Philos Stud
https://doi.org/10.1007/s11098-019-01252-w

Supertasks and arithmetical truth

Jared Warren[1] · Daniel Waxman[2]

Abstract This paper discusses the relevance of supertask computation for the determinacy of arithmetic. Recent work in the philosophy of physics has made plausible the possibility of supertask computers, capable of running through infinitely many individual computations in a finite time. A natural thought is that, if true, this implies that arithmetical truth is determinate (at least for e.g. sentences saying that every number has a certain decidable property). In this paper we argue, via a careful analysis of putative arguments from supertask computations to determinacy, that this natural thought is mistaken: supertasks are of no help in explaining arithmetical determinacy.

Keywords Arithmetical truth · Determinacy · Supertasks

Arithmetical truth *seems* determinate. There seems to be a fact of the matter about arithmetical claims, even those that humans will never prove or refute. However, it's at best unclear how this could be the case. In a recent paper, building on work by Hilary Putnam and Hartry Field, we presented a challenge for the view that arithmetical truth is determinate.[1] One possibility we didn't consider is that

[1] See Field (1994) and Putnam (1980).

✉ Jared Warren
jaredwar@stanford.edu

Daniel Waxman
danielwaxman@ln.edu.hk

1 Stanford University, Palo Alto, CA, USA

2 Department of Philosophy, Lingnan University, Tuen Mun, Hong Kong SAR

Published online: 18 February 2019

determinacy could be explained by computational supertasks. In this paper, we show that this approach fails.[2]

The appeal of supertasks is straightforward: if a computer could perform a countably infinite number of computations in finite time, we could check, for example, Goldbach's conjecture, by setting the computer to check whether each even number greater than 2 is the sum of two (possibly repeated) prime numbers. Either the computer finds a counterexample and halts before the computation finishes, and thus the conjecture is false, or not, in which case the conjecture is true.

The idea of using computational supertasks to secure determinacy often comes up in discussion. One prominent published appearance is in the widely read introduction to Benacerraf and Putnam (1983):

> If we take the stand that "nonconstructive" procedures—i.e., procedures that require us to perform infinitely many operations in a finite time—are conceivable, [Footnote: E.g., if one has an infinite series of operations to perform, say $S_1, S_2, S_3, \ldots$ and one is able to perform S_1 in 1 minute, S_2 in 1/2 minute, S_3 in 1/4 minute, etc.; then in 2 minutes one will have completed the whole infinite series.] though not *physically* possible (owing mainly to the existence of a limit to the velocity with which the physical operations can be performed), then we can say that there does "in principle" exist a verification/refutation procedure for number theory. For instance, to "verify" that an equation $P(x, y, z) = 0$ has a solution using the "procedure", we *check each ordered triple x, y, z of integers*. (Of course, this requires working forever, or else completing an infinite series of operations in a finite time.)…What this shows is: The notion of "truth" in number theory is not a dubious one if the notion of a completed actually infinite series (of, say, definitely specifiable physical operations) is itself not dubious.[3]

More recently, Berry (2014) has appealed to supertask computation in the *epistemology* of arithmetic in an analogous manner. We are happy to grant the potential epistemological utility of supertask computation. Our criticisms will be aimed at the idea that, as Benacerraf and Putnam suggest, supertask computation can be used to explain the determinacy of arithmetical truth.

Benacerraf and Putnam suspected that supertask computation was physically impossible, but more recent work has perhaps called this into question. E.B. Davies has shown that such machines can be constructed in continuous Newtonian universes, and John Earman and John D. Norton have shown, drawing on work by David Malament and Mark Hogarth, that certain relativistic spacetimes allow computers on certain paths to run through infinitely many individual computations in a finite time from the perspective of an external observer.[4] Generalizations of these results by Hogarth show how more complex supertask computers could

[2] See the essays in Salmon (1970) for a collection of classic articles on supertasks.

[3] Quoted from Benacerraf and Putnam (1983), p. 20. Weyl (1949) also hints at use of supertasks in arithmetic, though negatively.

[4] See Davies (2001) and Earman and Norton (1993, 1996).

potentially decide *any* arithmetical problem whatsoever.[5] Without making any detailed claims about what these results do and don't show, we will assume for the sake of argument that supertask computation is physically possible. We mean this to be understood as allowing the *epistemic possibility* that such machines *may one day be actual*.

Before we discuss the supertask approach to determinacy in detail, some important points of clarification need to be made.

First: we are not here concerned to try and settle the larger question of whether or not arithmetical truth is determinate. Our goal is only to evaluate an attempted explanation of the determinacy of arithmetical truth based on the possibility of supertask computations.

