Church's Thesis after 70 Years

Peter Smith

July 11, 2007

In the section 'Further reading', I listed a book that arrived on my desk just as I was sending *IGT* off to the press, namely *Church's Thesis after 70 Years* edited by Adam Olszewski et al. On the basis of a quick glance, I warned that the twenty two essays in the book did seem to be of 'variable quality'.

But actually, things turn out to be a bit worse than that: the collection really isn't very good at all! After I sent my book to press, I gave a paper-by-paper review on my blog, at http://logicmatters.blogspot.com. It is probably more fun to chase up the reviews 'in the wild', so to speak, starting from the entry for May 14, 2007. But here they are wrapped up into a single document, only marginally tidied. Some of the points made here should help further explain and support the general line on the Thesis taken in *IGT*.

1 CTT, minds and supertasks

Since the editors rather unhelpfully refrain from suggesting a sensible reading order, I'm just going to dive in and read the contributed papers in the order they are printed (they are arranged alphabetically by the authors' names). Here goes, then, starting with Darren Abramson, 'Church's Thesis and Philosophy of Mind'.

Abramson identifies 'the Church-Turing Thesis' with the claim '[...] no human computer, or machine that mimics a human computer, can out-compute a universal Turing machine'. That doesn't strike me as a terribly helpful move, for it runs together two claims, namely (i) no human computer can (in a reasonable sense) compute something that is not effectively computable (by a finite, step-by-step, algorithmic process), and (ii) whatever is effectively computable is Turing-computable/recursive. In fact, in *IGT*, I call just (ii) 'the Church-Turing Thesis'. But irrespective of the historical justification for using the label my way (as many do), this thesis (ii) is surely to be sharply separated from (i). For a start, the two claims have quite different sorts of grounds. For example, (i) depends on the impossibility of the human performance of certain kinds of supertask. And the question whether supertasks are possible is quite independent of the considerations that are relevant to (ii).

I didn't know, till Abramson told me, that Bringsjord and Arkoudas have argued contra (i) by purporting to describe cases where people do hypercompute. Apparently, according to them, in coming to understand the familiar pictorial argument for the claim

that $\lim_{n\to\infty} 1/2^n$ is 0 we complete an infinite number of steps in a finite amount of time. Gosh. Really?

Abramson makes short shrift of the arguments from Bringsjord and Arkoudas that he reports. Though I'm not minded to check now whether Abramson has dealt fairly with them. Nor indeed am I minded to bother to think through his discussion of Copeland on Searle's Chinese Room Argument: frankly, I've never felt that that sort of stuff has ever illuminated serious issues in the philosophy of mind that I might care about. So I pass on to the next paper ...

2 What's an algorithm?

Andreas Blass and Yuri Gurevich's paper 'Algorithms: A Quest for Absolute Definitions' really covers too much too fast to be very satisfactory. The first part is a quick review of the separate histories of Church's Thesis and Turing's Thesis, followed by a quick overview of the path from Turing's original analysis of what we might call a classical algorithmic procedure to its generalization in the work of Kolmogorov and Uspenskii, and Schönhage. But they also say

> In fact the notion of algorithm is richer these days than it was in Turing's days. And there are algorithms ... not covered directly by Turing's analysis, for example, algorithms that interact with their environments, algorithms whose inputs are abstract structures, and geometric or, more generally, non-discrete algorithms.

And they go on to describe very briefly – or at least, too briefly for this reader – some of work on more abstract general notions of computation.

But, by my lights, once we go beyond Kolmogorov and Uspenskii we lose touch with discussions that are directly relevant to the Church-Turing Thesis, construed as a claim about effectively computable functions (where the notion of effective computability is elucidated in the traditional way, in terms of what can be done by a sequential, step-by-deterministic-step procedure, where each small step is available to a computing agent of limited cognitive abilities, etc.). And indeed, Blass and Gurevich themselves don't challenge the Thesis: their concern is more with *extensions* of the core concept of an algorithm – or at least, that's how I prefer to describe what they are up to.

3 Constructivism, informal proofs

The third, short, paper in the Olszewski collection is by Douglas S. Bridges – the author, with Fred Richman, of the terrific short book *Varieties of Constructive Analysis*. The book tells us a bit about what happens if you in effect add an axiom motivated by Church's Thesis to Bishop-style constructive analysis. This little paper says more on the same lines, but I don't know enough about this stuff to know how novel/interesting this is.

