Physical Computation: How General are Gandy's Principles for Mechanisms?

B. Jack Copeland, Oron Shagrir[1]

KEYWORDS: accelerating Turing machine, asynchronous computation, Church's thesis, Davies, determinism, Gandy, Hogarth, hypercomputation, Pitowsky, physical computation, supertask, Thesis M

Abstract

What are the limits of physical computation? In his 'Church's Thesis and Principles for Mechanisms', Turing's student Robin Gandy proved that any machine satisfying four idealised physical 'principles' is equivalent to some Turing machine. Gandy's four principles in effect define a class of computing machines ('Gandy machines'). Our question is: What is the relationship of this class to the class of all (ideal) physical computing machines? Gandy himself suggests that the relationship is identity. We do not share this view. We will point to interesting examples of (ideal) physical machines that fall outside the class of Gandy machines and compute functions that are not Turing-machine computable.

1. Introduction

We discuss a fundamental question in the Philosophy of Computer Science (PCS): What are the limits of physical computation? Do the limits of (idealised) physical computation coincide with the limits of Turing-machine computation? Or are there (idealised) physical machines that compute functions not computable by Turing machine?

In a famous paper, Robin Gandy argued that whatever can be calculated by any discrete, deterministic, mechanical device is Turing-machine computable (Gandy 1980). If Gandy is right, there can be no such thing as a discrete deterministic *hypercomputer*. (A hypercomputer is any information-processing machine, notional or real, that is able to compute functions or numbers, or more generally solve problems or carry out tasks, that

---

[1] Shagrir's research was supported by the Israel Science Foundation, grant 857/03-07.

lie beyond the reach of the universal Turing machine (Copeland and Proudfoot 1999, Copeland 2002b).) We challenge Gandy's conclusion and expose some hidden, and questionable, assumptions in his argument.

We proceed as follows. We give an outline of Gandy's argument (section 2) and then we explore the territory lying outside the class of machines conforming to Gandy's principles. We begin with a discussion of non-discrete (i.e. analogue) hypercomputers (section 3) and partially random (i.e. non-deterministic) hypercomputers (section 4), and then we turn to discrete deterministic hypercomputers, citing examples from both Newtonian and relativistic frameworks (sections 5-7). We discuss discrete machines that are deterministic in the sense that their final state is uniquely determined by their initial state, and the rules of operation, and yet fall outside Gandy's narrow characterisation of a deterministic machine (section 8). We then consider an alleged *reductio ad absurdum* argument for the conjunction of Gandy's principles I-IV (section 9).

2. Summary of Gandy's argument

Gandy seeks to provide an argument for the following thesis:

"**Thesis M**. *What can be calculated by a machine is [Turing machine] computable*" (1980: 124).

He immediately clarifies and narrows this statement, however, saying that he will consider only *deterministic discrete mechanical devices*; these are, he says, 'in a loose sense, digital computers' (1980: 126). Thus he is actually arguing for the following:

Gandy's Thesis: Any function that can be computed by a discrete deterministic mechanical device is Turing-machine computable.

The first step of Gandy's argument is to formulate the notion of a discrete deterministic mechanical device in terms of precise axioms, which Gandy called Principles I-IV. The first principle, which Gandy calls 'form of description', describes a deterministic discrete mechanical device as a pair $<S,F>$ where S is a potentially infinite set of states and F is a state-transition operation from S_i to S_{i+1}. Gandy chooses to define the states of the machines in terms of subclasses of hereditarily finite sets HF over a potentially infinite set of atoms that is closed under isomorphic structures (such subclasses are termed 'structural classes'); and he defines the transformations as structural operations over these classes. Putting aside the technicalities of Gandy's presentation, the first principle can be approximated as:

I. Form of description: Any discrete deterministic mechanical device M can be described by $<S,F>$, where S is a structural class, and F is a transformation from S_i to S_j. Thus, if S_0 is M's initial state, then $F(S_0)$, $F(F(S_0))$,… are its subsequent states.

Principles II and III place finiteness or boundedness restrictions on S. They can be informally expressed as:

II. Limitation of Hierarchy: Each state S_i of S can be assembled from parts, that can be assemblages of other parts, etc, but there is a finite bound on the complexity of this structure. In Gandy's terminology this comes down to the requirement that the states of a machine are members of a fixed initial segment of HF.

III. Unique Reassembly: Each state S_i of S is assembled from basic parts (of bounded size) drawn from a reservoir containing a bounded number of types of basic parts.

Principle IV, 'local causation', puts restrictions on the types of transition operation available. It says that each changed part of a state is affected by a bounded local 'neighbourhood':

IV. Local causation: Parts from which $F(x)$ can be reassembled depend only on bounded parts of x.

This principle is an abstraction of two 'physical presuppositions': 'that there is a lower bound on the linear dimensions of every atomic part of the device and that there is an upper bound (the velocity of light) on the speed of propagation of changes' (1980: 126). If the propagation of information is bounded, an atom can transmit and receive information in its bounded neighborhood in bounded time. If there is a lower bound on the size of atoms, the number of atoms in this neighborhood is bounded. Taking these together, each changed state, $F(x)$, is assembled from bounded, though perhaps overlapping, parts of x.

