

How Can Nature Help Us Compute?

S. Barry Cooper

School of Mathematics, University of Leeds
Leeds LS2 9JT, U.K.
pmt6sbc@leeds.ac.uk
<http://www.maths.leeds.ac.uk/~pmt6sbc>

Abstract. Ever since Alan Turing gave us a machine model of algorithmic computation, there have been questions about how widely it is applicable (some asked by Turing himself). Although the computer on our desk can be viewed in isolation as a Universal Turing Machine, there are many examples in nature of what looks like computation, but for which there is no well-understood model. In many areas, we have to come to terms with emergence not being clearly algorithmic. The positive side of this is the growth of new computational paradigms based on metaphors for natural phenomena, and the devising of very informative computer simulations got from copying nature. This talk is concerned with general questions such as:

- Can natural computation, in its various forms, provide us with genuinely new ways of computing?
- To what extent can natural processes be captured computationally?
- Is there a universal model underlying these new paradigms?

1 Introduction

Freeman Dyson, in his introduction to George Odifreddi's [27] *The Mathematical Century : The 30 Greatest Problems of the Last 100 Years*, divides scientists into Cartesians and Baconians:

“According to Bacon, scientists should travel over the earth collecting facts, until the accumulated facts reveal how Nature works. The scientists will then induce from the facts the laws that Nature obeys. According to Descartes, scientists should stay at home and deduce the laws of Nature by pure thought. . . . Faraday and Darwin and Rutherford were Baconians: Pascal and Laplace and Poincaré were Cartesians. Science was greatly enriched by the cross-fertilization of the two contrasting . . . cultures.”

When it comes to computability, an important role of Cartesians has been to theoretically map out the boundaries of what is practically computable, while Baconians may point to new computational paradigms in the real world, so challenging theoretical barriers. Here too there is a synergistic relationship between these two approaches, and many researchers (Alan Turing is an obvious example of a Baconian delimiter cum Cartesian practitioner) move between them, with varying degrees of ease and success.

In this short article we make some observations on the extent to which this is important for current attempts to take computing to radically new levels, and to try to give a modest but much needed Cartesian shove to the search for new ways of computing. And far from being put off by the difficulties hypercomputationalists (like Hava Siegelman, Jack Copeland, and Tien Kieu) have run into — with Martin Davis [13], like Tom Sawyer’s Aunt Polly, admonishing their foolishness and metaphorically packing them off to bed with no supper — we will argue for a positive role for the black box model of computation, despite its being wielded by Davis with such destructive effect.

Of course, whenever one attempts to characterise some process, one is imposing some kind of inductive structure on nature, often of a particularly simple kind. The argument here is that new computational paradigms are in evidence when nature goes beyond that induction. That homogeneity of information is unknown in nature with its variable divide between matter (information) and energy (algorithmic content). And that on this, and the breakdown of inductive structure, rests a powerful mechanism for elevating information content — one which may well be modelled in new kinds of computers.

2 Natural phenomena as discipline problem — or how we found out that nature computes differently to us

The relationship between nature and computation has always involved a two-way process comprised of observation, prediction and theory. For the scientist, caught by the dream of Laplace’s [23] predictive ‘demon’, the special contribution of nature to the way we think has not been an explicit one. Nature more discipline problem than role model. Implicit in the search for a theory of everything is the assumption that it is a short step from understanding to prediction:

“Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situations of the beings who compose it — an intelligence sufficiently vast to submit these data to analysis — it would embrace in the same formula the movements of the greatest bodies and those of the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes.”

But when Albert Einstein [14] wrote in 1950 (p.54 of *Out of My Later Years*):

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

he opens the door to a world in which a mainstream scientific theory may struggle for predictive consequences, and in which nature may determine observable phenomena, based on well-understood local mechanisms, which are not globally predictable. At the same time we have van Leeuwen and Wiedermann’s [39] observation that “the classical Turing paradigm may no longer be fully appropriate to capture all features of present-day computing.”

