Logically Possible Machines

ERIC STEINHART
Department of Philosophy, William Paterson University, Wayne NJ 07470, USA.
Email: <esteinhart1@nyc.rr.com>, <steinharte@wpunj.edu>.

Abstract. I use modal logic and transfinite set-theory to define metaphysical foundations for a general theory of computation. A possible universe is a certain kind of situation; a situation is a set of facts. An algorithm is a certain kind of inductively defined property. A machine is a series of situations that instantiates an algorithm in a certain way. There are finite as well as transfinite algorithms and machines of any degree of complexity (e.g. Turing and super-Turing machines and more). There are physically and metaphysically possible machines. There is an iterative hierarchy of logically possible machines in the iterative hierarchy of sets. Some algorithms are such that machines that instantiate them are minds. So there is an iterative hierarchy of finitely and transfinitely complex minds.

Key words: situation, fixed-point, convergence, limit, infinite algorithm, infinite machine, super-Turing computer, possible world, infinite mind.

1. Introduction

I aim to work out formal and general metaphysical foundations for computation. The metaphysical foundations of computation are currently in deep disarray. The categories are confused. Computers are both concrete and abstract or both physical and non-physical. Stillings et al. (1995: 347) say "the Turing machine is both a physical system and an abstract computing device". The confusion has led Cleland (1993, 1995) to insightfully question the coherence of the traditional metaphysical foundations of computation. The ontological ambiguity has infected discussions of software: programs are said to be "concrete abstractions" or "machines made of text" (Colburn, 1999). The software world is filled with "virtual machines" whose parts interact causally in space and time but are not physical; we hear that "these textual entities are not physical entities and do not exist in the physical computer" (Sloman, 1993: 74, 78). My favorite: "These devices [computer chips] . . . can be made indefinitely small because of a crucial distinction. While ordinary machines work by manipulating stuff, computers manipulate information, symbols which are essentially weightless" (Johnson, 2000). Go ahead: drop your computer.

I aim to define metaphysical foundations able to support both classical computers and non-classical computers (Giunti, 1997; Seiglemann, 1996; Copeland & Sylvan, 1999; Hogarth, 1994; Hamkins & Lewis, 2000). Many supertasks have consistent recursive definitions and converge to well-defined objects at infinite limits (Koetsier & Allis, 1996). We can use infinities to define supertask computations that exceed the limits of Turing machines. However: in order to talk about such supertasks we need to use transfinite mathematics — we need set theory. We need not therefore depart from physicality (Quine 1969: 147 - 152; 1976; 1978; 1981: 15 - 18).[1] These supertasks are typically performed on physically

[1]Quinean ontological reductions of physical things to mathematical things are well-known: physical fields are functions from space-time points to (n-tuples of) numbers; space-time is an n-dimensional manifold of points; regions are classes of points; points are n-tuples of numbers; numbers are pure classes. Any ontology that reduces all objects to mathematical objects is *pythagorean*. Modern physics is deeply pythagorean (Wilczek, 1999). Pythagoreanism opposes both platonism and nominalism. It asserts: (Δ0) there are some objects; (Δ1) all objects are mathematical; (Δ2) an object x is concrete if and only if x is a

possible structures (e.g. infinitely divided spaces and times). If our actual universe does not support super-Turing computation, then this is merely a contingent fact; it is not a physical necessity. Super-Turing computation is surely physically possible. We can make precise sense of such claims only if we take modality seriously by positing physically possible universes in which such machines do in fact exist. Analog computers that work with continuous quantities are often advanced as models of physically real computation despite the fact that the physical quantities involved are all known to be discrete in our universe (e.g. charge). If quantum mechanics is right, then analog computers are at best physical idealizations (Seigelmann, 1996: secs. 1 & 6). Lewis (1986: 24 - 27) shows how to make sense of physical idealizations using possible worlds. Since analog machines gain their power from the uncountable infinity of the continuum, we once more require transfinite set theory to deal with such machines. Deutsche (1985) uses Everett's many-universes interpretation of quantum mechanics to argue that quantum mechanical computers work in parallel across alternative histories. The metaphysical foundations of computation are once again modal. Chalmers (1996) argues that we need to use modal logic in order to make sense of how algorithms are implemented by machines. Doyle (1991) defines *rational psychology* as the mathematical study of all possible minds. Rational psychology aims to fix the "equivalence classes of possible minds" in terms of their computational complexities. It needs a theory of computation able to handle modality and transfiniteness. If these authors are right, then the concepts associated with computation require for their logical analysis transfinitely complex systems of physically possible universes.

2. Facts and Situations

Properties combine with individuals. A logical space is a combinatorial plenitude of individuals with properties (Pendlebury, 1986). Suppose $D = \{d_1, d_2, \ldots\}$ is a set of individuals and $P = \{p_1, p_2, \ldots\}$ is a set of properties. Both D and P may be infinite. A property that combines with n individuals is an n-place property; properties that combine with many individuals are relations. For example: D is {A, B} and P is {p, q} where both p and q are 1-place. A *fact* over D and P is a list of the form $(p, x_1, \ldots x_n)$ where p is an n-place property and all the x_i are in D. Example: (p, A) and (q, B) are facts. We are free to set up properties in P and individuals in D as we like. More physically: let D consist of space-time point-instants; let the relations (2-place properties) in P associate point-instants with numerical locations on coordinate axes. The coordinates of point-instants determine a system of spatio-temporal order and distance relations. Those relations organize the point-instants in D into geometrically well-formed space-times. Generally: the *logical space* F over D and P is the set of all facts over D and P. Example: F is {(p, A), (p, B), (q, A), (q, B)}. A *situation* over F is any subset of F (that is: a situation over F is any set of facts over D and P). The totality S of situations over F is the set of all situations over F. S is the power set of F. If we want to do use S to provide models for modal logic, then we have to pick situations to be *worlds*. Necessity is truth at all worlds and contingency is truth at some worlds. We can define counterpart relations across situations. Computers at one

---

particular object; (Δ3) an object x is abstract if and only if x is a universal object (a property of or relation among objects); (Δ4) some mathematical objects are concrete and others are not; (Δ5) some concrete objects are physical and others are not; (Δ6) some abstract objects are physical and others are not. Class-theoretic pythagoreanism says that all concrete objects (all particulars) are classes. While this is not the place to debate pythagorean ontology, I submit that pythagoreanism serves the metaphysical needs of physical and computational theory better than either platonism or nominalism.

world have counterparts at other worlds. The possibilities of actual computers are realized by their counterparts. Finite computers have transfinite counterparts.

We can set up D and P to produce a system physically possible situations (some of which will be possible physical worlds). I say "universe" for "physical world". I assume that a physically possible universe is some spatio-temporal-causal system. I consider only *simple field-theoretic universes*. There are many theories of computation over field-theoretic universes. For example: universal Turing machines exist in cellular automata like Conway's game of life (Poundstone, 1985) and billiard ball automata (Toffoli & Margolus, 1987); electronic digital computers are easily abstracted from idealized electromagnetic fields (Scheutz, 1999). So I will discuss computers in terms of series of situations in such universes. All simple field-theoretic universes are mathematically situated within the simple field-theoretic system. The *simple field-theoretic system* is a set D of individuals, a set P of properties, a set F of facts over D and P, and a set S of situations over F. The set D of individuals is an uncountably infinite set of point-instants. For every finite i, the set P of properties contains a *spatial location functions* L_i. Space in any simple field-theoretic universe is structured by finitely many spatial location functions $L_1, \ldots L_n$. Each L_i associates each point-instant with its location on the i-the spatial coordinate axis. All spatial axes are orthogonal and that locations are given by numbers from the real numbers. The set P of properties contains one *temporal location function* T that associates each point-instant with its location on the single temporal coordinate axis. The time axis is orthogonal to all spatial axes and that temporal locations are given by real numbers. Each simple field-theoretic space-time is an (n+1)-dimensional manifold. So the location of any point-instant (and its distance relations to all other point-instants) is determined by an (n+1)-tuple of the form $(s_1, \ldots s_n, t)$ where $s_1, \ldots s_n$ are spatial coordinates and t is a temporal coordinate. These coordinates are all real numbers. We can simply identify point-instants with their locations. For every finite i, the set P of properties contains *field intensity functions* Q_i. A field associates each point-instant with some list of real number physical quantities. A scalar field maps each point-instant onto one quantity (e.g. mass, charge, etc.). A vector field associates each point-instant with many quantities (e.g. an electromagnetic vector or force vector). Each Q_i associates each point-instant in D with some real number or tuple of real numbers. So the facts in some simple field-theoretic system have the form $(q_1, \ldots q_m, s_1, \ldots s_n, t)$ where each q_i is the strength of the Q_i field, the s_i are spatial coordinates, and t is the time coordinate. If Q_i is a vector field with z values in its vector, then its q_i is a list $q_{i,1}, \ldots q_{i,z}$. For any simple field-theoretic universe, each fact is just a list of real numbers $(r_1, \ldots r_n)$. All facts in one universe have the same length.

