Selmer Bringsjord, Konstantine Arkoudas*

On the Provability, Veracity, and AI-Relevance of the Church-Turing Thesis

After providing some brief background on the Church-Turing thesis, we discuss MENDELSON's much-publicized change of heart regarding the provability of that thesis, and defend the standard conception of the thesis as a mathematically unprovable proposition. Next, the first author (BRINGSJORD) offers an argument that aims to establish the outright falsity of the thesis. The argument is controverted by the second author (ARKOUDAS), who puts forth several objections against it. BRINGSJORD replies to these objections, and to a number of other potential objections, and proceeds to consider previous attacks on the Church-Turing thesis and compare them with his own. The final section of the paper is devoted to an examination of arguments for the computational conception of mind from the Church-Turing thesis. We distinguish between strong and weak forms of computationalism, and we analyze an argument for weak computationalism on the basis of the Church-Turing thesis, which we show, *contra* COPELAND, to be formally valid. Accordingly, *if* the thesis is true, then, by virtue of this argument, weak computationalism is validated.

*S. BRINGSJORD, K. ARKOUDAS, Department of Cognitive Science, Department of Computer Science, Rensselaer AI & Reasoning (RAIR) Lab, Rensselaer Polytechnic Institute, Troy NY 12180 USA, selmer@rpi.edu, arkouk@rpi.edu, <http://www.rpi.edu/~brings>.

1. Background

At the heart of the Church-Turing thesis (CTT from now on) is the notion of an *algorithm*, characterized in traditional fashion by MENDELSON as

> an effective and completely specified procedure for solving a whole class of problems. [...] An algorithm does not require ingenuity; its application is prescribed in advance and does not depend upon any empirical or random factors. [MENDELSON 1990, p. 225]

A function $f: A \rightarrow B$ is then called *effectively computable* iff there exists an algorithm that an idealized computing agent can follow in order to compute the value $f(a)$ for any given $a \in A$.[1] Without loss of generality, we can restrict attention to so-called *number-theoretic functions*, i.e., functions that take N to N (where N is the set of natural numbers). Briefly, the justification for this restriction is a technique known as *arithmetization*. Using ideas made popular by GÖDEL, one can devise encoding and decoding algorithms that will represent any finite mathematical structure (e.g., a graph, a context-free grammar, a formula of second-order logic, a Java program, etc.) by a unique natural number. By using such a scheme, a function from, say, Java programs to graphs, can be faithfully (and *effectively*) represented by a function from N to N. Similar techniques can be used to represent a function of multiple arguments by a single-argument function.

The notion of an effectively computable function is informal, since it is based on the somewhat vague concept of an algorithm. CTT also involves a more formal notion, that of a *Turing-computable* function. A (total) function $f: N \rightarrow N$ is Turing-computable iff there exists a Turing machine which, starting with n on its tape (perhaps represented by n |s), leaves $f(n)$ on its tape after processing, for any $n \in N$. (The details of the processing are harmlessly left aside for now; see, e.g., [LEWIS and PAPADIMITRIOU 1997] for a thorough development.) Given this definition, CTT amounts to:

CTT A function $f: N \rightarrow N$ is effectively computable if and only if it is Turing-computable.

[1] TURING [1936] spoke of "computists" and POST [1934] of "workers," humans whose sole job was to slavishly follow explicit, excruciatingly simple instructions.

The term "Church's thesis" will be used synonymously with "the Church-Turing thesis," and CT will serve as an abbreviation for the former.

2. Mendelson and the Provability of Church's Thesis

In a widely circulated and debated essay, [MENDELSON 1990] reversed his earlier position [MENDELSON 1963] on the provability of Church's thesis and went on to argue against the standard conception of the thesis as mathematically unprovable. He contended that it is wrong to say that the thesis is not mathematically provable "just because it states an equivalence between a vague, imprecise notion (effectively computable function) and a precise mathematical notion (partial-recursive function)," and gave what he considers to be a perfectly precise mathematical proof of the easier half of the thesis. The issue clearly boils down to what counts as a mathematical proof, and indeed the bone of contention here can be traced to different understandings of that concept.

The question of what exactly ought to count as a mathematical proof is somewhat controversial; there have been widely differing opinions. Some thinkers in the "quasi-empirical" tradition (a tradition that is largely associated with LAKATOS [1976], and which emphasizes the fallibility of mathematics and the impossibility of securing foundations for it) view mathematical proofs as informal arguments (LAKATOS called proofs "thought experiments"), and hold, as KITCHER [1977] puts it, that:

> The mistake is to regard proofs as instruments of justification. Instead, we should see them as tools of discovery, to be employed in the development of mathematical concepts and the refinement of mathematical conjectures.

Anti-foundationalists in general, particularly those who are sympathetic to social constructivism (e.g., [ERNEST 1998]), hold that mathematical proofs are essentially social constructs, inextricably tied to a particular time and culture. According to such views, a mathematical proof is any argument—indeed, any *thing*, e.g., a diagram—that can convince fellow mathematicians that a certain claim is true. Rigor, they claim, particularly of the formal axiomatic variety, is not a realistic model of mathematical discovery and can actually stifle innovation. "Too much rigor can lead to rigor mortis," as KLEINER

[1991] puts it. On the other end of the spectrum, we have formalists of various degrees, in the tradition of FREGE, RUSSELL, and HILBERT, who view proofs as they are treated in proof theory [BUSS 1998; TROELSTRA and SCHWICHTENBERG 1996]: as perfectly rigorous mathematical objects in and of themselves.

Now, if one concurs with the Fregean approach and regards proofs as crisp mathematical objects (e.g., sequences or trees of formulas) arising in the context of some formal system or other, then clearly there can only be proofs of statements that represent well-formed formulas of such systems. Accordingly, since CTT in its usual formulation is not expressed as a well-formed sentence of a formal system (since "effectively computable" is taken as a *pre*theoretical, informal concept), then there is *ipso facto* no mathematical proof for it. If, by contrast, one is willing to entertain a more liberal view, whereby a mathematical proof need not be tied to any particular formal system, but need only be a convincing argument possibly involving informal, intuitive notions, then clearly it *could* be that CTT is mathematically provable. However, it seems to us that when logicians and authors of computability and logic texts state that CTT is not provable, they do so under the former understanding, and so, *contra* MENDELSON, what they say is not only true but trivially so. In fact, such authors would not deny that the thesis might be mathematically provable *if* one adopts the latter interpretation of "mathematical proof" as any cogent piece of argumentation. Indeed, almost invariably such authors point out that there is ample empirical evidence for CTT. The issue is simply that they are not willing to acknowledge such evidence, however convincing, as constituting a *mathematical proof*. So the issue raised by MENDELSON has little to do with CTT *per se*, and more to do with the old debate about the nature of mathematical proof.

There is perhaps another way to understand MENDELSON's assertion that we can have mathematical proofs about informal concepts, a weaker interpretation. On that interpretation, one could understand MENDELSON as saying that given certain informal concepts $C_1, \ldots, C_n$, it is possible to incorporate $C_1, \ldots, C_n$ as undefined (i.e., *primitive*) notions in some appropriate formal theory T, postulate certain propositions about them, and then proceed to give rigorous proofs in the extended theory T', proofs that may involve $C_1, \ldots, C_n$. This amounts to an implicit and possibly partial char-

acterization of the new concepts instead of explicit, total definitions of them within T. The new theory T' will be a non-conservative extension of T, and might be regarded as a quasi-formalization of $C_1, \ldots, C_n$. We obviously take no issue with that possibility. Indeed, we believe that this is how MENDELSON's claim that he has given a perfectly precise mathematical proof of the easier half of CTT should be understood. That "proof" can be readily formalized in a typed higher-order logic where we have a theory of the natural numbers and functions on them already available as libraries (as we do, say, in HOL [GORDON and MELHAM 1993]), and where "effectively computable" is taken as a primitive notion in the form of a unary relation on number-theoretic functions. We could then define the partial recursive functions as the inductive closure of the initial functions under the operations of composition, recursion, and minimalization, and couch MENDELSON's argument as an inductive proof. We could also formalize MENDELSON's proof in a first-order setting within an extension of ZFC, again by taking "effectively computable" as a primitive notion in addition to the binary relation of set membership.

There is nothing wrong with doing this, if we are dealing with a new concept C that we want to elucidate and can think of no explicit definition for it inside an already established theory. Taking C as a primitive and postulating appropriate propositions about it is a good way to experiment with the concept. (Usually, of course, this will happen by extending other theories rather than by starting from scratch.) It forces us to make explicit assumptions about C that integrate it rigorously with other previously established concepts and results, and gain experience with the kind of reasoning and results that such assumptions permit. But we gain nothing if the concept is one that has already been explicitly defined within another formal theory in a way that has proven satisfactory, i.e., in a way that has been empirically validated. Why take something as primitive when we can evidently define it explicitly in terms of other primitives that appear more fundamental (more primitive, so to speak)? We only end up violating Occam's razor and increasing the chances of inconsistency, which is a risk that we take every time we postulate a new axiom (and we *have* to postulate some axioms about the primitives, as otherwise we will not be able to prove anything about

them). Conservative theory extensions are obviously preferable to non-conservative extensions.

This can be seen more clearly by analyzing MENDELSON's argument for the easy half of CTT in detail. The argument has the form of an inductive proof in which both the basis step and the inductive step are asserted instead of proved. The fact that it has the form of an inductive argument is appropriate, since the set of partial recursive functions is inductively defined as a closure system—the smallest set of number-theoretic functions that contains the initial functions and is closed under composition, recursion, and minimalization. Now any closure system (or equivalently, any subalgebra built by a generating set) has an associated principle of structural induction, stating that if the basis elements have a certain property, and if that property is preserved by the operations, then *every* element of the closure has the property. In the case of the partial recursive functions the principle takes the following form: Let $\mathcal{S}$ be any set of number-theoretic functions and suppose that the following two conditions hold:

1. All initial functions are in $\mathcal{S}$ (basis step).

2. If $f_1, \ldots, f_n$ are in $\mathcal{S}$ and g is obtainable from f_i through composition, recursion, and/or minimalization, then g is in S (inductive step).

We are then entitled to conclude that $\mathcal{S}$ contains all partial recursive functions. MENDELSON's argument can be understood as an instantiation of this pattern, where $\mathcal{S}$ is the set of all effectively computable number-theoretic functions, where "effectively computable" is represented by some undefined, primitive unary predicate E applying to such functions.

But MENDELSON's "proof" simply asserts both the basis step and the inductive step (where $\mathcal{S}$ here is the set of effectively computable functions). In other words, he postulates two axioms:

A_1: The initial functions are effectively computable.

A_2: Effective computability is preserved by composition, recursion, and minimalization.

Of course these propositions are not arbitrary. They seem true in the intended interpretation, and MENDELSON tries to justify them

with two informal and very brief parenthetical arguments. But A_1 and A_2 are asserted nevertheless, as no mathematical derivations are given for them, only informal appeals to intuition. (Of course, it is conceivable that A_1 and A_2 could be deduced from some other unspecified and perhaps more fundamental axioms about E, but MENDELSON does not do anything of the sort.) Now an inductive mathematical proof that postulates both the basis and the inductive steps is tantamount to simply asserting that the inductive closure at hand has the relevant property. Mathematically, it has very little value. So let us assess the present state of affairs from a formal perspective. We have obtained a new theory T' as a non-conservative extension of a formal theory T by introducing a new primitive notion E. And we have postulated two axioms in T' that amount to a formal underspecification of E: We have claimed that the extension of E is a superset of the partial recursive functions. We have gained nothing by doing so. We have proliferated our conceptual ontology without settling the question of the exact extension of E relative to the formally defined set of partial recursive functions. It would in fact be impossible to settle that question within T' without arbitrarily postulating yet more axioms.