The fourth paper is 'On the Provability, Veracity, and AI-Relevance of the Church-Turing Thesis' by Selmer Bringsjord and Konstantine Arkoudas. At least these authors get it right about what the core Thesis is: a numerical function is effectively computable if and only if it is Turing-computable. But that's about as good as it gets. The first main section is a bash against Mendelson's argument against 'the standard conception of the thesis as mathematically unprovable'. Now, although I am sympathetic to Mendelson's conclusion, I'd want to argue for it in a rather different way (and do so in the Gödel book). But Bringsjord and Arkoudas's objections just seem badly point-missing about the possibility of a Mendelsonian line. Their argument (p. 69) depends on a bald disjunction between proofs in formal systems and what they call 'empirical evidence' for CTT. But of course, tertium datur. Take, for example, the familiar theorem that there are effectively computable functions which aren't primitive recursive. I'm not being tendentious in calling that a theorem – that's how the textbooks label the result. And the textbooks, of course, give a proof using a diagonalization argument. And it is a perfectly good proof even though it involves the informal notion of an effectively computable function (the argument isn't a fully formalizable proof, in the sense that we can't get rid of the informal notions it deploys, but it doesn't just give 'empirical' support either). Now, the Mendelsonian line is – I take it – precisely to resist that dichotomy formal/formalizable proof vs mere quasi-empirical support and to remind us that there are perfectly good informal proofs involving informal concepts (and Mendelson invites us to be sceptical about whether the informal/formal division is in fact as sharp as we sometimes pretend). And the key question is whether it is possible to give an informal mathematical proof of CTT as compelling as the proof that not all effectively computable functions are primitive recursive. Reiterating the rejected dichotomy is of course no argument against that possibility.

That's not a great start to the paper (though the mistake is a familiar one). But things then go further downhill. For much of the rest of the paper is devoted to a discussion of Bringsjord's claim that membership of 'the set of all interesting stories" (!!) is effectively decidable but not recursively decidable (Gödel number the stories and we'd have a counterexample to CTT). And what, according to Bringsjord, is the rationale behind the claim that members of that set is effectively decidable? 'The rationale is simply the brute fact that a normal, well-adjusted human computist can effectively decide [membership]. Try it yourself!' Well, you might be able to decide in many cases (always? just how determinate is the idea of an interesting story?): but who says that it is by means of implementing a step-by-step algorithm? Effective decidability is a term of art! It doesn't just mean there is some method or other for deciding (as in: ask Wikipedia or use a cleverly tuned neural net). It means that there is an algorithmic procedure of a certain sort; and in trying to judge whether a story is interesting it most certainly isn't available to inspection whether I'm implementing a suitable algorithm. This whole discussion just seems badly misguided.

4 Computability by any means

The next paper is 'The Church-Turing Thesis. A last vestige of a failed mathematical program' by Carol E. Cleland. Oh dear. This really is eminently skipable. The first five sections are a lightning (but not at all enlightening) tour through the entirely familiar story of the development of analysis up to Weierstrass, Dedekind and Cantor, the emergence of a set theory as a foundational framework, the 'crisis' engendered by the discovery of the paradoxes, Hilbert's formalizing response, the *Entscheidungsproblem* as a prompt to the development of a theory of effective computation. No one likely to be reading the Olszewski collection needs the story rehearsing again at this naive level.

And when Cleland comes to the Church-Turing Thesis she without comment runs together two importantly different ideas. On p. 133 the claim is [A] one about the 'effectively computable' numerical functions – which indeed is the version of the Thesis relevant to the *Entscheidungsproblem*. But by p. 140 the Thesis is being read as a claim [B] about the functions which are 'computable (by any means)'. And these are of course distinct claims, requiring distinct arguments. For example, suppose you think that the kind of hypercomputation that exploits Malament-Hogarth spacetimes is in principle possible: then, on that view, there indeed can be computations which are not effective in the standard sense as explicated e.g. by Hartley Rogers, i.e. involving algorithmic procedures which terminate after some finite number of steps. And the questions we can raise about the Hogarth argument are highly relevant to [B] but not to [A].

Cleland's last section offers some weak remarks about whether computation 'by any means' goes beyond Turing computability; but (I'm afraid) nothing here seriously advances discussion of that topic.