Simple field-theoretic systems include cellular automata. The individuals in D are point-instants arranged in some 2-dimensional (2D) discrete space and 1D discrete time. Space and time are made discrete by restricting location functions to integral reals (e.g. 0.0, 1.0, 2.0). Integral reals may be regarded as integers. So the set of properties P contains functions L_1, L_2, and T that associate point-instants with integers. Functions L_1 and L_2 determine spatial locations; T determines temporal location. A point-instant is thus a triple (x, y, t) of natural numbers. The set P contains one field: a matter density field that takes on the value 0 at p if there is no matter at p and the value 1 if there is matter at p. So facts in the (2+1)-D cellular automata system are quads of the form (v, x, y, t). Simple field-theoretic universes include (non-relativistic versions of) Quine's Democritean universes. A *Democritean universe* is a "cosmic distribution of binary choices (occupied vs. empty) over the points of space-time" (Quine, 1969: 147 - 152). They also include Cresswell's (1972: 135 - 137) view that "a possible world is a set of basic particular situations" where a basic particular situation is a space-time point that is either occupied or not by matter. Both Quine and Cresswell treat universes as matter density fields over space-times. Simple field-theoretic systems include the universes of classical (Newton-Maxwell) physics. Say for the sake of illustration that a classical universe U has:[2] (1) some (3+1)-D space-time R^4; (2) some matter density field; (3) some velocity field; (4) some gravitational field; (5) some electromagnetic field. Each point in space-time is a 4-tuple (x, y, z, t) of real numbers. Suppose the matter density field is a scalar field (it maps R^4 to R). Suppose the other fields are all vector fields (say they map R^4 to R^3). Velocity is (v_x, v_y, v_z); gravity is (g_x, g_y, g_z); electromagnetism is (e_x, e_y, e_z). So every classical fact is a real number 14-tuple with the form $(v_x, v_y, v_z, g_x, g_y, g_z, e_x, e_y, e_z, m, x, y, z, t)$.

The simple field-theoretic system determines a set of logically possible physical facts. Any list of real numbers is a fact in the simple field-theoretic system. The logical space F for the simple field-theoretic system is the set of all n-tuples of real numbers. Technically: F is the union of all R^n such that n is finite. The set S of physical situations is the power set of F. It is the set of all sets of physical facts. S is organized by the subset relation into a lattice of situations. Within this lattice we find possible physical universes. Say that a *(2+1)-CA* is a cellular automaton with 2D discrete space, 1D discrete time, and one binary matter field. Every (2+1)-CA is a set of facts of the form (v, x, y, t). If we let N denote the set of positive integral real numbers, then the set of *basic physical facts* for (2+1)-CAs is the set $B = \{0, 1\} \times N^2 \times N$. The set B is a subset of the field-theoretic logical space F. The set of situations over B is S_B. Every (2+1)-CA is a situation in S_B. Conway's game of life is a (2+1)-CA; every game of life is a situation in S_B (whatever the size of its space-time and whatever patterns evolve within it according to the game of life rule). For every Turing machine and every input, there is a game of life situation that contains the entire history of that machine (whether it halts or not). So S_B contains all possible Turing computations. For Quine-Cresswell universes (with continuous (2+1)-D space-time), the set S_B is $\{0, 1\} \times R^2 \times R$. For the Newton-Maxwell universes defined above, the set S_B is R^{14}. For every kind of physical universe with (n+1)-D space-time and finitely many fields, the logical space F contains the set S_B of basic physical facts for that kind. Discrete situations are embedded in dense situations; the dense is embedded in the continuous.

A physical state is a set of simultaneous facts that fills some space. More precisely: a situation is a physical *state* if and only if there is exactly one fact in that situation for every point in some geometrically well-formed space and all facts in that set occur at the same time. A *process* is any temporally linearly ordered set of states. Processes may be discrete, dense, or continuous. They may extend finitely or infinitely through time. A situation in S is physically complete if and only if there is exactly one fact in that set for each point-instant in some space-time. Every physically complete situation is a process that is divisible into states. *Physically possible universes* are at least physically complete situations whose facts all have the same length. Every universe is at least some process. You are free to argue for additional physical necessities and to thereby further constrain the set of situations that are universes. If U is a universe (U is a set of facts), then the power set of U is the set of situations in U. The power set of U is a boolean lattice partially ordered by the subset relation. A situation in U is a subsituation of U. The subsituations of U are physically structured by three important relations: (1) a spatial part-whole relation; (2) a temporal part-whole relation; and (3) a causal relation. For any situations x and y in U, either (1) x is a spatial part of y or not; (2) x is a temporal part of y or not; and (3) x causes y or not. If U is a universe, then some subsituations of U are *things*. Let us agree that things are (at least) materially filled space-time worms. A thing in U is some situation in U that is spatially-temporally-causally continuous in the right way and internally materially dense in the right

[2]The Newton-Maxwell universe isn't quite like this; but this is just a quick example. The simple field-theoretic system does suffice for real Newton-Maxwell physics.

way. You say which ways are right. If U is a (3+1) space-time, then things in U are materially filled 4D worms. Spatially extended things are substates of U that have spatial parts. For instance: room x is in the same building as room y if and only if there exists some situation B in U such that B is a building and x is a spatial part of B and y is a spatial part of B. Temporally extended things are processes that have temporal parts (their stages). For example: person-stage x is in the same person as person-stage y if and only if there exists some situation P in U such that P is a person and x is a temporal part of P and y is a temporal part of P. Causality is a relation among situations: situation C causes situation E. Consider this crude scenario: two billiard balls are set in motion towards one another at some initial time; they collide at some later time; their paths from the initial time to the collision time are two 4D worms; these pre-collision worms are situations and their sum is a situation C; the paths of the balls from the collision time to some later final time are also two 4D worms; those post-collision worms are situations and their sum is a situation E. It's intuitively satisfying but formally crude to say C causes E. For formal detail, I rely on Lewis (1973) and Salmon (1984).[3]

3. Orderings of Situations

It's easy to define a variety of order relations on S. S is partially ordered by the subset relation (S is a *lattice* of situations). Situation u is less than situation v if and only if every fact in u is in v. If situation u is a subset of situation v, then u is a *subsituation* of v. Any universe that is larger than one fact has some subsituations: situations internal to that universe (local states of affairs or circumstances). Some collections of situations are well-ordered by the subset relation. Other order relations on situations are possible. Figures 1 to 4 illustrates a few early members of an endlessly increasing (i.e. well-ordered) series of situations. Other examples include Royce's (1899: 506 - 507) perfect map of England in England and series of situations converging to fractals (Mandlebrot, 1978). Consider as yet another example the (one-way infinite) tapes of classical Turing machines. Any such tape is an infinitely long series $\langle b_1, b_2, b_3, \ldots \rangle$ where each b_i is (say) either 0 or 1. If we regard each b_i as a binary digit, then each tape corresponds to a real number $0.b_1b_2b_3\ldots$ that lies between 0 and 1. So: tapes are situations ordered by magnitude.

