A New Gödelian Argument for Hypercomputing Minds Based on the Busy Beaver Problem*

Selmer Bringsjord
Owen Kellett, Andrew Shilliday, Joshua Taylor,
Bram van Heuveln, Yingrui Yang, Jeffrey Baumes, Kyle Ross
Department of Cognitive Science
Department of Computer Science
Rensselaer AI & Reasoning (RAIR) Lab
Rensselaer Polytechnic Institute (RPI)
Troy NY 12180 USA
selmer@rpi.edu
http://www.rpi.edu/∼brings

9.9.05 1245am NY time

Abstract

Do human persons hypercompute? Or, as the doctrine of *computationalism* holds, are they information processors at or below the Turing Limit? If the former, given the essence of hypercomputation, persons must in some real way be capable of infinitary information processing. Using as a springboard Gödel's little-known assertion that the human mind has a power "converging to infinity," and as an anchoring problem Rado's (1963) Turing-uncomputable "busy beaver" (or Σ) function, we present in this short paper a new argument that, in fact, human persons *can* hypercompute. The argument is intended to be formidable, not conclusive: it brings Gödel's intuition to a greater level of precision, and places it within a sensible case against computationalism.

1 Introduction

It's safe to say that human minds are information processors of *some* sort.[1] The question is: *What* sort? Are we operating at or below the level of Turing machines (= at or below the so-called *Turing*

*My co-authors, all researchers with me in the Rensselaer AI & Reasoning (RAIR) Laboratory, made contributions to, if you will, the busy beaver "substrate" of the present paper. While the argument given herein (section 6) is mine (i.e., Bringsjord's), a lot of formal analysis and software engineering has gone into the RAIR Lab's attack on the busy beaver problem, and this attack forms the general context of my argument. Special thanks are due to Owen Kellett, whose tireless, ingenious pursuit of (in the quadruple, implicit-halt, contiguous format) $\Sigma^B(7)$ and beyond still presses on. (For cognoscenti: The B superscript is explained later.) (The pursuit is chronicled at http://www.cs.rpi.edu/∼kelleo/busybeaver.) Applied research on the busy beaver problem, which, again, forms the context for this paper, is supported by a grant from the National Science Foundation. Special thanks are due to Yingrui Yang for insights regarding Gödel's thought, and Hao Wang's thought about that thought. We are also greatly indebted to two anonymous referee's for a number of learned comments and trenchant objections, and for a remarkably careful reading of our text.

[1]Contemporary cognitive psychology, cognitive science, and cognitive neuroscience — all are predicated on the view that the human mind is an embodied information processor. For a nice survey that brings this point across clearly, see (Goldstein 2005).

Limit), or above?

Bringsjord has given a book-length specification and defense of *supermentalism*, the view that human persons are capable of hypercomputing (Bringsjord & Zenzen 2003). In addition, with Konstantine Arkoudas, he has given a purported modal proof for the very same view (Bringsjord & Arkoudas 2004). None of this prior work takes account of Gödel's intuition, repeatedly communicated to Hao Wang, that human minds "converge to infinity" in their power, and for this reason surpass the reach of ordinary Turing machines.[2] The purpose of this paper is to flesh out and defend this intuition; doing so leads to a new and perhaps not uninteresting argument for the view that human persons can hypercompute, that is, that they are hypercomputers. This argument is intended to be formidable, not conclusive: it brings Gödel's intuition to a greater level of precision, and places it within a sensible case against computationalism. In short, while the Gödelian argument at the heart of this paper rests in part on research that falls squarely within logic and computer science, the argument itself is just that: a deductive *argument*, not a *proof*.

The plan for the paper is as follows. In the next section we clarify the view to be overthrown herein, viz., that minds are *ordinary* (= Turing machine-equivalent) computing machines. In section 3, we briefly take note of the fact that the essence of hypercomputation consists in some way of exploiting the power of the infinite. In the following section, 4, we recount and summarize Gödel's intuitive-and-never-formalized view that the "states" of the human mind "converge to infinity" — a rather cryptic view he repeatedly discussed, in persistently vague terms, with Hao Wang. Section 5 is devoted to setting the context for the argument intended to formalize Gödel's intuitive view; this context is based on the so-called uncomputable "busy beaver" (or Σ) problem in computer science. In section 6 we give our Gödelian argument, in the form of an indirect proof. In section 7 we rebut a series of objections to our argument. The final section is a short conclusion.

2 Clarifying Computationalism, The View to be Overthrown

The view to be overthrown by the Gödelian argument articulated in this paper is the doctrine of *computationalism*. Propelled by the writings of innumerable thinkers (this touches but the tip of a mammoth iceberg of relevant writing: Peters 1962, Barr 1983, Fetzer 1994, Simon 1980, Simon 1981, Newell 1980, Haugeland 1985, Hofstadter 1985, Johnson-Laird 1988, Dietrich 1990, Bringsjord 1992, Searle 1980, Harnad 1991), computationalism has reached every corner of, and indeed energizes the bulk of, contemporary AI and cognitive science. The view has also touched nearly every major college and university in the world; even the popular media have, on a global scale, preached the computational conception of mind. Despite all this, despite the fact that computationalism has achieved the status of a Kuhnian paradigm, the fact is that the doctrine is maddeningly vague. Myriad one-sentence versions of this doctrine float about; e.g.,

- Thinking is computing.
- Cognition is computation.
- People are computers (perhaps with sensors and effectors).
- People are Turing machines (perhaps with sensors and effectors).
- People are finite automata (perhaps with sensors and effectors).

[2] This intuition is not directly related to the idea that Gödel's first incompleteness theorem implies the falsity of computationalism. Gödel (1951/1995) himself, in his Gibbs lecture (1951), and in correspondence, did make remarks revealing that he believed his incompleteness results, when conjoined with certain plausible propositions, do imply that minds can exceed standard computing machines — but the present paper is concerned with an altogether different question.

- People are neural nets (perhaps with sensors and effectors).
- Cognition is the computation of Turing-computable functions.
- ⋮

As long as the doctrine remains vague, how can we rationally hope to determine the truth-value of computationalism on the basis, at least in part, of formal analysis and reasoning? The only solution is that we must idealize the concepts of minds and machines to provide, as Shapiro (2003) puts it, an "interface" between mathematical results and the doctrine of computationalism. Without such an idealization, we should probably simply close up shop and agree with George Boolos' pessimistic comment that "... it is certainly not obvious what it means to say that the human mind, or even the mind of some human being, *is* a finite machine, e.g., a Turing machine" (Boolos 1995, p. 293).