At the mathematical level, the 1930s had already seen the discovery of a whole range of observable, but not predictable, phenomena. As we all know,

Turing [36] showed we cannot predict in general whether a given computation of a computer will ever terminate. And (along with Church [6]) that recognising the non-validity of an argument may completely elude us, even though Gödel had given us a computable procedure for listing all valid mathematical arguments. But, as described in [9], the more natural the examples of incomputable sets in mathematics became, the more inured became the working scientist to their irrelevance to the real world. It is not so much that the thickening mathematical smoke (too much for even Martin Davis to explain away) has obscured the flames of real world incomputability — more that the anomalies, decoherence, and lack of persuasiveness at the margins of a number of the most basic of standard scientific models are very hard to characterise in a precise enough way. It is the nature of the connection which is incomplete. And this is often reflected in a parallel dichotomy between Baconians (including many computer scientists) and Cartesians (most mathematicians and logicians). Paradoxically, some of the most determined guardians of this situation are mathematicians, particularly those whose careers have been built on the study of incomputability. But a wide spectrum of scientists know something is wrong, if only they could explain what.

There are some obvious examples of Baconian confrontation with incomputability (or at least something which looks very like it), and Cartesian interpretations of them. For instance, as we commented in [8]:

“To find a single body of *empirical* evidence which is clearly inconsistent with a narrowly mechanistic Laplacian determinism, one must first look to the quantum level.”

While noting that quantum computation, as currently conceived, “appears to hold few surprises for the classical recursion theorist”, we went on to mention the problem of explaining why the so-called ‘collapse of the wave function’, with its associated probabilities, takes the particular form it does. This predictive incompleteness of quantum theory gives rise to different ‘interpretations’ which leave us a long way from characterising the algorithmic content of the events it seeks to describe. This is how Andrew Hodges sums up the situation (in his article *What would Alan Turing have done after 1954?*, from Teuscher [35]):

“Von Neumann’s axioms distinguished the **U** (unitary evolution) and **R** (reduction) rules of quantum mechanics. Now, quantum computing so far (in the work of Feynman, Deutsch, Shor, etc) is based on the **U** process and so computable. It has not made serious use of the **R** process: the unpredictable element that comes in with reduction, measurement, or collapse of the wave function.”

Above the quantum level, Etesi and Nemeti [15] describe how relativistic considerations (involving the actuality of such things as large rotating black holes in galactic nuclei) may lead to effectively computable functions which are not Turing computable. They have since set out to explain more thoroughly how and why such general relativistic computers work.

At all levels between these physical extremes we find chaotic phenomena and turbulence — difficult to handle computationally, but are superficially less

threatening to standard *models* of computation. One is reassured by the extent to which one understands the underlying local behaviour, and by the overall patterns emerging to constrain what appears to be merely *practical* unpredictability. If one was just a little cleverer at solving differential equations, one assumes, or had a large enough computer, one could get much closer to predicting the details of chaotic contexts.

Kreisel [21] was one of the first to separate *cooperative phenomena* (not known to have Turing computable behaviour), from classical systems and proposed [22] (p143, Note 2) a collision problem related to the 3-body problem as a possible source of incomputability, suggesting that this might result in “an analog computation of a non-recursive function (by repeating collision experiments sufficiently often)”. This was before the huge growth in the attention given to chaos theory, with its multitude of different examples of the generation of informational complexity via very simple rules, accompanied by the emergence of new regularities (see for example the two classic papers of Robert Shaw [33], [32]). We now have a much better understanding of the relationship between emergence and chaos, but this still does not provide the basis for a practically computable relationship. As described in Cooper and Odifreddi [11]:

“As one observes a rushing stream, one is aware that the dynamics of the individual units of flow are well understood. But the relationship between this and the continually evolving forms manifest in the stream’s surface is not just too *complex* to analyse — it seems to depend on globally emerging relationships not derivable from the local analysis. The form of the changing surface of the stream appears to constrain the movements of the molecules of water, while at the same time being traceable back to those same movements.”

Relevant here is the widely recognised link between structures in nature, and mathematical objects, such as the Mandelbrot and Julia sets, which provide a metaphor for the way real-world complexity is generated by the iteration of simple algorithmic rules. Recently, high-profile names (such as Roger Penrose, Steve Smale) have been associated with investigations of the computability of such objects. Penrose (p124) points to the apparent unpredictability of structure in computer generated approximations to the Mandelbrot set as indications of an underlying incomputability:

“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.”

So the extraordinary richness of structure we observe in nature is matched by the as yet unsolved problems of showing that aspects of structures such as the Mandelbrot and certain Julia sets are computable (for recent progress see [20], [31], [1] and [30]).