For example: suppose Achilles is running along Zeno's racecourse; facts about Achilles are events of the form "Achilles is at point p at time t". Zeno's racecourse is a series of events of the form "Achilles is at point 0 at time 0", "Achilles is at point 1/2 at time 1/2", "Achilles is at point 3/4 at time 3/4, and so on. So the series is a function that pairs every number n in ω with an event of the form "Achilles is at point $(2^n - 1)/2^n$ at time $(2^n - 1)/2^n$". The series of events is both spatially and temporally well-ordered; it converges to a least upper bound (a limit): "Achilles is at point 1 at time 1". It is also recursive: each event is changed into the next by an operation that moves Achilles half the previous distance.

[3] Lewis's (1973) analysis of causality in terms of counterfactual dependence seems necessary to handle the counterfactual conditionals found in Chalmers (1996); Salmon's (1984) "principle of propagation of causal influence" (p. 455) seems especially suited for computation since it is formulated in terms of structure transmission (p. 454) which is formulated in terms of information-theoretic mark transmission (p. 451).

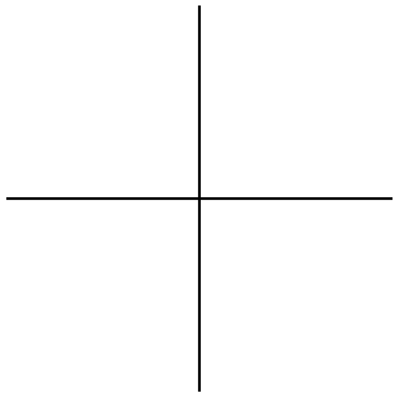

Figure 1. Cross with 1 level.

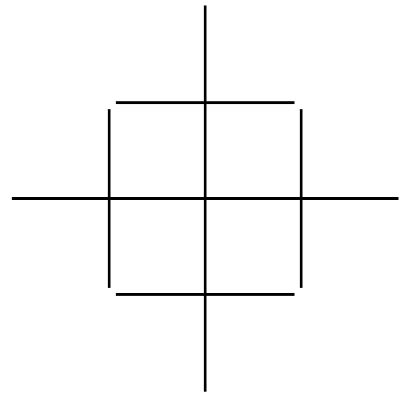

Figure 2. Cross with 2 levels.

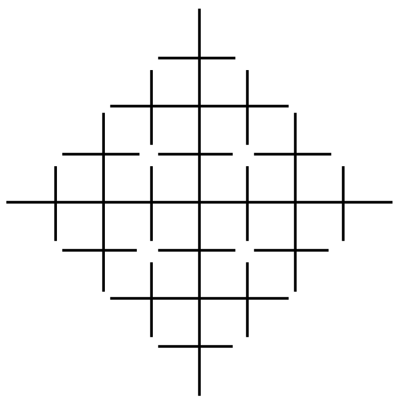

Figure 3. Cross with 3 levels.

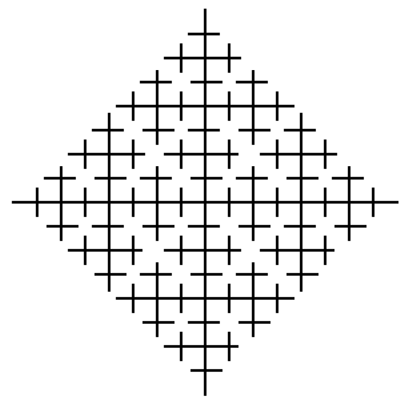

Figure 4. Cross with 4 levels.

4. Ordinals and Series of Situations

The finite ordinals are the natural numbers 0, 1, 2 and so on. Formally: 0 is an ordinal; the ordinal n+1 is the collection of all ordinals from 0 to n. So: 1 is {0}; 2 is {0, 1}; 3 is {0, 1, 2}, and n+1 is {0, 1, . . . n}. I follow von Neumann's identification of ordinals with sets: 0 is the empty set {}; if n is an ordinal, then the set (n $\cup$ {n}) is the ordinal (n+1). 0 is the *initial* ordinal. If n is an ordinal, then (n+1) is the *successor* of n. Some ordinals are transfinite. The first transfinite ordinal ω is defined recursively: {} is in ω; if n is in ω, then (n $\cup$ {n}) is in ω. So: ω is the set of all the finite ordinals. It is the set of the finite natural numbers: ω is {0, 1, 2, . . .}. ω is a *limit* ordinal; it is the *least upper bound* of the set of finite ordinals ordered by size. ω has a successor (ω+1). The ordinal ω+1 is the set {0, 1, 2, . . . ω}. There is an endlessly increasing series of transfinite ordinals.

A *series* is a function from an ordinal κ to some set V. If 5 = {0, 1, 2, 3, 4} and V = {a, b, c, d, e}, then *f* = {(0, a), (1, c), (2, d), (3, b), (4, e)} is a function from 5 to the set V; so *f* is a series. A *series* H *of situations* over some set S of situations is a function from some ordinal κ to S. The series H is *indexed* by κ. If H is a function from κ to S, then H is some set of ordered pairs (n, x) such that n is in κ and x is in S. A series H indexed by κ is finite if κ is finite and is infinite if κ is infinite. Suppose S is linearly ordered by $\leq$. A series H is *constant* if and only if for all n and m in κ, H(n) = H(m). All series I consider are non-constant unless otherwise mentioned. A series H is *increasing* if and only if for all n and m in κ, n is less than m if and only if H(n) $\leq$ H(m). A series H is *decreasing* if and only if for all n and m in κ, n is less than m if and only if H(n) $\geq$ H(m). A series H is *oscillating* if and only if it is neither increasing nor decreasing. For example: the motion of Achilles makes the increasing series$\{(0, 0), (1, 1/2), \ldots (n, (2^n - 1)/2^n), \ldots (\omega, 1)\}$.

5. Algorithmic Series of Situations

One component of the intuitive notion of an algorithm is that it is a kind of rule. We need both to define the concept "rule" and to determine which rules are algorithms. A *rule* is any formula in the language of set theory (Devlin, 1991). An operator is a (partial) function that instantiates a rule. An operator has some regularity. A function *f* is a *successor operator* if and only if *f* is a (partial) function from S to S and there is some rule R such that (x, y) is in *f* if and only if R(x, y). A series H is *recursive for successor ordinals* if and only if there is some successor operator *f* such that for every successor ordinal n less than κ, H(n) is *f*(H(n-1)). Let S^ω be the set of infinite series of situations over S. Let $\langle H(\alpha)\rangle$ denote any infinite series in S^ω. A function L is a *limit operator* if and only if (1) L is a (partial) function from S^ω to S; (2) there is some rule R such that ($\langle H(\alpha)\rangle$, y) is in L if and only if R($\langle H(\alpha)\rangle$, y); and (3) $\langle H(\alpha)\rangle$ somehow *converges* to y.[4] The set-theoretic least upper bound $\bigcup_{\alpha<\lambda} \langle H(\alpha)\rangle$ and the Cauchy limit $\lim_{\alpha\to\lambda} \langle H(\alpha)\rangle$ are both limit operators.[5] A series H is *continuous at limit ordinals* if and only if there is some limit operator L such that for every limit ordinal $\lambda < \kappa$, the limit situation H(λ) is L($\{ (h, \alpha) \in H \mid \alpha < \lambda\}$). Any series that is continuous at limit ordinals satisfies Koetsier & Allis's simple and general continuity principles (1997: 300-302).[6] A series is *well-behaved* if and only if (1) it is recursive for successor ordinals; and (2) it is continuous at limit ordinals.

