The Review of Modern Logic
Volume 11 Numbers 1 & 2 (June 2007–December 2008) [Issue 33], pp. 195–204.

Adam Olszewski, Jan Woleński, and Robert Janusz, editors
Church's Thesis after 70 Years
Frankfurt: Ontos Verlag, 2006
551 pp. ISBN 3938793090
Editor's note: The 2007 edition is illustrated.

REVIEW

R. GREGORY TAYLOR

Early on, a certain consensus established itself regarding the Church–Turing Thesis (CT), according to which every effectively computable function is Turing-computable (partial recursive). Nonetheless, a range of views regarding its status has developed over the intervening decades, and this anthology provides a timely survey. It consists of twenty-two articles by twenty-five different authors. The articles range in length from six to fifty-three pages and concern issues in the philosophy of mind, philosophy of mathematics, history of mathematical logic, theory of computation, or theory of programming languages. Only three of the papers—those by Blass and Gurevich, Odifreddi, and Sieg—have appeared previously.

The quality of the articles published here is very uneven. Five out of twenty-two—Bridges, Fitz, Horsten, McCarty, and Turner, to name names—are excellent in our opinion and even well written. Most of the others have something of interest. But a few range over very familiar territory without offering any clear point of view. Some—even one of our favorites—are patently the result of appending this or that regarding CT to a rehearsal of previously published ideas—however interesting—so as justify inclusion in this anthology. There are occasional problems with English. Oddly, the articles appear in lexicographic order using (first) author's last name with the result that the volume is devoid of internal organization. A two-paragraph preface helps not at all in this regard, and there is a name, but no topic, index.

We shall say at least a few words regarding each article and, along the way, group them thematically indicating dependencies. (Our classification is admittedly somewhat arbitrary due to the fact that several of the articles fall under at least two of our headings.) We start with the philosophy of mind.

I. *Minds, Machines, and (Hyper)Computation*

(1) B. Jack Copeland, *Turing's Thesis*, pp. 147–74.
(2) Andrew Hodges, *Did Church and Turing Have a Thesis about Machines?*, pp. 242–52.
(3) Darren Abramson, *Church's Thesis and Philosophy of Mind*, pp. 9–23.
(4) Selmer Bringsjord and Konstantine Arkoudas, *On the Provability, Veracity, and AI-Relevance of the Church–Turing Thesis*, pp. 66–118.
(5) Stanisław Krajewski, *Remarks on Church's Thesis and Gödel's Theorem*, pp. 269–80.

Logicians of the thirties used the term 'computability' to mean 'effective computability' exclusively, and 'computing machine' likewise meant a physical device mimicking the activity of human computers. The articles of Copeland and Hodges concern their recent debate regarding whether Church and Turing considered *hypercomputation* possible, whereby computation exceeding the capacity of a universal Turing machine is intended. According to Hodges both men excluded this possibility at least initially, holding that any physically computable function is Turing-computable (henceforth PCT). Copeland disagrees. According to him Church's pronouncements are noncommittal with respect to this issue, whereas Turing's remarks regarding 'digital computers with a random element' as well as machines that learn indicate that the behavior of such a machine might surpass that of any Turing machine.

Searle's Chinese Room Argument may be taken to show that our mental life—specifically, understanding Chinese—involves hypercomputation and, by CT, thereby goes beyond what is effectively computable. This in turn runs counter to the stronger claims of the Artificial Intelligence community, which was Searle's target. Abramson begins by arguing, following Copeland's work elsewhere, that a certain generalization of Searle's original argument conflates PCT with CT. Otherwise, he evaluates several arguments, based on other thought experiments, to the effect that human minds hypercompute. All these arguments fail according to Abramson.

In the longest article within the collection, authors Bringsjord and Arkoudas argue against Mendelson's claim that CT is a potentially provable proposition and consider arguments against its falsity and as well as objections to those arguments. In a final section the question whether CT entails a form of computationalism is taken up, whereby one means the view that cognition is computation. Against Copeland and siding with Searle the authors argue that CT has as consequence a weak version of computationalism according to which brain activity can be simulated by a Turing machine.