Of course, just as a turbulent stream is constrained within emergent flow patterns, different scientific disciplines are often associated with successive levels

of emergent physical reality, hierarchically resting one on the other. Again, the relationship can be described, but there is no correspondingly reductive framework to capture it computationally. Just as one cannot develop the theory of fluid dynamics on the basis of quantum mechanics, the life sciences extend over entirely new levels, each with their own distinctive parameters. As we shall see below, the different levels give rise to their own algorithmic content, from which computational paradigms can be extracted. But the higher one goes up the hierarchy, the more controversy there is about exactly how it has developed, and the less clear is the computational content of the links between local mechanisms and emergent global relations. This is Gregory Chaitin's [5] try at extracting incomputability from the complexities of biological evolution (while taking Ω to be the halting probability for a suitably chosen universal computer U):

“We have seen that Ω is about as random, patternless, unpredictable and incomprehensible as possible; the pattern of its bit sequence defies understanding. However with computations in the limit, which is equivalent to having an oracle for the halting problem, Ω seems quite understandable: it becomes a computable sequence. Biological evolution is the nearest thing to an infinite computation in the limit that we will ever see: it is a computation with molecular components that has proceeded for 10^9 years in parallel over the entire surface of the earth. That amount of computing could easily produce a good approximation to Ω , except that that is not the goal of biological evolution. The goal of evolution is survival, for example, keeping viruses such as those that cause AIDS from subverting one's molecular mechanisms for their own purposes.

This suggests to me a very crude evolutionary model based on the game of matching pennies, in which players use computable strategies for predicting their opponent's next play from the previous ones. I don't think it would be too difficult to formulate this more precisely and to show that prediction strategies will tend to increase in program-size complexity with time.

Perhaps biological structures are simple and easy to understand only if one has an oracle for the halting problem.” (italics added)

But the part of nature we are least able to make behave properly, and the part we are most familiar with (but understand least), is the human brain. Baconian experience of it comes first through our everyday experience of solving problems, while feeling nothing like a Turing machine. Such subjective impressions may not be scientific, but they can force themselves on us in a dramatic fashion. And can be the intuitive basis for the most informative of scientific work.

Jacques Hadamard [19] derived seminal observations on the role of intuition in mathematical thinking from this account of how Poincaré struggled unsuccessfully, and then successfully, to solve a problem:

“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.”

A few years earlier, Turing envisaged his technically complex 1939 paper [37] as an attempt to pin down the computable content of such creativity. He claimed to clarify there the relationship between ‘ingenuity’ (subsumed within his ordinal logics) and ‘intuition’ (needed to identify good ordinal notations for levels of the resulting hierarchy). Turing clearly regarded ingenuity as being what a clever Turing program is capable of, and intuition as something else. There was a clear implication that intuition is a feature of human mental processes, and to that extent Turing is certainly saying that his hierarchies have something to say about how the mathematician’s mind transcends his own model of machine computability – even if the results can be subsequently translated into proofs implementable by a Turing machine. This is what Turing ([37], pp.134–5) actually says about the underlying meaning of his paper:

“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.*”

My emphasis is to highlight the extent to which Turing was striving to bring mental processes within something approaching the standard model of computability, and failing.

An important role of such observation and analysis of mental higher functionality is to bring out, by contrast, differences with more obviously mechanical processes. The main problem with this approach is that because it does not really get us to grips with what underlies this higher functionality — that is, the particularities of the process of emergence — it is hard to fit the real world persuasively within any model derived from it. The temptation is to over-speculate and fudge the details, which is what some logicians think Roger Penrose [28] has succumbed to.

At the other end of the spectrum, bottom-up approaches involving trying to build intelligent machines, or developing models based on what we actually do understand about the physical workings of the brain, struggle to reproduce any recognisable or useful higher functionality. As Rodney Brooks [4] puts it “neither AI nor Alife has produced artifacts that could be confused with a living organism for more than an instant.”

But this does not mean that paradigm-stretching features are not strongly in evidence. For instance Smolensky [34, p.3], in his influential *Behavioral and Brain Sciences* paper, goes so far as to say:

“There is a reasonable chance that connectionist models will lead to the development of new somewhat-general-purpose self-programming, massively parallel analog computers, and a new theory of analog parallel computation: they may possibly even challenge the strong construal of Church’s Thesis as the claim that the class of well-defined computations is exhausted by those of Turing machines.”

We may be a long way from artificially performing the sort of mental marvels we observe, but there is plenty of evidence that the new ingredients on which to base a workable new computational discipline are already present.

