Non-Turing Computers and Non-Turing Computability[1]

Mark Hogarth

University of Cambridge

1. Introduction

Building on an idea by Pitowsky (1990), David Malament (private communications), Hogarth (1992) and Earman and Norton (1993) have shown how it is possible to perform computational supertasks—that is, an infinite number of computational steps in a finite span of time—in a kind of relativistic spacetime that Earman and Norton (1993) have dubbed a *Malament-Hogarth* spacetime[2].

Definition 1 A spacetime (M,g) is *Malament-Hogarth* just when there is a future endless curve $\lambda \subset M$ with past endpoint and a point $q \in M$ such that $\int_\lambda ds^2 = \infty$ and $\lambda \subset J^-(q)$.

(Hereafter, the symbols "q" and "λ" are assumed to have the properties they have in Definition 1. I shall also speak of a "λ-curve".)

Various examples of Malament-Hogarth (hereafter, *M-H*) spacetimes are given in Hogarth (1992), but the following artificial example from Earman and Norton (1993) is perhaps the simplest. Start with Minkowski spacetime (R^4, η) and choose a scalar field Ω on M such that $\Omega=1$ outside a compact set $C \subset M$ and Ω tends rapidly to infinity as a point $r \in C$ is approached. The spacetime $(R^4\text{-}r, \Omega^2\eta)$, depicted in Figure 1, is then M-H. (Although the region inside C *appears* quite small, it is in fact as large as the complement of C.)

Hogarth (1992) and Earman and Norton (1993) show how in a M-H spacetime, e.g., $(R^4\text{-}r, \Omega^2\eta)$, one might solve, e.g., the Goldbach conjecture. From a point $p \in \lambda$ launch a Turing machine along λ that is primed to first check if 2 is the sum of two primes, then likewise to check 4, then 6, and so on, *ad infinitum*. The Turing machine is also primed to signal to q if and only if it finds a counter-example to the conjecture, and then to halt operations. Since an observer, e.g. O in Figure 1, can travel from p to q in a finite span of proper time, she can discover the truth of the conjecture before her day's out. Fermat's last theorem is cracked in a similar fashion.

PSA 1994, Volume i, pp. 126-138

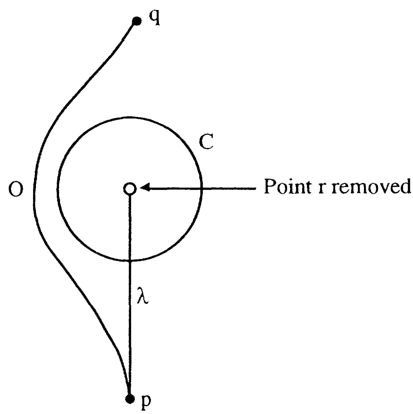

Figure 1. A toy Malament-Hogarth spacetime.

2. Solving the Turing Unsolvable

I move now to the objectives stated in the abstract. To simplify matters in this section, no account will be taken of the physical plausibility of the spacetimes under consideration or of the behaviour of the matter fields they support. Moreover, it will be assumed that every spacetime permits Turing machines of any size to operate unproblematically and that communication to future events is always possible. Note, however, that Hogarth (1992) has shown that M-H spacetimes cannot be globally hyperbolic (so they violate strong cosmic censorship), and that Earman and Norton (1993) have shown that in many M-H spacetimes photons travelling between some events suffer infinite blue shifts (which may indicate horizon instability).

A problem is said to be *Turing solvable* if there is a Turing machine (operating according to a finite instruction set, but having access to an infinite memory store) that can solve the problem after a finite number of steps. In this paper I shall say, somewhat informally, that a problem is *solvable in a spacetime (M,g)* if there is an observer $O \subset M$ who can initiate a procedure which is comprised of only Turing machines and ordinary communication devices and which will deliver the problem's solution to O after a finite span of O's proper time.

Thus the Goldbach conjecture and Fermat's last theorem are both solvable in the M-H spacetime in Figure 1, and indeed in any M-H spacetime. These two problems are Turing solvable (because in both cases the solution can be written into the finite program of a TM), so they do not prove a difference between this more general solvability and Turing solvability. I will now give a proof, by showing how a problem that is known to be Turing unsolvable is demonstrably solvable in any M-H spacetime. In fact I will give two examples: (see Boolos and Jeffery 1989.)