Fortunately, the idealization isn't hard to come by. In fact, it has already been firmly established on the machine side. Theoretical computer science provides us with a precise conceptual apparatus for making sense of 'computer' and 'computation.' For example, we can identify 'computer' with Turing machine (or, for that matter, with Register machine, abaci, etc.), and 'computation' with Turing machine computation. (And, as we do below, we can turn to logic/mathematics for models of 'hypercomputer' and 'hypercomputation.') This familiar identification allows us to rigorously refer to machines that never run out of memory or working space, and that infallibly obey instructions.

What about persons, or minds? What are we to take them to be? Here, of course, there is no convenient formalism to appeal to. However, it's clear that we must assume, for starters, that persons or minds can be quantified over along with Turing machines, and that they can viewed as entities capable of taking in inputs and returning outputs that reflect problem solving over these inputs. It's also clear from contemporary cognitive science, as we pointed out above, that the mind is *some* sort kind of information-processing device. Textbook after textbook and paper after paper in psychology, cognitive science, and cognitive neuroscience makes this plain as day (e.g., see, in addition to Goldstein's text: Ashcraft 1994, Baron & Kalsher 2001, Stillings, Weisler, Chase, Feinstein, Garfield & Rissland 1995).

Here, then, is how we can encapsulate computationalism so that some careful reasoning can be carried out: Given that p ranges over persons, and m over Turing machines, we simply say:[3]

$\mathcal{C}$ $\forall p \exists m\, p = m$

In addition, we can modify proposition $\mathcal{C}$ so that it carries some complexity information: Letting # be a function which, when applied to a Turing machine, simply returns the number of its states combined with the number of transitions used, we can say that computationalism is committed not only to the view that every person is some Turing machine, but that every person is some Turing machine whose complexity is at or below some threshold k; i.e.,

$\mathcal{C}'$ $\forall p \exists m (p = m \wedge \#(m) \leq k)$

[3]Some may desire to interpret the 'are' in 'People are (standard) TMs' not as the 'are' of '=,' but rather as some concept of "instantiation" (or "realization"), so that people *instantiate* (or *realize*) TMs. This will simply lead to a mere syntactic variant of the argument we give in section 6. This is so in light of the fact that, for all instantiated standard TMs, it is impossible that they hypercompute. (After all, theorems establishing that abstract, mathematically defined information-processing devices are unable to X entail that when these devices are built (i.e., instantiated), they cannot X. For example, we know that no instantiated standard Turing machine can solve the halting problem.)

3 The Essence of Hypercomputation: Harnessing the Infinite

As the papers in this volume make clear, we now understand well that there are information-processing machines that can exceed the Turing Limit (e.g., they can solve the halting problem); such machines just aren't standard TMs and the like (and of course there is ongoing debate as to whether these machines can be artifacts). There are in fact now many such machines in the literature. Indeed, just as there are an infinite number of mathematical devices equivalent to Turing machines (first-order theorem provers, Register machines, the λ-calculus, abaci, ...; many of these are discussed in the context of an attempt to define standard computation in Bringsjord 1994), there are an infinite number of devices beyond the Turing Limit. While Burgin (2001) is correct that the first fleshed-out account of such a machine (*trial-and-error machines*, as Kugel (1986) aptly calls them) appears to have been provided at the same time by Putnam (1965) and Gold (1965), and while we agree with his claim (2001) that his own more powerful *inductive TMs* (ITMs), introduced in 1983, have the distinct advantage of giving results in finite time, today's *infinite time TMs* (ITTMs) (Hamkins & Lewis 2000) can trace their lineage back to seminal theoreticians working well before 1965.[4] ITTMs result from extending the operation of TMs to infinite ordinal time, and produce a "supertask theory of computability and decidability" (Hamkins & Lewis 2000, p. 567). This should come as no surprise, because supertasks have long been discussed in theoretical treatments of computation. Intuitive supertask machines are called *Zeus machines*[5] (ZMs) by Boolos & Jeffrey (1989).[6] The simple point we want to make here is just that the mark of hypercomputers is that they find a way to somehow harness the power of the infinite. This should be completely uncontroversial, and this brute fact is something that Gödel clearly anticipated. Nothing, by the way, makes the "harnessing the infinite" mark of hypercomputation clearer than the aforementioned Zeus machines.

Zeus machines are based on the character Zeus, described by Boolos & Jeffrey (1989). Zeus is a superhuman creature who can enumerate $\mathbf{N}$, the natural numbers, *in a finite amount of time*, in one second, in fact. He pulls this off by giving the first entry, 0, in $\frac{1}{2}$ second, the second entry, 1, in $\frac{1}{4}$ second, the third entry in $\frac{1}{8}$ second, the fourth in $\frac{1}{16}$ second, ..., so that, when a second is done he has completely enumerated the natural numbers. Obviously, it's easy to adapt this scheme so as to produce a Zeus machine that can solve the halting problem: just imagine a machine which, when simulating an arbitrary Turing machine m operating on input u, does each step faster and faster ... (There are countably many Turing machines, and those that don't halt are trapped in

[4] E.g., in 1927 Hermann Weyl considered a machine able to complete

> an infinite sequence of distinct acts of decision within a finite time; say, by supplying the first result after $\frac{1}{2}$ minute, the second after another $\frac{1}{4}$ minute, the third $\frac{1}{8}$ minute later than the second, etc. In this way it would be possible ... to achieve a traversal of all natural numbers and thereby a sure yes-or-no decision regarding any existential question about natural numbers. (Weyl 1949, p. 42)

Actually, Bertrand Russell seems to be have been the first to grasp the essence of ITTMs/Zeus machines. In a lecture in Boston in 1914 he said about Zeno's paradox involving the race-course: "If half the course takes half a minute, and the next quarter takes a quarter of a minute, and so on, the whole course will take a minute" (Russell 1915, pp. 172–173). And later, when lampooning finitism as championed by Ambrose, Russell wrote:

> Ambrose says it is *logically* impossible [for a man] to run through the whole expansion of π. I should have said it was *medically* impossible. ... 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, pp. 143–144)

[5] Sometimes also called 'Zeno' machines, especially in the past.

[6] We thus leave aside discussion of other hypercomputational machines, such as: *analog chaotic neural nets* (Siegelmann & Sontag 1994), artificial neural nets allowed to have irrational numbers for coefficients; and *dial machines* (Bringsjord 2001); etc.

an unending sequence of the same cardinality as **N**.) If, during this simulation, the Zeus machine finds that m halts on u, then a 1 is returned; otherwise 0 is given.