Given that the weak interpretation of MENDELSON seems theoretically unsatisfactory and renders his proof trivial, we are inclined to think that what he has in mind is the stronger claim to which we alluded earlier, namely, that there are *bona fide* mathematical proofs that cannot be accurately couched in *any* formal system whatsoever. A mathematical proof of CTT, in particular, would have to lie outside the confines of any particular formal system in that "effective computability" would need to remain entirely unformalized, taken neither as an undefined term of a formal system nor as a defined one. Accordingly, if CTT is mathematically provable in the intuitive sense, then there exist informal *mathematical* proofs that are inherently unformalizable, in the sense that they cannot be captured in any axiomatic system, even in principle. That is an extremely strong claim. We are not aware of any evidence for it. We are aware of ample evidence for its negation. Certainly it is a widely held belief that any branch of modern mathematics can be developed within ZFC (e.g., HENSON [1984] states "it is an empirical fact that all of mathematics as presently known can be formalized within the ZFC system"). That is not simply an article of faith. As MENDELSON

is well aware, such development has been carried out in painstaking and tremendously extensive detail by many mathematicians. There may be doubts as to whether ZFC can capture everything that mathematicians are interested in talking about, but that mainly refers to intensionality issues (e.g., modal notions such as epistemic states at given points in time), not to extensional questions of proof existence. And here we are not even talking about the adequacy of ZFC specifically; we are talking about *all* formal axiomatic systems, the claim being that there are informal mathematical proofs that cannot be captured in *any* formal system. This contradicts what one might call "Hilbert's thesis": that the identification of the informal concept of mathematical proof with formal deduction in axiomatic systems is correct, meaning that any informal mathematical proof can be represented by some rigorous deduction in an appropriate formal system, and conversely, any such deduction can be viewed as representing an informal mathematical proof (this is "the easier half" of Hilbert's thesis).

In fact, there are increasing amounts of evidence to suggest that any style of argumentation capable of serving as genuine mathematical justification can be made perfectly rigorous. For example, for a long time diagrams were thought to be suspect and "rigorous" mathematicians would warn against the pitfalls of using such informal devices in mathematical proofs. But recent work (much of it sparked by the efforts of the late Jon BARWISE) has shown that Venn diagrams, higraphs, and Peirce diagrams can all be made perfectly precise [HAMMER 1995; SHIN 1995], with rigorous notions of syntax, semantics, soundness, completeness, etc. Even more recent results by the authors [ARKOUDAS 2005] suggest that it is possible to formalize *arbitrary* diagrammatic proofs; we have designed and analyzed a new domain-independent logical framework for heterogeneous natural deduction capable of combining diagrammatic and symbolic inference for arbitrary finite domains, we have proven it sound, and have given detailed algorithms for how to implement a proof checker for it.

Accordingly, given that the mathematical provability of CTT in that sense would contradict Hilbert's thesis, and given the overwhelming evidence in favor of that thesis and the absence (to the best of our knowledge) of any evidence against it, we conclude that CTT is indeed mathematically unprovable.

Even if one remains agnostic on the proper reading of "mathematical proof," however, there is still much to take issue with in MENDELSON's arguments. For instance, he writes:

> The concepts and assumptions that support the notion of partial-recursive function are, in an essential way, no less vague and imprecise than the notion of effectively computable function; the former are just more familiar and are part of a respectable theory with connections to other parts of logic and mathematics. (The notion of effectively computable function could have been incorporated into an axiomatic presentation of classical mathematics, but the acceptance of CT made this unnecessary.) The same point applies to [PT, FT, and TT]. Functions are defined in terms of sets, but the concept of set is no clearer than that of function and a foundation of mathematics can be based on a theory using function as primitive notion instead of set. Tarski's definition of truth is formulated in set-theoretic terms, but the notion of set is no clearer than that of truth. The model-theoretic definition of logical validity is based ultimately on set theory, the foundations of which are no clearer than our intuitive understanding of logical validity. [MENDELSON 1990, p. 232]

But MENDELSON doesn't establish these statements; he simply asserts them. By our lights—and by the lights of many others—the concept of a set and the relation of set membership, which are ultimately the only two primitive concepts underlying the notion of a Turing-computable (equivalently, partial recursive) function, are much clearer than the notion of an algorithm, which is the main concept underlying the informal notion of an effectively computable function.[2]

They are likewise clearer than the concepts of logical validity and entailment, limits and convergence, and all other examples mentioned by MENDELSON. Indeed, we claim that the set concept is inherently more foundational than—and hence it *is* "essentially dif-

[2]We do not think that the well-known independence results in ZFC, or the proliferation of unorthodox set-theoretic axioms and interpretations, undermine the clarity of the primitive concept any more than the existence of non-standard models of arithmetic or the incompleteness of Peano's axioms make the concept of natural number any less clear—or the existence of hyperbolic and elliptic geometries makes the Euclidean concepts of points and lines any less clear.

ferent" from—the concept of an algorithm. This is not a fringe view. MADDY [1997] calls the foundational view of sets "a pillar of contemporary orthodoxy," citing quotations such as the following: "All mathematicians do mathematics in set theory, consciously or unconsciously" [LEVY 1979]. Views like the following one by the mathematical logician MOSCHOVAKIS [1998] are common:

> I believe that most mathematical theories (and all the non-trivial ones) can be clarified considerably by having their basic notions modeled faithfully in set theory; that for many of them a (proper) set-theoretic foundation is not only useful but necessary—in the sense that their basic notions cannot be satisfactorily explicated without reference to fundamentally set-theoretic notions; and that set-theoretic foundations of mathematical theories can be formulated so that they are compatible with a large variety of views about truth in mathematics and the nature of mathematical objects. [MOSCHOVAKIS 1998, p. 9]

MENDELSON also claims that

> Another difficulty with the usual viewpoint concerning CT is that it assumes that the only way to ascertain the truth of the equivalence asserted in CT is to *prove* it.

But no such assumption is "usually" made, either explicitly or tacitly. Indeed, virtually all authors gladly accept the truth of the thesis even though they consider it mathematically unprovable. If they assumed that mathematical proof was the *only* way to ascertain the truth of the thesis, they would be in the rather embarrassing position of enthusiastically endorsing a proposition whose truth there is *no way*, by their own assumption, to ascertain. Rather, the usual viewpoint is that the thesis is as "ascertained" as any proposition with empirical import can hope to be: eminently likely, but in principle subject to refutation.

3. Is Church's Thesis True?

In this section BRINGSJORD presents an argument purporting to show that Church's thesis is false.[3] ARKOUDAS regards the thesis

[3] Alert readers will realize that if Church's thesis is false, it follows immediately that it's unprovable, since presumably nothing false can be proved. Such readers

as true and presents three objections to BRINGSJORD's argument, in Section 4.1, Section 4.2, and Section 4.3. BRINGSJORD replies to these objections, and proceeds to consider some additional possible objections. In Section 5 BRINGSJORD discusses previous attacks on Church's thesis.

BRINGSJORD's suspicion that CT is false first arose in connection with the concept of *productive* sets, which have two properties:

P1 They are classically undecidable (=no program, Turing machine, etc. can decide such sets).

P2 There is a computable function f from the set of all standard programs to any such set, a function which, when given a candidate program P (for deciding the set in question), yields an element of the set for which P will fail.

Put informally, a set A is productive iff it's not only classically undecidable, but also if any program proposed to decide A can be counter-exampled with some element of A. Clearly, if a set A' has these properties, then $A' \notin \Sigma_0$ and $A' \notin \Sigma_1$. If A' falls somewhere in AH, and is effectively decidable, then CTT falls. But what could possibly fit the bill? BRINGSJORD has become convinced that the set $\mathcal{S}$ of all interesting stories provides a perfect fit.

This no doubt catches you a bit off guard. Interesting stories? Well, let us first remind you that the view that there are productive sets near at hand is far from unprecedented. Douglas HOFSTADTER [1982], for example, holds that the set $\mathcal{A}$ of all As is a productive set. In order to satisfy P1, $\mathcal{A}$ must forever resist attempts to write a program for deciding this set; in order to satisfy P2, there must at minimum always be a way to "stump" a program intended to decide $\mathcal{A}$. That $\mathcal{A}$ satisfies both these conditions isn't all that implausible—especially when one faces up to the unpredictable variability seen in this set. For example, take a look at Figure 1, taken from *Graphic Art Materials Reference Manual* [1981].

---

will thus perhaps wonder why success in the present section doesn't render the previous section otiose. The answer is simply that, together, we have both endeavored to take MENDELSON seriously: To grapple directly with the provability issue, independent of other arguments.

Figure 1: Various Letter As

In order for a program to decide $\mathcal{A}$, it must capitalize on some rules that capture the "essence" of the letter in question. But what sorts of rules could these be? Does the bar in the middle need to touch the sides? Apparently not (see 2 A). Does there have to be a bar that *approximates* connecting the sides? Apparently not (see 7 G). And on and on it goes for other proposed rules.[4]

However, it must be conceded that no *argument* for the productivity of $\mathcal{A}$ has been provided by Hofstadter. For all we know, some company could tomorrow announce a letter recognition system that will work for all As. The situation is a bit different in the case of the mathematician Peter Kugel [1986], who makes clever use of an elementary theorem in unmistakably *arguing* that the set of all beautiful objects is located above Σ_1 in AH:

> We seem to be able to recognize, as beautiful, pieces of music that we almost certainly could not have composed. There is a theorem about the partially computable sets that says that there is a uniform procedure for turning a procedure for recognizing members of such sets into a procedure for generating them. Since this procedure is uniform—you can use the same

[4]Relevant here is Hofstadter's letter spirit program, which generates fonts from the first few letters in the font in question. For an argument that this program, and others, aren't really creative, see [Bringsjord, Ferrucci and Bello 2001].

> one for all computable sets—it does not depend on any specific information about the set in question. So, if the set of all beautiful things were in Σ_1, we should be able to turn our ability to recognize beautiful things into one for generating them [...] This suggests that a person who recognizes the Sistine Chapel Ceiling as beautiful knows enough to paint it, [which] strikes me as somewhat implausible. [KUGEL 1986, pp. 147–148]

The main problem with this line of reasoning is that it's disturbingly exotic. Beauty is perhaps a promising candidate for what KUGEL is after, but it must be conceded that most of those scientists who think seriously about human cognition don't think a lot about beauty. Indeed, they don't seem to think *at all* about beauty.[5] And this isn't (they would insist) because beauty is a daunting concept, one that resists recasting in computational terms. The stance would doubtless be that beauty is left aside because one can exhaustively analyze cognition (and replicate it on a machine) without bothering to grapple in earnest with this concept.

This claim about the irrelevance of beauty may strike some as astonishing, and it certainly isn't a view affirmed by each and every computationalist, but we gladly concede it for the sake of argument: for the record, we grant that ignoring beauty, in the context of attempts to model, simulate, and replicate mentation, is acceptable.[6] However, BRINGSJORD thinks there is another concept that serves our purposes perfectly: namely, the concept of a *story*. Stories are thought by many to be at the very heart of cognition. For example, in their lead target chapter in *Knowledge and Memory: The Real Story* [WYER 1995], Roger SCHANK and Robert ABELSON, two eminent scientists working in the area of cognition and computation, boldly assert on the first page that "virtually all human knowledge"

[5] A search for coverage of this concept in standard texts about cognition—e.g., [ASHCRAFT 1994] and [STILLINGS *et al.* 1995]—turns up nothing whatever.

[6] What argument could be mustered for ignoring beauty in the context of attempts to reduce cognition to computation, or to build an artificial agent capable of behaviors analogous to human ones typically taken to involve beauty? We envisage an argument running parallel to the one John POLLOCK [1995] gives for ignoring human emotions in his attempt to build an artificial person. POLLOCK's view, in a nutshell, is that human emotions are in the end just "time savers;" with fast enough hardware, and clever enough algorithms, artificial persons could *compute* the need to quickly flee (say) a lion, whereas we take one look and immediately feel a surge of fear that serves to spark our rapid departure.

is based on stories.[7] SCHANK and ABELSON go on to claim that since the essence of cognition inheres in narrative, we can jettison propositional, logic-based, rule-based, formal... schemes for knowledge representation. Among the 17 commentators who react to the target piece, 13 affirm the story-based view (the remaining four authors are skeptical). Moreover, this book is one of many in the same family. For example, SCHANK has devoted a book to the view that stories are at the very heart of human cognition: [SCHANK 1995]. For another example, note that DENNETT's [1991] *Consciousness Explained* can be read as a defense of the view (his "multiple drafts" view of consciousness) that thinking amounts to spinning out parallel stories.

The other nice thing about stories, from our perspective, is that apparently one of us knows a thing or two about them, in connection to computation. For over a decade, BRINGSJORD worked at creating an artificial agent capable of autonomously creating sophisticated fiction. BRINGSJORD first discussed this project in his *What Robots Can and Can't Be* [1992], in which he specifically discussed the challenge of characterizing, precisely, the class of *interesting* stories. (His main claim was that formal philosophy offers the best hope of supplying this characterization.) For those who seek to build agents capable of creative feats like good storytelling, this is a key challenge. It's easy enough to build systems capable of generating *un*interesting stories. For example, the world's first significant artificial story generator, TALE-SPIN [MEEHAN 1981], did a good job of that. Here, for example, is one of TALE-SPIN's best stories:

"Hunger"

> Once upon a time John Bear lived in a cave. John knew that John was in his cave. There was a beehive in a maple tree. Tom Bee knew that the beehive was in the maple tree. Tom was in his beehive. Tom knew that Tom was in his beehive. There was some honey in Tom's beehive. Tom knew that the honey was in Tom's beehive. Tom had the honey. Tom knew that Tom had the honey. There was a nest in a cherry tree. Arthur Bird knew that the nest was in the cherry tree. Arthur was in his nest. Arthur knew that John was in his cave. [...]