5 Effective computability, machine computability

The sixth paper in the Olszewski collection is Jack Copeland's 'Turing Thesis'. Readers who know Copeland's previous writings in this area won't be surprised by the general line: but truth trumps novelty, and this is all done with great good sense. To take up a theme in the last section, Copeland insists that 'effective' is a term of art, and that

> The *Entscheidungsproblem* for the predicate calculus [i.e. the problem of finding an effective decision procedure] is the problem of finding a humanly executable procedure of a certain sort, and the fact that there is none is consistent with the claim that some machine may nevertheless be able to decide arbitrary formulae of the calculus; all that follows is that such a machine, if it exists, cannot be mimicked by a human computer.

And he goes on to identify Turing's Thesis as a claim about what can be done 'effectively', meaning by finite step-by-step procedure where each step is available to a cognitive agent of limited (human-like) abilities, etc. Which seems dead right to me (right historically, but also right philosophically in drawing a key conceptual distinction correctly, as I've said above).

Copeland goes on to reiterate chapter and verse from Turing's writings to verify his reading, and he critically mangles Andrew Hodges's claims (later in this volume and elsewhere) that Turing originally had a wider thesis about mechanism and also that Turing changed his views after the war about minds and mechanisms. I'm not one for historical minutiae, but Copeland seems clearly to get the best of this exchange.

6 Concepts, extensions and proofs

I'll skip the contribution by Hartmut Fitz to return to in the next section, and next look at Janet Folina's 'Church's Thesis and the variety of mathematical justifications' because I'm particularly interested in her main topic, the variety in the idea of proof.

Folina's paper, though, gets off on the wrong foot. She writes: 'Rather than a claim about mathematical objects such as numbers or sets, CT (insofar as it is an assertion) is a claim about a concept. Assessing it requires conceptual analysis.' Which makes it sound as if arguing about CT is like arguing whether the concept of knowledge can be analysed as the concept of justified true belief. But whatever the history of this, nowadays CT is, precisely, a claim about mathematical objects (in the broad, non-Fregean, sense of 'object'): it's the claim that a function is effectively computable if and only if it is recursive – or equivalently, if and only if it is Turing computable. And the truth of that extensional claim doesn't depend on any claim about whether mere conceptual analysis can reveal e.g. that the effectively computable functions are Turing computable.

(Aside: In fact, it is as clear as anything is in this area that, pace Turing, mere conceptual analysis couldn't reveal such an equivalence, since there is nothing in the notion of an effective computation that demands that the 'shape' of our workspace stays fixed during a computation – quite the contrary, we often throw away scratch working, reassemble pages of a long computation etc. – whereas a Turing machine operates on a fixed workspace. It requires a theorem, not conceptual analysis, to tell us that what is computable in a more dynamically changeable workspace is also Turing computable.)

But as I say, Folina's main remarks are about the notion of 'proof' involved in the majority claim that CT is unprovable, and in the minority claim that it is, or might be, provable. She insists that there can be mathematical reasons for a proposition that aren't proofs, properly so called. Fair enough. But she seems to have in mind the sort of grounds we might have for believing Goldbach's conjecture. And it isn't clear to me what she'd say about e.g. the diagonal argument that there are effectively computable functions which aren't primitive recursive. This isn't just a quasi-inductive argument, and we'd indeed normally announce it as a proof even though it doesn't fall under what Folina calls the 'Euclidean' concept of mathematical proof, i.e. 'a deductively valid argument in some axiomatic (or suitably well defined) system', given that it involves the intuitive idea of an effectively computable function.

But say what you like here. For even if we allow Folina to reserve (hijack?) the term 'proof' for the 'Euclidean' cases, the question remains in place whether there can be an argument for CT which is as rationally compelling as the argument that shows that there are effectively computable functions which aren't p.r., or whether ultimately the

grounds are more like the quasi-inductive reasons that support a belief in Goldbach's conjecture. And that's the real issue at stake about the status of CT (not whether we call such an argument, if there is one, a proof).

7 Physical computability

On to 'Church's Thesis and physical computation' by Hartmut Fitz, and 'Did Church and Turing have a thesis about machines?' by Andrew Hodges. But I'll be very brief (and not very helpful).

Both papers are about what Fitz calls the Physical Church-Turing Thesis (a function is effectively computable by a physical system iff it is Turing machine computable), and its relative the Machine Church-Turing Thesis (a function is effectively computable by a machine iff it is Turing machine computable). Hodges argues that the founding fathers, as a matter of historical fact, endorsed MCT. I've already noted, though, that Copeland in his piece has vigorously and rather convincingly criticized Hodges's line, and I've nothing more to add.