Second: our question concerns the determinacy of our *actual* arithmetical language. If a counterfactual practice had grown up around a supertask computer, deference to its calculations could have been built into arithmetical concepts from the very beginning. The determinate truth of the analogue of Goldbach's conjecture in this counterfactual language would be tied to the result of a supertask computation. That arithmetical practice would be partly empirical, since the content of its concepts would be partly fixed by the output of physical computation. However, that practice is not our practice; our arithmetical concepts are not empirically dependent in that way. Our interest in this paper is in whether a supertask computer might be used to argue for the determinacy of our actual arithmetical practice, not in how it could be used to found an *alternative* practice.[6]

Third: we are not concerned to address those who challenge the physical possibility of supertask computations. There are many ways that this might be done. For instance, some might argue that the Malament–Hogarth solutions do not correspond to genuinely physically possible scenarios. And many constructivists and intuitionists reject the mathematical possibility of completed infinite totalities, and thus, presumably reject their physical possibility as well.[7] Whether or not they can do so coherently will depend upon whether they can develop a suitable non-classical spacetime theory vindicating their views, but we will not pursue this here.

Fourth: we waive worries to the effect that the machine might malfunction or get the intuitively "incorrect" answer because of interference or other operational quirks. We want to consider whether, even granting the flawless operation of the machine, the possibility of supertask computation can secure a determinate notion of arithmetical truth.

5 See Hogarth (2004); see also Button (2009) for discussion of how such generalizations relate to the Church–Turing thesis.

6 As a referee has pointed out, our current practice arguably does include dispositions to (a) "take certain combinations of physical data as evidence for the existence of a supercomputer" and (b) "form new mathematical beliefs on the basis of interactions with that apparent supercomputer". We don't deny that in such a scenario, mathematicians would likely take the results of the supertask computations as *evidence* for arithmetical claims. But although they might play such an epistemic role, our point here is that deference to a supertask computer is not part of the basic metasemantics of actual arithmetic. That is to say: the content of our actual arithmetical claims is surely not fixed by the outcome of any empirical process, let alone (merely possible) supertask computations.

7 See Weyl (1949), p. 42, Bishop (1967), and Dummett (1977), for example.

For the sake of focusing our discussion on a familiar case, let's consider how a supertask computer could be used to explain the determinacy of Goldbach's conjecture. Obviously, securing the determinacy of Goldbach's conjecture doesn't suffice for securing full arithmetical determinacy, but we will argue that supertask computers are *explanatorily* impotent even at this first step. Let "$GB(x)$" express that if x is even and greater than two, then x is the sum of two prime numbers. Using this notation, Goldbach's conjecture can be expressed simply as $\forall x GB(x)$.

Suppose that the Goldbach computation runs for two minutes from the perspective of an external observer. There are only two possibilities: either the computation halts or it doesn't.[8] Clearly, if a halt signal is sent, there is a counterexample and the conjecture is false. But what if the computation doesn't halt? Would that establish Goldbach's conjecture?

Not by itself. We know, by familiarity with the mechanism, that the Goldbach computation proceeds by checking each instance in succession. Consequently, the computer's failure to halt gives us reason to accept each of the infinitely many claims—$GB(0), GB(1), GB(2), \ldots, GB(n), \ldots$—and so on, for each standard numeral "n". But the conjecture itself is none of these individual instances; rather, it is the universal generalization that for *every* number x, $GB(x)$. Indeed, while the computation allows us to establish these infinitely many instances in a finite amount of time, in a precise sense it gives us nothing that is not already given by standard arithmetical theories such as Peano arithmetic (PA), for each of the individual instances is, if true, already provable in PA.[9] More is required to establish the truth of Goldbach's conjecture. We could transition from the truth of all of these claims to the truth of $\forall x GB(x)$, using the infinitary *omega rule*:

$$\textbf{Omega}: \frac{\varphi(0), \varphi(1), \varphi(2), \varphi(3), \ldots}{\forall x \varphi(x)}$$

But nothing about the Goldbach computation itself supports *this* step. To move from the truth of each of $GB(0), GB(1), GB(2), \ldots$ to the truth of $\forall x GB(x)$, is to assume that the omega rule is valid.[10] What is doing the explanatory work here is the relevant instance of the omega rule, together with the premises of this instance of the rule. The role of the computation here is only in showing that $GB(0), GB(1), GB(2), \ldots$ are all true. But since if these are true, they are already derivable in PA, the role of the computation is merely epistemological.

The Goldbach computation only "explains" the truth of Goldbach's conjecture if the appropriate instance of the omega rule is valid. And since the computation does nothing to tell us that this instance of the omega rule is valid, the computation itself isn't doing any fundamental explanatory work. The computation merely shows that each of $GB(0), GB(1), GB(2), \ldots$ are true; but if true, these are each derivable from PA. And this, together with the validity of the omega rule, would in itself explain

[8] Earman and Norton (1996), p. 251.