---

[4]If a series H is increasing, then it converges to U if and only if U is the least upper bound of H. So: the series $\langle 0, 1/2, 3/4, 7/8, \ldots\rangle$ converges to its least upper bound 1. If a series H is decreasing, then it converges to B if and only if B is the greatest lower bound of H. So: the series $\langle 1, 1/2, 1/4, 1/8, \ldots\rangle$ converges to its greatest lower bound 0. If a series H oscillates, then it converges to A if and only if A is the average of its differences. So: the series $\langle 100, 0, 99, 1, 98, 2, \ldots\rangle$ converges to 50.

[5]If Cauchy limits are used, then convergence is defined in terms of minimizing the differences between situations. Such minimization depends on the existence of a similarity metric on S that is based on some order relation on S. If $\langle h_n\rangle$ is some ω-series of situations, then $\langle h_n\rangle$ converges to limit situation U if and only if for any $\varepsilon > 0$, there is some N such that for all n > N, the *difference* between h_n and U < ε. If $\langle h_n\rangle$ converges to L, then the limit of $\langle h_n\rangle$ as $n \to \omega$ is U. This is written standardly as: $\lim_{n\to\omega} \langle h_n\rangle = U$.

[6]Benacerraf (1962) has argued conclusively that in the context of supertasks it is necessary to carefully consider continuity at limits. Since the Thompson lamp supertask is not continuous at limits, it is meaningless to ask about its state in the limit.

Another component of the intuitive notion of an algorithm is that any process that instantiates it *halts* (it produces an effect; it is *effective*). Any process that instantiates an algorithm may halt either violently (i.e. the process is stopped by an external cause) or naturally (i.e. the algorithm stops internally). Violent termination is irrelevant; an effective algorithm is one that halts naturally. An algorithm halts naturally if and only if it proceeds to some situation at which the algorithm ceases to determine any further changes. Since algorithms are defined in terms of functions, halting is defined in terms of functions. If f is any function, then x is a *fixed-point* of f if and only if $x = f(x)$. A function ceases to determine any further changes at its fixed-points. So: an algorithm halts naturally if and only if it *converges* to a fixed-point situation (some fixed-point of its operators). It may converge to a fixed-point after either finitely or transfinitely many operations (i.e. both finite and transfinite series can converge to fixed-points). It is easy to show that the halting states of classical Turing machines are fixed-points. Well-behaved supertasks have fixed-points in the limit. For example: consider the motion of Achilles in Zeno's Racecourse. At each step, Achilles moves half as far as he did before. His first step has length 1/2; his second has length 1/4; his n-th step has length $1/2^n$. So: as $n \rightarrow \omega$, Achilles goes to 1 and the length of his step goes to 0. The length of his step in the limit is 0. So in the limit he is at 1 and he takes steps of length 0 away from 1. Therefore: 1 is a fixed-point for Achilles. Well-behaved supertasks in the calculus converge to fixed-points. For example: Newton's method for finding the roots of polynomials is a rule-governed process that is effective if and only if it converges to a root (Blum et al., 1998, sec. 1.2.3; Thomas et al., 1998, sec. 3.8). It typically requires infinitely many operations for Newton's method to converge to a root. If Newton's method converges to a root, then it does not produce any further changes; the root is a fixed-point. One more example: the power series method for solving second-order linear differential equations is effective if and only if it converges to a solution (Boyce & DiPrima, 1977: sec. 4.2). Solutions are fixed-points for the method. Conversely: tasks and supertasks that fail to converge to fixed-points are not effective. The switching rule for the Thomson lamp (Thomson, 1954) does not take the state of the lamp to any fixed-point (the state of the lamp oscillates between on and off); so the switching rule for the Thomson lamp is not effective. Consequently: a series H indexed by κ is *algorithmic* (it is *effective*) if and only if (1) H is well-behaved and (2) there is some $\alpha < \kappa$ such that $H(\alpha)$ is a fixed-point (i.e. H is constant above α).

6. Machines are Models of Algorithms

A *program* is a text written in some (programming) language. Just as a sentence expresses a proposition, so a program expresses an *algorithm*. An algorithm is an *inductively defined property* on some ordinal κ. Just as sentences written in different languages can express the same proposition (e.g. "Schnee ist weiss" and "La neige est blanc" express the proposition that snow is white), so programs written in different programming languages (like C++, Smalltalk, or Lisp) can express the same algorithm.

An algorithm is an inductively defined property that is free in some variable H and is defined on some ordinal κ. The variable H ranges over series of situations. It ranges over functions from κ to the set S of situations over some logical space F. An algorithm has the form $(\forall n)((n < \kappa) \Rightarrow C(H, n))$ where $C(H, n)$ is the conjunction of 4 kinds conditions free in H and n. The conditions specify the values of H for each different type of ordinal $< \kappa$. The *initial condition* specifies the value of H for the initial ordinal 0. It specifies an initial situation. The *successor conditions* specify successor operations that determine H for successor ordinals $< \kappa$. The *limit conditions* specify limit operations that determine H for limit ordinals $< \kappa$. An algorithm may have free variables besides H. I refer to those as its *parameters*. The parameters of an algorithm are constants (e.g. they are inputs): they must be fixed before the algorithm is evaluated relative to any series. Figure 5 illustrates an inductively defined property I called *achilles* since it is true of the series that goes from 0 to 1 by always going half-way first. The property *achilles* has no parameters. It is true or false of some (ω+1)-sequence of real numbers. If $\langle s_n \rangle$ is an (ω+1)-sequence of real numbers, and if x is some real number, then *achilles*($\langle s_n \rangle$) is either true or false. For example: *achilles*({0, 1/2, 3/4, . . . $(2^n-1)/2^n$, . . . 1}) is true. Figure 6 illustrates an inductive defined property (sin x). The property (sin x) has one parameter (namely, x). It is free in one variable (namely, H). If $\langle s_n \rangle$ is an (ω+1)-sequence of real numbers, and if x is some real number, then (sin x)($\langle s_n \rangle$) is either true or false. If P is any program (written in some language) for computing the value of (sin x) using the Taylor power series method, then P expresses the inductively defined property (sin x).

An algorithm over some set S of situations is an inductively defined property of some series in S. A *machine* is a series of situations that *instantiates* (exemplifies, implements, realizes, or models) an algorithm.[7] An algorithm stands to its machine as a theory stands to its models. The instantiation relation is an *interpretation* (in the model-theoretic sense). For any algorithm P, H is a model of P if and only if there exists some interpretation I such that P is true of H on I. So: P is true (or false) at some universe w on some interpretation I if and only if there is some H in w such that P(H) on I. It is well-known that theories have standard and non-standard interpretations (and models). The distinction between *standard* and *non-standard* interpretations has been developed for algorithms (Copeland, 1996; Chalmers, 1996; Scheutz, 1999).[8] The distinction between standard and non-standard interpretations of algorithms motivates the distinction between machines that are *metaphysically possible* and machines that are *physically possible*. Metaphysical possibility is far less constrained than physical possibility. A *metaphysically possible machine* over some logical space is *any* algorithmic series of situations over that logical space. A *physically possible machine* (over some logical space) is minimally any algorithmic series of situations (1) whose members model the algorithm in some standard way; (2) whose members all lie within the same physical universe; (3) whose members are physically linearly ordered. You are free to add further constraints. For instance: you may want your physically possible machines (4) to be things; (5) to have their situations linearly ordered in time; (6) to follow causal regularities. A machine whose situations are linearly ordered in time is an algorithmic *process*.[9] Putnam (1988, pp. 120 - 125) argues that a rock

---

[7]If "to compute is to execute an algorithm" (Copeland, 1996), then machines are computers. However: I find the term "computer" so closely tied to classical Turing machines and "computation" so closely tied to "syntactic symbol manipulation" or "information processing" that I prefer not to talk about computers or computations.