(1) *The halting problem.* This is the problem of deciding if an arbitrarily given Turing machine, TM, will or will not eventually halt. Working in a M-H spacetime (M,g), adopt the following procedure. Launch TM along $\lambda \subset M$, having first primed TM to signal to $q \in M$ if and only if TM halts. The question will then be settled at q.

(2) *The decision problem for first-order logic.* This is the problem of deciding the validity or invalidity of an arbitrary sentence of first-order logic. First recall that

there is Turing machine, TM, that will halt after a finite number of steps if and only if a given sentence S of first-order logic is valid (ibid., p. 142). So, working in a M-H spacetime (M,g), launch TM along $\lambda \subset M$, having first primed TM to signal to $q \in M$ if and only if TM halts. Upshot: a signal at q means the sentence is valid, no signal at q means the sentence is invalid.

In general, the *decision problem* for a property P is Turing solvable if there is both a Turing machine TM that will halt after a finite number of steps if and only P holds and a Turing machine TM´ that will halt after a finite number of steps if and only if P does not hold. If only one (or both) of the pair TM, TM´ exists, then the decision problem for P is said to be *partially Turing solvable*. Problems (1) and (2) above are clearly of this kind. It is now evident that:

Result 1. Any decision problem that is partially Turing solvable is solvable in a M-H spacetime.

Attention is now turned to the *decision problem for arithmetic in the standard model*. In what follows, the word "sentence" is used as shorthand for "sentence in the language of arithmetic". The results below are standard (ibid.).

(i) Deciding (i.e. establishing whether true or false) arbitrary sentences in arithmetic is not partially Turing solvable. (This is a version of Gödel's first incompleteness theorem.)

(ii) Deciding arbitrary quantifier-free sentences is Turing solvable.

(iii) There is a Turing machine that will, in a finite number of steps, translate an arbitrary sentence S into a coextensive (i.e. a sentence with the same truth-value) sentence, S´, in prenex form (all quantifiers occurring at the extreme left).

(iv) There is a Turing machine that will, in a finite number of steps, translate an arbitrary sentence S (written in prenex form) containing two juxtaposed quantifiers of the same type into a coextensive sentence, S´, with one quantifier of that type in place of the previous two.

(v) If $\forall nS(n)$ is a sentence or $\exists nS(n)$ is a sentence, then the set of sentences $\{S(1), S(2), S(3),\ldots\}$ is recursively enumerable (that is, there is a Turing machine that will generate S(n) for any given n).

Because of (iii), it may be assumed that all sentences are in prenex form.

Because of (ii) and (v), it is clear that deciding arbitrary sentences of either the form $\exists nS(n)$ or $\forall nS(n)$, where S is quantifier-free, is partially Turing solvable. So by Result 1, they are both decidable in any M-H spacetime. This fact, together with (iv) above, implies that arbitrary purely existential or purely universal sentences in arithmetic are decidable in any M-H spacetime. (Incidentally, the Goldbach conjecture and Fermat's last theorem can both be stated as purely universal sentences.)

But because of (i), Result 1 cannot be used to show that arithmetic is decidable in any M-H spacetime. Indeed, although it cannot be "proved" that arithmetic is undecidable in, e.g., the M-H spacetime in Figure 1, there is no obvious way to construct a procedure in this particular spacetime that will decide even an arbitrary $\forall\exists$ type sentence. A more subtle kind of M-H spacetime is required, as I will now show. The following analysis contains a new piece of terminology. If a spacetime (M,g) contains non-intersecting open regions O_i, i=1,2,... such that (1) for all i $O_i \subset I^-(O_{i+1})$ and (2) there is point $q \in M$ such that for all i $O_i \subset I^-(q)$, then the O_is are said to form a *past temporal string*, or just *string* for short. See Figure 2 (i).

