Draft (diagrams incomplete)

Sheffield 13.viii.06

Non-Turing Computers are the new Non-Euclidean Geometries

Mark Hogarth

mhogarth@cantab.net

1. Introduction

While it goes without saying that computing devices need space and time (spacetime), nothing in Turing's theory of idealized computation prepares one for the rich nature of a computer's dependence on the underlying spacetime geometry. Nor does that theory prepare one for the vast array of non-Turing computers (and one computer à la Turing) obtained by painting ordinary Turing machine hardware on a variety of different geometries. Different geometries yield radically different computers, and the range of computers extends way beyond Turing's machine. Explicating and assessing this discovery is the object of this article.

The discovery itself derives from a physical theory (actually general relativity: Section 2); but its significance lies primarily with the pure mathematical/logical concept of computability (*C*omputability). According to the orthodox position, Computability is captured by Turing's picture of that concept. In response to the newly discovered deep connection between Computability and (the concept of) Geometry, I propose a different picture. In this picture Computability is a *two-sided* concept like the concept of geometry (Geometry), and the pure side contains Turing's picture as a detail. What I mean is this. Geometry began as Euclidean geometry, but with the emergence of the non-Euclidean geometries, Geometry became a two-sided concept, with the pure geometries on one side and physical geometry on the other. Similarly, I maintain, for Computability. Computability began as Turing computability, but with the emergence of non-Turing computers, Computability appears two-sided, with the pure computers on one side and physical computability on the other.

A consequence of adopting this new picture is that the non-Turing computers are viewed in the way one views non-Euclidean geometries—as bona fida models on a par with the original model (Euclidean, Turing). Pure computers no more compete with each other than do pure geometries. It follows that the Church-Turing thesis, which is an attempt to single out fundamentally one particular model, just dissolves. Granted more needs to be said; it is in Section 7.

This approach to Computability, this way of viewing the concept, is not something that lends itself to *proof*, any more than the two-sided view of Geometry can be proved. Issues of this nature are settled in court. A case is made, a verdict given. After a hearing which began with Gauss and ended decisively with Einstein, two-sided Geometry was finally upheld. In trying to reach a judgment on Computability, we should be guided by the Geometric case for the two cases are far from independent. When presented with an argument raised against the new computers, ask: does this argument have a counterpart in the Geometry case, i.e. an argument raised (unsuccessfully of course) against the non-Euclidean geometries? If the answer is positive, then the cross-conceptual link exposes the failure of the new argument.

This is the key to understanding Computability: viewing Computability through the lens of Geometry. It's the central message of this article. Computability is painted on Geometry. Computability is analogous to Geometry. Think Computability, think Geometry. Slogans aside, the idea is basic.

This, in outline, is the view presented here. The subject has a short history, which began with Pitowsky (1991), Hogarth (1992), and some unpublished work by David Malament of around the same time. The story began with a curious discovery about spacetime—so this is the place to start.

2. Spacetime: an overview

'Henceforth space by itself, and time by itself, are doomed to fade away into mere shadows, and only a kind of union of the two will preserve an independent reality.' Minkowski

The union of space and time is *spacetime*. Figure 1 depicts a spacetime diagram representing Newtonian spacetime. By convention, time is depicted vertically, space horizontally. A horizontal 'slice', as shown, is all of spacetime at one instant: a 'snapshot' of the world. The collection of all these (3-dimensional) spatial slices taken through all of time is spacetime. A point in the spacetime diagram represents a spacetime *event*, i.e. an instant of time at a point in space. (The rapid explosion of a tiny firecracker marks an event.) If an object persists in time, then it defines a path or *worldline* through spacetime. An object's worldline represents the entire 'history' (past and future) of the object. A persistent object might be referred to as an *observer* (but there is no implied act of observation). On the time axis itself, the space coordinate is zero by definition; so the time axis represents the worldline of a stationary object, or observer. Any worldline parallel to the time axis is also stationary; a non-parallel but straight line represents an observer moving at non-zero constant velocity (with respect to the stationary worldlines). A curved worldline represents an observer whose velocity changes in time, i.e. acceleration.

Insert Figure 1

The context of our discussion is Einstein's general theory of relativity. In this theory gravity is not a force but rather an aspect of spacetime curvature. Freely falling particles, i.e. particles influenced only by gravity, traverse geodesics of the geometry. This means that two particles, which appear to be drawn together by a gravitational force, are in fact following converging geodesics. The geometry itself is determined by the matter distribution, in accordance with the Einstein Field Equations. Since there are many possible matter distributions, there are many possible geometries.