3 Swimming with the tide

As Boris Kogan, a pioneer developer of the Soviet Union’s first analog and hybrid computers, comments (in an interview with Daniel Abramovitch, on pages 52–62 of the June 2005 issue of the IEEE Control Systems Magazine):

“Some of the great physical systems to be studied as objects of control are the dynamic processes in the living organisms, especially under pathological conditions.”

In the face of the sheer complexity of natural computational processes, one can take the Baconian outlook one step further, and allow Nature to take over the driving seat. This kind of abrogation of executive control can be quite fruitful. In April, 2001, Daniel Hillis, Chief Technology Officer of Applied Minds, Inc. (and ex-Vice President, Research and Development at Walt Disney Imagineering), was quoted as saying this about his experiences trying to make intelligent machines:

“I used to think we’d do it by engineering. Now I believe we’ll evolve them. We’re likely to make thinking machines before we understand how the mind works, which is kind of backwards.”

It is certainly true that the closest anyone has got so far to actual computers with recognisably hypercomputational ingredients is by surfing physical reality in some way. This is consistent with our Baconian suspicion that the world cannot be satisfactorily located within the standard computational model. Of course, it is not necessary for one to have any interest in hypercomputation for one to have an interest in new computational paradigms based on nature. As in the case of quantum computation, there may be very important operational benefits, even though there is an underlying classical model. But the above suspicion does get stronger the more difficult it is to divorce ones computational approach from its real-world origins.

The way forward adopted very widely now (as remarked in [10]), is to utilise the physical world’s rich potential for computation, without worrying too much about understanding the underlying rules of the game. The likely success of this approach may be limited — it takes ingenuity to get a natural process to compute more than itself — but may bring practically useful results and be the best we can do in the short to medium term. Here is the analogy suggested in [10]:

“The domestication of horses around five or six thousand years ago brought a revolution in transportation, only achieved through a creative interaction between humans and the natural world. At that time, trying to understand the principles underlying the equine organism in order to synthesise an artificial horse was unthinkable. But a few thousand years later there was enough understanding of scientific basics to underpin the invention of the ‘iron horse’, leading, amongst other things, to the opening up of many previously isolated parts of the world to people with no riding skills whatsoever.”

While Cartesian theorising may deliver computation with consciousness, wonderful things can still be achieved without consciousness. We would probably still have had present day computers even if Turing had not invented the universal Turing machine when he did. In our introduction to the CiE 2005 LNCS Proceedings volume, we referred to how Bert Hölldobler and Edward O. Wilson’s book on *The Ants* runs to over eight-hundred pages, and mentioned how ants and similar biological examples have inspired new problem-solving strategies based on ‘swarm intelligence’. But how the limits to what a real-life ant colony can achieve are very apparent, more so than those of recognisably conscious beings. For instance, as the constructors move in and tarmac over our richly structured ant colony, the ants have no hope of expanding their expertise to deal with such eventualities. In contrast, for us algorithmic content gives rise to new emergent forms, which themselves become victim to our algorithmic appetites, and even the inevitable limits on this inductive process we hope to decode. There is an important role for conscious and interventionist observation of our more ant-like everyday computational activities. It may well be that particular computational models expressing metaphors for natural processes, such as quantum and molecular computing, membrane computing, neural networks, evolutionary computation, relativistic computing, or evolving real-world models like grids and the internet, are currently the most exciting and practical examples of new computational paradigms. But we have to keep in mind the Holy Grail of synthesising and controlling in a conscious way that higher functionality which we observe in Nature but not in computers based on the standard Turing model. Conversely, we will never achieve this without engaging with the real world. To quote Rodney Brooks [2, p.139] again:

“I, and others, believe that human level intelligence is too complex and little understood to be correctly decomposed into the right subpieces at the moment and that even if we knew the subpieces we still wouldn’t know the right interfaces between them. Furthermore, we will never understand how to decompose human level intelligence until we’ve had a lot of practice with simpler level intelligences.”

In regard to connectionist models of computation based on the workings of the human brain — these have come a long way since Turing’s [38] discussion of ‘unorganised machines’, and McCulloch and Pitts’ early paper [24] on neural nets. But (quoting from [10]) “despite the growth of computational neuroscience

as an active research area, putting together ingredients from both artificial neural networks and neurophysiology, something does seem to be missing”. For Steven Pinker “. . . neural networks alone cannot do the job”. And focussing again on that elusive higher functionality, he describes [29, p.124] “a kind of mental fecundity called recursion”:

“We humans can take an entire proposition and give it a role in some larger proposition. Then we can take the larger proposition and embed it in a still-larger one. Not only did the baby eat the slug, but the father saw the baby eat the slug, and I wonder whether the father saw the baby eat the slug, the father knows that I wonder whether he saw the baby eat the slug, and I can guess that the father knows that I wonder whether he saw the baby eat the slug, and so on.”