It would take space we don't have, but would be exceptionally easy, to show, for each formalized hypercomputing paradigm, that it exploits the infinite in some manner. For those coming from the perspective of logic, rather than computation, the same point can be made about logics: Those logics corresponding to hypercomputation are *infinitary* logics. The simplest such logic is $\mathcal{L}_{\omega_1\omega}$.[7]

4 Gödel on Minds Exceeding (Turing) Machines by "Converging to Infinity"

As a significant number of readers will know, there is a rather extensive literature devoted to considering whether or not $\mathcal{C}/\mathcal{C}'$ fails in light of Gödel's first incompleteness theorem (full references are provided in Bringsjord & Arkoudas 2004). Thinkers central to this debate, as is widely known, include Lucas (1964, 1996) and, more recently, Penrose (Penrose 1989, Penrose 1994). This literature includes some of Gödel's own thoughts on the matter: Gödel (1951/1995) himself, in his Gibbs lecture (1951), and in correspondence, indicated that by his lights his incompleteness results do imply, when accompanied by some reasonable propositions, that minds can exceed standard computing machines. But the important point at the moment is simply this: the present paper is *not* concerned with these notions.[8] We are concerned, specifically, with Gödel's contention that the human mind can unendingly generate new ideas, techniques, representation schemes, and so on, and can in virtue of these inventions enter mental states that, in number, converge to infinity. Indeed, Gödel seems to have been of the opinion that the human mind *in fact* enters an infinite number of distinct states. Turing explicitly argued for the view that the human mind is capable of only a finite number of distinct states (see pp. 92–93 of Wang 1974), and though Gödel rejected this argument because it presupposed a materialist conception of mind (Turing assumed that the mind equals brain; Gödel wrote and said on many occasions that while the brain is fundamentally a finite digital computer, the mind includes non-physical powers and parts), he also specifically wrote that "there is no reason why the number of states of the mind should not converge to infinity in the course of its development" (Wang 1974, pp. 325–326). Here's the full quote:

> It would be a result of great interest to prove that the shortest decision procedure requires a long time to decide comparatively short propositions. More specifically, it may be possible to prove: For every decidable system and every decision procedure for it, there exists some formula of length less than 200 whose shortest proof is longer than 10^{20}. Such a result would actually mean that machines can't replace the human mind, which can give short proofs by giving a new idea. (Wang 1995, p. 187)

[7] The basic idea behind $\mathcal{L}_{\omega_1\omega}$ is straightforward. This system allows for infinite disjunctions and conjunctions, and hence allows for infinitely long derivations, where these disjunctions, conjunctions, and proofs are no longer than the size of the set of natural numbers. (We use ω to denote the "size" of the set of natural numbers: the niceties of cardinal numbers needn't detain us here.) Here is one simple formula in $\mathcal{L}_{\omega_1\omega}$ which is such that any interpretation that satifies it is finite (something that can't be expressed in ordinary first-order logic):

$$\bigvee_{n<\omega} \exists x_1 \ldots \exists x_n \forall y (y = x_1 \vee \ldots \vee y = x_n).$$

It is a well-known fact that the proposition captured by this formula cannot be captured by a formula in a system at or below Turing machines.

[8] More generally, this paper isn't concerned with any number of the many other points of intersection between computation, hypercomputation, and the mind — a cluster that Stannett (forthcoming), in the present issue of this journal, does a very nice job summarizing.

How are we to understand this? The basic idea would appear to be that as human minds develop through time over generations, they invent new concepts and techniques, which in turn allow previously resistant problems to be solved. There seems to be no upward bound whatsoever to this ascension. Gödel's examples in support of this view were invariably from mathematical logic. As Wang reports: "He appeals to our forming stronger and stronger axioms of infinity in set theory, and of defining computable well-orderings of integers that represent larger and larger ordinal numbers" (Wang 1995, p. 184). We're inclined to say that the recent proof of Fermatt's Last Theorem makes for an example Gödel would find congenial. We're also inclined to say that the formal development of hypercomputation is itself a case in point. Unfortunately, these examples are too indeterminate to be directly helpful in the present paper, in no small part because the gradual improvement through time, in and deriving from these examples, is hard to track. What we need is a problem, and corresponding intellectual toil, that conforms to a demonstrably incremental climb onwards and upwards. The busy beaver, or Σ, problem fits the bill perfectly, and to it we now turn.

5 Setting the Context: The Busy Beaver Problem

5.1 The Problem Defined

The "busy beaver" or Σ function is a mapping from $\mathbf{N}$ to $\mathbf{N}$ such that: $\Sigma(n)$ is the largest number of contiguous 1's that an n-state Turing machine with alphabet {0, 1} can write on its initially blank tape, just before halting with its read/write head on the leftmost 1, where a sequence

$$\overbrace{11\ldots11}^{m \text{ times}}$$

is regarded simply as m.[9] Rado (1963) proved this function to be Turing-uncompuable long ago; a nice contemporary version of the proof (which is by the way not based on diagonalization) is given in (Boolos & Jeffrey 1989). Nonetheless, the busy beaver problem is the challenge of determining $\Sigma(n)$ for ever larger values of n.

Of course, there are varying definitions of Turing machines. The formalism we prefer is the quadruple transition, implicit halt state one. In this scheme, each transition consists of four things: the state the machine is in, the symbol it's scanning, the action it is to perform (move right or left, or write a symbol), and the new state it is to enter. To make it clear that this is the specific scheme we're talking about, we write Σ^B, where the B is in honor of the excellent presentation in (Boolos & Jeffrey 1989).

5.2 RAIR Lab Approach Hitherto; Records

With assistance provided by the United States' National Science Foundation, and in an effort whose software engineering has been led by Kellett, our laboratory has been successful at marching upwards in a fashion Gödel apparently envisioned. While a number of new world records have been set in TM models other than B, we give here only the results for this scheme (see Table 1, which we now proceed to explain).

Consider for a moment the approach that must be used to establish the results and records for $\Sigma^B(n)$ defined in Table 1. Inspired by the impressive effort reported in (Lin & Rado 1964, Brady

[9] As we explain below, there are a number of variations in the exact format for the function. E.g., one can drop the conditions that the output 1's be contiguous.

1983, Machlin & Stout 1990, Marxen & Buntrock 1990), we have defined the following algorithm as a solution to $\Sigma^B(n)$:

1. Using a tree normalized approach,[10] enumerate a set S of n-state Turing machines that behaviorally represents the entire set.
2. For each member t of S:
   (a) Classify t as either a non-halter, or a halter
   (b) If t is a halter, run t to completion on an infinite bi-directional tape of all 0's. If the resulting tape satisfies the conditions specified in section 5.1, add t to our candidate set C.
3. Return the productivity of the most productive machine in C (i.e. the machine that produces the most contiguous 1's on the tape)

Clearly, the task in item $2a$ is menacing. While we can't hope to define a computational algorithm to compute this task (as this would be a solution to the halting problem below the Turing limit), we can still strive to prove that a particular machine does not halt on an individual basis by demonstrating that its behavior follows an infinite, repeatable pattern.