This restriction is weaker than Turing's 1936 conditions on human computability, since it allows state-transitions that result from changes in arbitrarily many bounded parts (in contrast, Turing allows changes in one bounded part). Gandy's characterisation encompasses *parallel* computation. '[I]f we abstract from practical limitations, we can conceive of a machine which prints an arbitrary number of symbols simultaneously', Gandy said: 'proofs of Thesis M must take parallel working into account' (1980: 124-125).

The second step of Gandy's argument is to prove a theorem asserting that any function computable by a device that satisfies principles I-IV is Turing-machine computable. We should emphasise that the proof goes much further than the (relatively trivial) reduction to the action of a single Turing machine of the actions of some given number of machines working in parallel. The class of 'Gandy machines' (i.e. machines conforming to Gandy's principles) includes machines with arbitrarily many processing parts that work on the same regions—e.g., printing on the same region of tape.

We cannot discuss the details of Gandy's principles and theorem here; we refer the reader to Sieg and Byrnes (1999) and Sieg (2002) for a detailed but simplified presentation of Gandy's argument. For our purposes, the argument can be summarised as:

STEP 1 '**Thesis P.** *A discrete deterministic mechanical device satisfies principles I-IV*.' (1980: 126)
STEP 2 '**Theorem.** *What can be calculated by a device satisfying principles I-IV is [Turing machine] computable*.' (1980: 126)
THEREFORE: **Gandy's Thesis.** What can be calculated by a discrete deterministic mechanical device is Turing-machine computable.

We argue that Gandy's characterisation of 'discrete deterministic mechanical device' is too narrow and that, his 1980 theorem notwithstanding, there are examples of (ideal) discrete deterministic mechanical devices that compute functions not computable by Turing machine.

We believe it is an open empirical question whether discrete deterministic machines contravening Gandy's principles are permitted by the physics of the real world.

That the question is indeed an empirical one (not a logical one) is evident, for such systems are permitted by Newtonian physics, as Gandy pointed out.

Gandy's discussion of the Newtonian case is tinged with regret, and rightly so, because there is a lacuna in his argument. Thesis P is falsified by Newtonian devices. Gandy remarked: 'I am sorry that Principle IV does not apply to machines obeying Newtonian mechanics' (1980: 145). (He pointed out that such machines may contain 'rigid rods of arbitrary lengths and messengers travelling with arbitrary large velocities, so that the distance they can travel in a single step is unbounded' (1980: 145).) Gandy left it as an open problem to find a reasonable alternative to Principle IV which is satisfied by Newtonian machines (1980: 145).

We proceed to catalogue interesting instances of (ideal) physical machines that lie outside the class of 'Gandy machines' and that compute (in a broad sense of 'compute') non-recursive functions. Our catalogue includes discrete, deterministic hypercomputers from both Newtonian and relativistic frameworks, as well as non-discrete and non-deterministic hypercomputers. We deal with the latter two categories first: while these are important sources of counterexamples to Thesis M, counterexamples of these types are of secondary interest here, since they do not bear on our critique of the forgoing two-step argument for Gandy's Thesis.

3. Non-discrete machines

It is straightforward to describe notional hypercomputers that make use of continuously-valued quantities (for example the neural networks described in Siegelmann and Sontag (1994) and the 'accumulator machines' described in Copeland (1997)). A very simple example of such a machine is Abramson's Extended Turing Machine, which is able to store a real number on a single square of its tape (Abramson 1971).

As Gandy was well aware, the possibilities inherent in non-discrete action throw up a zoo of prima facie counterexamples to Thesis M. In effect he dealt with this by fiat in his 1980 article: devices involving non-discrete action are not to be called 'mechanical'.

> I exclude from consideration devices which are *essentially* analogue ... I shall distinguish between 'mechanical devices' and 'physical devices' and consider only the former. (Gandy 1980: 125-6)

Mechanical devices, according to Gandy, are required to satisfy the two 'physical presuppositions' mentioned earlier: a lower bound on the linear dimensions of atomic parts of the device, and an upper bound on the speed of propagation of changes (1980: 126).

In unpublished work, Gandy faced the analogue challenge to Thesis M more directly. His 1993 manuscript 'On the impossibility of using analogue machines to calculate non-computable functions' discusses a number of specific proposals for hypercomputers, including da Costa and Doria (1991), Pour El and Richards (1979, 1989), and Penrose (1994); Gandy also mentions suggestions by Kreisel (1974, 1982).