Krajewski explores real and apparent links between CT and Gödel's First Incompleteness Theorem. He begins by reviewing Kleene's 1987 proof of a weak form of Gödel incompleteness that only appears to rely on CT. A second section discusses informal use of CT and its contrapositive "in connection with Gödel's Theorem." But the latter is playing no role here that we can see, and a strained analogy between CT and Gödel's Completeness Theorem is unsuccessful in our opinion. (What does the author mean by "translation of algorithms to the language of [number-theoretic] functions" on page 274?) A final section discusses the possible role of CT in Lucas–Penrose arguments.

II. *Alternative Formulations of* CT

(1) Charles McCarty, *Thesis and Variations*, pp. 281–303.
(2) Leon Horsten, *Formalizing Church's Thesis*, pp. 253–68.
(3) Douglas S. Bridges, *Church's Thesis and Bishop's Constructivism*, pp. 58–65.

In just what sense can a pocket calculator be said to compute the addition function given that permissible addends cannot exceed twenty decimal digits, say? This is the question that motivates McCarty's contribution. The problem with a counterfactual approach lies in delimiting a range of counterfactual circumstances and device behaviors within them that are strictly relevant. It is our calculator qua *logical machine* in Wittgenstein's sense that computes the full addition function, and McCarty's principal goal is formalization of a notion of physical computing device qua logical machine that validates both modal and nonmodal versions of Church's Thesis. Given denotational semantics for programming languages and the fact that there are but finitely many physical computing devices, he conjectures the existence of a primitive recursive predicate $M(e, m, n, p)$ having, roughly, the following property: for each device D, there is a function $\{e\}$ tracking D's resource requirements such that the function computed by D, assuming device input of length not exceeding $\log_{10} m$ and device output of length not exceeding $\log_{10} n$, takes m to n just in case $\exists p(M(e, m, n, p))$ holds. (We are to think of p as (the encoding of) a program with input m and output n.) Ultimately, the following modal version of Church's Thesis holds in the intended interpretation of a certain extension of

PA^2 with modal operators $\Box$ and $\Diamond$:

$$(\Box CT) \qquad \forall f \exists e \Box \forall m, n(\Diamond(\langle m, n\rangle \in f) \leftrightarrow \exists p(M(e, m, n, p))).$$

(Roughly, $\Box$CT says that each function computed by a physical device qua logical machine has a tracking function with index e such that, given any resource circumstance and any natural numbers m and n, there is a circumstance involving (possibly) expanded resources whereby $f(m) = n$ just in case $\exists p(M(e, m, n, p))$ holds.) Certain nonmodal structures can be derived from the modal structure (interpretation) mentioned above, and a nonmodal version of Church's Thesis holds in them (Corollary p. 294). McCarty takes his findings to justify investigation of modal arithmetic. In particular, his results suggest that when counterfactual features of physical devices concern anything but quantity of resources, then $\Box$CT fails.

Most, but not all, philosophers and logicians hold that CT is not mathematically provable simply because it is not a purely mathematical proposition. On the other hand, the situation would likely change greatly if CT were to be expressed in a suitably interpreted formal language. This would mean variables ranging over algorithms, which lack clear identity criteria. In light of this problem Horsten's stated goal is to seek approximations of CT, regarded as hypothesis, in formal contexts and to then see what propositions can be derived from the approximations. CT is approximated in an intuitionist setting by the schema

(ICT)

$$\forall x \exists y A(x, y) \to \exists e \forall x \exists m \exists n[T(e, x, m) \wedge U(m, n) \wedge A(x, n)],$$

where $A(x, y)$ is any formula of the language of first-order intuitionist arithmetic. Very briefly, ICT says that any method witnessing the truth of its antecedent may be uniformly transformed into a Turing machine (sequence of equations) e whose computation determines that $A(x, n)$ holds. Horsten argues that reasoning regarding ICT provides weak evidence that CT is conservative over PA and thus that informal appeals to CT in recursion-theoretic arguments are nonessential.