[7] An insightful review of this book has been written by Tom TRABASSO [1996].

How are things to be improved? How is one to go about building an agent capable of creating interesting stories? It was the sustained attempt to answer this question, in conjunction with the concept of productivity discussed above, that persuaded BRINGSJORD that CT is indeed false. Let us explain.

First, to ease exposition, let $\mathcal{S}^I$ denote the set of all interesting stories. Now, recall that productive sets must have two properties, P1 and P2; let's take them in turn, in connection with $\mathcal{S}^I$. First, $\mathcal{S}^I$ must be classically undecidable; i.e., there is no program (or TM, etc.) which answers the question, for an arbitrary story in $\mathcal{S}^I$, whether or not it's interesting. Second, there must be some computable function f from the set of all programs to $\mathcal{S}^I$ which, when given as input a program P that purportedly decides $\mathcal{S}^I$, yields an element of $\mathcal{S}^I$ for which P fails. It seems to us that $\mathcal{S}^I$ does have both of these properties—because, in a nutshell, Bringsjord and colleagues seemed to invariably and continuously turn up these two properties "in action." Every time someone suggested an algorithm-sketch for deciding $\mathcal{S}^I$, it was easily shot down by a counter-example consisting of a certain story which is clearly interesting despite the absence in it of those conditions regarded by the proposal to be necessary for interestingness. (It has been suggested that interesting stories must have inter-character conflict, but monodramas can involve only one character. It has been suggested that interesting stories must embody age-old plot structures, but some interesting stories are interesting precisely because they violate such structures, and so on.)

The situation we have arrived at can be crystallized in deductive form as follows.[8]

Arg$_3$

(9) If $\mathcal{S}^I \in \Sigma_1$ (or $\mathcal{S}^I \in \Sigma_0$), then there exists a procedure P which adapts programs for deciding members of $\mathcal{S}^I$ so as to yield programs for enumerating members of $\mathcal{S}^I$.

[8]Please note that the labeling in this argument is intentional. This argument is one Bringsjord is defending anew in the present chapter, and desires to preserve it precisely as it has been previously articulated [see BRINGSJORD and ZENZEN 2003]. The argument is now followed by new objections from ARKOUDAS, given in section 4.

	(10)	There's no procedure P which adapts programs for deciding members of $\mathcal{S}^I$ so as to yield programs for enumerating members of $\mathcal{S}^I$.	
$\therefore$	(11)	$\mathcal{S}^I \notin \Sigma_1$ (or $\mathcal{S}^I \notin \Sigma_0$).	10, 11
	(12)	$\mathcal{S}^I \in$ AH.	
$\therefore$	(13)	$\mathcal{S}^I \in \Pi_1$ (or above in the AH).	disj syll
	(14)	$\mathcal{S}^I$ is effectively decidable.	
$\therefore$	(15)	CT is false.	*reductio*

Clearly, Arg_3 is formally valid. Premise (9) is not only true, but necessarily true, since it's part of the canon of elementary computability theory. What about premise (10)? Well, this is the core idea, the one expressed above by KUGEL, but transferred now to a different domain: People who can *decide* $\mathcal{S}^I$, that is, people who can decide whether something is an interesting story, can't necessarily *generate* interesting stories. Student researchers in BRINGSJORD's laboratory have been a case in point: with little knowledge of, and skill for, creating interesting stories, they have nonetheless recognized such narrative. That is, students who are, by their own admission, egregious creative writers, are nonetheless discriminating critics. They can decide which stories are interesting (which is why they know that the story generators AI has produced so far are nothing to write home about), but *producing* the set of all such stories (including, as it does, such works as not only *King Lear*, but *War and Peace*) is quite another matter. These would be, necessarily, the *same* matter if the set of all interesting stories, $\mathcal{S}^I$, was in either Σ_0 or Σ_1, the algorithmic portion of AH.

But what's the rationale behind (14), the claim that $\mathcal{S}^I$ is effectively decidable? The rationale is simply the brute fact that a normal, well-adjusted human computist can effectively decide $\mathcal{S}^I$. Try it yourself: First, start with the sort of story commonly discussed in AI; for example:

"Shopping"

> Jack was shopping at the supermarket. He picked up some milk from the shelf. He paid for it and left.[9]

[9]From page 592 of [CHARNIAK and MCDERMOTT 1985]. The story is studied in the context of attempts to resolve pronouns: How do we know who the first

Well? Your judgement? Uninteresting, we wager. Now go back to "Hunger," and come up with a judgement for it, if you haven't done so already. Also uninteresting, right? Now render a verdict on "Betrayal," a story that can be produced by BRINGSJORD and FERRUCCI's [2000] BRUTUS:

> "Betrayal"
>
> Dave Striver loved the university. He loved its ivy-covered clocktowers, its ancient and sturdy brick, and its sun-splashed verdant greens and eager youth. He also loved the fact that the university is free of the stark unforgiving trials of the business world—only this *isn't* a fact: academia has its own tests, and some are as merciless as any in the marketplace. A prime example is the dissertation defense: to earn the PhD, to become a doctor, one must pass an oral examination on one's dissertation. This was a test Professor Edward Hart enjoyed giving.
>
> Dave wanted desperately to be a doctor. But he needed the signatures of three people on the first page of his dissertation, the priceless inscriptions which, together, would certify that he had passed his defense. One of the signatures had to come from Professor Hart, and Hart had often said—to others and to himself—that he was honored to help Dave secure his well-earned dream.
>
> Well before the defense, Dave gave Hart a penultimate copy of his thesis. Hart read it and told Dave that it was absolutely first-rate, and that he would gladly sign it at the defense. They even shook hands in Hart's book-lined office. Dave noticed that Hart's eyes were bright and trustful, and his bearing paternal.
>
> At the defense, Dave thought that he eloquently summarized Chapter 3 of his dissertation. There were two questions, one from Professor Rodgers and one from Dr. Teer; Dave answered both, apparently to everyone's satisfaction. There were no further objections.
>
> Professor Rogers signed. He slid the tome to Teer; she too signed, and then slid it in front of Hart. Hart didn't move.
>
> "Edward?" Rogers said.
>
> Hart still sat motionless. Dave felt slightly dizzy.
>
> "Edward, are you going to sign?"

---

occurrence of 'He' refers to in this story? And how do render the process of resolving the pronoun to Jack as a computational one?

> Later, Hart sat alone in his office, in his big leather chair, saddened by Dave's failure. He tried to think of ways he could help Dave achieve his dream.

This time, interesting, right?

Now at this point some readers may be thinking: "Now wait a minute. Isn't your position inconsistent? On the one hand you cheerfully opine that 'interesting story' cannot be captured. But on the other you provide an interesting story!—a story that must, if I understand your project, capitalize upon some careful account of interestingness in narrative."

"Betrayal" is based in significant part upon formalizations, in intensional logic, of definitions taking the classic form of necessary and sufficient conditions seen in analytic philosophy. These definitions are given for "immemorial themes;" in "Betrayal" the two themes are self-deception and, of course, betrayal. Here is approximately the definition of betrayal with which BRUTUS works:[10]

D Agent s_r betrays agent s_d at t_b iff there exists some state of affairs p and $\exists t_i, t_k \, (t_i \leq t_k \leq t_j \leq t_b)$ such that

- **1** s_d at t_i wants p to occur;
- **2** s_r believes that s_d wants p to occur;
- **3′** $(3 \wedge 6') \vee$
  - **6″** s_d wants at t_k that there is no action a which s_r performs in the belief that thereby p will not occur;
- **4″** there is some action a such that:
  - **4″a** s_r performs a at t_b in the belief that thereby p will *not* occur; and
  - **4″b** it's not the case that there exists a state of affairs q such that q is believed by s_r to be good for s_d and s_r performs a in the belief that q will not occur;
- **5′** s_r believes at t_j that s_d believes that there is some action a which s_r will perform in the belief that thereby p *will* occur.

[10] Note that the variables t_i range over times, and that $\leq$ means "earlier or simultaneous." Note also the following clauses, which appear in clause 3′.

3 s_r agrees with s_d that p ought to occur;

6′ s_d wants that there is some action a which s_r performs in the belief that thereby p *will* occur.

All of this sort of work (i.e., the gradual crafting of such definitions in the face of counter-example after counter-example; the crafting in the case of betrayal is described in Chapter 4 of [BRINGSJORD and FERRUCCI 2000]) is perfectly consistent with the absence of an account of 'interesting story.' *In fact*, this kind of philosophical analysis figures in the observation that proposed accounts of interestingness are invariably vulnerable to counter-example. For example, suppose we try (here, schematically) something Bringsjord and colleagues have tried: Let $c_1, \ldots, c_n$ enumerate the definitions of all the immemorial themes involved in narrative. Now suppose we venture a defintion having the following structure.

D′ A story s is interesting iff

1 [...]

⋮

k s instantiates (inclusive) either c_1 or c_2 or [...] or c_n.

$k+1$ [...]

⋮

p [...]

The problem—and, alas, BRINGSJORD has experienced it time and time again—is that along will come a counter-example; in this case, a story which explicitly fails to satisfy k from D′'s definiens will arrive. For example, an author can write a very interesting story about a phenomenon like betrayal as cashed out in definition D, except that instead of clause 4″, the following weaker clause is satisfied.

4′ there is some action a which s_r performs in the belief that thereby p will *not* occur.

The story here might involve a courageous, self-sacrificial mother who assures her addicted son that she will procure drugs to relieve his misery (as he desires), but intends only to confront the pusher and put an end to his destructive dealings. Ironically, clearly some of the interestingness in this story will derive precisely from the fact that the mother is not betraying her son. On the contrary, she plans to save him and others. In short, devising accounts like D' seems to be to fight a battle that can never be won; good narrative cannot be bottled.

Selmer will now endeavor to reply to several possible objections to his argument, the first three of which were expressed by his co-author.

4. Objections to $\mathbf{Arg}_3$

4.1. Objection 1

ARKOUDAS expressed his first objection as follows:

> The "set of all interesting stories" is an inherently fuzzy concept; it does not have a precise extension. Your argument rests on a confusion between *defining* a set and *computing* one. Questions of formal computability start after one has precisely defined a set S via arithmetization techniques as a set of natural numbers (or, more generally—and sans arithmetization—as a set of strings over some countable symbol set). Ideally, the definition should be given rigorously via a logical formula of the form
>
> $$x \in S \Leftrightarrow F(x) \tag{1}$$
>
> where F is a completely formal statement (no undefined symbols in it) of one free variable x. Of course, one need not descend to this level of detail and may instead offer a high-level definition that omits certain tedious details. But convincing remarks must be made to show that it is indeed possible (at least in principle) to go from the high-level definition sketch to a rigorous one of the form (1). Otherwise one cannot claim to have a mathematical object about which mathematical statements (such as Turing-computability or lack thereof) can be made.
>
> The vast majority of the sets in the arithmetic hierarchy (AH) are of course uncomputable. All of them, however, are precisely *definable*. What your own argument indicates is simply that the concept of an interesting story does not have a clear extension: Every time your students tried to come up with precise sufficient and necessary conditions to characterize it, someone came up with a counter-example. This has nothing to do with computability; it has to do with the set's definability. So to argue about exactly where in the AH the set of interesting stories resides (above or below the Turing limit) is to put the cart before the horse, as you have not even given us a single reason to believe that the set would be in the AH *at all*. To show that a set is in the AH, you need to convince us

that it can be defined by a sequence of alternating quantifier blocks over a recursive predicate. No such definition seems to exist for your set.

