers not only persuasion but computational and predictive content to the intuitive descriptions. The computational gives surety, gives ease of comparison between prediction and observation, and comes as a memetic package more easily passed between researcher and practitioner.

The Turing machine did for computational mathematics what Newton's computational mathematics did for his particle dynamics. The mathematics *disembodied* the science. It turned computation into *computer science*. Gone was the taxonomy of calculating machines built differently for different computational tasks. The hardware was trivial and did not need to be changed. The basic actions of the machine were as simple as could be. But the "machine" could compute anything a standard calculating machine could. While all the computing power lay in the *program*.

More generally, it enabled many to frame the familiar expectations of science encouraged by Newton — the so-called Laplacian model — within a precise *mathematical* model. Of course, the Newtonian model came with a "best before" date, one clear to the successors of the man who said (Albert Einstein, p. 54, "Out of My Later Years" 1950):

> "When we say that we understand a group of natural phenomena, we mean that we have found a constructive theory which embraces them."

Today, we take forward some of Turing's own questionings of the comprehensiveness of his disembodied computational model.

2. Universality, and Programs as Data

Of course, aspects of the 1936 Turing model were anticipated by others, such as Emil Post. The key extra ingredient was *universality*, based on the *coding* of machines as data. This essential feature of today's computer is often not understood — though was certainly recognised by John von Neumann, and implemented in his 1945 EDVAC report, which was so influential in the later development of the stored program computer. Von Neumann later acknowledged Turing's role in his 1948 Hixon Symposium lecture.

Photo taken in 1930 April, Bedruthan Steps, Cornwall
(photo in courtesy of Sherborne School archive)

Although the practical impact of Turing's universal machine is difficult to disentangle from the complexities of the early history of the computer, it established a hugely influential computing paradigm — that of the omnipotent computer. It encouraged the development of the functionalist perspective on human cognition and artificial intelligence, as in Hilary Putnam's *Minds and Machines* from 1960. The embodiment of human thinking is relegated to a subservient role, mirroring that of the Turing's universal machine. Turing himself is said by Andrew Hodges to have spoken to Donald Bayley in 1944 of "building a brain".

A more limited expression of the paradigm, in computing, is that of the *virtual machine* originally associated with IBM around 1965. The overriding concept is of varied computational environments being realisable independently of the particular hardware.

3. Programs as Data *Embodied*

Of course, a huge amount of work and ingenuity went into actually *building* universal machines, and Turing was very much part of this.

The early programmable machines were certainly not universal. The "program as data" handling facility of today's computers involves hard-won *embodied* elements of Turing's abstraction. The first stored-program computer that worked was the Manchester "Baby" from 1948. By this criterion, out go pioneering machines such as that of John Atanasoff ("the first electronic digital computer"), Charles Babbage (the Analytical Engine from 1837), Konrad Zuse, or the Turing Bombe, Colossus and ENIAC — all had their programming very much embodied via external tapes and the like. For instance, Tony Sale describes how the programming of Colossus was a far cry from the disembodiment of the universal Turing machine, depending as it did on a combination of tapes, telephone jack-plugs, cords and switches.

Turing became increasingly marginalised during these dramatic developments. A small version of his Automatic Computing Engine described in his 1945 report for the National Physical Laboratory was eventually built (the Pilot ACE) by 1950, by which time Turing had disappeared to Manchester.

What is striking is that Turing never shared the disdain or superficial reductionism of many mathematicians. He was fascinated by the actual building of computing machines, and always willing to engage with the physicality and sheer messiness of computational processes. And this was to pay dividends in his later work on mechanical intelligence and morphogenesis. Today, it is a willingness to engage with nature at the most basic level that informs some very necessary rethinking about computing in the real world, and gives mathematicians an important multidisciplinary role.