S. Shapiro's intensional (or epistemic) mathematics S4PA comprises PA with defining equations for primitive recursive functions together with the S4 axioms of modal logic formulated in the language of arithmetic with operator $\Box$. In this setting, Horsten considers

(ECT)

$$\Box \forall x \exists y \Box A(x, y) \to \exists e \forall x \exists m \exists n[T(e, x, m) \wedge U(m, n) \wedge A(x, n)],$$

where now $A(x, y)$ is any formula of the language of S4PA and operator $\Box$ is taken to mean "it is (reflexively) provable that." As it stands, ECT does not invoke the notion of algorithm. So Horsten suggests adopting a thesis (p. 259) stating that any proof witnessing the truth of $\Box\forall x\exists y\Box A(x, y)$ must involve presentation of an algorithm that, given natural number x, finds natural number y such that $A(x, y)$. Again, ECT taken as hypothesis within S4PA has been shown to be arithmetically conservative over PA, which Horsten takes to be weak evidence that the same is true of CT.

In the context of Bishop's constructive mathematics, the effect of Church's Thesis can be obtained by adjoining Richman's Axiom (CPF) stating that there is an enumeration of the set of all partial functions from $\mathbb{N}$ to $\mathbb{N}$ with countable domains. (Of course, "enumeration," "function," and "countable" must be understood constructively.) CPF can be used to show that, as a consequence of Specker's Theorem, there exists a bounded continuous mapping on $[0, 1]$ that is not uniformly continuous. It follows that one cannot hope to show in constructive mathematics unsupplemented by other principles that every bounded continuous mapping on $[0, 1]$ is uniformly continuous. Bridges cites several other examples of the way in which Church's Thesis in the guise of CPF reveals the limitations of constructive mathematics.

III. CT *and Conceptual Analysis*

(1) Andreas Blass and Yuri Gurevich, *Algorithms: A Quest for Absolute Definitions*, pp. 24–57.
(2) Wilfried Sieg, *Step by Recursive Step: Church's Analysis of Effective Calculability*, pp. 456–90.
(3) Stewart Shapiro, *Computability, Proof, and Open-Texture*, pp. 420–55.
(4) Oron Shagrir, *Gödel on Turing on Computability*, pp. 393–419.
(5) Adam Olszewski, *Church's Thesis As Formulated by Church—An Interpretation*, pp. 383–92.
(6) Roman Murawski and Jan Woleński, *The Status of Church's Thesis*, pp. 310–30.

The article of Blass and Gurevich is largely given over to an informal specification of their Abstract State Machine (ASM) model of sequential computation whereby each state of an ASM is a first-order structure. They begin with some remarks regarding CT, which they take to clarify the notion of computable function without fully determining the concept of an algorithm since "there is more to an algorithm than the function it computes." They have in mind a broad class of interactive algorithms including randomized algorithms, nondiscrete algorithms such as ruler-and-compass algorithms, as well as algorithms on abstract structures such as finite graphs. According to the authors, Kolmogorov–Uspensky machines followed by the pointer machines of Knuth and Schönhage come closer to capturing the class of sequential algorithms than does Turing's model. But it is left to the ASM model to formalize the full concept of sequential algorithm.

Sieg's oft-cited paper sets out to explain why Church's first published formulation of CT appeals to Gödel's notion of general recursiveness rather than Church's own notion of λ-definability. The latter's letters to Bernays in particular shed much light according to Sieg. In a final section Church's analysis is compared with that of Turing, and a tenuous analogy with Dedekind on geometric continuity is described. Sieg's article reminds the logic community of something important and forgotten concerning not Church, but rather, Turing, namely, that Turing's seminal paper of 1936 presents an *argument* for the cogency of his model of computation. Sieg's original discussion is found in his earlier "Mechanical Procedures and Mathematical Experience" (in A. George, ed., *Mathematics and Mind*, Oxford, Oxford University Press, 1994, pp. 71–140); a new postscript to the paper in the volume under review summarizes a subsequent shift in Sieg's thinking regarding the role of "Turing's central thesis" within Turing's argument. (According to that thesis any mechanical procedure can be carried out by a human computor satisfying certain determinacy, boundedness, and locality conditions.)