Two features that distinguish any 'relativistic' spacetime from Newtonian spacetime are particularly relevant to our discussion here. The first is the existence of a cosmic speed limit: no observer is permitted to travel faster than the speed of light, c. Units can be chosen so that $c=1$, in which case light moves on worldlines at 45° to the coordinate axis (Figure 2a). The light rays through an event p form a *lightcone* with apex p. An observer at p can (must) travel into the interior of the light cone, but not outside the cone, for that would require the observer to travel faster than c. Only light can (again, must) travel on the cone itself. The lightcone structure therefore provides the map of where observers may possibly travel. In some spacetimes this structure is simple (Figure 2b), but in others the lightcones may twist and turn (in response to curvature).

Actually 'relativistic spacetime' refers to *any* manifold which possesses a smoothly varying lightcone structure. So the physically significant relativistic spacetimes are those that satisfy the Einstein field equations for some physical matter distribution.

The second key feature concerns time itself. In Newtonian spacetime the time that elapses for an observer between two events does not depend on the observer's motion. In a relativistic spacetime, by contrast, this elapsed time—now called *proper time—is* dependent: two observers in relative motion will (in general) register different values for the elapsed time, or proper time. Gone is Newton's 'universal clock'; each observer is now equipped with her own individual 'clock', the clock which governs the rate at which local physical phenomena (e.g. her wristwatch, her body) unfold. This well-known yet counter-intuitive feature lies at the heart of the so-called twin-paradox (which is actually not a paradox at all (Lucas and Hodgson 1990)). In this thought experiment, one twin stays at home, while the other twin travels in a spaceship at constant velocity out to some distant galaxy (Figure 3). Upon arrival, the traveling twin swings round and returns to Earth, again at constant velocity. At the eventual reunion of the twins, it can be shown that the traveling twin has clocked-up less proper time than the stay-at-home twin. The traveling twin is therefore younger than the stay-at-home twin. A detailed analysis reveals that the greater the traveling-twin's velocity, the greater the discrepancy between their respective ages at the reunion. (This discrepancy creates the illusion that there is a violation of the principle of relativity; hence 'paradox'.) The relevance of this thought experiment will become clear in the next section.

3. Computing in spacetime

Consider how a Turing machine (*TM*) is ordinarily employed to solve a mathematical problem. In Figure 4, the computer user (*CU*) travels alongside the *TM* waiting for the *TM* to signal the solution. The event at which *CU* receives the solution (if one exists) I will call the *solution event*. Naturally one assumes that there is enough space and time to allow the *TM* to achieve these ends; Newtonian spacetime would do.

But one can consider TMs operating in other spacetimes, e.g. in the models of general relativity (GR). This is a natural exercise because GR is currently our best theory of spacetime in the large. A *TM* operating in the simplest spacetime, Minkowski spacetime, presents a situation analogous to the Newtonian case, except now the *CU* can reach the solution event in less proper time than it takes TM to reach the event (twin-paradox). Moreover, by taking a high-speed route, *CU* can make her elapsed proper time as short as she pleases. Impressive as this is, a rapidly completed Turing computable task is *still* a Turing computable task. If non-Turing computable tasks are possible then a different kind of spacetime is required.

The term *Malament–Hogarth (M–H)* was coined by Earman and Norton (1993). A spacetime is *M-H* if it possesses the property that there is some worldline of unbounded proper length that lies in the past of some event. Newtonian spacetime is not M-H; nor is Minkowski spacetime. But the latter can be deformed into a toy M-H. The idea is to start with Minkowski spacetime, choose a point *p* and mark a region *C* containing *p*. Define a new metric: outside *C*, the metric is the usual flat Minkowski metric; inside *C*, the metric is a scaled version of the old Minkowski metric, where the scalar rapidly approaches infinity as *p* is approached. Remove *p*. Figure ??? seems to suggest something untoward happens near *p* (isn't the TM having to shrink?), but this is an optical illusion. The interior of *C* really is just as large as the exterior, a fact made clearer in Figure ???.

CU and TM start together as before, but now CU can, in a finite span of her proper time, position herself to the entire future of the TM's immortal worldline. The setup allows for the implementation of a non-Turing computable task in the form of the halting problem. This is the problem of deciding whether an arbitrarily chosen TM will or will not halt. The procedure is simple. Following the hardware worldline, the chosen TM is primed to signal to the solution event if and only if TM halts. At the solution event CU can make no mistake: a signal means TM halted, no signal it didn't.