This technique, pioneered in (Lin & Rado 1964), and expanded upon in (Brady 1983, Machlin & Stout 1990), had previously produced a number of proven non-halter detection strategies and behaviors. Each of these behaviors encompass some subset of non-halting Turing machines within the scope of all non-halting Turing machines relevant to $\Sigma^B(n)$. The classifications range from Turing machines that can be proven non-halters by working backwards from their required halting conditions, to simple recurrence patterns, to even machines that mimic an abstraction of binary counters.

As a case in point, let us examine one of these patterns: so-called "Christmas Trees," discussed in (Brady 1983, Machlin & Stout 1990). For purposes of this discussion, consider the following notation used to represent the contents of a tape:

$$0^*[U][X][X][X][V_s]0^*.$$

Here 0^* denotes an infinite sequence of 0's that caps each end of the tape. Additionally, $[U]$, $[X]$, and $[V]$ represent some arbitrary sequence of characters on the tape while the subscripted s indicates that the machine is in state s with its read/write head at the leftmost character of the $[V]$ character sequence.

With this in mind, we can describe the Christmas Tree pattern in the context of the transformations that it makes on the tape. Machines exhibiting this behavior are classified as non-halters due to a repeatable back and forth sweeping motion which they exhibit on the tape. Observably, the pattern of the most basic form of Christmas Trees is quite easy to recognize. The machine establishes two end components on the tape and one middle component. As the read/write head sweeps back and forth across the tape, additional copies of the middle component are inserted, while maintaining the integrity of the end components at the end of each sweep.

Figure 1 displays a partial execution of a 4-state Christmas Tree machine which is representative of one sweep back and forth across the tape. As can be seen, at the beginning of this execution, the machine has established three middle or $[X]$ components capped by the $[U]$ and $[V]$ end components

[10] Our enumeration technique utilizes specific tree normalization filters adapted from the aforementioned works. The machines are enumerated using a tree-based approach and the filters specify conditions which allow us to prune machines and even entire subtrees from the overall tree, because they are proven behaviorally equivalent to either some other machine already in the tree, or, one that will be generated. Further discussion is beyond the scope of this paper, but for a complete description of the specific filters used, please refer to http://www.cs.rpi.edu/~kelleo.busybeaver.

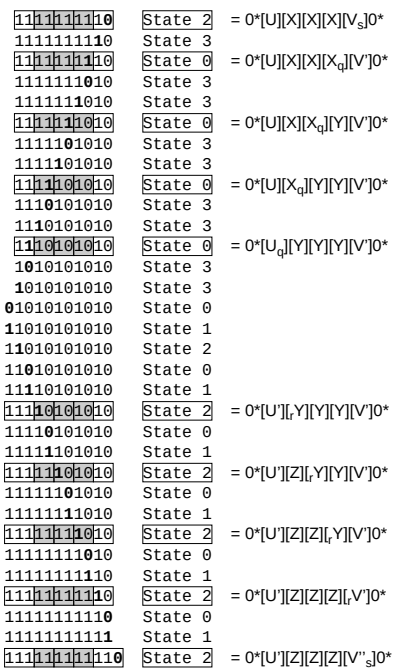

Figure 1: Christmas Tree execution

on each side. As the read/write head sweeps back and forth across the tape, it methodically converts the $[X]$ components into $[Y]$ components and then into $[Z]$ components on the return sweep. At the completion of the sweep, the tape is left in the state: $0^*[U'][Z][Z][Z][V_s'']0^*$ which can be shown to be equivalent to $0^*[U][X][X][X][V_s]0^*$. Thus each successive sweep across the tape performs similar transformations, adding an additional $[X]$ component in an infinite pattern.[11]

In light of these behavioral patterns, which by mathematical induction can be proved to guarantee non-haltingness, we can now modify our basic algorithm designed to solve $\Sigma^B(n)$:

1. Using a tree normalized approach, enumerate a set S of n-state Turing machines that behaviorally represents the entire set.
2. For each member t of S:
   (a) Attempt to show that t exhibits one of the defined non-halting behaviors
   (b) If the attempt fails, run t to a predetermined step limit on an infinite bi-directional tape of all 0's.
   (c) If t halts before or at this step limit, add it to our candidate set C if the resulting tape satisfies the conditions specified in section 5.1.
   (d) If it has not halted after the step limit has been reached, add it to our holdout set H.
3. Examine each machine in H and define new non-halting behaviors and detection techniques. Include these new behaviors in the next successive iteration of the algorithm until the size of H is reduced to 0.

[11] We don't include the specifics of the transformations that can be explicitly shown to be in an infinite pattern, since the Christmas Tree pattern is only one of several non-halting behaviors that have been defined. We again refer the reader to for a complete discussion on each of the non-halting behaviors that have been developed, including our own.

Category	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Standard Halt	6	80	2264	103095	6640133
Non-standard Halt	2	76	3844	271497	24911677
Back-tracker	23	865	49481	4008364	403910416
Subset loop	0	0	146	11013	2582783
Simple loop	5	130	5605	381736	48492276
Christmas tree	0	2	156	13987	2166668
Leaning Christmas Tree	0	0	0	69	23129
2-sweep Christmas Tree	0	0	23	2356	419598
3-Sweep Christmas Tree	0	0	0	470	77740
4-Sweep Christmas Tree	0	0	0	76	17345
5-Sweep Christmas Tree	0	0	0	0	2156
6-Sweep Christmas Tree	0	0	0	0	1352
7-Sweep Christmas Tree	0	0	0	0	345
8-Sweep Christmas Tree	0	0	0	0	65
Counter	0	0	0	113	25678
Holdout	**0**	**0**	**0**	**98**	**42166**
Total	36	1153	61519	4792874	489313527

Table 1: Distribution of Tree Normalized Machines for $\Sigma^B(n)$

4. Return the productivity of the most productive machine in C (i.e., the machine that produces the most contiguous 1's on the tape)

Utilizing this strategy, our laboratory has been able to extend the work done in (Lin & Rado 1964, Brady 1983, Machlin & Stout 1990) by defining several additional non-halt behaviors to add to the already established set. (These are candidates for what Gödel calls "new ideas.") These include behaviors such as multi-sweep Christmas Trees, leaning Christmas Trees, and modified binary counters. With the inclusion of these additional behaviors, we have explicitly confirmed the values of $\Sigma^B(n)$ up through $n = 4$ by classifying each of the machines in the relevant set S as either a halter, or a non-halter displaying one of the non-halting behaviors. For $n = 5$ and $n = 6$, the number of holdouts have been reduced to 98 and 42166, respectively. Soon these will all be classified as halters or non-halters; for if need be, because the number of machines is so small, we can resort to manual machine-by-machine examination. Our results are summarized in Table 1.