In the manuscript Gandy summarises his position as follows:

> A number of examples have been given of physical systems (both classical and quantum mechanical) which when provided with a (continuously variable) computable input will give a non-computable output. It has been suggested that such systems might allow one to design analogue machines which would calculate the values of some number-theoretic non-computable function. Analyses of the examples show that the suggestion is wrong. ... I claim that given a reasonable definition of 'analogue machine' it will always be wrong. The claim is to be read not so much as a dogmatic assertion, but rather as a challenge.

It is worth quoting Gandy's synopsis of his argument (which is hitherto unpublished):

> In the examples known to me it is proposed that there might be an analogue machine which with input j ($\varepsilon\ \mathbb{N}$ typesetter: this is a curly maths N for the set of all natural numbers) would output 'Yes' or 'No' to questions of the form ?$j \in A$? where A is some standard recursively enumerable non-recursive set—for example the set which represents the halting problem. ... There is a total computable function a: $\mathbb{N} \twoheadrightarrow \mathbb{N}$ which enumerates A without repetitions. ... The *waiting-time* function ν is defined by
>
> (1) $\nu(j)$ ™ $\mu n.a(n) = j$. Typesetter: ™ is a placeholder for a special math symbol which will be handwritten on the proofs

> This is a partial recursive function whose domain is A and which is not bounded by any total computable function. For any particular analogue machine there is an upper bound J on the inputs it can accept. I define
>
> (2) $\beta(J) = \text{Max}\ \{\nu(j): j < J\ \&\ j \in A\}$
>
> (with Max $\varnothing = 0$). This is a total function which is not computable; indeed it eventually majorises every computable function. ...
>
> Since a given machine cannot handle numbers greater than some bound we consider a given J and the questions $?j \in A?$ for $j < J$. Now I make the following
>
> CLAIM Let J be given. Then one cannot design an analogue machine (whose behaviour is governed by standard physical laws) which will give correct answers to all the questions $?j \in A?$ for $j < J$ unless one knows a bound B for $\beta(J)$.
>
> I call this a claim rather than a conjecture because I do not think one could prove it unless one placed severe restrictions on the notion of 'analogue machine', and this I do not wish to do. But I believe that if someone proposes an analogue machine for settling $?j \in A?$ for $j < J$ then it can be shown that either they have (surreptitiously?) made use of a bound for $\beta(J)$, or that not all the given answers will be correct. To illustrate the significance of the wording of the claim, suppose (what is quite plausible) that someone proves that $j \notin A$ for all $j < J = 10$; then he can design a machine which always outputs 'NO' for $j < J$. But, because of this proof he does in fact know that $\beta(J) = 0$. Of course, if one knows a B as above then one does not need an analogue machine to settle $?j \in A?$. One simply computes $a(n)$ ...

In some brief comments on the 1993 manuscript, Kieu (2002) argues that Gandy's conception of an analogue machine is too narrow. Contra Gandy's claim that 'it can be shown that either they have (surreptitiously?) made use of a bound for $\beta(J)$, or that not all the given answers will be correct', Kieu remarks (1) that in a probabilistic analogue hypercomputer, the probability of getting an incorrect result, while never zero, might be made arbitrarily small, and he suggests that this is so in the case of his own proposal for a quantum hypercomputer (based on the quantum adiabatic theorem[2]); (2) in his proposed machine the bound for $\beta(J)$, far from being known in advance, is in fact available as an *output* of the computation:

> Gandy argued that for any particular analogue machine there is an upper bound J on the inputs it can accept. He then claimed that, with given J, one cannot have an analogue machine which will always give correct answers to all the questions $j \in J?$, for $j < J$, unless one knows a bound B for $\beta(J) = \max\{v(j): j < J \ \& \ j \in A\}$, the maximum waiting time within the range J. ... Gandy ... recognised that his claim might not stand up to the kind of analogue machines whose behaviour depends on a single quantum (similar to our process involving Hamiltonians which have discrete spectra of integer values). ... Furthermore, our proposal is not subject to Gandy's treatment since ours is a kind of probabilistic computation, where the probability that we get a wrong answer can be made arbitrarily small but is not exactly zero. Also, his upper bound B turns out to be a product of our quantum hypercomputation: it is an output of the computation rather than a required input. (Kieu 2002: 558)

We will not pursue these matters here, but observe that the deep question of whether there are (ideal) analogue machines which satisfy usability constraints and which, when provided with a computable input, will give a non-computable output, is open. We believe that Gandy thought so too ('The claim is to be read ... as a challenge').

4. Partially random machines

Turing introduced the concept of a partially random machine in his paper 'Intelligent Machinery':

> It is possible to modify the above described types of discrete machines by allowing several alternative operations to be applied at some points, the alternatives to be chosen by a random process. Such a machine will be

[2] See Hagar and Korolev (2006, 2007) for a thorough critique of Kieu's proposed hypercomputer.