A word about a word. The term 'supertask' is used to denote an infinite number of tasks performed in a finite span of time. Earman and Norton (1993) use it in the title of their paper ('Forever is a day: Supertasks in Pitowsky and Malament-Hogarth spacetimes'). I prefer to omit the term in this context, partly because of its paradoxical connotations, but more to the point because both CU and TM only ever complete a *finite* number of steps in finite (proper) time. (One is tempted to say the beauty of the setup is that there is no supertask!)

4. Could our universe be M-H?

Earman and Norton (1995) raise the worry that M-H worlds may harbor a logical paradox, but they are quick to show the worry is unfounded. Assuming the logic of the situation is sound, the question raised by this section heading is a question of physics. Unfortunately it's a difficult question and there is currently no decisive answer one way or the other. What is known can be summarized as follows.

1. *Some of the standard textbook solutions to the Einstein field equation are M-H*. E.g. anti-de Sitter spacetime (in its universal covering form) and Reissner-Nordstrom spacetime (Hogarth 1994).

2. *M-H spacetimes may violate a fundamental law of physics*. M-H spacetimes possess a kind of 'naked singularity' (Hogarth 1992) and naked singularities are outlawed if the so-called 'cosmic censorship hypothesis' is correct (Penrose 1978). Still, no firm conclusions follow because the censor remains a conjecture, and the M-H singularity is actually rather benign and may not even count as 'naked' in the context of censorship (Earman and Norton 1993). On a further relationship between M-H spacetimes and cosmic censorship, see Etesi and Nemeti (2002).

3. *M-H spacetime tend to entail infinite blue shifts*. This kind of bad behavior was shown by Earman and Norton (1993) to crop up in a class of M-H spacetimes satisfying certain conditions relating to the signaling method. The result is not decisive though because other signaling methods are available. Actually, the issue here again relates to the cosmic censorship hypothesis, because part of the evidence for a censor is precisely that naked singularities tend to entail infinite blue shifts, which would in turn destroy spacetime.

5. The S*AD* computers and their axiomatization

Call the computer depicted in Figure 5 (a)/(b) SAD_1 (this stands for *first order arithmetical sentence deciding*: see Result 1 below). Another kind of *TM*–based computer can be constructed within some (but not all) *M–H* spacetimes. The basic idea is to 'string together' a sequence of 'first–order' computers to give a second–order computer (SAD_2). Continuing: SAD_3 is a string of SAD_2s, SAD_4 is a string of SAD_3s, and so on. A string with subcomponents SAD_1, SAD_2, SAD_3,… is called *AD* (*arithmetic deciding*).

Can these new computers actually be accommodated in spacetime? The answer is Yes – so long as one chooses the appropriate spacetime. For example SAD_2 cannot be accommodated in the spacetime in Figure 5(a) (Hogarth 1994), but a home can be constructed by following the procedure for creating that spacetime but this time one removes a line segment rather than just a point. This spacetime will also accommodate the other SAD_n (n>1) and *AD*. Intriguingly another home is a textbook GR solution, anti-de Sitter spacetime (in its universal covering form) (Hogarth 1996).

Now, textbook Turing computability theory usually begins with an argument aimed at showing that any finite number of ordinary computing machines in any configuration can always be mimicked by a single appropriately programmed *TM*. The theory then proceeds in terms of this one abstract and easily characterized machine, and thereby manages to transcend irrelevant hardware details. Indeed it is this very process of abstraction that makes Turing computability a branch of pure mathematics.

This approach can be applied to other *TM*–based computers. This will include the *TM* that operates à la Turing. I will call this the *ordinary TM* (*OTM*). The 'axiomatization' of a particular physical computer is achieved by providing a computationally equivalent schematic representation — a blue print, if you like. These are depicted in Figure ???.

Figure ???.

The diagrams are based on spacetime diagrams, but the reader can—and from a pure mathematical viewpoint *should*—view them as purely schematic. The lower filled dot represents the initial event; the upper filled dot a solution event; a line is a hardware worldline; an empty dot is 'the edge of spacetime at infinity'; three dots means that the sequence of hardware boxes continues indefinitely. First is the *OTM*: one *TM* travels without stopping and the solution event lies ahead of a finite part of the *TM*. The SAD_1 has one *TM* traveling without ever stopping, but in this case the whole of *TM*'s worldline lies below the solution event. SAD_2 is a string without end of SAD_1 computers: each box houses a SAD_1 computer. Two representations are given: the one on the left has a *TM* at the beginning to remind us where the (finite) program that drives each of the SAD_1s resides. The representation on the right is used when no such reminder is needed. Last in the Figure is the *AD* computer: a string of increasingly powerful *SAD* computers.