So while there does seem to be a great deal to be got from an ad hoc computational relationship with the real world, we should not be daunted by the sheer wonder natural structures inspire in us. It may be that the human brain, as an emergent phenomenon, has an intimate relationship with processes which are not easily simulable over significantly shorter time-scales than those to which natural evolution is subject. Maybe we will never build an artificial brain, anymore than we can make an artificial horse. But this does not mean we may not one day have a good enough understanding of basic hypercomputational principles to build computers — or firstly non-classical mathematical models of computation — which do things undreamt of today.

4 The constructive approach to computational barriers: In defense of the black-box model of computation

As Robin Gandy [16] points out in his article The confluence of ideas in 1936, Alan Turing did not set out in his 1936 paper to give a mathematical model of machine computation. That is not even a well-defined objective — machines as a part of nature require much more radical analysis. Odifreddi [26] (reporting on his discussions with Georg Kreisel, see pp. 101–123) sets out some of the underlying difficulties. What Turing had in mind was a model of how *humans* compute in a very specific manner:

‘The real question at issue is “What are the possible processes which can be carried out in computing a [real] number?” ’

What is different, and theoretically liberating, about Turing’s approach to characterising what a computable function is is his avoidance of the teleological constraints on how computation is viewed. Ultimate aims are put to one side in the interests of seeing and modelling atomic detail. Applying this approach to modelling how Nature computes is much more difficult, and certainly more Cartesian.

The first thing that strikes one about physical processes is their basis in the often illusive dichotomy between matter and energy. This has a parallel in various mathematical frameworks in which data and process are put on an equal footing,

with the distinction only re-emerging at the semantical level. To an extent, it is the way the universe is observed which leads to the observer seeing process or information. What science tells us is that energy in nature tends to express algorithmic content, implementable over a wide range of appropriate physical contexts which we seek to encapsulate in corresponding informational content. In fact, it appears that nothing interesting exists without this dichotomy, and this is bad news for those looking for the most reductive of foundational explanatory frameworks. At the same time, this observation gives an important role, corresponding to what we experience in the physical context, for algorithmic content — it provides the glue whereby local information content comes together to form a global entity which is more than the sum of its parts, and which is the source of information content qualitatively different from that of its origins. The classical counterpart of this picture is the so-called Turing universe, giving a framework based on oracle computation for mathematically analysing the computationally complex in terms of its algorithmic content. An important aspect of this way of structuring the Universe in accordance with the observed energy-matter dichotomy is the way in which simple global concepts (like definability and invariance) lead to explicit and structurally integrated counterparts to natural laws and large-scale formations whose origins were previously quite mysterious. As we argued in [7]:

“If one abstracts from the Universe its information content, structured via the basic ... fundamental laws of nature, one obtains a particular ... manifestation of the Turing universe ... , within which vague questions attain a precise analogue of quite plausible validity.”

This is useful not just in an explanatory role, but as a pointer to how we might achieve that control of higher-order computational structure that we observe in human thinking. The key ingredient here is just that local to global transfer and elevation of information content, based on quite elementary local interactive infrastructure. Here is how Antonio Damasio [12, p.169] describes the hierarchical development of a particular instance of consciousness within the brain (or, rather, ‘organism’), interacting with some external object:

“... both organism and object are mapped as neural patterns, in first-order maps; all of these neural patterns can become images. ... The sensorimotor maps pertaining to the object cause changes in the maps pertaining to the organism. ... [These] changes ... can be re-represented in yet other maps (second-order maps) which thus represent the relationship of object and organism. ... The neural patterns transiently formed in second-order maps can become mental images, no less so than the neural patterns in first-order maps.”

As we commented in [10]:

“Notice that what is envisaged is the re-representation of neural patterns formed across some region of the brain, in such a way that they can have a computational relevance in forming new patterns. This is where the clear demarcation between computation and computational effect becomes

blurred. The key conception is of computational loops incorporating these ‘second-order’ aspects of the computation itself. Building on this one can derive a plausible schematic picture of the global workings of the brain.”