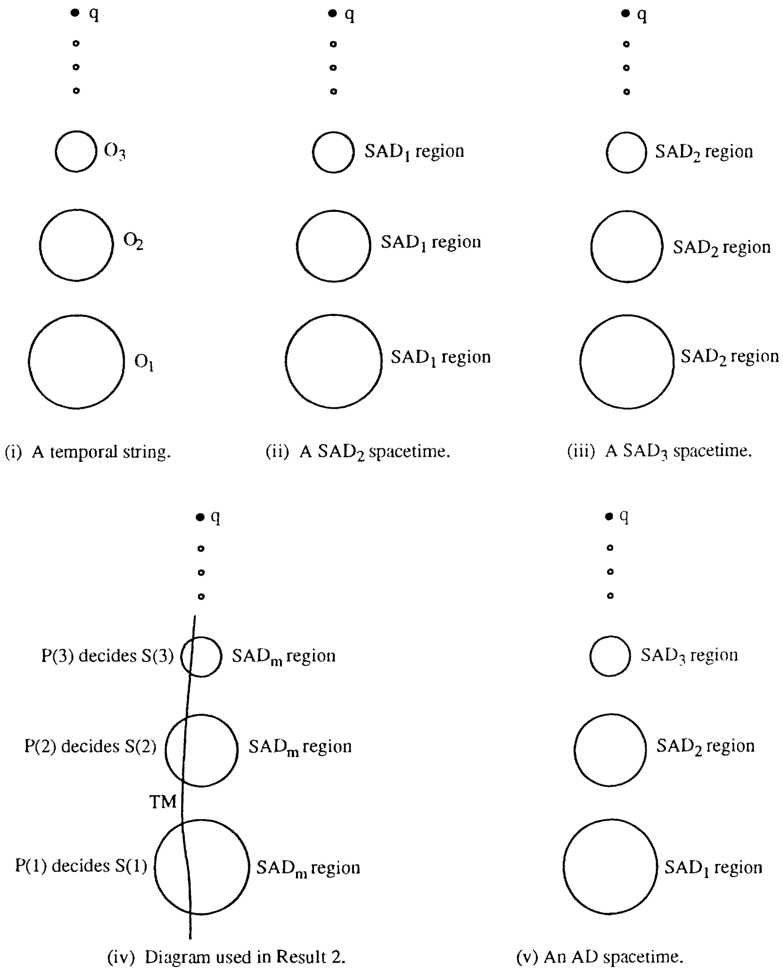

Figure 2

Definition 2. A spacetime (M,g) is an *nth-order arithmetical sentence deciding* (denoted SAD_n) spacetime if the n conditions contained in the following scheme are satisfied.

If n=1, (M,g) is a M-H spacetime.
If n>1, (M,g) admits a string of SAD_{n-1} spacetimes.

According to Definition 2, a SAD_1 spacetime is a M-H spacetime, a SAD_2 spacetime is a spacetime that contains a string of SAD_1 spacetimes (Figure 2 (ii)), a SAD_3 spacetime is a spacetime that contains a string of SAD_2 spacetimes (Figure 2 (iii)), and so on.

The efficacy of SAD spacetimes to decide sentences in arithmetic derives from the following result.

Result 2. Let (M,g) be a SAD_{m+1} spacetime and let $\exists nS(n)$ and $\forall nS(n)$ be two sentences in arithmetic. Suppose that for each $n \geq 1$, S(n) is decidable in any SAD_m spacetime.

Then $\exists nS(n)$ and $\forall nS(n)$ are both decidable in (M,g).

Proof. This consists of showing how appropriately chosen and appropriately located hardware can be used to decide $\exists nS(n)$ and $\forall nS(n)$. Part of this involves the Turing machine, TM, of (v) above travelling along a λ-curve that picks up one hour of proper time in each SAD_m component (this is possible because every such component admits a λ-curve), as depicted in Figure 2 (iv). TM is primed to generate S(1) in the lead up to the first component and to signal that sentence to the first component. TM is also primed to generate, for n>1, S(n) in the (n-1)th component and to signal that sentence to the nth component.

Now let P(n) denote the procedure that decides S(n). The procedure for deciding $\exists nS(n)$ consists of adding to each P(n) transmitting devices and receivers which operate as follows.

P(1) signals to q and P(2) if and only if S(1) is true.
For n>1, P(n) signals to P(n+1) if and only if P(n-1) signals to P(n).
For n>1, P(n) signals to q and P(n+1) if and only if S(n) is true and P(n) has not received a signal from P(n-1).

This procedure ensures that a *single* signal is sent to q if and only if there is an n such that S(n) is true. (The signal is actually sent by P(m), where m is the smallest integer for which S(m) is true.) Upshot: a signal at q means $\exists nS(n)$ is true, no signal at q means $\exists nS(n)$ is false.