Five self–explanatory rules apply.

(1) all boxes receive their instructions from the initial *TM*;

(2) the instruction set is finite;

(3) a lower box can signal to a higher box or to the solution event;

(4) each box and the signal event can receive only a finite number of signals (no swamping);

(5) downward signaling is forbidden.

6. The power of the *SAD* computers

This will be measured in terms of a computer's capacity to decide relations in the *arithmetical hierarchy* (Rogers (1987), Chapter 14). Such a relation is either recursive or can be obtained from a recursive relation by a finite sequence of operations involving the quantifiers $\exists$ or $\forall$. The term 'hierarchy' refers to a well-known theorem of Kleene: *the set of arithmetical relations with n quantifies form a proper subset of the set of arithmetical relations with (n+1) quantifiers* (so relations become genuinely more complex with increasing numbers of quantifiers). The term 'arithmetical' refers to the fact, established by Gödel, that a relation R is in the arithmetical hierarchy if and only if R is definable in elementary arithmetic (*ibid*.). This means one can investigate the decidability of relations (open sentences) in elementary arithmetic by investigating their counterparts in the arithmetical hierarchy.

It's a standard result that the *OTM* can decide arithmetical relations with no quantifiers (recursive relations), but not arithmetical relations with 1 (or more) quantifiers (*ibid*.). Briefly renaming 'OTM' 'SAD_0', renders this standard result as: SAD_0 can decide 0-quantifier arithmetic but not 1-quantifier arithmetic. You can guess what's coming.

Result 1. For $n>0$, SAD_n can decide n-quantifier arithmetic but not (n+1)-quantifier arithmetic.

The proof is in the Appendix. What the result shows is how neatly the SAD_ns map into **[onto?]** the Kleene hierarchy.

We also have two simple results regarding the *AD* computer.

Result 2. Arithmetic is decidable by *AD*.

Result 3.(Lucian Wischik [private communication]). *AD* can compute exactly $\aleph_0$ partial functions.

Since there are $\aleph_1$ partial functions, this shows that the *AD* computer is not omniscient. It is worth noting that the proof of Result 3 is essentially the proof of the corresponding *OTM* result (Rogers [1987], p. 22), for each *AD* computer is characterized by the finite program that drives it and so the set of *AD* computers can be given the usual Gödel-type numbering.

7. The concept of computability

The concept of computability, Computability, takes on a new appearance in the light of these investigations. I noted at the beginning of this article that the situation finds an analogy in the concept of geometry, Geometry, and that in the picture of Computability suggested by this analogy the Church-Turing thesis is notably absent. I maintain that once the analogy is grasped, many of the ‘old’ questions just answer themselves. Hence the brevity of this section. It is brief because the answers are brief.

To recapitulate Section 1. In the 19th century the Euclidean picture gradually gave way to a two-sided concept of geometry, with physical geometry on one side and pure geometry on the other. Physical geometry is concerned with modeling the geometry of the physical world. Pure geometry is concerned with the logical/mathematical structure of each of the many geometrical models now in the offing. Computability now also looks two-sided. Physical computability, like physical geometry, is a matter of physics. Pure computability is concerned with the logical/mathematical structure of each of the many computers now in the offing.

From a pure viewpoint geometric models do not compete with each other. We do not speak of Euclidean *v*. Riemannian.

And so it is with the pure models of computability.

The Church–Turing thesis is therefore explained away, for the very question it sought to answer—Which computational procedure or computer captures what is ‘intuitively computable’?—is premised on there being one fundamentally distinguished procedure or computer.

To put the point another way: the Church–Turing Thesis is like the outmoded claim: ‘Euclidean geometry is the true geometry’. There is not even a question of truth-value here. Those who think it true are blinkered, fixated on the Euclidean concept; those who think it false are championing another geometry, i.e. fixated on another geometry.

Those who hold that the Church–Turing Thesis is true are blinkered, fixated on the Turing machine; those who think it false are championing a different computer, i.e. fixated on another computer.

A supporter of the Church–Turing Thesis may draw attention to an interesting historical fact. In 1936, four seminal papers were published—by Church, Kleene, Turing, and Post—that each advanced a precise definition of Computability. At first, all these proposals appeared to be quite different from each other, but a detailed analysis soon showed them to be equivalent.