> described as 'partially random'. ... Sometimes a machine may be strictly speaking determined but appear superficially as if it were partially random. This would occur if for instance the digits of the number π were used to determine the choices of a partially random machine, where previously a dice thrower or electronic equivalent had been used. These machines are known as apparently partially random. (Turing 1948: 416)

It is clear that the outputs of a partially random machine may form a noncomputable sequence (in the sense of Turing 1936), since, as Church pointed out, if a sequence of digits r_1, r_2, ... r_n, ... is random, then there is no function $f(n)=r_n$ that is calculable by the universal Turing machine (Church 1940: 134-135). Partially random machines are not Gandy machines, of course. Are there partially random (ideal) machines which form interesting counterexamples to Thesis M? We do not pursue this matter here but emphasise that the question is open.

5. Time as a source of uncomputability

Machines that satisfy Gandy's principles may produce uncomputable output (in the Turing sense) from computable input if embedded in a universe whose physical laws have Turing-uncomputability built into them. We offer the following illustration, which rests on the idea that, even in a discrete deterministic universe, intervals of time between successive events need not form a computable sequence.

Asynchronous networks of Turing machines are described in Copeland and Sylvan (1999: 54). Because the Turing machines in the network operate asynchronously, there may be no computable function F such that the states of the network form a sequence $F(S_0)$, $F(F(S_0))$,... (where S_0 is the initial state of the network). This may be so in the case of a simple network of *two* non-halting Turing machines writing binary digits to a common, initially blank, single-ended tape. Let the machines in the network be m_1 and m_2 and let the additional common tape be T; m_1 and m_2 work uni-directionally along T, never writing on a square that has already been written on, and writing only on squares all of whose predecessors have already been written on. If m_1 and m_2 attempt to write simultaneously to the same square, a refereeing mechanism gives priority to m_1. If m_1

and m_2 operate in synchrony, the evolving contents of T can be calculated by the universal Turing machine. Where m_1 and m_2 operate asynchronously, the same is true if the *timing function* associated with each machine is Turing-machine computable. The timing function for m_1, Δm_1, is defined as follows (where n, $k \geq 1$): $\Delta m_1(n) = k$ if and only if k units of operating time separate the nth fundamental operation performed by m_i from the n+1th; m_2's timing function, Δm_2, is defined similarly.

If m_1 and m_2 are operating asynchronously and at least one of Δm_1 and Δm_2 is not Turing-machine computable, then the contents of T need not form a computable sequence in Turing's sense. For example, suppose that m_1 prints only 1s and m_2 prints only 0s; let h_i be 1 if the ith Turing machine halts when started with a blank tape and be 0 otherwise; and let Δm_1 and Δm_2 be such that the sequence of printings on T is 1 followed by h_1 occurrences of 0 (i.e. one or no occurrence) followed by 1 followed by h_2 occurrences of 0, and so on. This sequence is not Turing-machine computable.

Let us imagine a universe in which the existence of such temporal sequencing is a matter of brute fact; perhaps this Δm_1 is achieved simply by the user placing m_1 at a particular spatial location. m_1's time is 'deformed' relative to m_2's. m_1 still computes the same function: its states remain the computable sequence $F(S_0)$, $F(F(S_0))$,... . What changes is the way in which m_1 and m_2 interact, with the result that the evolving contents of T do not form a computable sequence. There are no reasonable grounds for insisting that a universe like this must either be not deterministic or not discrete.

Gandy's framework makes no special provision for representing the temporal dimension of computing. If a computation is governed by a clock operating in accordance with recursive laws, then all relevant temporal information can be encoded in the finite description of the initial state S_0 of the Gandy machine. However, this cannot be done in the present case, on pain of making the description of S_0 a Turing-uncomputable number (producing an Extended Turing Machine in Abramson's sense). If the asynchronous network is embedded in a universe in which no (Turing-) uncomputability lurks, then Gandy's 'form of description for discrete deterministic machines' is adequate to describe the network, but in the imagined non-recursive (but deterministic and discrete) universe, the network becomes an example of a discrete deterministic mechanical device that falls outside the Gandy characterisation.

We discuss the issue of determinism further in sections 7 and 8.

6. Supertasks

Next we discuss machines that compute non-recursive functions by performing supertasks, i.e., by proceeding through infinitely many steps in finite time. A simple example is the *accelerating* universal Turing machine or AUTM (Copeland 1998a, 1998b, 2002a). The AUTM speeds up in a manner described by Bertrand Russell:

> The opinion that the phrase 'after an infinite number of operations' is self-contradictory, seems scarcely correct. Might not a man's skill increase so fast that he performed each operation in half the time required for its predecessor? In that case, the whole infinite series would take only twice as long as the first operation. (Russell 1936: 144)

Since $1 + 1/2 + 1/4 + 1/8 + ... + 1/2^{n-1} + ...$ is less than 2, the AUTM requires less than two units of running time to do everything that the program on its tape instructs it to do. This is true even in the case of a program that does not halt—each of the infinite number of operations that the non-halting program instructs the machine to perform will be completed before the end of the second unit of running time. Given any Turing machine program, the AUTM is able to determine, in a finite amount of time, whether or not the program halts. The program is inscribed on the tape of the AUTM. The initial square of the AUTM's tape is reserved for a display of the outcome of the AUTM's computation, 0 (for 'does not halt') or 1 (for 'halts'). The AUTM begins its work by writing 0 on the initial square; it then proceeds, in the usual manner of the universal machine, to simulate the machine whose program it has been given. If the program halts then the scanner of the AUTM returns to the initial square of the tape and replaces the 0 written there during the setting-up procedure by 1. If, on the other hand, the program does not halt, the scanner of the AUTM never returns to the start of the tape. Either way, at the end of the second unit of operating time the initial square contains the desired answer.