At the present time, additional non-halt behaviors are being defined based on the holdouts for $\Sigma^B(5)$ and $\Sigma^B(6)$. In fact, each of the 98 holdouts for $n = 5$ has been individually manually examined and assigned to one of several new non-halt behaviors in development. These include additional counter variations (base-3 counters, alternating counters, resetting counters), several Christmas Tree variations (nested Christmas Trees, uneven Christmas Trees, partial sweep Christmas Trees), as well as combination behaviors (Christmas Trees with nested binary counting behavior).[12] We declare with extreme confidence that all 98 of these holdouts will soon be *automatically* proven non-halters as well: the case of 5, and then 6, will soon enough be determined by — to speak in Gödel's confessedly dramatic terms — the ever ascending human mind.

Thus the search continues. By incorporating the formal behaviors found from the $\Sigma^B(5)$ hold-

[12] A more thorough description of each of these behaviors can again be found on the RAIR Lab's Busy Beaver website.

outs, we hope to use these behaviors to significantly reduce the $\Sigma^B(6)$ holdout set and continue to adapt, combine, and formulate new behaviors for the remaining holdouts. (These new behaviors will certainly be defined in new representation schemes. For an example, see our reply to Objection #3, in section 7.) Considering this, we see no reason why the above algorithm cannot be sustained, and applied to $\Sigma^B(7)$, $\Sigma^B(8)$, and beyond. Specifically, we have every reason to believe that once n is cracked, lessons learned in that solution can be used to bootstrap up to $n+1$.

We turn now to the argument itself.

6 The New Gödelian Argument

First, note that, for generality, we drop the subscript B from Σ^B. Then here we go:

Proof. The argument is at its core a *reductio*. The following constitutes the argument's premises:

- There exists a human person who has determined the productivity of the initial segment of Turing machines (such people are in the RAIR Lab):
  (1) $\exists p(D(p, \Sigma(1)) \wedge \ldots \wedge D(p, \Sigma(6)))$
- There is a natural number at and beyond which Turing machines of size less than or equal to k fail (with respect to determining productivity):
  (2) $\exists n \forall m(\#(m) \leq k \rightarrow \neg D(m, \Sigma(n)) \wedge \neg D(m, \Sigma(n+1)) \wedge \neg D(m, \Sigma(n+2)) \ldots)$
- If a person can determine the productivity for n, he/she can determine it for $n+1$:
  (3) $\forall n \forall p(D(p, \Sigma(n)) \rightarrow D(p, \Sigma(n+1)))$

Now, to generate absurdity, suppose that computationalism holds, i.e., reiterating from above:

$\mathcal{C}'$ $\forall p \exists m(p = m \wedge \#(m) \leq k)$

Next, suppose that $p^\star$ is an arbitrary person who determines the initial segment of the busy beaver problem, i.e.,

(ρ) $(D(p^\star, \Sigma(1)) \wedge \ldots \wedge D(p^\star, \Sigma(6))$

By universal elimination on ($\mathcal{C}'$) we have:

(4) $\exists m(p^\star = m \wedge \#(m) \leq k)$

Next, we make the supposition, with $m^\star$ as arbitrary, that

(5) $(p^\star = m^\star \wedge \#(m^\star) \leq k)$

and likewise the supposition, with $n^\star$ arbitrary, that

(6) $\forall m(\#(m) \leq k \rightarrow \neg D(m, \Sigma(n^\star)) \wedge \neg D(m, \Sigma(n^\star + 1)) \wedge \neg D(m, \Sigma(n^\star + 2) \ldots)$

from which it follows by universal elimination that

(7) $(\#(m^\star) \leq k \rightarrow \neg D(m^\star, \Sigma(n^\star)) \wedge \neg D(m^\star, \Sigma(n^\star + 1)) \wedge \neg D(m^\star, \Sigma(n^\star + 2) \ldots)$

From (5) and (7) we can infer

$$\neg D(m^\star, \Sigma(n^\star)) \wedge \neg D(m^\star, \Sigma(n^\star + 1)) \wedge \neg D(m^\star, \Sigma(n^\star + 2) \ldots)$$

but by identity elimination and induction using (3), (5), and (ρ), we can deduce $\forall n D(m^\star, \Sigma(n))$ — contradiction. It follows that since (i) (as we've repeatedly noted) human persons are (or at least encompass) information processors somewhere in the Arithmetic Hierarchy, and (ii) (as we've also earlier noted) if persons are ordinary Turing machines they have a certain fixed size k, human persons are hypercomputers. QED

We have described this reasoning as a proof, and indeed in one sense it is,[13] but again, as we admitted at the outset, actually it's an *argument* — a provably valid chain of reasoning with plausible, rather than invulnerable, premises.[14] There are two reasons why this is so: (i) the key concept of "determining" is far from fully formal, and (ii) the premises are not unproblematic. We now briefly discuss these two issues in turn.

Obviously, the relation D is central to the argument. We say that $D(x, \Sigma(n))$ abbreviates 'x determines the value of $\Sigma(n)$,' but concede that 'determines' can't be completely specified by a set of necessary and sufficient conditions. However, the concept can be sufficiently characterized for present purposes, as follows. First, note that $D(x, \Sigma(n))$ makes reference to one particular natural number n, the number of states of the particular set of Turing machines in question. (For each n, there is only a finite set of n-state TMs.) D does *not* refer to the overall busy beaver problem, for to solve *that* problem is of course to determine productivity for every natural number. Note that in our case, as a matter of brute fact, we have *determined* the productivity of 6, 5, ..., 1. And notice specifically that in this case computational resources have been used; this should be clear from our earlier summary of our progress on attacking the busy beaver problem (section 5). So determining can include getting help from standard computation — but it must be *standard*: it wouldn't be permissible to consult a hypercomputational device, or an oracle, and it clearly wouldn't suffice to simply guess the answer. In short, to determine a value some genuine problem solving must take place. In addition, accompanying justification, in the form of a proof, must be provided (which is in keeping with Gödel's emphasis on humanity's progress in finding proofs, and with our approach to verifying our progress on the busy beaver problem: see our reply to Objection #3 in section 7).