Fitz, however, isn't so much interested in the historical question as in the stand-alone plausibility of MCT and PCT. He covers a lot of ground very fast in his discussion: some of what he says I found obscure, some points look good ones but need more development, and I'm not sure I'm getting a clear overall picture. But in any case, I can't myself get very worked about MCT and PCT once we've granted that neither is implied by the core Church-Turing Thesis, so I'm probably not paying Fitz quite enough attention. Anyway, I'm cheerfully going to skip on to the next papers.

8 Formalizing the Thesis

On, then, to the tenth paper in *Church's Thesis After 70 Years*, Leon Horsten's 'Formalizing Church's Thesis'.

Horsten explores two frameworks in which we can write down what look to be, in some sense, partial 'formalizations' of CT. Take first an intuitionistic framework. And consider a claim of the form $\forall x \exists y Axy$. Intuitionistically, Horsten suggests, this 'means that there is a method for finding for all x at least one y which can be shown to stand in the relation A to x. And this seems close to asserting that an algorithm exists for computing A.' But then, Church's Thesis tells us that there should be a recursive function that, given x, will do the job for finding a y such that Axy. So by an appeal to Kleene's Normal Form Theorem, it might seem that in the intuitionistic framework Church's Thesis implies this, call it ICT:

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

where T and U express the familiar p.r. Kleene functions.

Now, Horsten has seemingly sane things to say about why, however, ICT doesn't really capture CT – and so the common intuitionistic rejection of ICT as too strong

isn't really a strike against CT. But that isn't a novel thought. What was news to me is a theorem that Horsten credits to Anne Troelstra. Take Heyting Arithmetic plus ICT; then if this proves $\delta(\varphi)$, where this is the double-negation translation of φ, then Peano Arithmetic proves φ. Which is kind of cute. But what is the significance of that? Horsten's comment is: 'This conservativity phenomenon can be seen as a weak form of evidence for the thesis that CT is conservative over classical mathematics.' Why so?

OK, we often argue to some formal conclusion informally, using the assumption that a certain function is computable, and appeal to CT to avoid (i) bothering to prove the function in question is recursive and then (ii) turning our sketched argument into a stricter proof for the same formal conclusion. The assumption is that we can always do away with such an labour-saving appeal to CT: call that, if you like, the assumption that CT is conservative over classical mathematics. But it is equally, of course, the assumption that CT is *true*. For if CT weren't conservative, there would be a disconnect between claims about computability and claims about recursiveness and CT would be false. And conversely, if CT were false (in the only way it can be false, by there being a computable but not recursive numerical function) then it wouldn't be conservative: e.g., trivially, if f were such a computable but not recursive function, then an appeal to CT followed by KNFT would deduce a false equivalence between $f(x) = y$ and $\exists e \exists m[T(e, x, m) \wedge U(m, y)]$. So Horsten's claim comes to this: Troelstra's theorem is evidence that CT is true.

But is it? After all, suppose the theorem had failed: would that have dented our confidence in CT? Surely not: we'd just have concluded that ICT – which even intuitionists reject as too strong – really is far too strong (compare other cases where wildly optimistic constructivist assumptions can prove classically unacceptable claims, e.g. about the continuity of functions). But if failure of the theorem wouldn't have dented the classical mathematician's confidence in CT, in what sense does its success give any kind of evidential boost for CT?

After talking about 'formalizing' CT in an intuitionistic framework, Horsten goes on to discuss briefly an analogous shot at formalizing CT in the context of so-called epistemic arithmetic. Now, I didn't find this very satisfactory as a stand-alone section: but if in fact you first read another paper by Horsten, his 1998 *Synthese* paper 'In defense of epistemic arithmetic', then things do fall into place just a bit better. EA, due originally to Shapiro, is what you get by adding to first order PA an S4-ish modal operator $\Box$: and the thought is that this gives a framework in which the classicist can explicitly model something of what the intuitionist is after. So Horsten very briefly explores the following possible analogue of ICT in the framework of EA:

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

But frankly, that supposed analogue seems to me to have very little going for it (for a start, there look to be real problems understanding the modality when it governs an open formula yet is read as being something to do with knowability/informal provability). Horsten's own discussion seems thoroughly inconclusive too. So I fear that this looks to be an exercise in pretend precision where nothing very useful is going on.