The AUTM is an example of a machine obeying Newtonian mechanics (Copeland 2002a: 289, Earman 1986: 34) that falls outside the class of Gandy machines—since it contravenes Gandy's requirement 'that there is an upper bound (the velocity of light) on the speed of propagation of changes' (1980: 126). Another example: Davies (2001) shows how to construct an infinite series of machines in a continuous Newtonian universe without contravening this particular requirement. Davies considers a universe that obeys Newton's laws and in which matter can be divided more and more finely while retaining the same properties. Davies' intention is to construct a series of (ideal) machines whose physical operations are similar to those of Babbage's Analytical Engine and which are 'consistent with any physics known in the year 1850' (2001: 672). The construction does not depend on the speed of signal propagation being arbitrarily high—the speed of signal propagation is bounded. Nor does the construction depend on the availability of infinitely strong materials, or on the availability of an infinite amount of energy within a finite volume. It does depend, however, on the absence of a lower bound on the size of atomic components. Indeed, the gist of the construction is that each machine is a down-sized version of its predecessor. Thus, Gandy's requirement 'that there is a lower bound on the linear dimensions of every atomic part' (1980: 126) is not satisfied.

More precisely, Davies provides a recipe for constructing a series of machines M_1, M_2, M_3,… Each machine is a Babbage-type mechanical device, supplemented by a robotic factory that can produce a new version of the machine and of the factory. The new machine M_{n+1}, including its robotic factory, is smaller in size then its predecessor M_n, and thus it can perform the same operations faster. More generally, the sizes of the machines constitute a decreasing geometric series, and so does the speed of performing each operation-type; this includes the operations involved in producing a new machine in the robotic factory of the predecessor machine. The size of all the machines together is not much larger than the size of the first machine.[3]

An infinite series of universal machines can solve the halting problem as follows. Given an argument (*m*, *n*) our machines start simulating the operations of the *m*th Turing-

---

[3] Davies's construction is in fact more complex and elegant than described here, and, of course, lengthier (see 2001: 672-674).

machine operating on input *n*. Each machine performs *one* simulating operation, M_1 the first, M_2 the second, and so forth. In addition, each machine checks, after performing the simulating operation, whether the simulated machine reached its halting state. If it did, the simulating machine sends a signal to M_1, and if not, it produces the next machine which runs the next operation, and so on. Assuming the simulated machine does not halt, it takes the infinite series of simulating machines some finite time to complete the simulation, say a minute. M_1 therefore 'waits' at most a minute. If it received a signal during that time, it prints 'halts' and if not, it prints 'does not halt'.[4]

7. Supertasks in relativistic space-time

An important construction was proposed by Pitowsky (1990), who described a machine, based on extreme acceleration, that functions in accordance with Special Relativity. He suggested that similar set-ups could be replicated by space-time structures in General Relativity. Malament (in private communications) and Hogarth (1992, 1994) provided examples of such space-time structures (e.g., anti de Sitter spacetimes). Hogarth pointed out the non-recursive computational powers of such devices, and, more generally, that which functions are computable (in the broad sense) depends on the properties of the space-time.[5] More recently, Etesi and N´emeti (2002), Hogarth (2004) and Welch (forthcoming) further explore the computational powers of these devices, within and beyond the arithmetical hierarchy.

In essence these devices rest on the observation that there are solutions to Einstein’s equations according to which there are space-times with the following property. The space-time includes a future endless curve γ with a past endpoint *q*, and it also includes a point *p*, such that the entire stretch of γ is included in the chronological past of *p*. In each such space-time there can exist a machine, *PMH* (for Pitowsky-Malament-Hogarth) that can perform infinitely many computation steps in a finite span of time. PMH is made up of a pair of standard digital computers, T_A and T_B, that are in

---

[4] See Davies for a discussion of, and an argument for, the physical plausibility of this computation (2001: 677-679).