Though, as we've said, the second reason we have managed to articulate not a proof, but rather a rigorous argument, is that premises are problematic, the trio of propositions in question *is* strong: Premise (1) is an empirical fact. Premise (2) is indisputable, as it's merely a weaker proposition than that the busy beaver function is uncomputable. Premise (3) represents Gödel's intuitive view, fleshed out in connection to the busy beaver problem. The idea is really quite simple: If humans are smart enough to determine $\Sigma^B(n)$, they will eventually (perhaps after 100 years, perhaps after 1000, perhaps after 1,000,000,000, ...) be smart enough to determine $\Sigma^B(n+1)$: they will invent some new technique for economically representing the behavior of $n+1$ machines, and for detecting in that representation when activity will eventually cease, and when it will not: there will remain no holdouts. Though we aren't claiming that premise (3) is unassailable, it seems to us remarkable that one would maintain its denial: that is, that one would maintain that it is *impossible*, however long the analysis takes, to move from success on n to success on $n+1$. Gödel, we strongly suspect, was moved, at bottom, by the same perception of absurdity.[15]

Skeptics will of course remain, but they need to take account of the fact that, while (3) may not be unassailable, a variant would seem to be, viz., that if humans are smart enough to determine $\Sigma^B(n)$, it's then *mathematically possible* that they determine $\Sigma^B(n+1)$. Notice that this doesn't hold with respect to Turing machines: When we say that a Turing machine m of size k, as a matter of proof, must fail to determine the productivity of n and its successors, we are by implication saying that it is mathematically impossible that m determine the productivity of n and its successors.

[13] If given as input to a mechanical proof checker it comes back as verified; or, if the conclusion is negated and given to a standard resolution-based ATP along with the premises, a contradiction is returned.

[14] Let's leave aside the historical fact that some venerated proofs of today (e.g., from Cantor) were rejected when first shared with mathematicians and logicians.

[15] It should also be noted that if anything can count as evidence in favor of (3), it surely must be that, as a matter of empirical fact, our race continues to climb from n to $n+1$. This isn't *ironclad* evidence, but it *is* evidence. Remember, though we know that no standard Turing machine can continue indefinitely, the question on the table is whether persons are such machines themselves — and so it's not acceptable to reject the mounting evidence in the human sphere simply because we know ahead of time that every Turing machine will eventually get stuck.

(Every theorem τ is such that, using the necessity operator of modal logic (Hughes & Cresswell 1968, Chellas 1980), $\Box\tau$; and if some machine m cannot X, then, where the diamond can be read 'possibly,' $\neg\Diamond mXs$.)[16] If one were to develop these ideas, it would require a parallel of the find of careful investigation undertaken in (Bringsjord & Arkoudas 2004); such investigation will have to wait for another paper built atop the present one.

We turn now to setting out and rebutting five objections to our argument.

7 Objections

We set out and rebut five objections in this section.

Objection #1; Reply

The first objection can be expressed as follows: "The third premise in your argument is in the spirit of saying that given enough time, humans can solve anything. This premise is shown to be false by parody, because this assertion resembles the claim that the human record for the 100 meter run will converge to 0 seconds, since we (the human race) achieves a new record with each passing year. The point is that solving the busy beaver problem for input up to 6, as you have done (after much research and development) doesn't tell us much about the limits of the human mind."

In reply, first, note that our argument most assuredly doesn't even *flirt* with the notion that given enough time, humans can solve anything. Problems unsolvable by infinite time Turing machines, for example, are problems that, for all we say in the paper, are unsolvable by human persons, period. The premise in question says only that if the human race gets to n in the Σ problem, it can get to $n+1$. The formal reasoning over this premise that is part and parcel of our argument would in fact be obviated if we made the naive, blanket claim this objection ascribes to us.

Second, there is here an acute disanalogy, since we know that it's physically impossible that a human (under standard conditions: e.g., the human's body can't be replaced in significant part by a robotic one, etc.) run a 100 meter race in, say, 1 second (let alone zero!). But we *don't* know that it's impossible for a human being to always be able to move to the next increment in the Σ challenge. What we *do* know is that no computing machine at the Turing Limit or below can always move up one increment $n+1$ from success on a prior n.

Third, Machlin and Stout (1990) point out that Rado himself pronounced, at a symposium in the same year that his proof of the uncomputabiliy of the arbitrary case was published, that 4-state machines were "entirely hopeless." Given that we have now passed beyond the hopeless, and see further progress on the horizon, the busy beaver problem is in point of fact telling us that our race's reach is beyond where some rather smart people figured it would stop.

Objection #2; Reply

The second objection runs as follows: "The usual proof of the uncomputability of the busy beaver function shows that any program of size j must fail to compute the right value for some αj, where

[16] The point here is not the same as saying that there is no proof that human mental power coincides with processing at and below the level of standard Turing machines. The point is simply that when a standard Turing machine is unable, as a matter of proof, to X, then we can say, using the box operator of modal logic (assuming, say, the system KT45), it's logically necessarily the case that (= $\Box$) that TM can't X. Things appear to be quite different in the case of human persons: it seems odd to maintain that it's logically necessary that persons can't X, where X relates to some Turing-uncomputable task. After all, where is the contradiction that can be derived from supposing that humans can X? (If X-ing is logically impossible, then a contradiction must be derivable from supposing that X-ing happens.

α is a very small constant, say 8. In other words, any program of size j can only work for inputs $< 8j$. Now, the number of possible human brain states may be something like $2^{10^{10}}$. So the theorem in question only implies that humans will get stuck billions of orders of magnitude past were current research on computing the function stands."

This counter-argument is ambiguous between two possible interpretations, both of which are fallacious. One possibility is this reasoning:

Interpretation 1

(1) Any program (at or below the Turing Limit) of size j can only determine the Σ value of αj and less, for some constant α.
(2) The human brain is of size j.
$\therefore$ (3) Persons are unable to determine the Σ value of any number greater than αj.

This argument is formally invalid, as can be immediately seen when it's represented in elementary logic. The fact is, proposition (3) doesn't follow from (1) and (2). Interpretation 2 corresponds to the argument resulting from the addition of two key premises, viz.,

(4) The human brain is a program at or below the Turing Limit.

(5) Persons are brains.

Now, it must be agreed that $\{(1), (2), (4), (5)\} \vdash (3)$; this is Interpretation 2. But we nonetheless still have fallacious reasoning: we have *petitio principii*, for the simple reason that the (4), (5) pair is precisely what's at issue. Any counter-argument that takes as premises this pair is circular reasoning, since, again, the purpose of the present inquiry is to see if we can provide a rigorous case for Gödel's rejection of this conjoined pair, in favor of the view that persons operate beyond the Turing Limit.[17]

Objection #3; Reply

The third objection: "There is a hidden assumption behind the reasoning you put forth: that once we've recorded an answer for $\Sigma(m)$, for a certain input m (5, say), we will never change it. History has shown this assumption to be perilous, because humans are far from immune to errors."