4. Information — Hiding and Unhiding

However one views mathematics, there is no doubting its important role in decoding the world we live in. To Winston Churchill, Alan Turing and the thousands who gave up years of their lives to secret activity at Bletchley Park were "the geese that laid the golden eggs but never cackled". In retrospect, it is battery hens that come to mind. Increasingly, scientists are misunderstood and given ill thought out hoops to jump through. Great science is organised according to algorithms which Turing's science tells us are unlikely to be intelligent. Bletchley Park was central to Turing's career, and must have been an intense and personally formative part of his life, and of many others. Things would never be the same after. Of course, their machines and their lives there made as if they had never happened. It would be nearly two decades after Turing's passing before the world started to decode the achievements of those years.

5. The Discovery of Unsolvability

If Alan Turing was peculiarly misunderstood as a one of the worlds great scientists, incomputability may be a correspondingly important and misunderstood part of his scientific legacy. If people know who Turing was, it is for Turing machines, decoding the Enigma, or computers. Or it could be for his ending, 1950s "normality" fractured by a coming together of events of startling unpredictability. But few will make the connection with the mathematics of incomputability.

Only six years before Turing's "computable numbers" paper, David Hilbert had famously proclaimed in Königsberg, during an opening address to the Society of German Scientists and Physicians, 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."

Turing's unsolvable problem was that of deciding whether his universal machine would successfully compute or not. And the corollary, known for many years as "Church's Theorem", was the counter-intuitive fact that there is no computer program for deciding of a given sentence of first-order logic whether it is logically valid or not.

These are quite striking and interesting facts, with clever proofs. But there is no obviously embodied counterpart. And — as the proof-theorists have managed to show — most of the *interesting* mathematical problems reside well within this so-called "Turing barrier". But challenges to computability fascinated Turing, and the mathematics of incomputability was not to be so easily sidelined.

6. Mapping the Road to the Incomputable

Of all Turing's papers, his 1939 one on *Systems of logic based on ordinals* is the least understood. There was an underlying idea that we might be able to explore the incomputable via iterated approximation, maybe even to find a way to compute beyond the Turing (machine) barrier. What he found was that there might exist computable routes into the incomputable. But it was the *finding* of the routes that defeated the machine. Of course, the mathematician is very familiar with this phenomenon. There is the well-known story of Poincaré getting stuck on a problem, leaving off to go on a bus journey, and the solution coming to him complete and memetic independently of conscious rational thought. How often do we solve a problem according to some very personal process, only to convert the solution into something formal and communicable to our peers? Turing's mathematics gives us an explanation of why written proofs often do not tell us how the proof was discovered. The question arose — does the brain somehow support non-algorithmic thought processes?

7. Oracles and Interactivity

Buried away in this long 1939 paper is a single page which had a huge impact on the mathematics of the incomputable. The world around us is a world of information, and we cannot be sure all this information originated computably — for instance, it might have been delivered via a quantum random phenomenon, which by recent work of Calude and Svozil may well involve incomputability. Turing devised a machine to compute using real numbers which were not necessarily computable, and in so doing provided a model for computation relative to embodied information. How prescient. Our computers are no longer just Turing machines. They are part of a hugely complex computational world which collectively creates and exchanges new information. And our material universe is inhabited by computable causality within an embodied environment of great informational complexity, a computational context demanding proper analysis.

Strangely, despite Turing's later interest in interactive computation, he never seems to have returned to his oracle Turing machine model. The mathematical development was left to Emil Post and Stephen Kleene and their successors, and has since become a rich field of research which promises real-world returns Turing would find fascinating. The key to these is a reclaiming of the incomputable via the sort of embodied hierarchical development Turing envisaged back in the late 1930s. Achieved with the benefit of what we know now about global relations and their links to observed emergence.

8. Modelling the Brain

Some of Turing's most interesting work — sadly cut off in 1954 — was done in his last few years. For Turing, the human brain had ever been both inspiration and challenge to his work on computing machines. And he attempted to bring a characteristically basic approach to both the physical and the mental, those two irksome companions of the philosopher of mind. Here is Jaegwon Kim (in *Physicalism, or Something Near Enough*, Princeton, 2005) setting out the problem:

> "...the problem of mental causation is solvable only if mentality is physically reducible; however, phenomenal consciousness resists physical reduction, putting its causal efficacy in peril."