The procedure for deciding $\forall nS(n)$, given below, is similar except this time a single signal at q means $\forall nS(n)$ is false, no signal at q means $\forall nS(n)$ is true.

P(1) signals to q and P(2) if and only if $\neg S(1)$ holds.
For n>1, P(n) signals to P(n+1) if and only if P(n-1) signals to P(n).
For n>1, P(n) signals to q and P(n+1) if and only if $\neg S(n)$ holds and P(n) has not received a signal from P(n-1).

Thus $\exists nS(n)$ and $\forall nS(n)$ are seen to be decidable in (M,g). QED

We have seen already how single quantifier sentences can be decided in SAD_1 (=M-H) spacetimes. By Result 2, double quantifier sentences can be decided in SAD_2 spacetimes. Applying Result 2 again shows that triple quantifier sentences can be decided in SAD_3 spacetimes. And continuing in this way we see that n-tuple quantifier sentences can be decided in SAD_n spacetimes.

In fact, these different order SAD spacetimes can be fitted into a single spacetime.

Definition 3. A spacetime (M,g) is an *arithmetic deciding* (AD) spacetime just when (M,g) admits a string of open regions O_1, O_2, O_3,.. such that for each n≥1, (O_n,g) is a SAD_n spacetime.

In Figure 2(v), an observer can decide an arbitrary sentence S by communicating it to the SAD_n region that decides sentences of that order. Arithmetic is therefore decidable in an AD spacetime.

It is natural to wonder how the computing hardware necessary to decide all of arithmetic gets installed in the AD spacetime. What follows is a *prima facie* reasonable method of performing that task.

The various paths of the hardware through spacetime are represented by worldlines, and the idea of the method is to begin with one worldline at the initial event p in Figure 3 and to have a process of worldline branching that results in each SAD_1 component being populated by a λ-curve and every other component of every string being populated by at least one worldline. (Recall that the SAD_1 spacetimes accommodate Turing machines travelling on λ–curves, while all the other components accommodate communicating devices à la Result 2.)

(1) A worldline that meets a SAD_n component, n>1, must bifurcate, with one worldline branch extending to the next component of that string and the other branch entering the component and extending to the first component of the string on the "next level down".

(2) Two kinds of worldline must follow the first available λ-curve: the one that enters the first SAD_1 component and any one that is constrained by (1) to enter a SAD_1 component.

Figure 3 illustrates the process at work on the SAD_1 and SAD_2 stages of the AD spacetime. The tree-like property is an obvious attraction, but this system also guards against one potential disaster, namely, that an infinite amount of hardware mass might be forced to reside in a compact set. I omit the formal proof of why this cannot happen. It relies on the fact that if a region R contains worldlines whose lengths sum to infinity, then it must contain at least one future endless curve. But according to Proposition 6.4.7 in Hawking and Ellis (1973), this can only occur if R is non-compact or violates strong causality. Part of the reason why I chose from the outset to consider only strongly causal spacetimes was to ensure that in this case R must be non-compact.

Admittedly, there remains the worry that an unbounded mass might reside in a region which is non-compact but of finite volume. I am not sure whether or not this can happen. (Are there any SAD_1 spacetimes with finite volume?) In any case, the example in the next section suffers no such pathology.

We have also shown that the hardware for the AD spacetime can be built up using a finite set of instructions. Roughly: start at an event p, then manoeuvre and bifurcate according to (1) & (2), while creating communication devices and the Turing machines of (iv), and having them operate according to Result 2. One other instruction states that the instructions of the previous sentence must be issued to each new piece of hardware.

A word about terminology. In what follows, a *SAD spacetime* will be used as a generic term to cover SAD_n spacetimes of all orders and AD spacetimes. (Of course in this case SAD=SAD_1, but the term "SAD_1" sounds rather specific.) Also, a *simple* SAD_n spacetime is a SAD_n spacetime that is not SAD_{n+1}. Finally, a SAD_n (respectively, SAD, AD, etc.) *computer* will refer to a computing device whose underlying spacetime is SAD_n (respectively, SAD, AD, etc.).