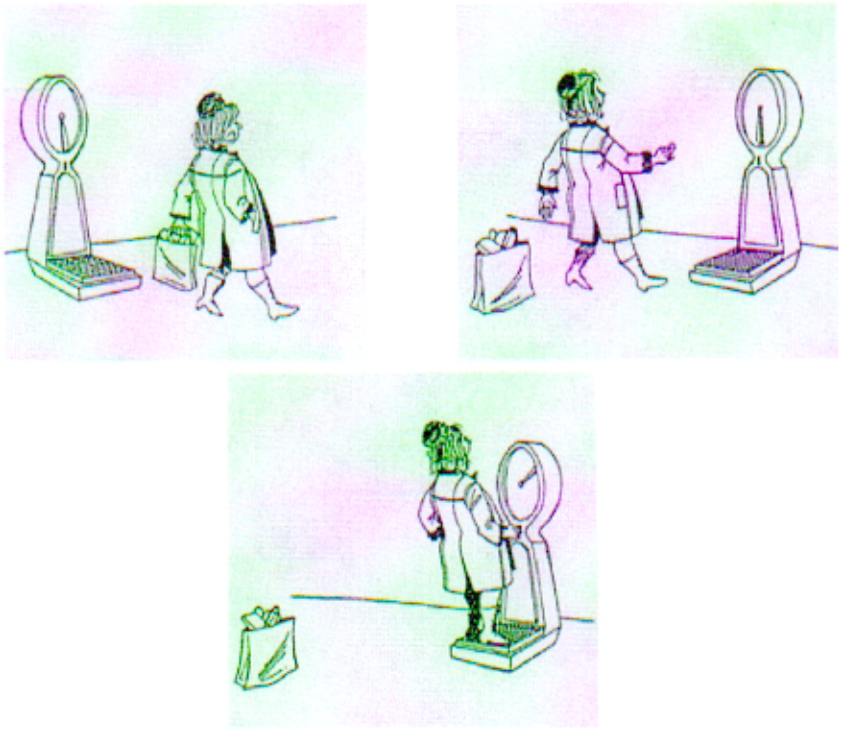


Figure 2: What is the story that can be constructed from these snapshots? (Reprinted here with paid permission from Psychological Corporation.)

We routinely apply concepts like 'Turing-computable' to objects that aren't defined in the narrow way described in this objection. In fact, we have already seen reference to such an object above, in Figure 1: the set of all A's. This set isn't defined in the rigorous way ARKOUDAS venerates. Now of course he requires that such definitions be achievable only *in principle*. But can the set of all A's be narrowly defined, given more insight, time, and energy? No one really knows. Nonetheless, we still specify and implement computer programs that take as input various A's, and we still (witness HOFSTADTER) try to determine whether the set of all A's is Turing-decidable. The attack on CTT should not be forced to abide by

constraints more stringent than those guiding the practice of computer science.

It is worth noting, as well, that, like A's, stories can be visual. For example, see Figure 2. How does computing over visual objects work, relative to a formal scheme (recursion theory) that is purely linguistic in nature? Again, no one yet knows. Therefore, to repeat, the constraints ARKOUDAS has in mind are too stringent. Nonetheless, to simplify the dialectic that follows, we will pretend that stories are invariably textual in nature.

4.2. Objection 2

ARKOUDAS' second objection runs as follows:

> My second main criticism of your argument concerns your claim that S^I is effectively decidable. (I'll disregard for now the inherent fuzziness of S^I, since the points I want to make here are orthogonal to that issue.) Quoting from your text:
>
> > But what's behind the rationale for (14)? [That's the premise that S^I is effectively decidable.] The rationale is simply the fact that a normal, well-adjusted human computist can effectively decide S^I.
>
> But, to falsify CT, you need to come up with a non-Turing-computable set that 'a normal, well-adjusted human computist' can nevertheless decide *by way of an algorithm*. By definition, to show that a set A is effectively decidable you need to *demonstrate the existence of an algorithm* that an idealized computist could use to decide the membership problem for A. Ideally, you would do this constructively: you would *show us* the algorithm whose existence you are claiming. Perhaps you could also argue indirectly for the algorithm's existence, by trying to derive some sort of contradiction from the assumption that no such algorithm exists.
>
> But you do not do that. What you show is that for the two or three particular short story excerpts that you cite, most people would come to the expected judgment. No one would doubt that, but it is quite irrelevant. It says nothing about the existence or non-existence of an algorithm. Consider an analogous hypothetical argument:
>
> > What is the rationale for the claim that the halting problem is effectively decidable? The simple fact that a normal, well-adjusted human computist

> can effectively decide whether any given program always halts or not. Consider the program
>
> ```
> x := y * z;
> ```
>
> Well? Your judgment? A halter, we wager. Ok, now try this:
>
> ```
> if true then
> x := 1
> else
> x := 2;
> ```
>
> Also a halter, right? Now try this one:
>
> ```
> while true do
> ;
> ```
>
> This time a non-halter, right? Ergo, there is an algorithm for deciding the halting problem.

In fact even if you presented millions of positive and negative examples of programs that were correctly classified by humans with regard to termination, we could still infer nothing whatsoever about the existence of an algorithm for the halting problem.

Moreover, I would claim that, by your line of reasoning, *all* sets are effectively decidable. Consider any set A whatsoever. Now it is clear that "a normal, well-adjusted human" will be able to correctly identify some objects as belonging to A and some objects as not belonging to A, for otherwise they can hardly be said to understand what A is. By your reasoning, this would appear to licence the conclusion that A is effectively decidable.

Indeed, if we reject Church's thesis, as you do, and refuse to identify "algorithm" with any precise set-theoretic notion, then we can never deduce that a given set is *not* effectively decidable. So if you claim that a decision algorithm for a certain set exists (as you do for S^I) and yet you refuse to present the alleged algorithm, no one could possibly falsify your claim. Accordingly, if we take the standard view that an unfalsifiable statement is not scientific, then we ought to conclude that your assertion that "S^I is effectively decidable" is not a scientific statement—it is an article of faith.

This objection mistakenly conflates two senses of 'effectively computable.' One sense, invoked in our background section, and a direct reflection of the framework and language used by TURING and POST

(see note 1), is based on what a "computist" or "worker" can do; i.e., on what a human being, working mechanically, can accomplish. The second sense, which is clear in what the objection states, is that of what is *algorithmically* computable. Let e.c.$_1$ refer to the former sense, and e.c.$_2$ refer to the latter. ARKOUDAS is certainly correct that S^I cannot be said to be e.c.$_2$ on the strength of what a computist can do: one needs in this case to present the algorithm. But when one has in mind e.c.$_1$, as I do, the one and only piece of evidence to bring forward is the observation that it is transparent that a computist can handle the task at hand—and I mean the *arbitrary* task as hand. This is clearly the case in the story domain: read it, judge it, spit out the verdict.

ARKOUDAS goes on to offer programs designed to confirm his objection, and to generalize his objection. But the sample programs are irrelevant to the case at hand, for the simple reason that we can put on display programs that stump human computists with respect to haltingness.[11] But no such counter-examples can be provided in the case of interesting stories. Finally, as to the generalization to the proposition that we can never be sure that any set is *not* effectively decidable, which is purported to go through if my position is assumed, I do clearly follow TURING and POST (and many others in the relevant tradition, e.g. [SIEG and BYRNES 1996]) in holding that it must be self-evident that the computist or worker can prevail in all cases. Even young students realize, for example, that long division, when time and energy is unbounded, is perfectly reliable. This realization comes not only because particular examples like $456 \div 8$ are unproblematic, but also because it's evident that the trick will work for any relevant pair, and hence no counter-example (unlike the case of the halting problem) is forthcoming.

4.3. Objection 3

Konstantine's third objection is actually a pair of related objections. The first of the pair is this:

[11] E.g., at least in the days before WILES changed the landscape [WILES 1995, WILES and TAYLOR 1995], we could stump computists with a Turing machine M such that it halts iff Fermat's Last Theorem is true and provable, and spins forever iff FLT is false. Any number of Turing machines like this could be dreamed up now.

You use Kugel's argument to justify your claim that S^I is not Turing-computable. Unfortunately, Kugel's argument is enthymematic and flawed. Kugel starts by making the following two assumptions: (a) there is an algorithm for recognizing "beautiful objects;" and (b) there is an algorithm for generating the set of all "objects," beautiful or not. (In my view both assumptions are problematic for various reasons—e.g., both "beautiful" and "object" are ill-defined—but in any event these are two assumptions that Kugel *must* make because they are needed by the result from elementary computability theory to which he appeals, so let us go along for the sake of the argument.) He then glibly cites the aforementioned result (omitting, as we will see, a key assumption of the result), which states that for any set $A \subseteq U$ whose elements are drawn from some countable universe U, *if* (i) there is an algorithm for deciding the membership problem for A (namely, given any member $x \in U$, do we have $x \in A$ or $x \in U \setminus A$?); and (ii) there is an algorithm for generating the universe U; *then* there is an algorithm for enumerating A (to wit: Use the algorithm from (ii) to start listing the elements of the universe U, deploying the procedure from (i) to weed out elements which are not members of A). Using (a) and (b) for (i) and (ii), Kugel concludes that the set of all beautiful objects is algorithmically enumerable, which means that (idealized) persons could generate arbitrarily large numbers of beautiful objects even if they had zero creativity, as long as they could effectively recognize beauty. In Kugel's words: "This suggests that a person who recognizes the Sistine Chapel Ceiling as beautiful knows enough to paint it, [which] strikes me as somewhat implausible." Since he views this conclusion as counter-intuitive, he rejects assumption (a) via *reductio ad absurdum*, inferring that the set of beautiful objects cannot possibly be effectively decidable (or Turing-computable, by Church's thesis).

But the oddness which Kugel attributes to the conclusion actually lies in assumption (b), an assumption which Kugel neglects to state even though it is a crucial premise of the theorem he invokes. If one assumes, as Kugel does, that a person has an algorithm for generating all objects, then that person already "knows enough to paint the Sistine Chapel Ceiling"—as well as compose Beethoven's ninth symphony, write Tolstoy's War and Peace, and so on. Hence, assumption (b) is just as counter-intuitive as the conclusion that Kugel finds implausible, and therefore, by his own lights, we have just as good grounds to reject it. If we reject it, however, we can no longer

> appeal to the theorem that is the centerpiece of Kugel's reasoning, and his whole argument collapses.

ARKOUDAS is of course perfectly right about the situation: Any standard typographic set A that is Turing-enumerable can presumably be enumerated by even a dim human being: just follow the relevant instructions. Even dim human beings, after all, can locate entries in a dictionary; they do so by essentially following the standard algorithm for lexicographic ordering, which takes as a starting place the ordering on the starting alphabet. (In English, 'A' comes before 'B' comes before 'C,' and so on.) Computists needn't be humans: they could be, say, pigeons. Thus, a trained pigeon could write *King Lear*, sooner or later. All of this is completely uncontroversial. However, it doesn't tell in the least against KUGEL's point, which is based on the real-life fact that (e.g.) SHAKESPEARE.[12] *created King Lear*. He imagined the characters, arranged the narrative, wrote the dialogue, and so on.

The second part of the pair is expressed as follows:

> Since you (BRINGSJORD, not KUGEL) actually claim that the set of interesting stories S^I *is* effectively decidable, the foregoing theorem from computability theory can be adapted to show that, contrary to what you claim, S^I is effectively enumerable, i.e., there is an algorithm for generating all and only the interesting stories. Let A_I be the algorithm that you claim can decide S^I. We can of course represent every element of S^I by a string of English letters and punctuation characters. And there is an obvious algorithm A_U that effectively enumerates the set of all strings of English letters and/or punctuation characters; call that set U (this is our universe here, $U \supseteq S^I$). Now here is an algorithm for cranking out interesting stories: start enumerating the set U by using algorithm A_U; as each string in U is generated, use algorithm A_I to decide if it represents an interesting story. If it does, keep it in the list, otherwise strike it out. It is easy to see that if one accepts your own assumptions, then this algorithm generates all and only the elements of S^I.

[12] I (BRINGSJORD) shy away from speaking of MICHELANGELO, for the simple reason that while I can write fiction, I can't paint. I do suspect that painters would confirm, in the case of MICHELANGELO, what I say about SHAKESPEARE.

This objection founders for reasons already canvassed. What we know is that S^I, the set of interesting stories, is *effectively* decidable. We know this, again, because we ourselves can be the verifying computists. It hardly follows from this (as has been previously noted) that we have on hand the *algorithm* (and that is ARKOUDAS' word: algorithm) A_I. This inference succeeds only if the two previously distinguished senses of effectively computable (e.c.$_1$ and e.c.$_2$) are erroneously conflated.