Does this striking confluence of ideas support the Church–Turing Thesis? The answer is: does the work of Euclid, Playfair, Wallis, Saccheri, each of whom provided an alternative characterization of Euclidean geometry, suggest that Euclidean geometry is fundamentally distinguished?

And the reason why Church, Kleene, Turing and Post arrived at the same model is essentially the same as the reason why Euclid, Playfair, Wallis and Saccheri arrived at the same model: their ideas came from a common pool.

Still, one may ask: how then did the Church–Turing Thesis come to prove so useful in establishing theorems about Turing computability? The answer is: if one's 'intuitions' have been shaped by a single system, then what is 'intuitively true' can guide one to what is in fact true *of that system*.

Again, look to Geometry. When there was just one geometry what was intuitively true about the Geometry acted as a guide to truths about Euclidean geometry. It is intively true that the angles of a triangle sum to 180, and it is true

Similarly when there was just one computational procedure or computer, what was intuitively true about Computability acted as a guide to truths about the Turing machine. That was how the thesis 'worked'.

Of course intuition plays no formal role in two-sided Geometry. We now say: any theorem in Euclidean geometry must derive solely from the logical structure of that geometry. And similarly for the other geometries.

Intuition plays no formal role in two-sided Computability. Theorems about the *OTM* must derive solely from the logical structure of that computer. And similarly for the other computers.

Appendix: Proof of Result 1

Let Σ_n (respectively Π_n) denote the set of relations expressible by a series of *n* quantifiers that begin with $\exists$ (respectively $\forall$) and act on a recursive relation. For example, if the 2–ary relation $R(u,v)=\exists x\forall yS(x,y,u,v)$, where *S* is recursive, then $R\in\Sigma_2$. Let Δ_n denote the set of relations that are expressible in both such forms; hence $\Delta_n=\Sigma_n\cap\Pi_n$.

I will prove the positive part first, then the negative part.

The positive part i.e. $R\in\Sigma_n\cup\Pi_n$ implies R is SAD_n decidable.

Proof: We need only prove '$R\in\Sigma_n$ implies *R* is SAD_n decidable', because if that is true then $S\in\Pi_n$ implies $\neg S\in\Sigma_n$ implies $\neg S$ is SAD_n decidable implies *S* is SAD_n decidable.

The proof is by induction.

The case n=1. Let *S* be a 1-ary relation of the form $S(z)=\exists xR(x, z)$, where *R* is recursive. For an arbitrarily chosen but fixed *z*, the *TM* of the SAD_1 runs recursively through $R(1, z)$, $R(2, z)$,…, and signals (to the solution event) if and only if such a relation is first reached that holds. Thus a signal implies $S(z)$ holds; no signal implies $S(z)$ does not hold.

Case $n=m$. Suppose then that $T\in\Sigma_m$ implies *T* is SAD_m decidable, where *T* can be taken without loss of generality to be 2-ary relation, i.e. $T=T(x,z)$.

Members of Σ_{m+1} take the form $\exists xT(x, z)$. The claim is that the following procedure will decide $\exists xT(x, z)$.

Let $SAD_m(y)$ denote the *y*th SAD_m component of SAD_{m+1} and let *P* denote the program used by SAD_m to decide $T(x, z)$ for any given *x, z*. Suppose *z* is arbitrarily chosen but fixed. We then program SAD_{m+1} with a set of instructions that consists of the program P together with the following instructions.

(i) $SAD_m(x)$ is assigned to decide $T(x, z)$.

(ii) SAD_m (1) signals to the solution event and to SAD_m (2) if and only if $T(1, z)$ holds.

(iii) For $x>1$, $SAD_m(x)$ signal to $SAD_m(x+1)$ if and only if $SAD_m(x\text{-}1)$ signal to $SAD_m(x)$.

(iv) For $x>1$, if $SAD_m(x)$ has not received a signal from $SAD_m(x\text{-}1)$ and $T(x, z)$ holds then $SAD_m(x)$ signals to both the solution event and $SAD_m(x+1)$

This procedure ensures that a *single* signal is sent to the solution event if and only if $\exists xT(x, z)$ holds. Hence all relations in Σ_{m+1} are SAD_m decidable. That completes the proof of the positive part.

To prove the negative part, I will first rehearse some textbook theory.

These three results are standard: $\Sigma_n \cup \Pi_n \subset \Sigma_{n+1} \cap \Pi_{n+1} = \Delta_{n+1}$(note '$\subset$' means proper subset); $(\Sigma_n - \Pi_n) \neq \varnothing$; $(\Pi_n - \Sigma_n) \neq \varnothing$. It follows that the classes Σ_0, Σ_1, Σ_2, Σ_3,… and the classes Π_1, Π_2, Π_3,… each form a strictly increasing sequence. This is the Kleene *arithmetical hierarchy theorem*.