There are two reasons why this objection fails. First, it seems to imply that, in general, humans can't trust the output given them by ordinary computing machines. But why should work on the busy beaver problems be held to standards higher than those in play in other domains? Every time we use a calculator to do anything more intricate than what we can do mentally we are trusting that the computation in question is correct. The second reason why the third objection fails is that our research on the busy beaver can be recast to fall within the declarative approach, which produces proofs for every result recorded. In this appproach, we first re-represent Turing machines in declarative fashion, using one of the standard routes toward proving the undecidability of first-order logic (Boolos & Jeffrey 1989)e.g., see the undecidability proof in. This produces, for each TM m, a set of formulae $\Delta \cup \phi_h$ such that $\Delta^m \vdash \phi_h^m$ iff m halts (started on the 0-filled tape). We then add formulae corresponding to repetitive behavior. E.g., using the denotational proof language Athena (Arkoudas n.d.) we can say:

[17] Either because human brains are information processors operating above the Turing Limit ($\neg(4)$), or because persons are not brains ($\neg(5)$).

```
Can the machine enter a state in which it will read
what's under the read head, write the same symbol, stay
in the same place, and transition to the same state?

(define WillLoop
  (exists ?t ?q ?x ?c
    (and (Transition ?q ?x ?x Stay ?q)
         (InState ?t ?q)
         (ReadingCell ?t ?c)
         (Contents ?t ?c ?x))))
```

Using Athena's proof-search constructs, we then write theorem-proving procedures that utilize state-of-the-art systems such as Vampire (Voronkov 1995) in order to analyze the Turing machines in question. A representation of the problem in logic immediately raises the level of trust we are justified in having in the results obtained with help from a computer (Arkoudas & Rinard 2004).[18] In fact, the idea is to record a value for $\Sigma(m)$ only if that value is accompanied by a (machine- *and* human-) checkable proof that that value is in fact correct.

Objection #4; Reply

The fourth objection: "There is some confusion of a specific person with all of humanity, past, present, and future. The relevant claim in your argument, premise (3), refers to individual persons, since you're quantifying over the set of persons (as a sort), and yet in your further justification after presenting that argument, you refer to humanity ('If humans ... will eventually ... after 1,000,000,000 [years] ...')."

We do indeed shift freely between individual persons, and humanity, but this is harmless, and accords with what Gödel apparently had in mind. It is harmless because just as there are no restrictions on time and energy when we consider what a standard Turing machine can compute (a nice discussion of this well-known fact, which enlivens computability theory, is provided in Boolos & Jeffrey 1989), there are likewise no restrictions with respect to time and energy when we're talking, in the present context, about a particular person.[19] So we could talk of one idealized, immortal person "converging to infinity" in cognitive power, or — following Gödel directly — the race itself "converging to infinity" as generation after generation ascends in mental power. Either way, premise (3) suitably encapsulates the basic idea that, so to speak, each rung on the ladder can be reached from the one just beneath it.[20] In addition, we are free to ignore the fact that some persons alive today have insufficient cognitive power to even grasp the Σ function. We can ignore this fact, again, because we are speaking of persons in the ideal case. Put concretely, quantification in the

[18] This approach also raises the possibility of automating the inductive proofs that show some particular repetitive behavior does in fact guarantee non-haltingness (these proofs are currently done by hand). We are currently considering this possibility.

[19] In addition, just as in computability theory we consider machines working on and on without being impeded by natural events, so too we are considering people who work on and on, despite the fact that we know it's physically possible that the entire race become extinct (because of some cataclysmic event).

[20] The generation-to-generation approach does assume that there is a way to manage the ongoing project, and that that way is itself computable. But there are myriad existence proofs of the fact that such multi-generation management can be pulled off, and is straightforwardly mechanical. Lying closest to home is the existence proof based on the busy beaver work carried out thus far by *homo sapiens*, since here we are as a group in 2005 working as a natural extension of minds that have unfortunately passed away. And then, further afield, there are myriad confirming cases. E.g., NASA routinely runs projects that provide confirmation, as in, at present, a manned mission to Mars.

argument over persons is thus quantification over those persons able to fully grasp the challenge, and attack it with full cognitive power. This is really standard practice, when you think about it. For example, when we say that humankind was smart enough to put a man on the Moon, we are fully aware of the fact that, actually, only a relatively small percentage of humans were cognitively equipped (because of the right training, the right nutrition, genes, and so on) to be members of a team able to pull this feat off.

Objection #5; Reply

The fifth and final objection is expressed as follows: "There is in your case against computationalism a hidden assumption that people are unaffected by nature. But one can affirm computationalism ($\mathcal{C}/\mathcal{C}'$), while still allowing humans to solve problems beyond the reach of standard Turing machines — because they can solve such problems by virtue of external hypercomputational effects. After all, historically, many scientific breakthroughs happen serendipitously."

It may well be that hypercomputation takes place in nature (for a defense see Cleland 1993), but surely we don't yet know how to *harness* such naturally occurring phenomena (Bringsjord & Zenzen 2002). In light of this situation, it would indeed have to be serendipitous if naturally occurring hypercomputation affects some human project in a favorable manner. However, luck is not part of problem solving; this can be seen by consulting accounts of problem solving viewed as an area within cognitive psychology (e.g., see Ashcraft 1994, Goldstein 2005), or accounts of problem solving in the form of means-ends (or goal) analysis popular in logic and artificial intelligence (e.g., see the traditional account of goal analysis in Bergmann, Moor & Nelson 1997). Reflective of the fact that problem solving includes only what the agent in question can receive at least some credit for, the key relation D in our argument, as we have said earlier, excludes luck.

8 Conclusion

Our purpose herein was to articulate a new, credible case against $\mathcal{C}'$ and $\mathcal{C}$, and hence for the view that human persons hypercompute, on the basis of Gödel's intuition regarding the ever-ascending human mind. We didn't take on the onus of *proving* that Gödel is right. So, not everyone will be persuaded by the case we have presented. But one thing should be indisputable even at this point in the dialectic we hope to have started: With the advent of hypercomputation, and an increase in the level of our race's understanding of this mode of information processing beyond the Turing Limit, there arises the great and irresistable question as to whether or not, as information processors ourselves, we are above this limit.

References

Arkoudas, K. (n.d.), Athena. http://www.cag.csail.mit.edu/~kostas/dpls/athena.

Arkoudas, K. & Rinard, M. (2004), Deductive runtime certification, *in* 'Proceedings of the 2004 Workshop on Runtime Verification', Barcelona, Spain, pp. 39–56.

Ashcraft, M. (1994), *Human Memory and Cognition*, HarperCollins, New York, NY.

Baron, R. & Kalsher, M. (2001), *Psychology (Fifth Edition)*, Allyn and Bacon, Boston, MA.

Barr, A. (1983), Artificial intelligence: Cognition as computation, *in* F. Machlup, ed., 'The Study of Information: Interdisciplinary Messages', Wiley-Interscience, New York, NY, pp. 237–262.

Bergmann, M., Moor, J. & Nelson, J. (1997), *The Logic Book*, McGraw Hill, New York, NY.