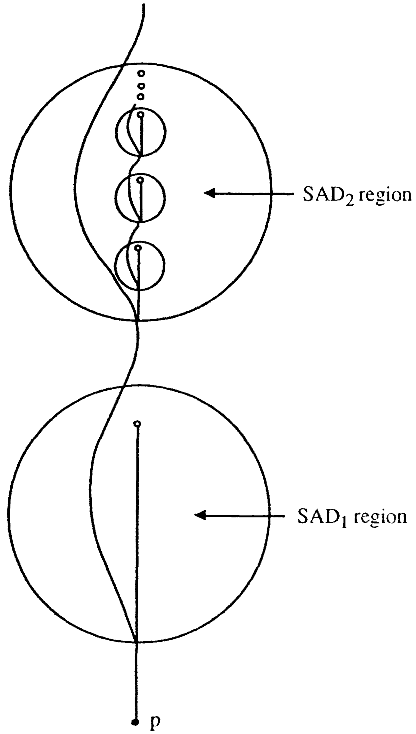

Figure 3. The "hardware tree" grows up into the AD spacetime.

3. An Example of an AD Spacetime[3]

Start with Minkowski spacetime (R^4, η) and choose a compact set $C \subset R^4$. Now draw a closed inertial line segment $v \subset C$. About this v, define regions O_1, O_2, O_3,... with inclusion relations appropriate for all the strings of an AD spacetime, in such a way that v intersects every component, as depicted in Figure 4. Then choose a scalar field Ω on M such that Ω=1 outside C and Ω tends rapidly to infinity as the line v is approached. Remove v. Then $(R^4\text{-}v, \Omega^2\eta)$ is an AD spacetime because O_1 is a SAD_1 spacetime, O_2 is a SAD_2 spacetime, and so on. Moreover, it not difficult to show that (despite appearances!) every component of every string has infinite volume.

Although the corresponding AD computer consists of an infinite number of infinitely large regions, each with its own communication devices and Turing machines, it can still be contained in a box, e.g. the one depicted in Figure 4, with *finite* spatio-temporal surface area. So in this regard this AD computer is no different to an ordinary desktop computer.

I do not yet have a proof, but I think that anti-de Sitter spacetime (see Hawking and Ellis 1973, p.132) is probably an AD spacetime. This spacetime is usually not viewed as a reasonable solution of the Einstein equations, but the fact that its structure is so simple hints, perhaps, that reasonable AD solutions may indeed exist.

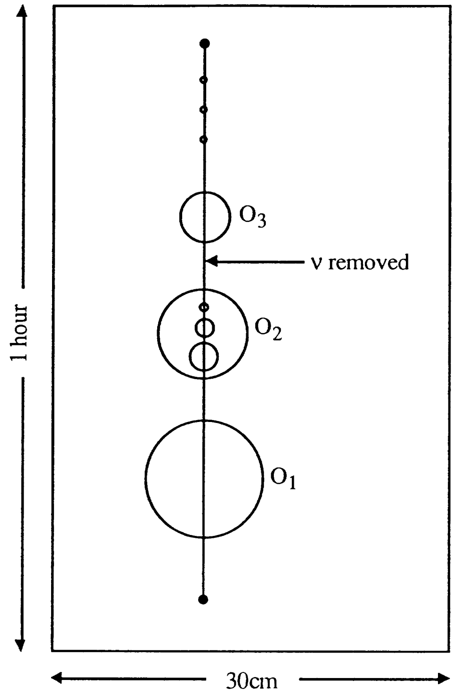

Figure 4. An AD spacetime.

4. The Impact on Church's Thesis

A typical way of stating Church's thesis (CT) is the following: In an ideal world the limit of computation is exactly captured by Turing computability. As it stands, this statement of CT is vague: it has no truth-value since we are not told which set of worlds, W, the "ideal" worlds are chosen from. Different choices of W give different truth-values, but I want to choose a W which gives CT a real fighting chance of success. (In that way, the doubt I manage to cast on CT will not be seen as a hollow victory.) This means W should not be the set of logically possible worlds; for the "Zeus machine" in Boolos and Jeffery (1989, p. 14) is perfectly consistent and is able to perform Turing unsolvable tasks. Nor, on the other hand, should W be a set of worlds all of which are very similar to our universe in such matters as possessing only a finite amount of material (as our universe might) or being temporally finite (as our universe might be). For that would again make CT fail—this time because the arbitrarily massive Turing machines entailed by Turing computability would be impossible, leaving the ideal computing limit somewhere short of Turing computability.