[5] See Hogarth (1994:127-133).

communication. T_B moves along γ, and T_A along a future directed curve that connects the beginning point q of γ with p. The time it takes T_A to travel from q to p is finite, while during that period T_B completes the infinite time trip along γ.[6]

This physical set-up permits the computation of non-recursive functions. One feeds T_A with input n. T_A sends a signal with the pertinent input n to T_B. T_B is a universal machine that mimics the computation of the nth Turing machine operating on input n. In other words, T_B calculates the Turing-computable function $f(n)$ that returns the output of the nth Turing machine (operating on input n) if this Turing machine halts, and returns no value if this Turing machine does not halt. If $\mathrm{T_B}$ halts it immediately sends a signal back to T_A; if T_B never halts, it never sends a signal. Meanwhile T_A 'waits' during the time it takes T_A to travel from q to p (say, a minute). If T_A received a signal from T_B it prints '1'; if it received no signal from T_B it prints '0'. Thus the machine *PMH* computes a function that is not recursive, for *PMH* computes the function h, where $h(n)$=1 if $f(n)$ is defined, and $h(n)$=0 otherwise. h is not recursive, since h characterizes the self-halting states of Turing machines..

Naturally there are open questions about the *physical* possibility of these machines. Earman and Norton (1993) discuss some of these in detail, pointing out that the physical plausibility of *PMH* and, more generally, Malament-Hogarth space-times, are open questions which depend on 'a resolution of some of the deepest foundations problems in classical general relativity, including the nature of singularities and the fate of cosmic censorship' (1993: 40-41) There are also logical issues about the notion of computation assumed here, for example, concerning the non-repeatability of the computation process. These issues are addressed in Shagrir and Pitowsky (2003).

8. 'Deterministic' versus 'Gandy-deterministic'

PMH is of interest with respect to Gandy's arguments since it seemingly complies with his principles: (a) *PMH* is a discrete state machine: in fact, it consists of two communicating standard digital computers; (b) *PMH* is consistent with the laws of physics as Gandy conceives them, i.e., is consistent with the principles of Relativity. In

---

[6] For details see Hogarth (1992, 1994).

particular, *PMH* satisfies the two 'physical presuppositions' that motivate local causation (Principle IV), i.e., that there is a lower bound on size of atomic parts and an upper bound on the speed of signal propagation. We might wonder then which of Gandy's postulates is *not* satisfied.

We note that the same question arises with respect to the other two types of super-task machine described in section 6. It is true that both the AUTM and Davies' sequence of shrinking machines violate a physical claim that motivates local causation. The AUTM permits signal propagation at arbitrarily high speed, and the shrinking machines allow arbitrarily small components. Still, they both seem to comply with local causation itself. In each case there is a bound on the number of components that affect each changed state. So which of Gandy's postulates is *not* satisfied?

Merely appealing to the fact that infinitely many steps are involved does not provide an answer. Gandy's postulates also allow processes that consist of infinitely many steps. The difference, rather, is that *PMH* allows *terminating* processes that consist of infinitely many steps, whereas Gandy's proof assumes that processes consisting of infinitely many steps do not terminate. That *PMH* countenances such processes is apparent: if the simulated Turing machine is not halting, T_A does halt, after the infinitely-many-steps simulation by T_B, producing '0' as its output. But which of Gandy's principles is violated by *PMH*?

The answer lies in an ambiguity in the term 'deterministic'. Gandy says that by 'deterministic' he means that 'the subsequent behaviour of the device is uniquely determined once a complete description of its initial state is given' (1980: 126). *PMH* is certainly deterministic in this sense: its halting state, whether T_A halts on '0' or '1', is uniquely determined once a complete description of its initial state is given. But Gandy assumes more than that. He requires that the configuration of each state (step) is to be uniquely determined by the configuration of *the* previous state. This stronger requirement is present in the formulation of Principle I, which requires that the process can be described as a sequence S_0, $F(S_0)$, $F(F(S_0))$,… (where S_0 is the initial state and F is the state-transition function).

On this notion of 'deterministic', *PMH* is *not* deterministic. For consider the halting state of T_A. If T_A receives a signal from T_B, then its subsequent behaviour is deterministic in Gandy's sense. But if it receives no signal from T_B, its behaviour is no longer Gandy-deterministic. To count as deterministic, the state of T_A-halting-on-0 should be determined, in part, by the no-signal message of *the* last state of T_B. However, T_B, a non-halting Turing machine, does not have a last state. There is no state of T_B which is the one that comes just before the state of T_A-halting-on-0. For after each state of T_B, there are infinitely many others at which no signal is sent to T_A. Thus the state of T_A-halting-on-0 is undetermined in Gandy's sense.[7]

However, there is an extremely reasonable account of determinism according to which *PMH is* deterministic. It is deterministic in that the state of T_A-halting-on-0 is uniquely determined by the initial state of the machine. This is because the state of T_A-halting-on-0 is a *limit* of previous states of T_B (and T_A), of which the relevant feature is their not sending a signal to T_A. On this account, *PMH* is deterministic in that the T_A-halting-on-0 state is, in part, the limit of the previous no-signal-being-sent states of T_B. This sense of determinism is in good accord with the physical usage whereby a system or machine is said to be deterministic if it obeys laws that invoke no random or stochastic elements.