[8]For Copeland (1996) and Chalmers (1996), the standard models are those that satisfy certain constraints on the mapping of abstract states to concrete states; for Scheutz (1999), the standard models are the ones that "realize functions" in his sense. While Scheutz aims to sharply distinguish his account from those of Copeland and Chalmers, I see much in common. Look at Scheutz's use of "obeying the laws of physics".

[9]It is surely not logically necessary for a physical machine to be a process. It is far from clear what time might be apart from a linear ordering. So it may only be physically necessary for a machine to be some physically well-ordered series of situations. If time is dropped, we may drop causality too. It is surely logically possible for there to be an algorithmic series of static situations arranged motionlessly in space. Such a series is a

instantiates every finite automaton. Searle (1990, pp. 25 - 27) claims that his office wall instantiates a word processing program. If Copeland and Chalmers are right, then Putnam's rock and Searle's wall are non-standard models of algorithms. They are metaphysically but not physically possible machines. A series of situations that starts in a physical universe U may extend into logical space beyond U (it may extend into supersituations of U). So: a finitely complex universe U may contain a finite algorithmic series that logically continues to infinity beyond U; an infinitely complex universe U may contain an infinite series of finitely complex machines that converges to an infinitely complex machine in U or that converges to an infinitely complex machine in some supersituation of U. A machine in one universe has counterparts in other universes. The possibilities of any machine (e.g. its possible extensions) are realized by its counterparts. Computational counterfactuals are realized by counterparts. For example: if there weren't contingent physical limits to this machine (limits on space, time, and energy), then it would have converged to the exact solution. The limited classical Turing machines at our universe have unlimited counterparts (with actually infinite tapes) at other universes.

$$\begin{aligned}\text{achilles} = \lambda H.(\forall n)((n \leq (\omega+1)) \Rightarrow (&\\ &((n = 0) \Rightarrow (H(n) = 0))\ \&\\ &((0 < n < \omega) \Rightarrow (H(n) = H(n\text{-}1) + (1/2^n)))\ \&\\ &((n = \omega) \Rightarrow (H(n) = \lim_{k \to \omega} H(k))))).\end{aligned}$$

Figure 5. The inductive definition of property *achilles*.

$$\begin{aligned}\sin x = \lambda H.(\forall n)((n \leq (\omega+1)) \Rightarrow (&\\ &((n = 0) \Rightarrow (H(n) = x))\ \&\\ &((0 < n < \omega) \Rightarrow (H(n) = H(n\text{-}1) + \frac{(-1)^n x^{2n+1}}{(2n+1)!}))\ \&\\ &((n = \omega) \Rightarrow (H(n) = \lim_{k \to \omega} H(k))))).\end{aligned}$$

Figure 6. The inductive definition of property (sin x).

7. Classical Turing Machines

The *discrete boolean system* is one substructure of the simple field-theoretic system. It includes all Turing machines. The (restricted) set D of individuals is an infinite set of point-instants. Point-instants have properties that organize them into a 1D discrete space and 1D discrete time. The (restricted) set P of properties includes: one spatial location function L from D to the integers Z; one temporal location function M from D to the natural numbers N; and one value field Q from D to {0, 1}; one head field H from D to {0, 1}; one machine state field V from D to $\{v_0, \ldots v_n\}$. The logical space F in the discrete boolean system is the set of all 5-tuples (v, h, q, x, t) in $V \times \{0,1\} \times \{0,1\} \times Z \times N$.

machine; I see no scientific reason to deny that it is physical. I do not see why all machines must be dynamical. Static machines are logically and physically possible.

The set S of situations over F is the power set of F. A situation in S is a *boolean state* if and only if there is exactly one fact in that set for every point in space and all facts in that set occur at the same time. Let C be the set of boolean states over S. A boolean state is a *tape* if and only if (1) there is at most one (v, h, q, x, t) in that state such that h is 1 and (2) the machine state v is non-zero if and only if h is non-zero. If h is 1 in some fact (v, h, q, x, t) in some tape, then the machine head is located at the tape position x at time t. Since there is only at most one fact in any tape for which h is 1, there may be no fact for which h is 1. If some tape contains no fact for which h is 1, then the machine head is nowhere. A tape is headless iff the head is nowhere. There is likewise at most one (v, h, q, x, t) such that s is not zero. A headless tape is stateless. Let T denote the set of tapes in S. Clearly not all states are tapes: T is a subset of C. One natural way to order the tapes is to think of their positive places as the digits in a real number between 0 and 1; we can then think of the tapes as ordered by magnitude < like those real numbers. A *process* is any temporally linearly ordered set of tapes. Since time is discrete, every process is well-ordered. A *discrete boolean universe* is a set of facts that assigns field values to all point-instants in space-time (it is a physically complete situation over F). A discrete boolean universe is divisible into a temporally linearly ordered set of boolean states. Every boolean universe is some process. Since time is discrete, it is a series of tapes. We can directly index states in any discrete boolean universe by the times at which they occur. The result is time-indexed series of tapes: $\langle H_n \mid n < \omega \rangle$. The time-indexed series of states is a function from ω to states. Every discrete boolean universe is some time-indexed but ordinally incomplete series of states (it is "potentially" but not "actually" infinite).

Some but not all discrete boolean universes are classical Turing machines (CTMs). The dynamics of CTMs are well-known; I do not review them here. Any CTM consists of a tape and a control device (a read/write head and a switching table). The switching table in the control device associates every condition pair (state x, read i) pair with an action triple (write j, move the head m, go into state y). Head motions are +1, 0, -1. There is some finite upper bound on the number of states. Any CTM starts in some initial state v_0 and the head over position 0. Any CTM halts if it associates the input pair (x, i) with the action triple (i, 0, x). It halts if and only if, upon reading i in state x, it writes i, it does not move the head, and it remains in state x. Halting is convergence to a fixed-point. So a discrete boolean universe H is a *classical Turing machine* if and only if it satisfies the following Turing constraints (see Gandy, 1980; Sieg & Byrnes, 1999): (1) every state in H is a tape; (2) there are finitely many distinct states in facts in H; (3) the initial state H(0) places the head over the initial position 0 and has initial machine state v_0; (4) the value of the tape always changes only under the head; (5) the change from state H(n) to H(n+1) always moves the head at most one step in any direction; (6) there is some $\alpha < \omega$ such that $H(\alpha)$ is a fixed-point. If some discrete boolean universe H is a CTM, then there exists some operator f from C to C such that for every n, H(n+1) is f(H(n)). The function f over H corresponds to the control device. Since a CTM defines f only on tapes, there are many operators in the discrete boolean system that contain f. There are many operators that are equivalent to f when restricted to tapes. Every CTM is clearly an algorithmic process: it is a time-indexed series; it is well-behaved; it converges to a fixed-point. It is not necessary to consider the machine state and head position. A *pure tape series* is a series of headless (hence stateless) tapes. For every CTM, there is a pure finite tape series whose tapes correspond exactly to the distinct tapes of that CTM. The correspondence establishes an equivalence relation on CTMs. A CTM x is equivalent to a CTM y if and only if the pure tape series that corresponds to x is the same as the pure tape series that corresponds to y.