Thus the set W should be neither very large (all logically possible worlds) nor very small (worlds very similar to our own). A middle way is needed. One's initial reaction might be to take W as the set of worlds that are "beefed up" versions of our world, i.e.

worlds based on our world but with enough added space, time and material to allow the realisation of a Turing machine. But of course this is naive, for spacetimes like the Robertson-Walker k=+1 (big bang, big crunch) model, which is a good candidate for the cosmological structure of our universe, cannot be spatio-temporally extended in any natural way and cannot be plied with an unbounded amount of material. Another and more promising proposal is that W should be the set of physically possible worlds, i.e. worlds that share our universe's fundamental laws of physics, but not necessarily our universe's boundary conditions. Of course given our ignorance of what these laws are, this proposal provides at best a working characterisation of W. Nevertheless, this somewhat ill-defined W results in a version of CT that is physically relevant, is not refuted by the hypothetical Zeus machine, and does not fail on account of material limitation—assuming of course that material limitation is not a requirement of physical law.

Moreover, this CT stands firm against two putative counter-examples that have appeared in the literature (Grünbaum 1967, Earman and Norton 1993). Both are machines that can perform an infinite number of computations in a finite time. In one version the machine's components are accelerated to infinite speed in a finite time; in the other, the components shrink continuously to nothing in a finite time. But neither machine seems to cut any ice with CT, for whatever the laws of physics in fact are, the requirements of speed and shrinkage presumably violate them.

But is this CT true? Well, it will be if there is at least one physically possible Turing machine *and* if there are no physically possible non-Turing computers (=SAD computers in this context). The first part of the condition is almost certainly true because there are some thoroughly reasonable solutions of the Einstein equations (e.g. the Robertson-Walker k=0 big bang model) which have both spacetime enough and matter enough to realise a Turing machine. What about the second part of the condition, that all SAD computers are physically impossible? This seems *prima facie* false because there are known simple SAD_1 solutions of the Einstein equations with spacetime enough for the SAD_1 hardware. The Reissner-Nordström black hole solution (see Hogarth 1992) is a case in point. Admittedly this is a vacuum solution, but its existence suggests that simple SAD_1 is a not uncommon property among reasonable models, including ones with infinite material. Against this however is the fact that all SAD spacetimes violate various versions of cosmic censorship. That is to say, physicists have conceived of laws that would effectively outlaw SAD spacetimes and *a fortiori* SAD computers. But of course it is not yet known whether *any* kind of censorship laws really exist, still less whether one exists that will outlaw SAD computers and save CT. Far be it from me to predict the outcome of the debate on the cosmic censorship hypothesis, but I will note that, in concluding his survey of the subject, Earman (1994) remarks that "it is much too soon to pronounce the cosmic censorship hypothesis dead, but the prognosis is not particularly cheerful". Perhaps then the same could be said of CT.

At this juncture, advocates of CT might try to advance their case from a quite different angle, by appealing to the fact that several apparently independent explications of the concept of computability—by Markov, Church, Post, Kleene etc.—have all been found to be exactly equivalent to Turing computability. Of course, I concede this equivalence. But as Horowitz (1992) has rightly stressed, all these explications rest on a shared assumption: that it is impossible to perform a computational super-task. Thus the very assumption that threatens Turing computability, also threatens these other explications. Any attempt, therefore, to base a case for CT on this equivalence is futile: if one explication falls, they all fall.

To summarise, the physically possible computing limit does not "hold sway above the flux", like the concepts of pure mathematics, but is firmly tied to some contingent and as yet unknown facts about the world. I have suggested that this elusive limit will extend at least as far as the Turing machine, but that it may extend yet further, to the simple SAD_1 computer and perhaps even to the AD computer. Indeed the limit point could even lie somewhere between these last two computers, e.g. at some simple SAD_n computer, where n>1. In any case, this particular issue can only be settled with a deeper understanding of singularities and the status of the cosmic censorship hypothesis.