The notion of a computing *oracle* is due to Turing. One can think of it as a 'black box' that possesses by fiat certain well–defined powers of computation (e.g. it can decide only recursively enumerable sets). By Π_1–*oracle*, for example, I mean a device that will deliver correct answers to any question of just the form 'does $R(x)$ hold?', where $R \in \Pi_1$. And by a Π_1–*oracle computer*, I mean an *OTM* which, in addition to its ordinary operations, can consult a Π_1–*oracle* at any step in the operations.

Oracles and the arithmetical hierarchy come together in this version of *Post's theorem*. A Π_n–oracle computer ($\equiv \Sigma_n$–oracle computer) can decide relations in precisely Δ_{n+1}. This shows that if one had complete 'access' to a particular level of the arithmetical hierarchy, then one could recursively access no more than a small subset of the next level up.

I will adopt the standard practice of abbreviating relations like $(\exists x)(\forall y)(\exists z)R(u,x,y,z)$ & $(\exists x)S(u,x)$, where R and S are recursive, to $\exists\forall\exists$ & $\exists$; that is, I will indicate only the quantifier symbols and logical connectives. $\exists_n$ (respectively, $\forall_n$) will denote an alternating string of n–quantifiers beginning with $\exists$ (respectively, $\forall$).

Using the *Tarski–Kuratowski algorithm* any relation expression involving quantifiers and logical connectives can be reduced to an equivalent series of alternating quantifiers, which in turn allows one to establish where a given relation is located in the arithmetical hierarchy. E.g. $\neg\forall$ is equivalent to $\exists$; $\neg(\exists\ \&\forall)$ is equivalent to $\neg(\exists\forall)$ and to $\neg\exists\neg\forall$ and finally to $\forall\exists$ (*ibid*., p. 309).

Now for the proof of the negative part of *Result* 1, i.e. there are relations in Π_{n+1} which are not SAD_n decidable.

Suppose $S(z)$ is 1-ary relation that is SAD_n-decidable. It will be shown that $S(z)$ must be of the form $S_1(z) \vee S_2(z)$, where $S_1(z)\in\Pi_n$ and $S_2(z)\in\Sigma_n$.

This will complete the proof because by the Kleene hierarchy theorem $\mathrm{S}(z)\in\Pi_n\cup\Sigma_n \subset\Sigma_{n+1}\cap\Pi_{n+1}\subset\Pi_{n+1}$

The proof is by induction.

Case of n=1. Following Earman and Norton [1995], if $S(z)$ is decidable by SAD_1 then it must perform in one of four ways:

(1) for all z, there is a signal (to arrive at the solution event);

(2) for all z, no signal means that $S(z)$ does not hold;

(3) for all z, no signal means that $S(z)$ holds;

(4) for some z no signal means $S(z)$ does not hold, and for some z no signal means $S(z)$ holds.

(In case (4) it must be a Turing-computable matter which z means 'signal implies does not hold' and which z means the opposite.)

Case (1) implies that the *TM* program must halt for each *z*, so *S*(*z*) must take the form $S(z)=R_a(z)$, where R_a is a recursive relation. Thus $S(z) \in \Pi_0$.

Case (2) implies that when *S*(*z*) holds, the *TM* halts after a finite number of steps. This means that for each *z* a number is reached which halts the *TM* program. Thus *S*(*z*) is expressible as $S(z)=\exists y R_b(y,z)$ for some recursive relation R_b; i.e. *S*(*z*) is $\exists$.

Case (3) implies that the *TM* will halt only if S(*z*) does not hold, that is, when $\neg S(z)$ does hold. From case (2), this implies that $\neg S(z)$ is $\exists$, i.e. S(*z*) is $\forall$.

Case (4) implies that for some *z* *S*(*z*) is as in case (2), so *S* is $\exists$; and for some *z* *S*(*z*) is as in case (3), so *S*(*z*) is $\forall$. Thus *S*(*z*) is $\exists\vee\forall$.

Collecting together cases (1) through (4) proves the case *n*=1.

The case n=m. Suppose that '*S*(*z*) is SAD_m–decidable, then *S*(*z*) is a 1-ary relation of the form $\exists_m \vee \forall_m$

Consider the SAD_{m+1} computer: this consists of a string of SAD_m components. Relations that a SAD_m component can decide can, by the hypothesis of the previous paragraph, be decided by a Π_m–oracle. Thus by replacing each SAD_m component by a Π_m–oracle, there is no loss of computational power. Call this new computer X. The question is: what is the power of X?