The philosopher Friedrich Waismann noted that most of our empirical concepts are "open-textured" to the extent that "there are always ... directions in which [such concepts have] not been defined" ("Verifiability" in A. Flew, ed., *Logic and Language*, Oxford, Basil Blackwell, 1968, p. 120). Thus Waismann held that there is no need to decide whether Einstein's use of the word "simultaneous" constitutes introduction of an altogether new concept, a change in a concept in current use, or extension of that concept to new cases: the word "time" is not in fact governed by precise rules. Shapiro argues that mathematical concepts such as number, set, computability, and recursiveness (Turing-computability) have been characterized by a similar openness at least in the past. By implication he follows Mendelson in rejecting the idea that CT cannot be proved mathematically because it pairs an imprecise philosophical notion with a precise mathematical notion. Indeed, Shapiro concludes that CT is now "established with as much rigor as anything in (informal) mathematics," crediting the analyses of Turing, Gandy, and Sieg for the "sharpening" of a pre-theoretic notion of computability.

After reviewing Turing's argument for CT, Shagrir reminds us that Gödel initially rejected CT but was later convinced by Turing's work that CT is in fact true. Then, in 1972, Gödel published a very short paper containing a section entitled "A Philosophical Error in Turing's Work" in which he appears to retract earlier endorsements. Shagrir concurs with J. Webb's suggestion that Gödel believed that "all Turing was really *analyzing* was the concept of 'mechanical procedure,' but in his *arguments* for the adequacy of his analysis he overstepped himself by dragging in the mental life of a human computer" ("Introductory Note to Remark 3 of Gödel 1972a" in K. Gödel, *Collected Works*, Vol. II, S. Feferman *et al.*, eds., New York, Oxford University Press, 1990, p. 302). Specifically, Gödel rejected Turing's finiteness constraint on the number of states of mind. But Gödel never ceased to believe that Turing's analysis of computability was on the mark.

Olszewski notes that Church's 1936 formulation of CT may be interpreted in two ways. According to one interpretation it is an empirical proposition identifying two concepts whose extensions are somehow fixed. On a second normative reading, CT stipulates that the concept *effectively calculable function* should henceforth be identified with the concept *partial recursive function*. Olszewski stresses the fact that the effectively calculable functions properly contain the effectively calculable number-theoretic functions. But it seems clear that, even in his abbreviated statements, Church has in mind only the latter. Olszewski bemoans the lack of progress with respect to the standard interpretation of CT as an empirical hypothesis and concludes that the usual formulation of CT is therefore inadequate. But he makes no suggestions for how philosophers and logicians might better proceed.

Murawski and Woleński opt for the standard view that CT is best regarded as an "explication" of an intuitive concept of effectively calculable function. (Their discussion seems to rule out its simultaneously being an empirical claim.) They distinguish several ways in which the analytic–synthetic and *a priori–posteriori* distinctions can be understood and then claim that CT is analytic in one sense and *a priori* in one sense. But their discussion indicates no engagement with a large philosophical literature. For example, following Kant one typically holds that it is true propositions that are analytic (or not) and judgments that are *a priori* (or not). According to Murawski and Woleński, in contrast, it is sentences, specifically theorems of first-order logic, that are analytic (in their "absolute" sense). This means that English "all bachelors are male" fails the test. Further, CT is claimed to be analytic in a theory-relative sense without its every becoming clear what the relevant theory is supposed to be.

IV. *Arguments for and against* CT

(1) Janet Folina, *Church's Thesis and the Variety of Mathematical Justifications*, pp. 220–41.
(2) Elliott Mendelson, *On the Impossibility of Proving the "Hard Half" of Church's Thesis*, pp. 304–309.
(3) Carol E. Cleland, *The Church–Turing Thesis: A Last Vestige of a Failed Mathematical Program*, pp. 119–46.

As for the status of CT, many positions have been staked out: it is held to be proved, provable but not yet proved, potentially provable, unprovable but true, contingent but possibly true, and even false. Folina argues that this situation is the result of CT's being understood in different ways as well as there being different senses in which a proposition can be "proved."

Generally speaking, those who doubt that CT is the sort of thing that can be proved have cited its pairing of a "precise" with an "imprecise" concept. Mendelson asks that we consider the nature of the relativism inherent in this notion of precision: it concerns not the concepts basic to the theoretical context within which a notion is defined but, rather, the actual definition of that notion in the context. He suggests adding to the language of ZF a predicate with intended meaning "is an (effectively) computable function" and then introducing axioms expressing facts regarding such functions. Accordingly, says Mendelson, the concept *(effectively) computable function* would not be imprecise in such a context, and both it and its converse might be provable. (What we are calling CT is what Mendelson refers to as the "hard half" of CT.)