5. Towards a Theory of Non-Turing Computability

In his famous paper of 1937, Alan Turing set out the details of what he believed to be the most general computer. The result was surprisingly simple. Just a machine comprised of an ordinary mechanical device[4] and a infinite supply of paper tape for memory storage. This Turing machine is intuitive, for sure, but it is also fantastic: it possesses an unbounded quantity of material and is capable of operating for all eternity. This might lead us to worry that such a machine is physically impossible (how, otherwise, could it be set apart from the incredible shrinking computer of the previous section?), but for Turing and most of his followers there was no such concern. They simply grasped the idea of a finite computing device, closed their eyes, and extrapolated like mad.

Now Turing's intuition was, I presume, based on Newtonian spacetime. Had his intuition been deepened by exposure to SAD_1 spacetimes then he might have arrived at the SAD_1 computer in Figure 1. Of course that is only a guess. But it does show that if the intuitive approach of Turing and his followers—roughly, the global mechanics of the machine are unproblematic—is applied not to (the now defunct) Newtonian spacetime but to a spectrum of relativistic spacetimes, the result is not only Turing machines but also various kinds of non-Turing computers. This provides the initial impetus for elaborating a theory of non-Turing computability. The theory's *raison d'être* is further discussed in the final section. But now I turn to some results and conjectures of the theory itself.

We first recall that Turing computability theory ordinarily begins with an argument aimed at showing that any number of Turing machines in any configuration can always be mimicked by a single appropriately programmed Turing machine. The theory then proceeds in terms of this one abstract and easily characterised machine, and thereby manages to transcend irrelevant hardware details. I now use this approach with the simple SAD_1 computer and the AD computer (i.e. the least and the most powerful of the SAD computers, roughly speaking). Thus, an *ideal* simple SAD_1 computer is a simple SAD_1 computer that can mimic any other simple SAD_1 computer; and an *ideal* AD computer is defined analogously. Now consider the following five claims.

(a) The SAD1 computer underpinned by the spacetime in Figure 1 and fitted with a single Turing machine that follows the l-curve is an ideal simple SAD1 computer. I shall refer to this particular computer as the *naked Turing machine*.

(b) All partially Turing solvable problems are solvable by the naked Turing machine. They included: all Turing solvable problems, the Halting problem, first-order predicate logic (i.e. Hilbert's Entscheidungsproblem), Diophantine problem (i.e. Hilbert's tenth problem) and the word problem for semi-groups.

(c) Arithmetic is not decidable by the naked Turing machine.

(d) A decision problem that is not partially Turing solvable is not solvable by the naked Turing machine.

Statements (b), (c) and (d) are true ((b) is Result 1, (c) and (d) are easily proved), but statement (a) is a conjecture. My evidence for (a) derives only from the fact that all my attempts to construct a simple SAD_1 computer that cannot be mimicked by this putative ideal computer have failed.

Now to AD computers.

(e) The AD computer underpinned by the spacetime in Figure 4 and fitted with the AD solving hardware described in Section 2 is an ideal AD computer. I shall refer to this particular computer as the *multi-string computer*.

(f) Arithmetic is decidable by the multi-string computer.

(g) The multi-string computer can mimic any computer underwritten by a relativistic spacetime and containing only Turing machines and simple communications devices. In other words, the multi-string computer is the "ideal relativistic computer".

Statement (f) is true, but statements (e) and (g) are conjectures. Because (g) is stronger than (e), I will only try to justify the former. Two arguments support its case. The first is that *if* SAD_1 computers are the basic building blocks of relativistic computers (as they seem to be), and *if* forming strings is the best method of connecting computers together (as it seems to be), then the multi-string computer is king because it possesses strings of every order. The other reason is that all my various attempts to construct a machine that cannot obviously be mimicked by this computer have failed. For example, one can show that a single multi-string computer can mimic a string of multi-string computers; it can also mimic a countably infinite number of multi-string computers working "in parallel". I know of no problems that are not solvable by the multi-string computer. But some are sure to exist.

6. Concluding Dialogue

Frank, who works on the theory of computability by means of Turing machines, reckons these new computers are not worth the candle. Isabel disagrees.

Frank I like the idea of these non-Turing computers, but frankly I can't see them catching on. They're just too, well, fantastic.

Isabel Surely a Turing machine is fantastic. At least, an infinitely massive device capable of computing to eternity sounds pretty fantastic to me.

Frank Well that's one way of putting it. I prefer to think of a Turing machine as just the natural extension of an everyday computer.