Horsten's next section on 'Intensional aspects of Church's Thesis' is unsatisfactory in another way. He talks muddily about 'an ambiguity at the heart of the notion of algorithm'. But there is no such ambiguity. There is the notion of an algorithmic procedure (intensional if you like); and there is the notion of a function in fact being computable by an an algorithmic procedure (so an extensional notion in that, if f has that property, it has it however it is presented). The distinction is clear and unambiguous.

9 Kleene and Gödel

Let's move on, then, to the next paper, in the Olszewski collection, 'Remarks on Church's Thesis and Gödel's Theorem' by Stanislaw Krajewski. This is too short to be very helpful, so I won't comment on it – except to say he mentions the unusually overlooked argument from Kleene's Normal Form Theorem and (a dispensable use of) Church's Thesis to Gödel's Theorem that I give in *IGT*. The argument, as Krajewski points out, is due to Kleene himself, a fact which I now realize I didn't footnote in my book. Oops!

10 Precision and pretension

We are half way through, and things aren't going brilliantly! The next paper – Charles McCarty's 'Thesis and Variation' – doesn't exactly raise my spirits either. The first couple of sections are pretentiously written, ill-focused, remarks about 'physical machines' and 'logical machines' (alluding to Wittgenstein). The remainder of the paper is unclear, badly expounded, stuff about modal formulations of CT (in the same ball park, then, as Horsten). Surely, in this of all areas of philosophy, we can and should demand direct straight talking and absolute transparency: and I've not the patience to wade through authors who can't be bothered to make themselves totally clear.

At least the following piece, five brisk sides by Elliott Mendelson, is clear. He returns to the topic of his well known 1990 paper, and explains again one of his key points:

> I do not believe that the distinction between 'precise' and 'imprecise' serves to distinguish 'partial recursive function' from 'effectively computable function'.

To be sure, we offer more articulated definitions of the first notion: but, Mendelson insists, we only understand them insofar as we have an intuitive understanding of the notions that occur in the definition. Definitions give out at some point where we are (for the purposes at hand) content to rest: and in the end, that holds as much for 'partial recursive function' as for 'effectively computable function'.

Mendelson's point then is that the possibility of establishing the 'hard' direction of CT can't be blocked just by saying that the idea of a partial recursive function is precise, the idea of an effectively computable function is isn't, so that there is some sort of categorial mismatch. (Actually, though I take Mendelson's point, I'd want stress a somewhat different angle on it. For CT is a doctrine about the co-extensiveness of two concepts. And there is nothing to stop one concept having the same extension as

another, even if the first is in some good sense relatively 'imprecise' and the second is 'precise' – any more than there is anything to stop an 'imprecise' designator like 'those guys over there' in the circumstances picking out exactly the same as 'Kurt, Stephen, and Alonzo'.)

As to the question whether the hard direction can actually be proved, Mendelson picks out Robert Black's 'Proving Church's Thesis', *Philosophia Mathematica* 2000, as the best recent discussion. I warmly agree, and I take up Robert's story in the last chapter of my book.

11 The status of the Thesis, once more

As you can imagine, it is going to be pretty difficult, by this stage in the game, to say something both novel and illuminating about 'The status of Church's Thesis'. And certainly, Roman Murawski and Jan Wolenski don't really succeed in their paper of that title.

They don't help their cause by seeming to vacillate from the start about how to understand Church's Thesis itself. Their official story is clear enough: CT is the claim that a function is effectively computable if and only if it is (partial) recursive. Fine: by my lights, that's the right thing to say. But note, this makes CT a doctrine in the realm of reference. But then Murawski and Wolenski immediately offer, without any comment on the mismatch, quotations from Kalmar and Kleene which talk of CT as stating 'the identity of two notions' or as supplying 'previously intuitive terms ... with a certain precise meaning' – talk which is only really appropriate if CT is taken as a doctrine in the realm of sense. Of course, our belief that the extensions of 'effectively computable function' and 'recursive function' are the same may well be grounded in beliefs about the relation between the senses of those expressions: but we surely ought to be clear here when we are talking about sense, and when we are talking about reference.

Anyway, Murawski and Wolenski proceed to distinguish four lines that have been taken about the status of CT, of increasing plausibility according to them: (1) it is an empirical hypothesis, (2) 'is an axiom or theorem', (3) it is a definition, (4) it is an explication.