4.4. Objection 4

In the next objection, we see a variant of ARKOUDAS' final objection:

> "Look, Bringsjord, you must have gone wrong somewhere! Stories are just strings over some finite alphabet. In your case, given the stories you have put on display above, the alphabet in question is { `Aa, Bb, Cc, [...], :, !, ;, [...]`}, that is, basically the characters we see before us on our computer keyboard. Let's denote this alphabet by 'E.' Elementary string theory tells us that though E^*, the set of all strings that can be built from E, is infinite, it's *countably* infinite, and that therefore there is a program P which enumerates E^* (P, for example, can resort to lexicographic ordering). From this it follows that your $\mathcal{S}$, the set of all stories, is itself countably infinite. (If we allow, as no doubt we must, all natural languages to be included—French, Chinese, and even Norwegian—the situation doesn't change: the union of a finite (or for that matter a countably infinite) number of countably infinite sets is still just countably infinite.) So what's the problem? You say that your students are able to decide $\mathcal{S}^I$? Fine. Then here's what we do to enumerate $\mathcal{S}^I$: Start P in motion, and for each item S generated by this program, call your students to pass verdict on whether or not S is interesting. This composite program—call it P': P working in conjunction with your students—enumerates $\mathcal{S}^I$. So sooner or later, P' will manage to write *King Lear*, *War and Peace*, and even more

> recent belletristic narrative produced by Bringsjord's favorite author: Mark Helprin."[13]

The reasoning here is fallacious. The reason is straightforward, and decisive: An assumption is made here that the composite information processing, i.e., P plus what the computist is doing in judging stories, falls at the level of Turing machines. While it's of course true that P is at this level (it's just lexicographic ordering yet again) we don't know that what computists are doing as literary critics (if you will) is Turing-computable. In fact, that's exactly the issue at hand, and hence the objection is nothing more than a *petitio*.

4.5. Objection 5

The next objection is an attempt to resurrect ARKOUDAS' first objection:

> "I now see your error, Selmer: premise (12) in Arg_3. If $\mathcal{S}^I$ is to be in AH, then your key predicate—'Interesting'; denote it by 'I'—must be a bivalent one. (More precisely, I must be isomorphic to a predicate that is built via quantification out of the totally computable bivalent predicates of Σ_0.) But a moment's reflection reveals that I *isn't* bivalent: different people have radically different opinions about whether certain fixed stories are interesting! Clearly, though Jones and Smith may share the same language, and may thus be able to fully understand 'Shopping,' 'Hunger,' 'Betrayal,' *King Lear*, and *War and Peace*, their judgements may differ. "Shopping" might be downright thrilling to an AInik interested in determining how, upon reading such a story, humans know instantly that the pronoun 'He' refers to Jack."[14]

It is important to realize that we are talking about stories *qua* stories; stories as narrative. Hence a better way to focus the present objection is to note that Jones may find *Kind Lear* to be genuine

[13] BRINGSJORD has responded to this objection in an earlier publications (see the chapters on Church's thesis in BRINGSJORD & FERRUCCI [2000] and BRINGSJORD & ZENZEN [2003]). The following response is a new one, and supplants previous ones, which are confessedly inadequate.

[14] This intelligent objection is originally due to Michael MCMENAMIN [1992], though a number of thinkers have conveyed its gist to us.

drama, but monstrously boring drama (because, he says, King Lear is but a lunatic), while Smith is transfixed. It's undeniable that differences of opinion like those existing between Jones and Smith are common. But this fact is not a threat to BRINGSJORD's argument. First, note that such differences are present in *all* domains, not just in the domain of narrative. WITTGENSTEIN, remember, teased much out of a clash between someone who says that $2+2=4$ and someone who flatly denies it—so even the arithmetical realm, if Objection 3 goes through, would lack bivalent properties, and if anything is suffused with bivalence, it's arithmetic. Moreover, there is nothing to prevent us from stipulating that these agents come decked out with some fixed "value system"—for judging stories. In fact, let us heretofore insist that I be read as not just interesting *simpliciter*, but interesting given (what must surely be one of the world's most refined systems for gauging stories) the knowledge and ability of none other than Umberto ECO.[15] Our new predicate, then, can be I_{UE}.

The objection could perhaps be sustained as follows:

> "I seriously doubt that Umberto Eco *has* a fixed effective decision system by which *he* decides. I take it this is an illusion predicated on the fact that Eco has the *authority* to say what interests him (*a la* Wittgenstein on the incorrigibility of 'introspection'). *Whatever* Eco sincerely pronounces 'interesting' *is* interesting for Eco; what he says goes. This seems akin to what you two envision your 'decked out' agents doing (just reading and pronouncing); this seems unlike effective deciding. You might as well say that each of us has an effective procedure for deciding the set of things that will be said by us in our lifetime: just by saying that we do we 'enumerate the set.' You might as well say the U.S. Supreme Court has a rote procedure for deciding cases: in deciding them they 'enumerate the set' of Supreme Court decisions. Eco's own infallibility being a matter of authority, nothing guarantees that identically 'decked out' agents—lacking authority—will decide the same as him (or each other for that matter)."

[15] Those unfamiliar with ECO's non-fiction work, might start with his surprising reasons for finding Ian FLEMING's 007 (James Bond) series to be very interesting; see "Chapter Six: Narrative Structures in Fleming," in [ECO 1979].

This is a decidedly weak objection. Clearly, one claim made against us is simply that Eco has no system by which he judges interestingness. But this claim is wrong. The reason is that Eco doesn't rely on mere authority: he presents the system: again, we refer interested readers to: [Eco 1979]. (One might say that Eco has become an authority *because* he has described his system.) Given this, the analogies to the Supreme Court, and to what we say in our lifetimes, fail. In neither of these domains is there even the hint of a description of the scheme by which verdicts are produced; the situations are therefore disanalogous. We do suspect that *individual* members of the Supreme Court *would* be analogous to Eco. Indeed, analyses of and careful commentaries on Supreme Court opinions routinely contain descriptions of the scheme deployed by a particular member of the Court.

4.6. Objection 6

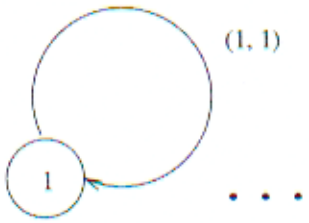
"At the start of this chapter you affirmed Mendelson's characterization of 'algorithm.' Let me remind you that according to that characterization, 'An algorithm does not require ingenuity.' Are you not now bestowing remarkable ingenuity upon the readers/judges you have in mind?"

Recall that in order to parse 'effectively computable,' as we have noted, it's necessary to invoke the generic concept of an agent, either Turing's "computist" or Post's "worker." (At the very least, the standard way to unpack 'effectively computable' is through this generic concept.) The agent in question, as none other than Elliot Mendelson reminded us nearly forty years ago [Mendelson 1963], needn't be a *human* agent, because, following the mantra at the heart of computability theory, we impose no practical restrictions on the length of calculations and computations. It follows immediately that the agents we have in mind have enough raw time and energy to process the longest and most complex contenders in $\mathcal{S}$. Furthermore, if we are going to seriously entertain CTT, we must, all of us, allow the agents in question to have certain knowledge and ability, for example the knowledge and ability required to grasp the concepts of number, symbol, change, movement, instruction, and so on. The agents we have in mind are outfitted so as to be able to grasp stories, and the constituents of stories. And in deploying I, and in moving to I_{UE}, we assume *less* on the part of agents (workers, computists, etc.)

than what even defenders of CT through the years have assumed. This is so because such thinkers freely ascribe to the agents in question the knowledge and ability required to carry out sophisticated proofs—even proofs which cannot be formalized in first-order logic. The agents capable of deciding $\mathcal{S}^I$ need only read the story (and, for good measure, read it n subsequent times—something mathematicians routinely do in order to grasp proofs), and render their decision.

4.7. Objection 7

"Yes, but what your computists do is not decomposable into smaller, purely mechanical steps, which is the hallmark of an algorithm. They are supposed to read a story (and, if I understand you, perhaps read it again some finite number of times), and then, just like that, render a judgment. This is more like magic than mechanism."



(The node here reflects the start state.)

Figure 3: A Flow-Diagram Fragment That Entails Non-Halting

To see the problem with this objection, let's prove, in a thoroughly traditional manner, that a certain well-defined problem is effectively solvable. Recall that all Turing machines can be recast as flow diagrams (e.g., see [BOOLOS and JEFFREY 1989]). Next, note that any TM represented by a flow diagram having as part the fragment shown in Figure 3 would be a non-halting TM (because if started in state 1 with its read/write head scanning the leftmost 1 in a block of 1s—and we can assume the alphabet in question to be a binary one consisting of {0,1}—it will loop forever in this fragment). Let m be a fixed TM specified for computist Smith in flow diagram form, and let this diagram contain the fragment of Figure 3. Suppose that Brown looks for a minute at the diagram, sees the relevant fragment, and declares: "Nonhalter!" In doing this, Brown assuredly decides m, and his performance is effective. And

yet what's the difference between what Brown does and what our "Eco-ish" agents do? The activity involved is decomposable in both cases. There are innumerable "subterranean" cognitive processes going on beneath Brown's activity, but they are beside the point: that we don't (or perhaps can't) put them on display does not tell against the effectiveness in question. The fact is that Brown simply looks at the diagram, finds the relevant fragment, assimilates, and returns a verdict.[16] The same is true of our agents in the case of stories.

Before turning to consider other attacks on CT, we point out that the predicates I and I_{UE} really aren't exotic, despite appearances to the contrary. All those who try to harness the concepts of theoretical computer science (concepts forming a superset of the formal ones canvassed in this book) in order to get things done end up working with predicates *at least* as murky as these two. A good example is to be found in the seminal work of John POLLOCK, which is based on the harnessing of theoretical computer science (including AH) so as to explicate and implement concepts like warrant, defeasibility, *prima facie* plausibility, and so on.[17]

5. Arg_3 in Context: Other Attacks on CT

Over the past six decades, the possibility of CT's falsity has not only been raised,[18] but CT has been subjected to a number of outright attacks. While we obviously don't have the book-long space it would take to treat each and every attack, we think it's possible to provide a provisional analysis that is somewhat informative, and serves to situate BRINGSJORD's own attack on CT. What this

[16] Our example is perfectly consistent with the fact that the set of TMs, with respect to whether or not they halt, is not Turing-decidable.

[17] Here is one example from [POLLOCK 1995]: POLLOCK's OSCAR system is designed so as to constantly update that which it believes in response to the rise and fall of arguments given in support of candidate beliefs. What constitutes correct reasoning in such a scheme? POLLOCK notes that because a TM with an ordinary program can't decide theorems in first-order logic (the set of such theorems isn't Turing-decidable), answering this question is quite tricky. He ingeniously turns to super-computation for help: the basic idea is that OSCAR's reasoning is correct when it generates successive sets of beliefs that approach the ideal epistemic situation in the limit. This idea involves AH, as POLLOCK explains.

[18] BOOLOS and JEFFREY, for example, in their classic textbook *Computability and Logic* [1989], provide a sustained discussion of CT—and take pains to leave the reader with the impression that CT can be overthrown.

analysis shows, we think, is that Arg_3 is the most promising attack going.

Following R.J. NELSON [1987], we partition attacks on CT into three categories:

CAT1 Arguments against the arguments for CT;

CAT2 Arguments against CT itself; and

CAT3 Arguments against doctrines (e.g., the computational conception of mind) which are said (by some, anyway) to presuppose CT.

Consider CAT3 first. Perhaps the most promising argument in this category runs as follows. Assume for the sake of argument that all human cognition consists in the execution of effective processes (in brains, perhaps). It would then follow by CT that such processes are Turing-computable, i.e., that computationalism is true. However, if computationalism is false, while there remains incontrovertible evidence that human cognition consists in the execution of effective processes, CT is overthrown.

Attacks of this sort strike us as unpromising. For starters, many people aren't persuaded that computationalism is false (despite some careful arguments we have ourselves given; e.g., see BRINGSJORD & ARKOUDAS [2004]). Secondly, this argument silently presupposes some sort of physicalism, because the evidence for the effectiveness of cognition (in the sense that *all* cognition is effective; only this view can support an overthrow of CT in CAT3) no doubt derives from observation and study of processes in the central nervous system. Thirdly, it is certainly at least an open question as to whether the processes involved *are* effective. Indeed, by BRINGSJORD's lights, some of the processes that constitute cognition aren't effective.

What about CAT1? The main issue with the work of all those who intend to attack CT by attacking the time-honored rationales for it is that such work can at best expose flaws in particular arguments for the thesis, but cannot refute the thesis itself. For example, William THOMAS [1973] seeks to capitalize on the fact (and it *is* a fact, that much is uncontroversial) that the main rationale behind CT involves empirical induction—a form of reasoning that has little standing in mathematics. Unfortunately, THOMAS' observations don't threaten CT in the least, as is easy to see. Most of us believe, *unshakably* believe, that the universe is more than 3 seconds old—but what mathematical rationale have we for this belief? As RUSSELL

pointed out, mathematics is quite consistent with the proposition that the universe popped into existence 3 seconds ago, replete not only with stars, but with light here on Earth from stars, and also with minds whose memories include those we have. More generally, of course, from the fact that p doesn't follow deductively from a set of propositions Γ, it hardly follows that p is false; it doesn't even follow that p is the slightest bit implausible.