Many discrete boolean universes are not CTMs. There are many discrete boolean universes at which the CTMs constraints do not hold. There may be states in the universe that are not tapes. The number of heads may be infinite or endlessly increase. The head may move nonlocally in ways not reducible to CTM head motions. It may move left or right by any rule-defined distance. It may move to recursively inaccessible squares. Giunti (1997) has argued that allowing the head to directly move to recursively inaccessible squares produces a super-Turing machine. For example: the head may move from any square rightwards to the square indexed by the next higher Rado number (this is a +Rado move) or leftwards to the square indexed by the next lower Rado number (this is a -Rado move). It is easy to show how an otherwise classical Turing machine that is able to make +Rado and -Rado moves is able to compute the n-th Rado number for any n. States may be present on the tape whether or not the head is there too. The states distributed over the tape may encode a random real. The number of states may be infinite or endlessly increase. Any finite or infinite number of values may change in one time step. Siegelmann (1996) has shown how changing infinitely many values leads to super-Turing machines. The set of discrete boolean universes includes many algorithmic processes that exceed the computational power of CTMs. Super-Turing machines (STMs) are physically possible in the discrete boolean universe. CTMs have STM counterparts. Suppose X is a CTM; the computational counterfactual "If X were able to make Rado moves, then X would be able to compute function g" is true if and only if there is some discrete boolean universe in S at which X has a counterpart that does make Rado moves and does compute g.

8. Accelerating Turing Machines

The simple field-theoretic system includes the *dense boolean system*. The dense boolean system extends the discrete boolean system to the limit ordinal ω. It contains series defined on the ordinal $(\omega+1)$. The dense boolean system contains accelerating processes. An *accelerating process* locates its successive states at fractional times. For instance: an accelerating agent performs an initial act in 1/2 time-unit (e.g. 1/2 second), the next act in 1/4 time-unit, and so on (Weyl, 1963: 42; Grunbaum, 1969; Boolos & Jeffrey, 1980). Accelerating processes include *accelerating Turing machines* (ATMs; Copeland, 1998a). Copeland (1998b) has shown that ATMs are more powerful than CTMs. Hamkins & Lewis (2000) have developed a theory of (non-classical) Turing machines indexed by infinite ordinals; the countable versions of their machines can run in dense space-times.

The set D of individuals in the dense boolean system is a countably infinite set of point-instants. Point-instants have properties that organize them into a 1D space and 1D time. The set P of properties includes: one spatial location function L from D to the set X of spatial fractions {-1, . . . -7/8, -3/4, -1/2, 0, +1/2, +3/4, +7/8, . . . +1}; one temporal location function M from D to the set of Y of temporal fractions {0, 1/2, 3/4, 7/8, . . . 1}; one value field Q from D to {0, 1}; one head field H from D to {0, 1}; one machine state field V from D to $\{v_0, \dots v_n\}$. The logical space F in the dense boolean system is the set of all 5-tuples (v, h, q, x, t) in $V \times \{0,1\} \times \{0,1\} \times X \times Y$. The set S of situations over F is the power set of F. Dense boolean universes are physically complete situations. A dense boolean universe is a process that associates each moment in {0, 1/2, 3/4, 7/8, . . . 1} with some situation. The moments in {0, 1/2, 3/4, 7/8, . . . 1} correspond to the ordinals in $\{0, 1, \dots \omega\}$. If *f* is any operator for a CTM, then we can iterate *f* on the moments in {0, 1/2, 3/4, 7/8, . . . 1}. We can associate the n-th tape of any CTM with the n-th fractional moment $(2^n-1)/2^n$. We extend a CTM to an ATM by associating the limit moment 1 with some limit tape of the CTM. We do not need to worry too much about the location of the head in the limit. For example: if the head moves up without bound, then in the limit there is no finite place at which the head is located; so, in the limit, the tape is headless. A finite series of

CTM tapes with the head at finite locations can converge in the limit to a headless and stateless tape (i.e. it can converge to a pure tape).

An ATM is a CTM with an adjoined limit state. As the number of iterations of a CTM increases without bound, there are three possible situations in the limit (at ω). First: there is some finite k such that the CTM *finitely converges* at k to a fixed-point tape H(k). If the CTM converges at some finite k to a fixed-point tape H(k), then the limit tape H(ω) of the ATM is H(k). A CTM that finitely converges has an ATM counterpart. However: its ATM counterpart does nothing more than it does. Second: the CTM does not finitely converge, but *infinitely converges* to a Cauchy limit tape. A CTM infinitely converges to a Cauchy limit tape if and only if there is some tape E such that, for all $\varepsilon > 0$, there is some k such that for all $n > k$, the absolute value of the difference between H(k) and E is less than ε. If such a tape E exists, then the limit tape H(ω) of the ATM is E. A CTM that infinitely converges has an ATM counterpart that does *infinitely* more than it does. Third: if the CTM fails to converge in the limit, then it is not extendible to an ATM. If a CTM oscillates or otherwise diverges, then it is not algorithmic in the limit. A CTM that fails to converge does not have any ATM counterpart; it is not extendible to the infinite. Discrete machines that are more powerful than CTMs also fall into one of the three limit categories.

A series of binary fractions like {0, 1/2, 3/4, 7/8, . . . 1} is a Zeno-series. Each point in a Zeno-series is a Z-point. If space and time are dense, then between any two Z-points in any Zeno-series, there exists another Zeno-series. For instance: between 0 and 1/2 there is the Zeno-series {0, 1/4, 3/8, 7/16, 15/32, . . . 1/2}. It is possible to compress countably infinitely many Zeno-series between any two Z-points (Rucker, 1995, p. 67). ATMs nest countably infinitely deeply. Such nesting corresponds to embedding an infinite loop within an infinite loop. Any dense space-time is able to contain algorithmic processes whose algorithms are signified by programs containing infinitely deeply nested infinite loops. Copeland (1998b) has shown how an ATM can compute the halting function for any CTM on any input. If we have some canonical enumeration of CTMs, then we may suppose that $Q_k(n, m)$ is the k-th iteration of the application of the n-th CTM to the input m. If Copeland is right, then for each Q(n,m) is an ATM such that $\lim_{k \to \omega} Q_k(n, m) = 1$ if Q(n, m) halts; $\lim_{k \to \omega} Q_k(n, m) = 0$ otherwise. There is an algorithmic process in the dense boolean system whose algorithm is signified by the following nested loops: for $0 \leq n \leq \omega$ do { for $0 \leq m \leq \omega$ do { $H(n, m) = \lim_{k \to \omega} Q_k(n, m)$; }}. Therefore: there is a machine in the dense boolean system that converges to a matrix that encodes the halting function for CTMs. The process that produces the halting matrix is effective in the generalized sense of effectiveness I have proposed here; it makes use of no mysterious oracles. We may likewise compose ATMs to generate the look-up table used by Shagrir's (1997) supersystem.

9. Algorithmic Series of Sets

Algorithms (as inductively defined properties) plainly have models that are not series of situations in the simple field-theoretic system. An algorithm, in the least restricted sense, is an inductively defined property of some series of sets. An algorithmic series of sets is any series of sets that is well-behved (recursive at successor ordinals, continuous at limits) and that converges to some fixed-point. A *logically possible machine* is any algorithmic series of sets. Just as finite machines go on into their infinite counterparts, just as physically possible machines go on into their metaphysically possible counterparts, so metaphysically possible machines go on into their logically possible counterparts.

An example from the theory of transfinite cardinal numbers (Drake, 1974) shows how algorithmic series vastly exceed Turing computability. If X and Y are sets, then say X $\precsim$ Y if and only if there is some 1-1 function *f* such that *f* pairs each x in X with exactly one y in Y. If X $\precsim$ Y, then there is some 1-1 function *f* whose domain is X and whose range is some subset of Y. If x is infinite, then *the aleph function* $\aleph(x)$ is the next cardinal greater than x. $\aleph(x)$ is the set of all y such that y $\precsim$ x. So: $\aleph(x) = \{ y \mid y \precsim x\}$. The first *aleph* is $\aleph_0 = \omega$. The next cardinal greater than $\aleph_0$ is $\aleph(\aleph_0)$. There is a series of increasingly large alephs: $\aleph_0$, $\aleph(\aleph_0)$, $\aleph(\aleph(\aleph_0))$, and so on. Figure 7 defines the aleph series. The series of alephs is recursive at successor ordinals and is continuous at limit ordinals. It is well-behaved. The aleph series conties on into the "super-alephs".