But they seem just confused in supposing that (1) people have supposed that CT – as they have officially defined it – is an empirical hypothesis. Rather, it is claims like 'every humanly computable function is recursive', or 'every function computable by a physically realizable machine is recursive', that have been thought to be empirical. And as I've stressed before, CT is to be distinguished from those quite different claims. So put (1) aside.

Now, as Mendelson famously stressed, the easy half of CT looks to have a perfectly good informal proof (it is an informal theorem, if you like). Running the argument for total functions: the initial functions are plainly effectively computable; definitions by composition and recursion preserve effective computability; and definition by regular minimization preserves effective computability too. But all total recursive functions are definable from initial functions by composition, recursion and/or regular minimization.

So all total recursive functions are effectively computable. Mutatis mutandis for partial recursive/partial computable functions. QED. So if we are to reject (2) we need a good reason to suppose that it is impossible to run a comparable argument for the difficult half of CT. But Murawski and Wolenski change the subject twice. First, they talk about what would be needed for a formal proof of CT and the problem of showing that 'the axioms of the adopted axioms for computability do in fact reflect exactly the properties of computability (intuitively understood)' which looks too much like the problem of establishing CT. But whatever the force of those remarks about formal provability, they don't in themselves sabotage the possibility of an argument for the hard half of CT which, while informal, is as cogent as the familiar argument for the easy half. Then secondly, Murawski and Wolenski go off at a tangent talking about the role of CT in proofs: but that's irrelevant to (2).

We can skip over (3) as such remarks about CT as a definition that we find in the Founding Fathers actually don't seem to distinguish the thought from (4). The real options do indeed seem to be (2) – understood as a claim about there being an informal proof for CT – versus (4). Murawski and Wolenski worry a bit about the kind of explication CT could provide, given that they say 'it replaces a modal property of functions by one that is non-modal'. But while there may be worries about the notion of explication here, that isn't one of them. For 'f is effectively computable' is in the current context equivalent to 'there is an algorithm which computes f'. So there isn't anything excitingly modal going on.

12 Kreisel, Church

I'm going to pass over the next three papers in the Olszewski collection pretty quickly, for various reasons.

First, there's a piece on 'Analog computation and Church's Thesis' by Jerzy Mycka: I just don't know anything about this stuff, and am certainly not in a position to comment on this piece, though it looks decent enough.

Then there is a paper by Piergiorgio Odifreddi on 'Kreisel's Church'. Now, I find Kreisel's writings intriguing and irritating in roughly equal measure (all sorts of interesting things seem to be going on, but he just can't or won't write flat-footed technical exposition and direct, transparent, plain-spoken, philosophical commentary). So I'd hoped that Odifreddi might take up one or two of Kreisel's themes from his discussions of CT and give them a careful development and exploration – for Odifreddi's wonderful book on Recursion Theory shows how clear and judicious he can be. But regrettably, he does something quite different: he takes us on a lightening survey of Kreisel's many relevant articles, with a lot of quotations and some references to related later work. But if you were puzzled by Kreisel before, you'll stay pretty puzzled – though you'll have a longer bibliography!

Next, Adam Olszewski contributes a short piece on 'Church's Thesis as interpreted by Church'. Here Olszewski does at least pick up on the point that I noted that Murawski and Wolenski slurred over without comment. CT is nowadays usually taken to

be a claim about the co-extensiveness of the two notions of effective computability and recursivess; but the Founding Fathers were wont to talk of the notions being identical, or of one notion being a definition of the other. Church in 1936 himself uses both identity talk and definition talk. But actually, it isn't too likely that – at that early date – Church had a clearly worked out position in mind on the status of the correlation he was proposing between computability and recursivess, and I don't think we can read too much into his choice of words here. The headline is that Church, at least originally, seemed to take the modest view of the correlation as having something like the status of a Carnapian explication, while Turing thought, more boldly, that it could be forced on us by an analysis of the very notion of a computation. This is familiar stuff, however, and Olszewski's brief remarks don't in any way change the familiar picture.

13 Gödel on Turing

Although initially Gödel was hesitant, by about 1938 he is praising Turing's work as establishing the 'correct definition' of computability. Yet in 1972 he writes a short paper on undecidability, which includes a section headed 'A philosophical error in Turing's work', in which Gödel takes issue with Turing's implicit assumption of the finitude of the human mind. Question: is this a change of view on Gödel's part (from seeing Turing's work as successful to seeing it as fatally flawed)?