Boolos, G. (1995), Introductory note to: "Some basic theorems on the foundations of mathematics and their implications", *in* 'Collected Works III', Oxford University Press, Oxford, UK, pp. 290–304.

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

Brady, A. (1983), 'The determination of the value of rado's noncomputable function $\Sigma(k)$ for four-state turing machines', *Mathematics of Computation* **40**(162), 647–665.

Bringsjord, S. (1992), *What Robots Can and Can't Be*, Kluwer, Dordrecht, The Netherlands.

Bringsjord, S. (1994), 'Computation, among other things, is beneath us', *Minds and Machines* **4.4**, 469–488.

Bringsjord, S. (2001), 'In computation, parallel is nothing, physical everything', *Minds and Machines* **11**, 95–99.

Bringsjord, S. & Arkoudas, K. (2004), 'The modal argument for hypercomputing minds', *Theoretical Computer Science* **317**, 167–190.

Bringsjord, S. & Zenzen, M. (2002), 'Toward a formal philosophy of hypercomputation', *Minds and Machines* **12**, 241–258.

Bringsjord, S. & Zenzen, M. (2003), *Superminds: People Harness Hypercomputation, and More*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Burgin, M. (2001), 'How we know what technology can do', *Communications of the ACM* **44**(11), 83–88.

Chellas, B. F. (1980), *Modal Logic: An Introduction*, Cambridge University Press, Cambridge, UK.

Cleland, C. (1993), 'Is the Church-thesis true?', *Minds and Machines* **3**, 283–312.

Dietrich, E. (1990), 'Computationalism', *Social Epistemology* **4**(2), 135–154.

Fetzer, J. (1994), 'Mental algorithms: Are minds computational systems?', *Pragmatics and Cognition* **2.1**, 1–29.

Gödel, K. (1951/1995), Some basic theorems on the foundations of mathematics and their implications, *in* 'Collected Works III', Oxford University Press, Oxford, UK, pp. 304–323.

Gold, M. (1965), 'Limiting recursion', *Journal of Symbolic Logic* **30**(1), 28–47.

Goldstein, E. B. (2005), *Cognitive Psychology: Connecting Mind, Research, and Everyday Experience*, Wadsworth, Belmont, CA.

Hamkins, J. D. & Lewis, A. (2000), 'Infinite time Turing machines', *Journal of Symbolic Logic* **65**(2), 567–604.

Harnad, S. (1991), 'Other bodies, other minds: A machine incarnation of an old philosophical problem', *Minds and Machines* **1**(1), 43–54.

Haugeland, J. (1985), *Artificial Intelligence: The Very Idea*, MIT Press, Cambridge, MA.

Hofstadter, D. (1985), Waking up from the Boolean dream, *in* 'Metamagical Themas: Questing for the Essence of Mind and Pattern', Bantam, New York, NY, pp. 631–665.

Hughes, G. & Cresswell, M. (1968), *An Introduction to Modal Logic*, Methuen, London, UK.

Johnson-Laird, P. (1988), *The Computer and the Mind*, Harvard University Press, Cambridge, MA.

Kugel, P. (1986), 'Thinking may be more than computing', *Cognition* **18**, 128–149.

Lin, S. & Rado, T. (1964), 'Computer studies of turing machine problems', *Journal of the Association for Computing Machinery* **12**(2), 196–212.

Lucas, J. R. (1964), Minds, machines, and Gödel, *in* A. R. Anderson, ed., 'Minds and Machines', Prentice-Hall, Englewood Cliffs, NJ, pp. 43–59. Lucas' paper is available online at http://users.ox.ac.uk/∼jrlucas/mmg.html.

Lucas, J. R. (1996), Minds, machines, and Gödel: A retrospect, *in* P. Millican & A. Clark, eds, 'Machines and Thought: The Legacy of Alan Turing, Volume I', Oxford University Press, Oxford, UK, pp. 103–124.

Machlin, R. & Stout, Q. (1990), 'The complex behavior of simple machines', *Physica D* **42**, 85–98.

Marxen, H. & Buntrock, J. (1990), 'Attacking the busy beaver 5', *Bulletin of the European Association for Theoretical Computer Science* **40**, 247–251.

Newell, A. (1980), 'Physical symbol systems', *Cognitive Science* **4**, 135–183.

Penrose, R. (1989), *The Emperor's New Mind*, Oxford, Oxford, UK.

Penrose, R. (1994), *Shadows of the Mind*, Oxford, Oxford, UK.

Peters, R. S., ed. (1962), *Body, Man, and Citizen: Selections from Hobbes' Writing*, Collier, New York, NY.

Putnam, H. (1965), 'Trial and error predicates and a solution to a problem of mostowski', *Journal of Symbolic Logic* **30**(1), 49–57.

Rado, T. (1963), 'On non-computable functions', *Bell System Technical Journal* **41**, 877–884.

Russell, B. (1915), *Our Knowledge of the External World as a Field for Scientific Method in Philosophy*, Open Court, Chicago, IL.

Russell, B. (1936), 'The limits of empiricism', *Proceedings of the Aristotelian Society* **36**, 131–150.

Searle, J. (1980), 'Minds, brains and programs', *Behavioral and Brain Sciences* **3**, 417–424. This paper is available online at http://members.aol.com/NeoNoetics/MindsBrainsPrograms.html.

Shapiro, S. (2003), 'Mechanism, truth, and Penrose's new argument', *Journal of Philosophical Logic* **32**(1), 19–42.

Siegelmann, H. & Sontag, E. (1994), 'Analog computation via neural nets', *Theoretical Computer Science* **131**, 331–360.

Simon, H. (1980), 'Cognitive science: The newest science of the artificial', *Cognitive Science* **4**, 33–56.

Simon, H. (1981), 'Study of human intelligence by creating artificial intelligence', *American Scientist* **69**(3), 300–309.

Stannett, M. (forthcoming), 'The case for hypercomputation', *Applied Mathematics and Computation* .

Stillings, N., Weisler, S., Chase, C., Feinstein, M., Garfield, J. & Rissland, E. (1995), *Cognitive Science*, MIT Press, Cambridge, MA.

Voronkov, A. (1995), 'The anatomy of vampire: Implementing bottom-up procedures with code trees', *Journal of Automated Reasoning* **15**(2).

Wang, H. (1974), *From Mathematics to Philosophy*, Keagan Paul, London, UK.

Wang, H. (1995), On computabilism' and physicalism: Some sub-problems, *in* J. Cornwell, ed., 'Nature's Imagination: The Frontiers of Scientific Vision', Oxford University Press, Oxford, UK, pp. 161–189.

Weyl, H. (1949), *Philosophy of Mathematics and Natural Science*, Princeton University Press, Princeton, NJ.