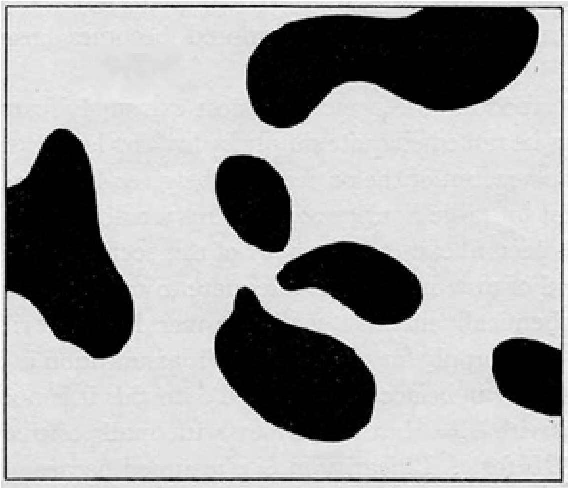
How can mentality have a causal role in a world that is fundamentally physical? And what about "overdetermination" — the problem of phenomena having both mental and physical causes? The most that most philosophers of mind can agree on is a degree of *supervenient* of mental properties on physical ones.

Turing in 1948 came up with his "unorganised machines" which provided a neural net model alternative to the better known predecessor of Warren McCulloch and Walter Pitts. Christof Teuscher gives an account of the innovative nature of "Turing's Connectionism" in his book of that name.

Connectionist models have provided the basis for a large research field, and exhibited interesting features in keeping with what one might expect from the human brain. Paul Smolensky, for instance, talks in his 1988 paper *On the proper treatment of connectionism* 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."

9. The Turing Test and AI

At the other end of the scale we have Turing's famous 1950 paper in *Mind* astutely narrowing down what one can sensibly say about human intelligence, and discussing in some detail his observer-based test for a thinking machine. The resulting "Turing Test" still dominates people's



Turing computer-generated dappled pattern

thinking on the issue. The paper joins the other two most cited papers of Turing. One of these is the 1936 paper of course, which many might expect to be the most frequently cited of his papers. But no …

10. How Nature Computes

To the surprise of those outside of biology and medicine, the most cited of Turing's papers is the final 1952 *The Chemical Basis of Morphogenesis*. And in many ways this is one of his most original and maybe visionary foray into the world of computation. He was not to know that the mathematics of sunflowers and patterns on animal coats would connect up with today's recognition of the importance of emergence, and throw light on a whole range of intractable foundational questions across a wide range of research areas in science and the humanities. Computationally simple rules, connectivity, emergent forms at the edge of computability, and definable in terms of the rules, just like Turing's patterns. Turing's coherence of vision, at the end of his short life, giving us morphogenesis — inhabiting the same fractal world as the Mandelbrot set; the same computational world as the halting problem for the universal Turing machine; the same large scale structure as found in the observable universe; and perhaps the key to Kim's world of supervenience.

11. The Alan Turing Year

So, what will we be celebrating in 2012? Above all, it should be the continued influence of the Turing vision on some of the most important research directions today. Turing had an amazing instinct for recognising big questions about how the world works. He was like another famous 20th century scientist, Paul Dirac, in having a very down-to-earth grasp of the what-makes-the-world-tick, combined with a brilliant grasp of abstract structures. Turing's work on the nature of computation has defined the computer revolution that has changed our world. And his groundbreaking explorations of processes beyond what a computer can handle look likely to provide key elements of the next trans-computer developments. We should celebrate how Turing combined the practical and the visionary, and gave us both technological breakthroughs and a continuing sense of the mystery of what lies beyond.

S Barry Cooper
University of Leeds, UK

Barry Cooper is Professor of Mathematical Logic at the University of Leeds. He is President of the Association Computability in Europe and a Managing Editor of the journal "Computability". He is currently Chair of the Turing Centenary Committee and Co-Chair of the Turing Centenary Conference in Cambridge in June 2012. The 2nd edition of his book *Computability Theory* is due out in early 2012. He is editing with Turing's biographer Andrew Hodges "The Once and Future Turing" for Cambridge University Press.