Surely not. What Gödel was praising in 1938 was Turing's analysis of a finite step-by-step computational procedure. (Recall the context: Turing was originally fired up by the *Entscheidungsproblem*, which is precisely the question whether there is a finitistic procedure that can be mechanically applied to decide whether a sentence is a first-order logical theorem. So it is analysis of such procedures that is called for, and that was of concern to Gödel too.) What the later Gödel was resisting in 1972 was Turing's further thought that, in the final analysis, human mental procedures cannot go beyond such finite mechanical procedures (he was inclined to think that, in the words of his Gibbs Lecture, the human mind 'infinitely surpasses the powers of any finite machine'). There is no change of mind, just a change of topic.

That's at any rate what I have previously taken to be the natural view, without doing any careful homework to support it (but it is of course the view that fits with the important distinction I've been stressing in these comments, between CT as a claim about effective computability, and bolder theses about what can be computed by machines and/or minds). And so it is very good to see that now Oron Shagrir has given a careful and convincing defence of this view, with lots of detailed references, in his (freely downloadable!) paper 'Gödel on Turing on computability'. Good stuff!

14 Open texture and computability

We've now reached a very nice paper by Stewart Shapiro, 'Computability, Proof, and Open-Texture', written with his characteristic clarity and good sense. One of the few 'must read' papers in the collection.

But I suspect that Shapiro somewhat misdescribes the basic logical geography of the issues in this area: so while I like many of the points he makes in his paper, I don't think they support quite the conclusion that he draws. Let me explain.

There are three concepts hereabouts that need to be considered. First, there is the inchoate notion of what is computable, pinned down – in so far as it is pinned down – by examples of paradigm computations. Second, there is the idealized though still informal notion of effective computability. Third, there is the notion of Turing computability (or alternatively, recursive computability).

Church's Thesis is standardly taken – and I've been taking it – to be a claim about the relation between the second and third concepts: they are co-extensive. And the point to emphasize is that we do indeed need to do some significant pre-processing of our initial inchoate notion of computability before we arrive at a notion, effective computability, that can reasonably be asserted to be co-extensive with Turing computability. After all, 'computable' means, roughly, 'can be computed': but 'can' relative to what constraints? Is the Ackermann function computable (even though for small arguments its value has more digits than particles in the known universe)? Our agreed judgements about elementary examples of common-or-garden computation don't settle the answer to exotic questions like that. And there is an element of decision – guided of course by the desire for interesting, fruitful concepts – in the way we refine the inchoate notion of computability to arrive at the idea of effective computability (e.g. we abstract entirely away from consideration of the number of steps needed to execute an effective step-by-step computation, while insisting that we keep a low bound on the intelligence required to execute each particular step). Shapiro writes well about this kind of exercise of reducing the amount of 'open texture' in an inchoate informal concept and arriving at something more sharply bounded.

However, the question that has lately been the subject of some debate in the literature – the question whether we can give an informal proof of Church's Thesis – is a question that arises after an initial exercise of conceptual refinement has been done, and we have arrived at the idea of effective computability. Is the next move from the idea of effective computability to the idea of Turing computability (or some equivalent) another move like the initial move from the notion of computability to the idea of effective computability? In other words, does this just involve further reduction in open texture, guided by more considerations ultimately of the same kind as are involved in the initial reduction of open texture in the inchoate concept of computability (so the move is rendered attractive for certain purposes but is not uniquely compulsory). Or could it be that once we have got as far as the notion of effective computability – informal though that notion is – we have in fact imposed sufficient constraints to force the effectively computable functions to be none other than the Turing computable functions?

Sieg, for example, has explored the second line, and I offer arguments for it in my Gödel book. And of course the viability of this line is not in the slightest bit affected by agreeing that the move from the initial notion of computability to the notion of effective computability involves a number of non-compulsory decisions in reducing open texture. Shapiro segues rather too smoothly from discussion of the conceptual move from the

inchoate notion of computability to the notion of effective computability to discussion of the move from effective computability to Turing computability. But supposing that these are moves of the same kind is in fact exactly the point at issue in some recent debates. And that point, to my mind, isn't sufficiently directly addressed by Shapiro in his last couple of pages to make his discussion of these matters entirely convincing.