Among the articles collected here that of Cleland, which summarizes her earlier work, comes as the greatest surprise. She argues against the received view according to which Turing's analysis of computability succeeds brilliantly and that, consequently, CT is true. Specifically, she aims to show that "Turing's account is based upon problematic assumptions that are very specific to Hilbert's program" and that "the credibility of the Turing account is significantly diminished if these dubious formalist assumptions are rejected" (p. 121). Cleland correctly sees the impetus for Turing's work in Hilbert's *Entscheidungsproblem*. (Both Church and Turing want to say what counts as a formal proof instance, which involves saying whether a given formula counts as a substitution instance of a given schema.) Cleland places great weight on difficulties that arise when one attempts to realize a given Turing machine physically. We find the details of her discussion puzzling at many points. Further, we can accept her claim, up to a point, that a "mineral crystal could instantiate a Turing machine just as well as a laptop computer" (p. 138) since in our view neither rock nor laptop, both impressively finite, manages to do this in a straightforward manner (see the foregoing review of McCarty). Cleland suggests at one point (pp. 139–40) that the failure of Hilbert's program entails that Turing's account, which it spawned, must be inadequate. But her reasoning here is obscure. She emphasizes physical machines and processes and sees a range of problems in the relations between them and Turing machines. But the implications for CT remain unclear. Why not identify the idealized human computor, whose computational abilities are embodied by a universal Turing machine, with the ideal mathematician whose proof-verification abilities were assumed by Hilbert and Ackermann? Why bring in physical machines and processes?

V. CT *and Physical Computation*

(1) Hartmut Fitz, *Church's Thesis and Physical Computation*, pp. 175–219.
(2) Piergiorgio Odifreddi, *Kreisel's Church*, pp. 353–82.
(3) Jerzy Mycka, *Analog Computation and Church's Thesis*, pp. 331–52.
(4) Karl Svozil, *Physics and Metaphysics Look at Computation*, pp. 491–517.

The *Physical Church–Turing Thesis* (PCT) asserts that a number-theoretic function is effectively computable by a physical system just in case it is Turing-computable. Fitz' article investigates the status of PCT, specifically a variety of attempts to falsify it by presenting alleged instances of hypercomputation on the part of physical systems. The concept of implementation, Fitz argues, is inherently vague although not arbitrary. Further, the concept of physical computation cannot be detached from a human observer; it is not a natural phenomenon but, rather, a mode of description of the behavior of physical systems. Consequently, purely theoretical arguments for observationally inaccessible instances of hypercomputation on the part of such systems miss their mark.

Odifreddi sets himself the ambitious task of surveying the many remarks regarding the Church–Turing Thesis scattered throughout Kreisel's published papers and reviews. Some concern the question whether PCT is true or false (pp. 374–79), and Kreisel's remarks point in both directions.

Mycka reviews three models of analog computation and their relations to one another: the General-Purpose Analog Computer (GPAC) model of Shannon and Pour–El, the Extended Analog Computer (EAC) model of Rubel, and the real recursive function model of Moore. As for CT, Mycka thinks it unrelated to analog computation despite the promise of his title (p. 348). (English "efficiency" is used here whereas "effectiveness" is intended.) In a few final paragraphs devoted to physical systems that may hypercompute, Mycka invokes the possibility that PCT is false. This article was perfectly interesting up to page 348, but its inclusion here is a stretch nonetheless.

Svozil considers plausible examples of hypercomputation—in particular, quantum computation—and concludes, rather weakly, that assertion of PCT "appears highly speculative at least for the time being and maybe forever" (p. 507).

VI. *Church's λ-Calculus and Programming Languages*

(1) David Turner, *Church's Thesis and Functional Programming*

Turner's contribution is an introduction to Church's calculus of λ-conversion and the functional programming languages (Haskell, Miranda, and ML) derived from it. A single-page introduction concerning CT does not really justify its inclusion here—the link between CT and ML, say, is less conceptual than historical. What might justify its inclusion is the quality of this article and the intrinsic interest of its topic.

Manhattan College, Riverdale, NY 10471 USA
E-mail address: gregory.taylor@manhattan.edu