The super-aleph series is: $\aleph_0$, $\aleph_{\aleph_0}$, $\aleph_{\aleph_{\aleph_0}}$, . . . Figure 7 also defines the super-aleph series. You make the higher super-alephs κ_n by always replacing the lowermost "0" with "$\aleph_0$". The series of super-alephs is recursive at successor ordinals and is continuous at limit ordinals. It is transfinitely algorithmic. The limit super-aleph θ is an endless series of subscripted $\aleph$'s. So what happens if we form $\aleph(\aleph_\theta)$? $\aleph(\aleph_\theta) = \aleph_{\aleph_\theta}$. But $\aleph_{\aleph_\theta}$ is an $\aleph$ subscripted by an endless series of $\aleph$'s; but that is just $\aleph_\theta$! So: $\aleph(\aleph_\theta) = \aleph_\theta$. Since $\aleph_\theta = \theta$, $\aleph(\theta) = \theta$. So the super-aleph $\aleph_\theta$ is a *fixed-point* of the aleph function. If you start from $\aleph_0$ and work up by super-aleph iteration, then you'll stop at $\aleph_\theta$. Since θ is a fixed-point of the $\aleph$ function, the $\aleph$ function *transfinitely halts* at θ. Every series halts at its fixed-points. Since the aleph-series has a fixed-point, the aleph series is algorithmic. Axiom F (Every normal function has a regular fixed point = every normal function halts) is an essential "halting theorem" for transfinite machinery (Drake, 1974: 115).

The Aleph Series.	The Super-Aleph Series
$\aleph_0 = \omega$;	$\kappa_0 = \aleph_0$;
$\aleph_{n+1} = \aleph(\aleph_n)$ if n is a successor ordinal;	$\kappa_{n+1} = \aleph_{\kappa_n}$.
$\aleph_\lambda = \bigcup_{\beta<\lambda} \aleph_\beta$ if λ is a limit ordinal.	$\theta = \cup\{ \kappa_n \mid n \in \omega\}$.

Figure 7. The Aleph and Super-Aleph series.

10. Supermachines and Superminds

If the computational theory of mind is correct, then for every mind there exists some algorithm such that any machine instantiating that algorithm is that mind. Some algorithms are such that finitely complex machines that instantiate them are finitely complex minds. The class of finitely complex minds surely includes (1) many of the minds realized by living things on Earth and (2) all of the minds realized by artifacts on Earth. If AI has achieved only the level of intelligence of an insect or chicken, that is nevertheless some non-zero level of intelligence. While human minds may not lie within the class of finitely complex minds, it would be absurd to argue that the cognitive power of the human animal exceeds every degree of complexity. There is some degree of cognitive and computational complexity that serves as an upper bound for minds realized by human bodies. Doyle (1991: 41 - 44) argues that "Human beings and Turing-equivalent machines need not exhaust the range of entities in which to realize psychologies". The upper bound for minds realized by human bodies is hardly the upper bound for cognitive complexity.

I have argued for the existence of an iterative hierarchy of machines within the iterative hierarchy of sets. The hierarchy of machines includes classical Turing machines and super-Turing machines (supermachines). If there are any algorithms for CTMs that are minds, then there are algorithms for ATMs that are minds. Finite logical operations extend to the transfinite: if there are any algorithmic processes that perform finitely complex reasoning, then there are algorithmic processes that perform transfinitely complex reasoning. It is easy to extend both classical symbol-processing AI programs and connectionist networks to the transfinite. If there are finitely complex minds, then there are transfinitely complex minds. If there is an iterative hierarchy of machines, then there is an iterative hierarchy of universal machines. At every level of the machine-hierarchy, there is a universal machine able to simulate all machines of lesser or equal complexity. So: at every level, there exists a machine that has all the mental powers of all machines of lesser or equal complexity. I conclude that there exists an endless progression of minds of ever-greater complexity and power. There is a hierarchy of *supermachines* and so a hierarchy of *superminds*. If human minds are more powerful than CTMs, then they lie somewhere within the hierarchy of transfinitely complex minds. If the modern theory of computation (and cognition) logically requires metaphysical foundations that are both modal and transfinite, then it is reasonable to infer the existence of this hierarchy of minds.

The classical Neoplatonism of Plotinus and Proclus posits a hierarchy of increasingly powerful and complex superhuman minds.[10] It is an unfortunate historical fact that the Neoplatonic hierarchy of minds and superminds is presented in absurdly superstitious terms. It is possible to argue for the existence of a Neoplatonic hierarchy of minds and superminds while arguing against all the religious mythology with which that hierarchy was classically associated (e.g. by Proclus and the Pseudo-Dionysius). The hypothesis of an infinite mind crops up repeatedly in modern philosophy — most often as a literary device, but sometimes more seriously. Dedekind used the reflexivity of self-consciousness to argue for the infinity of his own mind (Royce, 1899: 506 - 507). Boolos & Jeffrey (1980: 13 - 16) talk about how Zeus (essentially an accelerating machine) computes some function

[10]One might couple the modal theory of counterparts of persons with the theory of the hierarchy of minds to argue for a kind of Neoplatonic personal immortality. The theory of counterparts of persons is well-known (Lewis, 1986: ch. 4). If modal realists like Leibniz and Lewis are right, then our possibilities are realized by our counterparts. Our actual lives are finitely temporally extended: human minds are mortal. However: if human minds are algorithmic processes, and if finite algorithmic processes have transfinite extensions, then we have counterparts that are transfinitely continue our cognitive life-processes. For example: any human person who enumerates the first n terms of some infinite series has a counterpart (at other possible universes) who enumerates the first (n+1) terms of that series. So it is clear that for any series whose finite members can be produced by a human enumerator, there is a series of counterparts whose members collectively enumerate the entire infinite series. That series converges to a superhuman enumerator who enumerates the entire infinite series. The series of mortal enumerators converges to an immortal enumerator. If our lives are algorithmic processes, then all the transfinite possibilities of our finite lives are realized by supermachines.

of the positive integers by computing the n-th entry in $1/2^n$ seconds. Moravec (1988: Apx. 3) talks about superminds. Rucker (1995) portrays mathematical reality as existing in the intellect of an infinite mind (in a "Mindscape"). More seriously: Takeuti (1985: 255) says that "Modern mathematics is the world of the infinite mind". It would be interesting to use the tools of cognitive science and the theory of super-Turing computation to study the psychological powers of transfinitely complex minds.

Classical Neoplatonism also says that the progression of increasingly complex and powerful minds converges to an upper bound. It converges to a limit mind, to a mind than which there is none more powerful or intelligent. This Maximal Mind (the "Divine Mind") possesses all mental or cognitive perfections. If minds are algorithmic processes, then there is an upper bound to the progression of increasingly complex minds. The upper bound on the progression of minds is at most the limit of the constructible universe of sets. If my earlier reasoning is right, then this Maximal Mind exists. If the Maximal Mind is truly maximal, then it is computationally universal. All the cognitive operations of all human and superhuman minds are processes that are parts of it. It is well-known that Berkeley and Spinoza argued for the existence of some Maximal Mind. Traditional theists are likely to argue that the Maximal Mind is the Mind of God. Neoplatonists are likely to deny most of the personal or anthropomorphic features of the theistic Mind of God. They are likely to argue instead that the Maximal Mind is simply an impersonally objective feature of an eternally and necessarily existing mathematical world order. I think it would be both logically and metaphysically interesting to investigate the properties of this Maximal Mind. I do not see any logical objections to the existence and cognition of all the minds in the hierarchy of minds and to the existence and cognition of the Maximal Mind. As Kant studied the limits of pure reason, we might use the formal tools of modern mathematics and cognitive science to study the logical limits of mind qua mind.