But read his paper and judge for yourself!

15 Three last papers

The next paper in the Olszewski collection is Wilfried Sieg's 'Step by recursive step: Church's analysis of effective computability'. And if that title seems familiar, that's because the paper was first published ten years(!) ago in the *Bulletin of Symbolic Logic*. The historical fine detail is indeed interesting, but not (I think) particularly exciting if your concerns are about the status of Church's Thesis now the dust has settled. So, given that the piece is familiar, I don't feel moved to comment on it further here.

Sieg's contribution is disappointing because it is old news; the last two papers are disappointing because neither says anything much about Church's Thesis (properly understood as a claim about the coextensiveness of the notions of effective computability and recursiveness). Karl Svozil, in 'Physics and Metaphysics Look at Computation', instead writes about what physical processes can compute, and in particular says something about quantum computing (and says it too quickly to be other than fairly mystifying). And David Turner's 'Church's Thesis and Functional Programming' really ought to be called 'Church's Lambda Calculus and Functional Programming'.

Which brings us to the end of the collection. A very disappointing – at times, rather depressing – read, I'm afraid. My blunt summary suggestion to philosophers (for others, perhaps their mileage will vary!): read the papers by Copeland, Shagrir, and Shapiro and you can really give the other nineteen a miss ...

Cite this article. Hansson, S.O. Adam Olszewski, Jan Wolenski, and Robert Janusz (eds): Churchâ€™s Thesis After 70 Years. Erkenn 69, 421â€“425 (2008). https://doi.org/10.1007/s10670-008-9129-7. Download citation. Start by marking â€œChurch's Thesis After 70 Yearsâ€ as Want to Read: Want to Read savingâ€¦ Want to Read.Â Church's Thesis (CT) was first published by Alonzo Church in 1935. CT is a proposition that identifies two notions: an intuitive notion of an effectively computable function defined in natural numbers with the notion of a recursive function. Despite the many efforts of prominent scientists, Church's Thesis has never been disproven. There exists a vast literature concerning Church's Thesis (CT) was first published by Alonzo Church in 1935. CT is a proposition that identifies two notions: an intuitive notion of an effectively computable function defined in natural numbers with the notion of a re Abramson identies â€˜the Church-Turing Thesisâ€™ with the claim â€˜[...] no human com-puter, or machine that mimics a human computer, can out-compute a universal Turing machineâ€™. That doesnâ€™t strike me as a terribly helpful move, for it runs together two claims, namely (i) no human computer can (in a reasonable sense) compute something that is not eectively computable (by a nite, step-by-step, algorithmic process), and (ii) whatever is eectively computable is Turing-computable/recursive. In fact, in IGT, I call just (ii) â€˜the Church-Turing Thesisâ€™.Â On, then, to the tenth paper in Churchâ€™s Thesis After 70 Years, Leon Horstenâ€™s â€˜Formalizing Churchâ€™s Thesisâ€™. Horsten explores two frameworks in which we can write down what look to be, in some sense, partial â€˜formalizationsâ€™ of CT. The Churchâ€“Turing thesis is a statement that characterizes the nature of computation and cannot be formally proven. Even though the three processes mentioned above proved to be equivalent, the fundamental premise behind the thesisâ€”the notion of what it means for a function to be "effectively calculable" (computable)â€”is "a somewhat vague intuitive one".[3] Thus, the "thesis" remains an hypothesis.[3].Â Church uses the words â€œeffective calculabilityâ€ on page 100ff. ^ In his review of Churchâ€™s Thesis after 70 Years edited by Adam Olszewski et al. 2006, Peter Smith's criticism of a paper by Muraswski and Wolenski suggests 4 "lines" re the status of the Churchâ€“Turing Thesis: (1) empirical hypothesis (2) axiom or theorem, (3) definition, (4) explication. Churchâ€™s Thesis and Functional Programming - Free download as PDF File (.pdf), Text File (.txt) or read online for free. David Turner gives a condensed summary of the lambda calculus and functional programming.Â This becomes clearer in Churchâ€™s restatement of the thesis the following year, after he had seen Turingâ€™s paper, see below. For a fuller discussion see Hodges (this volume). Three definitions of the effectively computable functions of the natural num- bers (non-negative integers, hereafter N), developed nearly contemporaneously in the early to mid 1930â€™s, turned out to be equivalent.