We are left, then, with CAT2—the category into which Bringsjord's attack on CT falls. How does Arg_3 compare with other attacks in this category? To support the view that Bringsjord's attack is superior, let us consider a notorious argument from four decades back, one due to László Kalmár [1959] (and rejected by none other than Elliott Mendelson [1963]), and the only other modern attack on CT that we know of, one given by Carol Cleland [1993; 1995].[19]

5.1. Kalmár's Argument against CT

Here's how Kalmár's argument runs. First, he draws our attention to a function g that isn't Turing-computable, given that f is:[20]

$$g(x) = \mu_y(f(x,y) = 0) = \begin{cases} \text{the least } y \text{ such that } f(x,y) = 0 \text{ if } y \text{ exists} \\ 0 \text{ if there is no such } y \end{cases}$$

Kalmár proceeds to point out that for any $n \in N$ for which a natural number y with $f(n,y) = 0$ exists, "an obvious method for the calculation of the least such y [...] can be given," namely, calculate in succession the values $f(n,0), f(n,1), f(n,2), \ldots$ (which, by hypothesis, is something a computist or TM can do) until we hit a natural number m such that $f(n,m) = 0$, and set $y = m$.

> On the other hand, for any natural number n for which we can prove, not in the frame of some fixed postulate system

[19] Perhaps we should mention here something that students of CT and its history will be familiar with, viz., *given an intuitionistic interpretation of 'effectively computable function,'* CT can be disproved. The basic idea is to capitalize on the fact that any subset of N is intuitionistically enumerable, while many such sets aren't effectively enumerable. (A succinct presentation of the disproof can be found on page 592 of Nelson [1987].) The main problem with such attacks on Church's thesis, of course, is that they presuppose (certain axioms of—see e.g., Kreisel [1965; 1968]) intuitionistic logic, which most reject.

[20] The original proof can be found on page 741 of [Kleene 1983].

> but by means of arbitrary—of course, correct—arguments that no natural number y with $f(n, y) = 0$ exists, we have also a method to calculate the value $g(n)$ in a finite number of steps: prove that no natural number y with $f(n, y) = 0$ exists, which requires in any case but a finite number of steps, and gives immediately the value $g(n) = 0$. [KALMÁR 1959, p. 74]

KALMÁR goes on to argue as follows. The definition of g itself implies the *tertium non datur*, and from it and CT we can infer the existence of a natural number p which is such that

(i) there is no natural number y such that $f(p, y) = 0$; and

(ii) this cannot be proved by any correct means.

KALMÁR claims that (i) and (ii) are very strange, and that therefore CT is at the very least implausible.

This argument is interesting, but really quite hopeless, as a number of thinkers have indicated. For example, as MENDELSON [1963] (see also MOSCHOVAKIS' [1968] review of both KALMÁR's paper and MENDELSON's reaction) points out, KALMÁR's notion of 'correct proof,' for all KALMÁR tells us, may fail to be effective, since such proofs are outside the standard logical system (set theory formalized in first-order logic). This is surely historically fascinating, since—as we have seen—it would be MENDELSON who, nearly thirty years later, in another defense of CT (the one we examined earlier), would offer a proof of the 'only if' direction of this thesis—a proof that he assumes to be correct but one that he admits to be beyond ZF. But the root of KALMÁR's problem is that his proofs, on the other hand, are wholly hypothetical: we don't have a single one to ponder. And things get even worse for KALMÁR (as NELSON [1987] has pointed out), because even absent the proofs in question, we know enough about them to know that they would vary for each argument to g that necessitates them, which would mean that KALMÁR has failed to find a *uniform* procedure, a property usually taken to be a necessary condition for a procedure to qualify as effective.

Though KALMÁR does anticipate the problem of lack of uniformity,[21] and though BRINGSJORD personally happens to side with

[21] He says:

> By the way, [the assumption that the procedure in question] must be uniform seems to have no objective meaning. For a school-boy, the method for the solution of the diverse arithmetical problems he

him on this issue, it is clear that his argument against CT fails: If Kalmár's argument is to succeed, (ii) can be supplanted with

(ii′) this cannot be proved by any effective means.

But then how can the argument be deductively valid? It is not, at bottom, a *reductio*, since (i) and (ii′) surely are not absurd, and this is the only form a compelling version of the argument could at core be. KALMÁR himself, as we have noted, confesses that his argument is designed only to show that CT is implausible, but this conclusion goes through only if (i) and (ii′), if not absurd, are at least counter-intuitive. But are they? For some, perhaps; for others, definitely not.

Our own take on Kalmár's argument is that it can be rather easily *shown* to be flawed as follows: First, let

$$m_1, m_2, \ldots, m_n, m_{n+1}, \ldots$$

enumerate the set of Turing machines. Now substitute for Kalmár's g the following function.

$$h(m_i) = \begin{cases} 1 & \text{if } m_i \text{ halts} \\ 0 & \text{if } m_i \text{ doesn't halt} \end{cases}$$

Recall that if a TM halts, simulating this machine will eventually reveal this fact. This allows us to produce an exact parallel to KALMÁR's reasoning: Start with m_1; proceed to simulate this machine. Assuming it halts, return 1, and move on to m_2, and do the same for it; then move to m_3, and so on. While this process is running, stand ready to prove "not in the frame of some fixed postulate system but by means of arbitrary—of course, correct—arguments" that the machine m_i fails to halt, in which case 0 is returned. The parody continues as follows. Given CT, and the law of the excluded middle (which the definition of the function h presupposes), we infer two implausible propositions—propositions so implausible that CT is itself cast into doubt. They are:

($\mathbf{i}_h$) there exists an m_k such that $h(m_k) = 0$; and

---

has to solve does not seem uniform until he learns to solve equations; and several methods in algebra, geometry and theory of numbers which are now regarded group-theoretic methods were not consider as uniform before group-theory has (sic) been discovered. [KALMÁR 1959, p. 73]

($\mathbf{ii}'_h$) this cannot be proved by any effectively computable means.

This is a parody, of course, because both of these propositions are fully expected and welcomed by all those who both affirm CT and have at least some familiarity with the formalisms involved.

Now, what about Bringsjord's case against CT? First, the narrational case is deductive, as Arg_3 makes plain. Second, the process of reading (and possibly rereading a finite number of times) a story, assimilating it, and judging whether or not it's interesting on a fixed evaluation scheme—this process is transparently effective. (Indeed, related processes are routinely requested on standardized tests containing reading comprehension problems, where stories are read, perhaps reread, and judged to express one from among n "main ideas.") Third, the process we're exploiting would seem to be uniform.[22]

5.2. Cleland's Doubts about CT

CLELAND [1993; 1995] discusses three variants on our CT:

$\mathbf{CT}_1$ Every effectively computable number-theoretic function is Turing-computable.

$\mathbf{CT}_2$ Every effectively computable function is Turing-computable.

$\mathbf{CT}_3$ Every effective procedure is Turing-computable.

Before evaluating CLELAND's arguments against this trio, some exegesis is in order. First, each of these three theses is a conditional, whereas CT, as we have explained, is a *bi*conditional. There should be no question that the biconditional is more accurate, given not only MENDELSON's authoritative affirmation of the biconditional form, but also given that CHURCH himself originally refers to his thesis as a *definition* of "effectively calculable function" in terms of "recursive function" [CHURCH 1940].[23] However, since we have happily conceded the 'if' direction in CT, there is no reason to worry about this aspect of CLELAND's framework. The second point is that by

[22] No doubt test designers are correct that a uniform procedure needs to be followed in order to excel in their reading comprehension sections. So why wouldn't the process at the heart of Arg_3 be uniform as well?

[23] On the other hand, CHURCH then immediately proceeds to argue for his "definition," and the reader sees that he is without question urging his readers to affirm a *thesis*.

'number-theoretic' function CLELAND simply means a mapping from N to N. We thus now understand function *simpliciter*, as for example it's used in CT_2, to allow functions from the reals to reals.[24] There is of course no denying that CHURCH and TURING failed to advocate CT_2, but CT_1 is certainly the "left-to-right" direction of our CT.

Now, what does CLELAND say against CT_1-CT_3? She claims, first, that CT_3 can be disproved; the argument is simply this. One type of effective procedure coincides with what CLELAND calls "mundane procedures," which are "ordinary, everyday procedures such as recipes for making Hollandaise sauce and methods for starting camp fires; they are methods for manipulating physical things such as eggs and pieces of wood" [CLELAND 1995, p. 11]. Turing machine procedures, on the other hand, are "methods for 'manipulating' abstract symbols" [CLELAND 1995, p. 11]. Since mundane procedures have "causal consequences," and TMs (*qua* mathematical objects) don't, it follows straightaway that mundane procedures aren't Turing-computable, that is, $\neg CT_3$.[25]

CLELAND's reasoning, when formalized, is certainly valid. The problem is that CT_3 (on that reading) has next to nothing to do with those propositions placed in the literature under the title "Church's Thesis"! CT_3 is a variant that no one has ever taken seriously. It may *seem* to some that CT_3 has been taken seriously, but this is only because one construal of it, a construal at odds with CLELAND's, has in fact been recognized. On this construal, that a procedure is Turing-computable can be certified by either a relevant design (e.g., a TM flow-graph for making Hollandaise sauce, which is easy to come by or by a relevant artifact (e.g., an artificial agent capable of making Hollandaise sauce, which again is easy to come by). At any rate, we're quite willing to concede that CT_3, on CLELAND's idiosyncratic reading, is provably false. (Note that we have known for decades that even CT_1, on an intuitionistic (and hence idiosyncratic) reading of "effectively computable function," is provably false. See note 20.) It's worth noting that CLELAND herself has sympathy for those who

[24] It will not be necessary to present here the formal extension of computability with number-theoretic functions to computability with functions over the reals. For the formal work, see, e.g., [GRZEGORCZYK 1955; 1957].

[25] In [BRINGSJORD and ZENZEN 2002] we explain why CLELAND's placing recipes for such things as cheese balls alongside mathematical accounts of computation is unacceptable.

hold that her reading of CT_3 is not a *bona fide* version of Church's Thesis [CLELAND 1995, p. 10]. What then, about CT_2 and CT_1?

Here CLELAND no longer claims to have a refutation in hand; she aims only at casting doubt on these two theses. This doubt is supposed to derive from reflection upon what she calls "genuinely continuous devices" [CLELAND 1995, p. 18], which are objects said to "mirror" Turing-uncomputable functions [CLELAND 1995, pp. 16–17]. An object is said to *mirror* a function iff (a) it includes a set of distinct objects which are in one-to-one correspondence with the numbers in the field of the function, and (b) the object pairs each and every object corresponding to a number in the domain of the function with an object corresponding to the appropriate number in the range of the function. CLELAND takes pains to argue, in intuitive fashion, that there are objects which mirror Turing-uncomputable functions (e.g., an object moving through a 2-dimensional Newtonian universe). She seems unaware of the fact that such objects *provably* exist—in the form, for example, of analog chaotic neural nets and, generally, analog chaotic dynamical systems [SIEGELMANN and SONTAG 1994, SIEGELMANN 1995]. (These objects are known to exist in the mathematical sense. Whether they exist in the corporeal world is another question, one everyone—including CLELAND—admits to be open.) We will be able to see CLELAND's fundamental error (and, indeed, the fundamental error of anyone who attacks CT by taking her general route) if we pause for a moment to get clear about the devices in question. Accordingly, we'll present here an analog dynamical system via the "analog shift map," which is remarkably easy to explain.