How one synthesises in practice the sort of representational mechanisms integral to intelligent thought is a problem which goes far beyond any schematic picture of the underlying structures, but these structures give a reassuring solidity to our attempts. The sort of current developments which are brought to mind are the sort of large interactive structures such as the internet and large computing grids. One already observes global phenomena emerging in such contexts, initially as problems, such as those which threaten economic planning, but potentially with computational outcomes which are more ‘new paradigm’ than generally expected. Robin Milner commented in his 1991 Turing Award lecture [25, p.80] that:

“Through the seventies, I became convinced that a theory of concurrency and interaction requires a new conceptual framework, not just a refinement of what we find natural for sequential computing.”

Such observations have been taken up by Goldin and Wegner [18] in support of new thinking concerning models of today’s highly interactive non-linear computation. This takes us beyond thinking of intelligence as something that resides purely within the autonomous brain. As Brooks [3] points out:

“Real computational systems are not rational agents that take inputs, compute logically, and produce outputs . . . It is hard to draw the line at what is intelligence and what is environmental interaction. In a sense, it does not really matter which is which, as all intelligent systems must be situated in some world or other if they are to be useful entities.”

Particularly relevant to future computing capabilities is Brooks’ [2, p.139] argument that there is a realistic approach to AI involving no internally generated representations, but rather using “the world as its own model”. Which brings us back to Danny Hillis’ idea that “we’ll evolve” intelligent machines rather than “do it by engineering”. A Baconian enterprise, no doubt, but one in which we should be prepared for Cartesian surprises.

References

1. M. Braverman, Hyperbolic Julia Sets are Poly-Time Computable. In *Proceedings of the 6th Workshop on Computability and Complexity in Analysis, Wittenberg, Germany, August 16–20, 2004* (Brattka V., Staiger L. and Weihrauch K., eds.), ENTCS v.120, Elsevier, Amsterdam, 2005, pp.17–30.
2. R. A. Brooks, Intelligence Without Representations, *Artificial Intelligence*, **47** (1987), 139–159.
3. R. A. Brooks, *Intelligence Without Reason*, A.I. Memo No. 1293, The A.I. Laboratory, M.I.T., Cambridge, MA, April 1991.
4. R. A. Brooks, The relationship between matter and life, *Nature* **409** (2001), 409–411.

5. G. J. Chaitin, *Algorithmic Information Theory*, Cambridge University Press, Cambridge, New York, 1987.
6. A. Church, A note on the Entscheidungsproblem *J. Symbolic Logic*, **1** (1936), 40–41 and 101–102.
7. S. B. Cooper, Beyond Gödel’s Theorem: Turing nonrigidity revisited. In *Logic Colloquium ’95* (eds. Makowsky, J. A. and Ravve, E. V.), Proceedings of the Annual European Summer Meeting of the Association of Symbolic Logic, Haifa, Israel, August 9–18, 1995, Lecture Notes in Logic, vol. 11, Springer, Berlin, New York, Tokyo, 1998, pp. 44–50.
8. S. B. Cooper, Clockwork or Turing U/universe? – remarks on causal determinism and computability. In *Models and Computability* (S. B. Cooper and J. K. Truss, eds.), London Mathematical Society Lecture Note Series 259, Cambridge University Press, Cambridge, 1999, pp. 63–116.
9. S. B. Cooper, The incomputable Alan Turing. In *Turing 2004: A celebration of his life and achievements*, electronically published by the British Computer Society, 2005.
10. S. B. Cooper, Definability as hypercomputational effect, *Applied Mathematics and Computation*, to appear.
11. S. B. Cooper and P. Odifreddi Incomputability in Nature. In *Computability and Models: Perspectives East and West* (S. B. Cooper and S. S. Goncharov, eds.), Kluwer Academic/ Plenum Publishers, New York, Boston, Dordrecht, London, Moscow, 2003, pp. 137–160.
12. A. Damasio, *The Feeling Of What Happens*, Harcourt, Orlando, FL, 1999.
13. M. Davis, The myth of hypercomputation. In Teuscher, 2004, pp. 195–211.
14. A. Einstein, *Out of My Later Years*, Philosophical Library, New York, 1950.
15. G. Etesi and I. Németi, Non-Turing computations via Malament-Hogarth spacetimes, *Int. J. Theoretical Phys.* **41** (2002), 341–370.
16. R. O. Gandy, The confluence of ideas in 1936. In *The Universal Turing Machine: A Half-Century Survey* (Herken R., ed.), Oxford University Press, New York, 1988, pp. 51–102.
17. Gandy R. O. and Yates C. E. M. eds. (2001) *Collected Works of A.M. Turing: Mathematical Logic*. North-Holland, Amsterdam, New York, Oxford.
18. D. Goldin and P. Wegner, Computation Beyond Turing Machines: seeking appropriate methods to model computing and human thought, *Communications of the ACM* **46** (2003), 100–102.
19. J. Hadamard, *The Psychology of Invention in the Mathematical Field*, Princeton Univ. Press, Princeton, 1945.
20. P. Hertling, Is the Mandelbrot set computable? *Math. Logic Quarterly* **51** (2005), 5–18.
21. G. Kreisel, Mathematical logic: What has it done for the philosophy of mathematics? In *Bertrand Russell, Philosopher of the Century* (R. Schoenman, ed.), Allen and Unwin, London, 1967, pp. 201–272.
22. G. Kreisel, Church’s Thesis: a kind of reducibility axiom for constructive mathematics. In *Intuitionism and proof theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968* (A. Kino, J. Myhill and R. E. Vesley, eds.) North-Holland, Amsterdam, London, 1970, pp. 121–150.
23. P. S. de Laplace [1819], *Essai philosophique sur les probabilités*, English trans. by F. W. Truscott and F. L. Emory, Dover, New York, 1951.
24. W. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5**, 115–133.