The same goes for the other two types of super-task machine, the accelerating and shrinking machines. The halting state is uniquely determined by the previous state if the simulated machine halts. But if the simulated machine does not halt, then the halting state of the super-task machine is not determined in Gandy's sense. For there is no such thing as *the* previous state which uniquely determines the configuration of the halting state. Nevertheless, these machines are deterministic in the sense that the configuration of their halting state is uniquely determined by the configuration of the initial state. This, again, is due to the fact that the configuration of the halting state is a limit of the configurations of the preceding states.

---

[7] One could define, of course, the previous state of T_A-halting-on-0 to be the state of T_B-ceasing-to-exist. But the stipulation only shifts the problem to the state of T_B-ceasing-to-exist. For now the state of T_B-ceasing-to-exist must be determined by a previous state. Yet there is not such state, for in between each state of T_B and the state of T_B-ceasing-to-exist there are infinitely many other states of T_B.

9. An independent argument: the alleged *reductio*

Concerning his four principles, Gandy argued: 'if any of the principles be significantly weakened in (almost) any way then every function becomes calculable' (1980: 130). He appeared to consider this a *reductio ad absurdum*. As Israel puts it, each of Gandy's conditions

> is necessary to avoid a certain kind of absurdity or vacuity. The form of the result is as follows: if we allow machines that satisfy any three of the ... conditions, but not all four, we can show that for any number-theoretic function f, there is a machine M_f such that for any given (notation for) n, M_f will output (a notation for) $f(n)$. That is ... every number-theoretic function is computable. (Israel 2002: 197)

We do not agree that there is any absurdity here. Each number-theoretic function *is* computable, relative to some set of capacities and resources. We believe that statements concerning computability should always be indexed, explicitly or implicitly, to a set of capacities and resources (see also Copeland and Sylvan 1999). When classicists say that some functions are absolutely uncomputable, what they mean is that some functions are not computable relative to the capacities and resources of a standard Turing machine. That particular index is of paramount interest when the topic is computation by effective procedures. In the wider study of what is computable, other indices are of importance.

Allowing that every number-theoretic function is computable (relative to some index) does not trivialise computability theory. Some indexed statements of computability are indeed trivial—for example, the statement that each number-theoretic function is computable relative to itself—but this is not generally so. Theorems of the form 'f is computable relative to r' are often hard-won. Questions about which functions are computable relative to certain physical theories are seldom trivial; and the question of which functions are computable relative to the theories that characterise the real world is of outstanding interest.

From this point of view the independent argument endorsed by Israel for the necessity of Gandy's principles is fallacious.

10. Conclusion

Gandy's notion of determinism and his 'form of description for discrete deterministic machines' are not broad enough to apply to discrete deterministic mechanical computation in general. We described several types of discrete computing machine that can reasonably be described as deterministic yet are not Gandy machines. Gandy's narrow notion of determinism fails to include some physical systems that would normally be said to obey deterministic physical laws. We also observed that Gandy's notion of a 'mechanical device' is unduly narrow. By focusing on discrete deterministic systems, Gandy leaves out interesting instances of ideal-physical computing systems. We mentioned analogue devices and partially random machines.

A more general conclusion is that one must be careful about the notions 'machine' and 'mechanical'. Turing famously provided a precise characterisation of the notion of mechanical process, in terms of Turing machines. Turing's quarry was the notion of mechanical process as it appears in the context of logic and mathematics—that is to say, a process carried out 'by human clerical labour, working to fixed rules, and without understanding' (Turing 1945: 386). This notion of mechanical process was subsequently incorporated into theoretical computer science, but with a twist: the paradigm case of a computing agent shifted from a human clerk to an artefactual physical machine (Copeland 2000, Shagrir 2002).

It is this logico-mathematical notion of 'mechanical' that Gandy had primarily in mind. Gandy wished to generalise Turing's analysis, which, as Gandy pointed out, contains 'crucial steps' where appeal is made 'to the fact that the calculation is being carried out by a human being' (1980: 124). It is because Gandy was generalising Turing's treatment that discreteness and determinism play such a central role in his presentation, and it is Turing's requirement 'that the action depend only on a bounded portion of the record' that Gandy sought to generalise in his principle of local causation (1980: 135).

The availability of robust examples of (ideal) physical machinery that fails to conform to Gandy's definition is a telling indication that the logico-mathematical notion is just *different* from the notion of 'mechanical' present in physical (and psycho-physical) discussion.[8]

REFERENCES

Abramson, F.G. (1971), 'Effective Computation over the Real Numbers', *Twelfth Annual Symposium on Switching and Automata Theory*, Northridge, Calif.: Institute of Electrical and Electronics Engineers.

Church, A. (1940), 'On the Concept of a Random Sequence', *American Mathematical Society Bulletin* 46, pp. 130-135.

Copeland, B.J. (1997), 'The Broad Conception of Computation', *American Behavioral Scientist* 40, pp. 690-716.

(1998a), 'Super Turing-Machines', *Complexity* 4, pp. 30-32.