Suppose *T*(*z*) is a 1-ary relation that is decidable by X. Then, as above, X must perform in one of four ways:

(1′) for all *z*, there is a signal;

(2′) for all *z*, no signal means that *T*(*z*) does not hold;

(3′) for all *z*, no signal means that *T*(*z*) does hold;

(4′) for some *z* no signal means *T*(*z*) does not hold, and for some *z* no signal means *T*(*z*) does hold.

By analogy with the *n*=1 case and applying Post's theorem, we see that:

Case (1′) implies that $T(z)$ is

both $\exists(\exists_m \vee \forall_m)$ *and* $\forall(\exists_m \vee \forall_m)$; $\exists(\exists_m \vee \forall_m)$; $\exists_m \vee \forall_m$; $\exists_m \vee \exists_{m+1}$; $\exists_{m+1}$

Case (2′) implies that $T(z)=\exists[\exists_m \vee \forall_m]$. This implies that $T(z)$ is $\exists[\exists_m \vee \forall_m]$; $\exists\exists_{m+1}$; $\exists_{m+1}$

Case (3′) implies that $T(z)$ is $\forall[\exists_m \vee \forall_m]$; $\forall[\forall_{m+1}]$; $\forall_{m+1}$

Case (4′) implies that $T(z)$ is $[\exists_m \vee \forall_m] \vee [\exists_m \vee \forall_m]$; $\exists_m \vee \forall_m$; $\exists_{m+1}$

Collecting together cases (1′) through (4′) shows that $T(z)$ X-decidable implies $T(z)$ is $\exists_{m+1} \vee \forall_{m+1}$. But we know that $T(z)$ SAD_{m+1}–decidable implies $T(z)$ X–decidable. Thus $T(z)$ SAD_{m+1}-decidable implies $T(z)$ is $\exists_{m+1} \vee \forall_{m+1}$.

Hence the $n=m$ case implies the $n=m+1$ case and, since the $n=1$ holds, the proof is complete. ♦

References

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

Earman, J. [1995]: *Bangs Crunches Whimpers and Shrieks: Singularities and Acausalities in Relativistic Spacetimes,* Oxford, Oxford 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. and Norton, J. [1995]: ‘Infinite Pains: The Trouble with Supertasks’, in A. Morton and S. Stich (eds) *Benacerraf and his Critics,* Cambridge, MA, Blackwells, 231–261.

Etesi, G. and Nemeti, I. [2002]: ‘Non–Turing computability via Malament–Hogarth space–times’, *International Journal of Theoretical Physics* **41**(2), 341–370.

Hogarth, M. [1992]: ‘Does General Relativity Allow an Observer to View an Eternity in a Finite Time?’, *Foundations of Physics Letters,* **5**, 173–181.

Hogarth, M. [1994]: ‘Non–Turing Computers and Non–Turing Computability’, in D. Hull, M. Forbes, and R. M. Burian (eds), *PSA* 1994, Vol. 1. East Lansing, Philosophy of Science Association, 126–138.

Rogers, H. [1987]: *Theory of Recursive Functions and Effective Computability*, Cambridge, MA, MIT Press.

Penrose, R. (1978). ‘Singularities of Spacetime’, in N. R. Lebovitz, W. H. Reid, and P. O. Vandervoort (eds), *Theoretical Principles in Astrophysics and Relativity*. Chicago: University of Chicago Press, 217-243.