First let's get clear on the general framework for the "shift map." Let A be a finite alphabet. A *dotted sequence* over A is a sequence of characters from A^* wherein one dot appears. For example, if A is the set of digits from 0 to 9, then 3.14 is a dotted sequence over A. Set $A^{\cdot}$ to the set of all dotted sequences over A. Dotted sequences can be finite, one-way infinite (as in the decimal expansion of π), or bi-infinite. Now, let $k \in N$; then the shift map

$$S^k : A^{\cdot} \rightarrow A^{\cdot} : (a)_i \rightarrow (a)_{i+k}$$

shifts the dot k places, negative values for a shift to the left, positive ones a shift to the right. (For example, if $(a)_i$ is 3.14159, then with $k = 2$, $S^2(3.14159) = 314.159$.) Analog shift is then defined as

the process of first replacing a dotted substring with another dotted substring of equal length according to a function $g\colon A^{\cdot} \to A^{\cdot}$. This new sequence is then shifted an integer number of places left or right as directed by a function $f\colon A^{\cdot} \to Z$. Formally, the analog shift is the map

$$\Phi\colon a \to S^{f(a)}(a \ominus g(a)),$$

where $\ominus$ replaces the elements of the first dotted sequence with the corresponding element of the second dotted sequence if that element is in the second sequence, or leaves it untouched otherwise. Formally:

$$(a \oplus g)_i = \begin{cases} g_i & \text{if } g_i \in A \\ a_i & \text{if } g_i \text{ is the empty element} \end{cases}$$

Both f and g have "finite domains of dependence" (DoDs), which is to say that they depend only on a finite dotted substring of the sequence on which they act. The domain of *effect* (DoE) of g, however, may be finite, one-way infinite, or bi-infinite. Here is an example from [SIEGELMANN 1995, p. 547] which will make things clear, and allow us to see the fatal flaw in CLELAND's rationale for doubting CT_2 and CT_1. Assume that the analog shift is defined by (where π^2 is the left-infinite string ...51413 in base 2)

DoD	f	g
0.0	1	π^2
0.1	1	.10
1.0	0	1.0
1.1	1	.0

and that we have a starting sequence of $u = 000001.10110$; then the following evolution ensues:

000001.00110

0000010.0110

$\pi^2.0110$

$\pi^2 0.100$

$\pi^2 0.100$

$\pi^2 01.00$

$$\pi^2 1.00$$

$$\pi^2 01.00$$

At this point the DoD is 1.0 and hence no changes occur; this is a *fixed point*. Only the evolution from an initial dotted sequence to a fixed point counts.[26] In this case the input-output map is defined as the transformation of the initial sequence to the final subsequence to the right of the dot (hence in our example u as input leads to 00). The class of functions determined by the analog shift includes as a proper subset the class of Turing-computable functions (the proof is straightforward: SIEGELMANN [1995]). Moreover, the analog shift map is a mathematical model of idealized physical phenomena (e.g., the motion of a billiard ball bouncing among parabolic mirrors). From this it follows that we provably have found exactly what CLELAND desires, that is, a genuinely continuous device that mirrors a Turing-uncomputable function. So, if CLELAND can establish that

(16) If x mirrors a function, then x computes it,

she will have overthrown both CT_2 and CT_1. Unfortunately, given our analysis of the analog shift map, we can see that CLELAND doesn't have a chance; here is how the reasoning runs. Recall, first, the orthodox meaning of 'effectively computable function,' with which we started this chapter: a function f is effectively computable provided that, an agent having essentially our powers, a computist (or worker), can compute f by following an algorithm. So let's suppose that you are to be the computist in the case of the analog shift map. There is nothing impenetrable about the simple math involved; we'll assume that you have assimilated it just fine. So now we would like you to compute the function Φ as defined in our example involving π. To make your job as easy as possible, we will guarantee your immortality, and we will supply you with an endless source of pencils and paper (which is to say, we are "idealizing" you). Now, please set to work, if you will; we will wait and observe your progress...

What happened? Why did you stop? Of course, you stopped because you hit a brick wall: it's rather challenging to write down and manipulate (or imagine and manipulate mentally) strings like π in base 2! (Note that the special case where the DoE of g is finite in

[26] For a nice discussion of the general concept of a fixed point in connection with supertasks, see [STEINHART 2002].

the analog shift map generates a class of functions identical to the class of Turing-computable ones.) Yet this is precisely what needs to be done in order to attack CT_2 and CT_1 in the way CLELAND prescribes.

CLELAND sees the informal version of the problem, for she writes:

> Is there a difference between mirroring a function and computing a function? From an intuitive standpoint, it seems that there is. Surely, falling rocks don't compute functions, even supposing that they mirror them. That is to say, there seems to be a difference between a mere representation of a function, no matter how detailed, and the computation of a function. [Q:] But what could this difference amount to? [CLELAND 1995, p. 20]

She then goes on to venture an answer to this question:

> A natural suggestion is that computation requires not only the mirroring of a function but, also, the *following* of a procedure; falling rocks don't compute functions because they don't follow procedures. [CLELAND 1995, p. 20]

CLELAND then tries to show that this answer is unacceptable. The idea is that since the answer doesn't cut it, she is entitled to conclude that (16) is true, that is, that there *isn't* a difference between mirroring a function and computing a function,[27] which then allows the mere existence of (say) an idealized billiard ball bouncing among parabolic mirrors to kill off CT_2 and CT_1.

What, then, is CLELAND's argument for the view that the "natural suggestion" in response to Q fails? It runs as follows:

> Turing machines are frequently construed as *purely* mathematical objects. They are defined in terms of the same kinds of basic entity (viz., sets, functions, relations and constants) as other mathematical structures. A Turing machine is said to *compute* a number-theoretic function if a function can be *defined* on its mathematical structure which has the same detailed structure as the number-theoretic function concerned;

[27] This reasoning is certainly enthymematic (since it hides a premise to the effect that there are no other answers that can be given to question Q), but we charitably leave this issue aside.

> there isn't a distinction, in Turing machine theory, between computing a function and defining a function [...] If computing a function presupposes following a procedure, then neither Turing machines nor falling rocks can be said to compute functions. [CLELAND 1995, p. 21]

This argument is an enthymeme; its hidden premise is that 'compute' is used univocally in the relevant theses, i.e., that 'compute' means the same thing on both the left and right sides of CT, CT_1, and CT_2. This premise is false. The locution 'f is effectively computable,' on the orthodox conception of Church's Thesis, does imply that there is an idealized agent capable of *following* an algorithm in order to compute f. But it hardly follows from this that when 'compute' is used in the locution 'f is Turing-computable' (or in the related locution 'TM M computes f'), the term 'compute' must have the same meaning as it does in connection with idealized agents. Certainly anyone interested in CT, and in defending it, would hasten to remind CLELAND that the term 'compute' means one thing when embedded within CT's left side, and another thing when embedded within CT's right side.[28] Having said this, however, and having implicitly conceded the core mathematical point (viz., that at least some definitions of TMs and Turing-computability deploy 'compute' in the absence of the concept of "following"[29]), we should probably draw CLELAND's attention to the formal approach we took, where in order to characterize information-processing beyond the Turing Limit, we distinguished between a TM as a type of architecture, and a program which this architecture *follows* in order to compute.

CLELAND never intended to literally refute CT_1 and CT_2. (As we have seen, she did intend to refute the heterodox CT_3, and for the

[28] Unexceptionable parallels abound: We can say 'My friend told me that Burlington is a nice city,' and we can say 'My CD-ROM travel program told me that Burlington is a nice city,' but we needn't accept the view that 'told me' means the same in both utterances.

[29] Consider, e.g., one BRINGSJORD uses in teaching mathematical logic: A *Turing machine* is a quadruple (S, Σ, f, s) where

1. S is a finite set of *states*;
2. Σ is an alphabet containing the black symbol —, but not containing the symbols $\Leftarrow$ ("go left") and $\Rightarrow$ ("go right").
3. $s \in S$ is the *initial state*;
4. $f : S \times \Sigma \longrightarrow (\Sigma \cup \{\Leftarrow, \Rightarrow\}) \times S$ (the *transition function*).

sake of argument we agreed that here she succeeds.) But she fails even in her attempt to cast doubt upon these theses, and therefore CT is unscathed by her discussion.

6. Church's Thesis and Computationalism

In this final section we briefly discuss the relationship between CTT and computationalism, the view, roughly, that cognition is computation. The plan is as follows. We start by clarifying computationalism, and end up distinguishing between "weak" and "strong" versions of the doctrine. Next, we consider an argument deemed by Copeland to be fallacious. We show that the argument is formally valid once neatened, and is aimed at validating weak computationalism.[30] Of course, since this argument has CTT as a premise, it is sound only if Bringsjord's argument against CTT given in Section 3 fails.

6.1. What is 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. However, this isn't to say that the view has been once and for all defined. The fact is, the doctrine is exceedingly vague. Myriad one-sentence versions of it float about; e.g.,

1. Thinking is computing.
2. Cognition is computation.
3. People are computers (perhaps with sensors and effectors).
4. People are Turing machines (perhaps with sensors and effectors).

[30] This is as good a place as any to point out that earlier, we brought to your attention that some have maintained that computationalism presupposes CTT. This view would amount to

$$\mathcal{C} \rightarrow \text{CTT},$$

and if CTT is false, it would of course follow by *modus tollens* that $\mathcal{C}$ is false. However, we also pointed out that this conditional is questionable. But we focus in this final part on the converse: whether CTT implies (perhaps in conjunction with some additional premises) computationalism.

5. People are finite automata (perhaps with sensors and effectors).
6. People are neural nets (perhaps with sensors and effectors).
7. Cognition is the computation of Turing-computable functions.
8. ⋮

For present purposes, such a list isn't particularly helpful. We need to settle on one proposition, so that we can have some hope of productively discussing the relationship between CTT and computationalism. The one that we pick (unsurprisingly, given that we have have anchored our discussion of Church's Thesis to CTT) is the fourth, and we unpack it into a "strong" and "weak" version, and simply drop the parenthetical:

$\mathcal{C}^s$ People are Turing machines.

$\mathcal{C}^w$ People can be simulated by Turing machines.

We can put these doctrines perspicuously: Given that p ranges over persons, and m over Turing machines, we simply say:

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

$\mathcal{C}^w$ $\forall p \exists m\, S(m, p)$

Having on hand these versions of computationalism will prove valuable when it comes time to consider whether this doctrine is entailed by CTT.[31] Of course, as you might expect, more will need to be said about what 'simulation' means here.

6.2. The 'Simulation Fallacy'—Isn't a Fallacy

COPELAND tells us that one commits the 'Simulation Fallacy' "by believing that the Church-Turing thesis, or some formal result proved by Turing or Church, secures the truth of the proposition that the brain can be simulated by a Turing machine" [1998, p. 133]. As a paradigmatic example of the fallacy at work, Copeland gives us this passage from John SEARLE:

[31] We do not consider in this chapter the various attacks on $\mathcal{C}^s$ in the literature, many of which have been published by BRINGSJORD (e.g., to pick just one, see [BRINGSJORD 1999]), joined in some cases by ARKOUDAS (e.g., see [BRINGSJORD and ARKOUDAS 2004]).

> Can the operations of the brain be simulated on a digital computer [read: Turing machine—B.J.C.]? [...] The answer [...] seems to me [...] demonstrably 'Yes' [...] That is, naturally interpreted, the question means: Is there some description of the brain such that under that description you could do a computational simulation of the operations of the brain. But given Church's thesis that anything that can be given a precise enough characterization as a set of steps can be simulated on a digital computer, it follows trivially that the question has an affirmative answer. The operations of the brain can be simulated on a digital computer in the same sense in which weather systems, the behavior of the New York stock market, or the pattern of airline flights over Latin America can. [1992, pp. 200–201]

It seems to us that SEARLE's reasoning here is perfectly valid. The proposition he seeks to establish is this one:

(S) Under some description d, the activity of the brain is Turing-computable.

His argument for (S) is really quite straightforward. He points out that under some descriptions, the activity of the brain can be given a "precise enough characterization as a set of steps," that is, can be characterized in algorithmic, or effectively computable, steps. Since that which is algorithmic is Turing-computable by CTT, (S) follows. In stark sequence, SEARLE's argument is

Arg_4
Under some particular description d, the activity of the brain is effectively computable.
CTT
$\therefore$ Under some particular description d, the activity of the brain is Turing-computable.

Not only does this seem to be a perfectly valid argument, but there would seem to be descriptions that could be set to d to make the first premise true. We have in mind the common, general description (seen in the fields of cognitive science and AI) of the brain in terms of (conventional) artificial neural networks, which provably only process information in effectively computable ways.[32]

[32]See, for example, the discussion of the mindbrain and neural nets in [RUSSELL and NORVIG 2002; BRINGSJORD and ZENZEN 1997].

According to COPELAND, other well-known thinkers have been led astray by the Church-Turing fallacy. Victims supposedly include the Churchlands, and JOHNSON-LAIRD. But their arguments, characterized by Copeland as bald attempts to "deduce from Church's Thesis [= our CTT] that the mindbrain can in principle be simulated by a Turing machine" [COPELAND 1998, p. 133], can all be charitably read as formidable instances of Arg_4.