(1998b), 'Even Turing Machines Can Compute Uncomputable Functions', in Calude, C., Casti, J., Dinneen, M. (eds) *Unconventional Models of Computation*, London: Springer-Verlag.

(2000), 'Narrow Versus Wide Mechanism', *Journal of Philosophy* 96, pp. 5-32.

(2002a), 'Accelerating Turing Machines', *Minds and Machines* 12, pp. 281-301.

(2002b) 'Hypercomputation', in Copeland (ed.) 2002-3.

(ed.) (2002-3) *Hypercomputation*. Special issue of *Minds and Machines* (Vols 12(4), 13(1)).

(ed.) (2004), *The Essential Turing*, Oxford: Oxford University Press.

[8] With thanks to two anonymous referees for helpful comments.

(ed.) (2005), *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle to Build the Modern Computer*, Oxford: Oxford University Press.

Copeland, B.J., Proudfoot, D. (1999), 'Alan Turing's Forgotten Ideas in Computer Science', *Scientific American* 280 (April), pp. 76-81.

Copeland, B.J., Sylvan, R. (1999), 'Beyond the Universal Turing Machine', *Australasian Journal of Philosophy* 77, pp. 46-66.

da Costa, N.C.A., Doria, F.A. (1991), 'Classical Physics and Penrose's Thesis', *Foundations of Physics Letters* 4, pp. 363-374.

Davies, B. (2001), 'Building Infinite Machines', *British Journal for the Philosophy of Science* 52, pp. 671-682.

Earman, J. (1986), *A Primer on Determinism*, Dordrecht: D. Reidel.

Earman, J., Norton, J.D. (1993), 'Forever is a Day: Supertasks in Pitowsky and Malament-Hogarth Spacetimes', *Philosophy of Science* 60, pp. 22-42.Etesi G. and N´emeti I. (2002), 'Non-Turing Computations via Malament-Hogarth Space-times', *International Journal of Theoretical Physics* 41, pp. 341–370.

Gandy, R. (1980), 'Church's Thesis and Principles for Mechanisms', in Barwise, J., Keisler, H.J., Kunen, K. (eds) *The Kleene Symposium*, Amsterdam: North-Holland.

Hagar A., Korolev A. (2006), 'Quantum Hypercomputability?', *Minds and Machines* 16, pp. 87-93.

(2007), 'Quantum Hypercomputation: Hype or Computation?', *Philosophy of Science* (forthcoming).

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

(1994), 'Non-Turing Computers and Non-Turing Computability', *PSA* 1, pp. 126-38.

(2004), 'Deciding arithmetic using SAD computers', *British Journal for the Philosophy of Science* 55, pp. 681–691.

Israel, D. (2002), 'Reflections on Gödel's and Gandy's Reflections on Turing's Thesis', *Minds and Machines* 12, pp. 181-201.

Kieu, T.D. (2002), 'Quantum Hypercomputation', *Minds and Machines* 12, pp. 541-561.

Kreisel, G. (1974), 'A Notion of Mechanistic Theory', *Synthese* 29, pp. 11-26.

(1982), Review of Pour-El and Richards, *Journal of Symbolic Logic* 47, pp. 900-902.

Penrose, R. (1994), *Shadows of the Mind: A Search for the Missing Science of Consciousness*, Oxford: Oxford University Press.

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

Pour-El, M.B., Richards, J.I. (1979), 'A Computable Ordinary Differential Equation Which Possesses No Computable Solution', *Annals of Mathematical Logic* 17, pp. 61-90.

(1989), *Computability in Analysis and Physics*, Berlin: Springer.

Russell, B.A.W. (1936), 'The Limits of Empiricism', *Proceedings of the Aristotelian Society* 36, pp. 131-50.

Shagrir, O. (2002), 'Effective Computation by Humans and Machines', *Minds and Machines* 12, pp. 221-240.

Shagrir, O., Pitowsky, I. (2003), 'Physical Hypercomputation and the Church-Turing Thesis' *Minds and Machines* 13, pp. 87-101.

Sieg, W. (2002), 'Calculations by Man & Machine: Mathematical Presentation', *Proceedings of the Cracow International Congress of Logic, Methodology and Philosophy of Science*, Synthese Series, Kluwer Academic Publishers.

Sieg, W., Byrnes, J. (1999), 'An Abstract Model for Parallel Computations: Gandy's Thesis', *Monist* 82, pp. 150-164.

Siegelmann, H.T., Sontag, E.D. (1994), 'Analog Computation via Neural Networks', *Theoretical Computer Science* 131, pp. 331-360.

Turing, A.M. (1936), 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society* (Series 2) 42, pp. 230-265; in *The Essential Turing* (Copeland (ed.) 2004).

(1945), 'Proposed Electronic Calculator', in *Alan Turing's Automatic Computing Engine* (Copeland (ed.) 2005).

(1948), 'Intelligent Machinery', in *The Essential Turing*.

Welch P.D. (forthcoming) 'The Extent of Computation in Malament-Hogarth Spacetimes'.