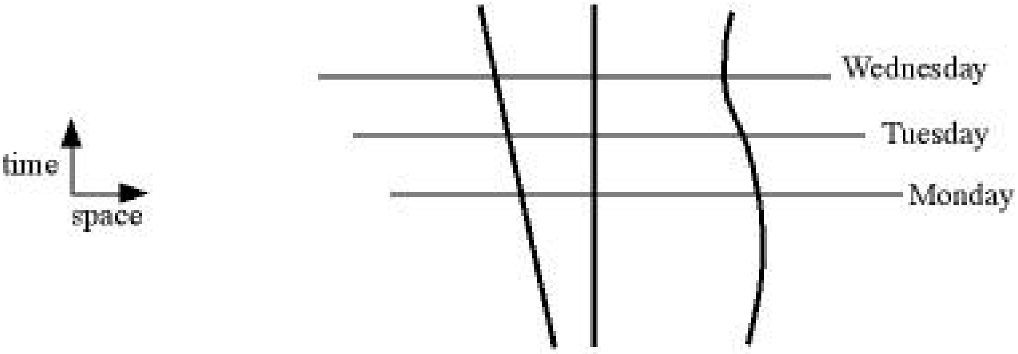

Figure 1

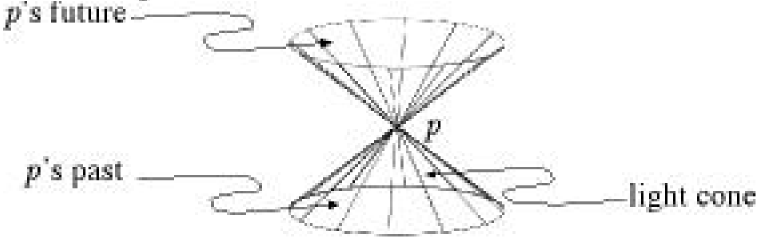

Figure 2(a)

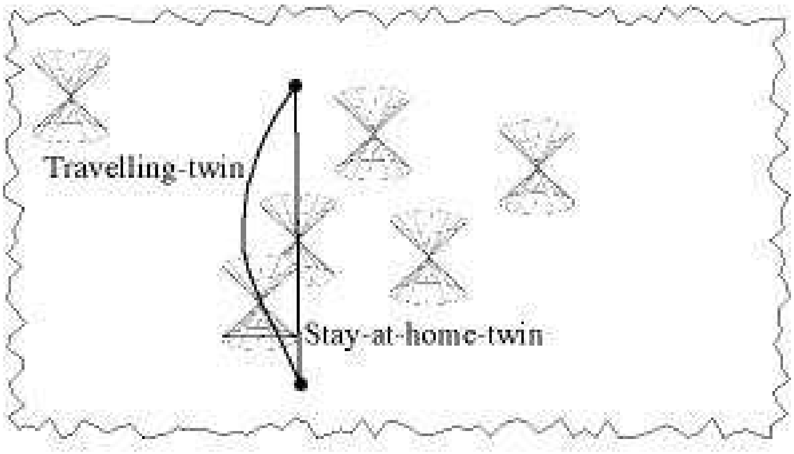

Figure 3

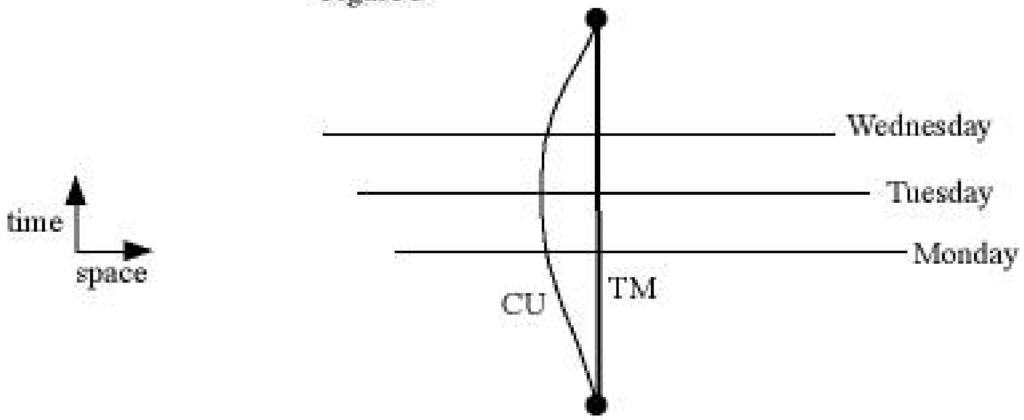

Figure 4

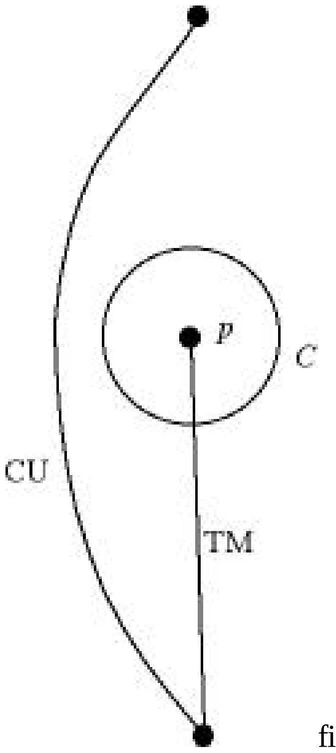

figure 5