It's very important to realize that Searle and others, when speaking here of 'simulation,' do *not* have in mind the technical sense introduced early in the study of computability. This is the technical sense used in simulation proofs, for example that a multi-tape Turing machine can be shown by simulation to be equivalent in power to a standard one-tape Turing machine (a readable version of this proof is given in [LEWIS and PAPADIMITRIOU 1981]). Instead, what these thinkers have in mind is the sense of 'simulation' at work when, for example, the weather is simulated.[33] Consider the case of a hurricane H. H exists in the real world, and can of course wreak tremendous havoc. The virtual version of H—call it H^{sim},—on the other hand, is quite benign, and is thoroughly digital, regulated by standard computation. More importantly, H^{sim} is built by selectively attending to only *some* of the attributes possessed by H; the former is far from an atom-by-atom representation of the latter in some computational system. What Searle has in mind is only *some description*: there is a description of the brain that admits of a Turing-computable simulation. This is a very weak claim; COPELAND evidently fails to appreciate just how weak it is. Such a claim, in the case of the weather, is in fact easy to concretize courtesy of what many of us routinely consult: radar. Figure 4 shows a snapshot of a simulation of a storm that, as I type this sentence, is about to come crashing through Troy and then over the Taconic mountains just east of the city. Each of these snapshots can be chained together to make a temporally extended simulation; and this simulation, more assuredly, is Turing-computable. The same kind of thing can without question be done for the brain.

[33] Note that SEARLE's [1980] famous Chinese Room Argument against strong AI explicitly invokes a sense of simulation matching the sense in use in the realm of weather.

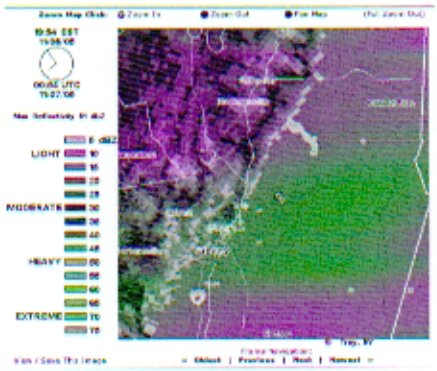


Figure 4: Snapshot of Virtual Storm Front Approaching Troy

Notice as well the rather obvious connection between (S) and $\mathcal{C}^w$. First, instead of speaking of human brains, SEARLE could speak of human *persons*. Second, to say that under some description d, x is Turing-computable, is just to say that x can be simulated by a Turing machine. So, there is a simple variant of SEARLE's argument that runs like this:

Arg'_4
Under some particular description d,
the (cognitive) activity of persons is effectively computable.
CTT
$\therefore$ $\mathcal{C}^w$

We conclude that, *contra* COPELAND, "Weak" computationalism (or, as it's sometimes called, "Weak" AI) does indeed follow from CTT. Of course, if BRINGSJORD's argument against Church's thesis in Section 3 is sound, then the case in question, while based on formally valid reasoning, nonetheless fails for the simple reason that one of the premises in it, CTT, is false.

References

Arkoudas, K. [2005], "Combining Diagrammatic and Symbolic Reasoning", *Technical Report 2005–59*, MIT Computer Science and Artificial Intelligence Lab, Cambridge, USA.

Ashcraft, M. [1994], *Human Memory and Cognition*, Harper-Collins, New York, NY.

Barr, A. [1983], "Artificial Intelligence: Cognition as Computation", in *The Study of Information: Interdisciplinary*

Messages, (F. Machlup ed.), Wiley-Interscience, New York, NY, pp. 237–262.

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

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

Bringsjord, S. [1999], "The Zombie Attack on the Computational Conception of Mind", *Philosophy and Phenomenological Research* **59.1**, 41–69.

Bringsjord, S. and Arkoudas, K. [2004], "The Modal Argument for Hypercomputing Minds", *Theoretical Computer Science* **317**, 167–190.

Bringsjord, S. and Ferrucci, D. [2000], *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine*, Lawrence Erlbaum, Mahwah, NJ.

Bringsjord, S., Ferrucci, D., and Bello, P. [2001], "Creativity, the Turing Test, and the (Better) Lovelace Test", *Minds and Machines* **11**, 3–27.

Bringsjord, S. and Zenzen, M. [1997], "Cognition is not Computation: The Argument from Irreversibility?", *Synthese* **113**, 285–320.

Bringsjord, S. and Zenzen, M. [2002], "Toward a Formal Philosophy of Hypercomputation", *Minds and Machines* **12**, 241–258.

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

Buss, S. [1998], "Introduction to Proof Theory", in *Handbook of Proof Theory*, Studies in Logic and the Foundations of Mathematics, **137**, (S. Buss ed.), Elsevier.

Charniak, E. and McDermott, D. [1985], *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, MA.

Church, A. [1940], "A Formulation of the Simple Theory of Types", *Journal of Symbolic Logic* **5**, 56–68.

Cleland, C. [1993], "Is the Church-Thesis True?", *Minds and Machines* **3**, 283–312.

Cleland, C. [1995], "Effective Procedures and Computable Functions", *Minds and Machines* **5**, 9–23.

Copeland, B.J. [1998], "Turing's O-Machines, Searle, Penrose and the Brain", *Analysis* **58**(2), 128–138.

Davis, M.D., Sigal, R., and Weyuker, E.J. [1994], *Computability, Complexity, and Languages*, 2nd edn, Academic Press.

Dennett, D. [1991], *Consciousness Explained*, Little, Brown, Boston, MA.

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

Eco, U. [1979], *The Role of the Reader: Explorations in the Semiotics of Texts*, Indiana University Press, Bloomington, IN.

Ernest, P. [1998], *Social Constructivism as a Philosophy of Mathematics*, State University of New York Press.

Fetzer, J. [1994], "Mental Algorithms: Are Minds Computational Systems?", *Pragmatics and Cognition* **2.1**, 1–29.

Gordon, M.J.C. and Melham, T.F. [1993], *Introduction to HOL, a Theorem Proving Environment for Higher-Order Logic*, Cambridge University Press, Cambridge, England.

Graphic Art Materials Reference Manual [1981], Letraset, New York, NY.

Grzegorczyk, R. [1955], "Computable Functionals", *Fundamentals of Mathematics* **42**, 168–202.

Grzegorczyk, R. [1957], "On the Definitions of Computable Real Continuous Functions", *Fundamentals of Mathematics* **44**, 61–71.

Hammer, E.M. [1995], *Logic and Visual Information*, CSLI Publications, Stanford, California.

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.

Henson, C.W. [1984], Review of Set Theory: An Introduction to Independence Proofs by K. Kunen, *Bulletin of the American Mathematical Society* (New Series) **10**, 129–131.

Hofstadter, D. [1982], "Metafont, Metamathematics, and Metaphysics", *Visible Language* **14**(4), 309–338.

Hofstadter, D. [1985], "Waking Up from the Boolean Dream", *Metamagical Themas: Questing for the Essence of Mind and Pattern*, Bantam, New York, NY, pp. 631–665.

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

Kalmár, L. [1959], "An Argument Against the Plausibility of Church's Thesis", in *Constructivity in Mathematics*, (A. Heyting ed.), North-Holland, Amsterdam, The Netherlands, pp. 72–80.

Kitcher, P. [1977], "On the Uses of Rigorous Proof", *Science* **196**, 782–783.

Kleene, S.C. [1983], "General Recursive Functions of Natural Numbers", *Math. Annalen* **112**, 727–742.

Kleiner, I. [1991], "Rigor and Proof in Mathematics: A Historical Perspective", *Mathematics Magazine* **64**(5), 291–314.

Kreisel, G. [1965], "Mathematical Logic", in *Lectures in Modern Mathematics*, (T. Saaty ed.), John Wiley, New York, NY, pp. 111–122.

Kreisel, G. [1968], "Church's Thesis: A Kind of Reducibility Thesis for Constructive Mathematics", in *Intuitionism and Proof Theory*, Proceedings of a Summer Conference at Buffalo, N.Y., (A. Kino, J. Myhill, and R. Vesley eds.), North-Holland, Amsterdam, The Netherlands, pp. 219–230.

Kugel, P. [1986], "Thinking May Be More Than Computing", *Cognition* **18**, 128–149.

Lakatos, I. [1976], *Proofs and Refutations: the Logic of Mathematical Discovery*, Cambridge University Press.

Levy, A. [1979], *Basic Set Theory*, Springer.

Lewis, H.R. and Papadimitriou, C.H. [1981], *Elements of the Theory of Computation*, Prentice Hall, Englewood Cliffs, NJ.

Lewis, H.R. and Papadimitriou, C.H. [1997], *Elements of the Theory of Computation*, Prentice Hall.

Maddy, P. [1997], *Naturalism in Mathematics*, Oxford University Press.

McMenamin, M. [1992], *Deciding Uncountable Sets and Church's Thesis*, [unpublished manuscript].

Meehan, J. [1981], "Tale-spin", in *Inside Computer Understanding: Five Programs Plus Miniatures*, (R. Schank and C. Reisbeck eds.), Lawrence Erlbaum, Englewood Cliffs, NJ, pp. 197–226.

Mendelson, E. [1963], "On Some Recent Criticism of Church's Thesis", *Notre Dame Journal of Formal Logic* **4**(3), 201–205.

Mendelson, E. [1990], "Second Thoughts about Church's Thesis and Mathematical Proofs", *Journal of Philosophy* **87**(5), 225–233.

Moschovakis, Y. [1968], "Review of Four Recent Papers on Church's Thesis", *Journal of Symbolic Logic* **33**, 471–472. One of the four papers is Kalmár [1959], "An Argument Against the Plausibility of Church's Thesis", in *Constructivity in Mathematics*, (A. Heyting ed.), Amsterdam, The Netherlands: North-Holland, pp. 72–80.

Moschovakis, Y.N. [1998], "On Founding the Theory of Algorithms", in *Truth in mathematics*, (H.G. Dales and G. Oliveri eds.), Oxford Science Publications, pp. 71–104.

Nelson, R.J. [1987], "Church's Thesis and Cognitive Science", *Notre Dame Journal of Formal Logic* **28**(4), 581–614.

Newell, A. [1980], "Physical Symbol Systems", *Cognitive Science* **4**, 135–183.

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

Pollock, J. [1995], *Cognitive Carpentry: A Blueprint for How to Build a Person*, MIT Press, Cambridge, MA.

Post, E. [1944], "Recursively Enumerable Sets of Positive Integers and their Decision Problems", *Bulletin of the American Mathematical Society* **50**, 284–316.

Rogers, H. [1967], *Theory of Recursive Functions and Effective Computability*, McGraw-Hill Book Company.

Russell, S. and Norvig, P. [2002], *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.

Schank, R. [1995], *Tell Me a Story*, Northwestern University Press, Evanston, IL.

Searle, J. [1980], "Minds, Brains and Programs", *Behavioral and Brain Sciences* **3**, 417–424.

Searle, J. [1992], *The Rediscovery of the Mind*, MIT Press, Cambridge, MA.

Shin, S.-J. [1995], *The Logical Status of Diagrams*, Cambridge University Press.

Sieg, W. and Byrnes, J. [1996], "K-Graph Machines: Generalizing Turing's Machines and Arguments", in *Gödel 96, Lecture Notes in Logic*, Springer-Verlag, New York, NY, pp. 98–119.

Siegelmann, H. [1995], "Computation Beyond the Turing Limit", *Science* **268**, 545–548.

Siegelmann, H. and 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.

Steinhart, E. [2002], "Logically Possible Machines", *Minds and Machines* **12**(2), 259–280.

Stillings, N., Weisler, S., Chase, C., Feinstein, M., Garfield, J., and Rissland, E. [1995], *Cognitive Science*, MIT Press, Cambridge, MA.

Thomas, W. [1973], "Doubts about Some Standard Arguments for Church's Thesis", *Papers of the Fourth International Congress for Logic, Methodology, and Philosophy of Science, Bucharest*, D. Reidel, Amsterdam, The Netherlands, pp. 13–22.

Trabasso, T. [1996], "Review of Knowledge and Memory: The Real Story", *Minds and Machines* **6**, 399–403.

Troelstra, A.S. and Schwichtenberg, H. [1996], *Basic Proof Theory*, Cambridge University Press, Cambridge, England.

Turing, A.M. [1936], "On Computable Numbers with Applications to the *Entscheidungsproblem*", *Proceedings of the London Mathematical Society* **42**, 230–265.

Wiles, A. [1995], "Modular Elliptic Curves and Fermat's Last Theorem", *Annals of Mathematics* **141**(3), 443–551.

Wiles, A. and Taylor, R. [1995], "Ring-Theoretic Properties of Certain Hecke Algebras", *Annals of Mathematics* **141**(3), 553–572.

Wyer, R.S. [1995], *Knowledge and Memory: The Real Story*, Lawrence Erlbaum, Hillsdale, NJ.