Acknowledgements

I want to especially thank Carol Cleland for her valuable comments during the many drafts of this paper. Jim Moor gets the credit for inspiring this project. I also wish to thank Jack Copeland, Luciano Floridi, and Pete Mandik. The support of the philosophy departments at William Paterson University and Dartmouth College is greatly appreciated.

Notes

References

Benacerraf, P. (1962) 'Tasks, Super-tasks, and the Modern Eleatics', *Journal of Philosophy 59* (24), pp. 765 - 784.

Blum, L., Cucker, F., Shub, M., & Smale, S. (1998) *Complexity and Real Computation*. New York: Springer-Verlag.

Boolos, G. & Jeffrey, R. (1980) *Computability and Logic* (2nd Edition). New York: Cambridge University Press.

Boyce, W. & DiPrima, R. (1977) *Elementary Differential Equations* (3rd Edition). New York: John Wiley & Sons.

Chalmers, D. (1996) 'Does a Rock Implement Every Finite-State Automaton?', *Synthese 108*, pp. 309 - 333.

Cleland, C. (1993) 'Is the Church-Turing Thesis True?, *Minds & Machines 3,* pp. 283 - 312.

Cleland, C. (1995) 'Effective Procedures and Computable Functions', *Minds & Machines 5,* pp. 9 - 23.

Colburn, T. (1999) 'Software, Abstraction, and Ontology', *The Monist 82* (1), pp. 3 - 20.

Copeland, B. J. & Sylvan, R. (1999) 'Beyond the Universal Turing Machine', *Australasian Journal of Philosophy 77* (1), pp. 46 - 66.

Copeland, B. J. (1996) 'What is Computation?', *Synthese 108* (3), pp. 335 - 359.

Copeland, B. J. (1998a) Super-Turing Machines', *Complexity 4* (1), pp. 30 - 32.

Copeland, B. J. (1998b) 'Even Turing Machines can Compute Uncomputable Functions', in C. Calude, J. Casti, and M. Dinneen (Eds.), *Unconventional models of Computation*. New York: Springer-Verlag.

Cresswell, M. J. (1972) 'The World is Everything that is the Case', in M. Loux (Ed.) (1979) *The Possible and the Actual*. Ithaca, NY: Cornell University Press, pp. 129 - 145.

Deutsch, D. (1985) 'Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer', *Proceedings of the Royal Society,* Series A, 400, pp. 97 - 117.

Devlin, K. (1991) *The Joy of Sets: Fundamentals of Contemporary Set Theory*. New York: Springer-Verlag.

Doyle, J. (1991) 'The Foundations of Psychology: A Logico-Computational Inquiry into the Concept of Mind' in R. Cummins & J. Pollock (Eds.) *Philosophy and AI: Essays at the Interface*. Cambridge, MA: MIT Press, pp. 39 - 78.

Drake, F. (1974) *Set Theory: An Introduction to Large Cardinals*. New York: American Elsevier.

Gandy, R. (1980) 'Church's Thesis and Principles for Mechanisms', in J. Barwise, H. Keisler, K. Kunen (Eds.), *The Kleene Symposium*. New York: North-Holland Publishing, pp. 123 - 148.

Giunti, M. (1997) *Computation, Dynamics, and Cognition*. New York: Oxford University Press.

Grunbaum, A. (1969) 'Can an Infinitude of Operations be Performed in a Finite Time?', *British Journal of the Philosophy of Science 20,* pp. 203 - 218.

Hamkins, J. D. & Lewis, A. (2000) 'Infinite Time Turing Machines', *Journal of Symbolic Logic 65* (2), pp. 567-604.

Hogarth, M. (1994) 'Non-Turing Computers and non-Turing Computability', *Philosophy of Science Association 1*, 126-138.

Johnson, G. (2000) Untitled article. *New York Times*, Sunday, 31 December 2000, sec. 4 (Week in Review), p. 4.

Koetsier, T. & Allis, V. (1997) 'Assaying Supertasks', *Logique et Analyse 159*, pp. 291 - 313.

Lewis, D. (1973) 'Causation', *Journal of Philosophy 70,* pp. 556 - 567.

Lewis, D. (1986) *On the Plurality of Worlds*. Cambridge, MA: Blackwell.

Mandlebrot, B. (1978) *Fractals: Form, Chance, and Dimension*. San Francisco: W. H. Freeman.

Moravec, H. (1988) *Mind Children: The Future of Robot and Human Intelligence*. Cambridge, MA: Harvard University Press.

Pendlebury, M. (1986) 'Facts as Truthmakers', *The Monist 69* (2), 177 - 188.

Poundstone, W. (1985) *The Recursive Universe: Cosmic Complexity and the Limits of Scientific Knowledge*. Chicago: Contemporary Books, Inc.

Putnam, H. (1988) *Representation and Reality*. Cambridge, MA: MIT Press.

Quine, W. V. (1969) *Ontological Relativity and Other Essays*. New York: Columbia University Press.

Quine, W. V. (1976) Wither physical objects? *Boston Studies in the Philosophy of Science 39,* 497 - 504.

Quine, W. V. (1978) Facts of the matter. *Southwestern Journal of Philosophy 9* (2), 155 - 169.

Quine, W. V. (1981) *Theories and Things*. Cambridge, MA: Harvard University Press.

Royce, J. (1899/1927) *The World and the Individual* (First Series; Supplementary Essay). New York: Macmillan. Lectures originally delivered 1899.

Royce, J. (1900/1976) *The World and the Individual* (Second Series). Gloucester, MA: Peter Smith. Lectures originally delivered 1900.

Rucker, R. (1995) *Infinity and the Mind*. Princeton, NJ: Princeton University Press.

Salmon, W. (1984) 'Causal connections', in J. Kim & E. Sosa (Eds.) (1999) *Metaphysics*. Malden, MA: Blackwell, pp. 444 - 457.

Scheutz, M. (1999) 'When Physical Systems Realize Functions', *Minds and Machines 9*, pp. 161 - 196.

Searle, J. (1990) 'Is the Brain a Digital Computer?', *Proceedings and Addresses of the American Philosophical Association 64,* pp. 21 - 37.

Shagrir, O. (1997) 'Two Dogmas of Computationalism', *Minds and Machines 7* (3), pp. 321 - 344.

Sieg, W. & Byrnes, J. (1999) 'An Abstract Model for Parallel Computations: Gandy's Thesis', *The Monist 82* (1), pp. 150 - 164.

Siegelmann, H. (1996) 'The Simple Dynamics of Super Turing Theories', *Theoretical Computer Science 168,* pp. 461 - 472.

Sloman, A. (1993) 'The Mind as a Control System', in C. Hookway & D. Peterson (Eds.) *Philosophy and Cognitive Science*. Royal Institute of Philosophy Supplement 34. New York: Cambridge University Press.

Stillings, N., Weisler, S., Chase, C., Feinstein, M., Garfield, J., Rissland, E. (1995) *Cognitive Science: An Introduction* (2nd Edition). Cambridge, MA: MIT Press.

Takeuti, G. (1985) 'Proof Theory and Set Theory', *Synthese 62,* pp. 255 - 263.

Thomas, G., Finney, R., & Weir, M. (1998) *Calculus and Analytic Geometry* (9th Edition). Reading, MA: Addison-Wesley.

Thomson, J. (1954) 'Tasks and Supertasks', *Analysis 15,* pp. 1 - 13.

Toffoli, T. & Margolus, N. (1987) *Cellular Automata Machines: A New Environment for Modeling*. Cambridge, MA: MIT Press.

Weyl, H. (1963) *Philosophy of Mathematics and Natural Science*. New York: Atheneum. (Original work in German 1927).

Wilczek, F. (1999) Getting its from bits. *Nature 397* (28 January), 303 - 306.