25. R. Milner, Elements of interaction:Turing award lecture, *Communications of the ACM* **36** (1993), 78–89.
26. P. Odifreddi, *Classical Recursion Theory*, North-Holland, Amsterdam, New York, Oxford, Tokyo, 1989.
27. P. Odifreddi, *The Mathematical Century : The 30 Greatest Problems of the Last 100 Years* (trans. A. Sangalli), Princeton University Press, Princeton, 2004.
28. R. Penrose, *Shadows of the Mind: A Search for the Missing Science of Consciousness*. Oxford University Press, Oxford, 1994.
29. S. Pinker, *How the Mind Works*. W. W. Norton, New York, 1997.
30. R. Rettinger, A fast algorithm for Julia sets of hyperbolic rational functions. In *Proceedings of the 6th Workshop on Computability and Complexity in Analysis, Wittenberg, Germany, August 16–20, 2004* (Brattka V., Staiger L. and Weihrauch K., eds.), ENTCS v.120, Elsevier, Amsterdam, 2005, pp.145–157.
31. R. Rettinger and K. Weihrauch, The computational complexity of some Julia sets. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 9–11, 2003* (Goemans M. X., ed.), ACM Press, New York, 2003, pp.177–185.
32. R. Shaw, Strange attractors, chaotic behaviour, and information flow, *Z. Naturforsch.* **36A** (1981), 80–112.
33. R. Shaw, *The dripping faucet as a model chaotic system*, The Science Frontier Express Series, Aerial Press, Santa Cruz, CA, 1984.
34. P. Smolensky, On the proper treatment of connectionism, *Behavioral and Brain Sciences* **11** (1988), 1–74.
35. C. Teuscher (ed.), *Alan Turing: Life and legacy of a great thinker*, Springer-Verlag, Berlin, Heidelberg, 2004.
36. A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proc. London Math. Soc.* (2) **42** (1936–7), pp. 230–265. Reprinted in A. M. Turing, *Collected Works: Mathematical Logic*, pp. 18–53.
37. A. M. Turing, Systems of logic based on ordinals, *Proc. London Math. Soc.* (2) **45** (1939), pp. 161–228. Reprinted in A. M. Turing, *Collected Works: Mathematical Logic*, pp. 81–148.
38. A. M. Turing, *Intelligent machinery*, National Physical Laboratory Report, 1948. In *Machine Intelligence 5* (B. Meltzer and D. Michie, eds.), Edinburgh University Press, Edinburgh, 1969, pp. 3–23. Reprinted in A. M. Turing, *Collected Works: Mechanical Intelligence* (D. C. Ince, ed.), North-Holland, Amsterdam, New York, Oxford, Tokyo, 1992.
39. J. van Leeuwen and J. Wiedermann, The Turing Machine Paradigm in Contemporary Computing. In *Mathematics Unlimited — 2001 and Beyond* (Enquist B. and Schmidt W., eds.), Lecture Notes in Computer Science, Springer-Verlag, 2000.