Isabel In a way it is. And that's why the hardware of these new computers is chosen to be essentially nothing but Turing machines. A simple SAD_1 computer, for example, is just a Turing machine to the past of a point. Or in more picturesque terms, it's a naked Turing machine.

Frank Yes, O.K., when I said the non-Turing computers are fantastic I didn't mean the hardware so much as the spacetime supporting the hardware. Of course I believe in Turing machines; that's my job! No, my unease stems from the wild spacetimes you employ.

Isabel They are not all wild. For example, there is a spacetime that represents a charged black hole which is SAD_1.

Frank Yes, but surely the spacetime underlying our universe is not like that. These solutions are just idealisations.

Isabel That's beside the point. You don't want to rubbish a hypothetical computer—Turing or non-Turing—simply because it can't fit into our universe. If you do, you'll leave your precious Turing machine to the mercy of the cosmologists, because according to one of their theories, the universe and all it contains, will crunch to nothing in a few billion years. Your Turing machine would be cut-off in mid-calculation!

Frank O.K. I grant you that the particular spacetime structure of our universe has little bearing here. But isn't it true that while lots of really nice spacetimes can house Turing machines, all the spacetimes that might house non-Turing computers, including that black hole spacetime you just mentioned, are in some way grossly unphysical? I've heard you say yourself that these spacetimes are prone to infinite photon blue shifts, horizon instabilities, and the red pencil of the cosmic censor. They don't seem to have a hope.

Isabel On the contrary, they have lots of hope. For one thing, the problems you mention are probably just aspects of the putative cosmic censor; they're not additional problems. And the verdict on whether there is a cosmic censor could go either way: the jury is still out. I take it that you're prepared to accept that. In that case, am I to understand that you want to argue against non-Turing machines solely because the spacetimes that underwrite them may fall prey to a censor?

Frank Pretty much, yes.

Isabel Then let me put this to you. Just suppose that tomorrow we read in *Nature* that a new law of physics has been discovered that forbids spacetime to extend to temporal infinity. It would be a kind of "temporal infinity censor", if you like. Now that would censor your Turing machines, and so my question is: would the fact of this censor destroy the Turing machine's *raison d'être*? I mean: be honest, would you quit your line of research?

Frank Well probably not.

Isabel And I say "fair enough". But just as the prospect of temporal censorship need not affect your attitude towards Turing computers, so also the prospect of cosmic censorship need not put us off non-Turing computers.

Notes

[1]I would like to thank Gordon Belot, George Boolos, Rob Clifton, John Norton, Adrian Stanley and particularly Jeremy Butterfield for their helpful suggestions.

[2]I shall follow the standard notational conventions of Hawking and Ellis (1973). All spacetimes are assumed to be causally well-behaved in the sense that they satisfy strong causality.

[3]There are other known AD spacetimes, but this beautifully simple example is due to John Norton.

[4]In Turing (1937), the "mechanical device" is actually a man faithfully executing a finite set of instructions.

References

Boolos, G.S. and Jeffrey, R.C. (1989). *Computability and Logic*. Cambridge: Cambridge University Press.

Earman, J. and Norton, J. (1993). "Forever is a Day: Supertasks in Pitowsky and Malament-Hogarth Spacetimes" *Philosophy of Science* 5: 22-42.

Earman, J. (1994). *Bangs, Crunches, Wimpers, and Shrieks: Singularities and Acausality in Relativistic Spacetimes*. Forthcoming book.

Grünbaum, A. (1967). *Modern Science and Zeno's Paradoxes*. London: George Allen and Unwin.

Hawking, S.W. and Ellis, G.F.R. (1973). *The Large Scale Structure of Space-Time*. Cambridge: Cambridge University Press.

Hogarth, M.L. (1992) "Does General Relativity Allow an Observer to View an Eternity in a Finite Time?" *Foundations of Physics Letters* 5:173-181.

Horowitz, T. (1992) "Computability as a Physical Modality" Unpublished manuscript.

Pitowsky, I. (1990) "The Physical Church Thesis and Physical Computational Complexity", *Iyyun* 39: 81-99.

Turing, A.M. (1937) "On Computable Numbers, with an Application to the *Entscheidungsproblem*", *Proc. Lond. Math. Soc.* 42: 230-65; a correction 43, 544-6