[9] This is because PA (and even weaker arithmetical theories such as Robinson Arithmetic) are Σ_1-complete.

[10] Obviously, for this particular case we could work with a restricted version of the rule requiring that φ contain no unbounded quantifiers.

the truth of Goldbach's conjecture. The supertask computation only enables us to see what follows from PA so as to secure the premises for an omega-inference. With the instances guaranteed by PA, and the omega rule already in hand, there is no explanatory work for the supertask computation to do. So even for Goldbach's conjecture alone, the supertask computation is explanatorily impotent.

The point generalizes to more complex arithmetical sentences. Supertask computations are inert in explaining arithmetical determinacy in general, not just Π_1 claims like Goldbach's conjecture. Closing standard arithmetical theories, such as PA, under the omega rule, results in true arithmetic: the complete and non-recursive first-order theory of the standard model of arithmetic.[11] This means that if the omega rule is valid in our arithmetical language, then arithmetic is determinate. The omega rule, together with PA, is doing all the work. Supertask computation seems irrelevant to explaining arithmetical determinacy.

So far we've assumed that the omega rule was needed to reason from the non-halt of the Goldbach computation to the truth of Goldbach's conjecture. But perhaps there are other routes to the same result. An alternative approach might be to reason directly from the non-halt to the truth of Goldbach's conjecture using mathematical induction:

$$\textbf{Induction :} \ ((\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(S(x))) \rightarrow \forall y \varphi(y))$$

Perhaps the reasoning would run as follows: the computation checks $GB(0)$, and if it checks any number n, it also checks the successor of n, so by induction, we conclude that the computation checks *every* number. Assuming that successfully checking $GB(m)$ gives us reason to think $GB(m)$ true, then we can conclude that Goldbach's conjecture is true on the basis of the non-halt. This approach doesn't require dealing with the omega rule at all.

But does it do any better with respect to explaining determinacy? Not as far as we can see. The situation here is actually slightly worse than before. The machine establishes $GB(0), GB(1), GB(2), \ldots$—and since the conditional here is material, this suffices to establish each of the conditionals, $GB(0) \rightarrow GB(1)$, $GB(1) \rightarrow GB(2)$, $GB(2) \rightarrow GB(3), \ldots$—however, this is not yet enough to establish the crucial induction step $\forall x(GB(x) \rightarrow GB(S(x)))$. To move from the instances to the generalization, you again need to appeal to the omega rule, or something just as strong. So the move to induction is not helpful.

Even waiving this concern and assuming (falsely) that the computation directly establishes the induction step, the move to the conclusion would still require an induction principle involving predicates describing physically existing supertask computers. In order to justify this application, appeal must be made to a powerful principle of induction, for example, fully open-ended schematic induction or even a second-order induction principle.[12] But it is widely argued that strong induction principles of this kind, by themselves, can be used to secure arithmetical

[11] For a proof of this that doesn't require indexing proofs with infinite ordinals, see Tennant (1978). It is worth noting that the result also holds with weak arithmetical theories, such as Robinson arithmetic, in place of PA.

[12] See Kreisel (1967) for relevant discussion.

determinacy.[13] If this is right, then the computation again plays no deep explanatory role; the mathematical principles are themselves sufficient to explain determinacy.

To avoid misunderstanding, let us stress that we are *not* saying that these arguments—using the omega rule and second-order/open-ended induction—are invalid. Nor are we claiming that they are somehow *epistemically* problematic. Rather, our claim is that, as the uses of the omega rule and induction in these arguments illustrates, the supertask computation itself is not doing any *explanatory* work: it does not play an essential role in the ultimate philosophical explanation of the determinacy of arithmetical truth.

Returning to the Goldbach example, we can shed some light on the general mathematical situation here using some basic model-theoretic reasoning. Let's assume that the recursive commitments of our actual arithmetical practice are represented by PA. If Goldbach's conjecture is independent of PA, so that $\mathsf{PA} \nvdash \forall x GB(x)$, then the theory $\mathsf{PA} + \neg\forall x GB(x)$ is consistent just in case PA is consistent (which it must be, since we're assuming it doesn't prove $\forall x GB(x)$). A version of the completeness theorem tells us that every consistent theory has a model, so if PA is consistent and does not prove Goldbach's conjecture, then $\mathsf{PA} + \neg\forall x GB(x)$ has a model. And in model-theoretic terms, to claim that our arithmetical language is determinate is to claim that all of its models are elementarily equivalent: any models have the same first-order theory.[14] Above we assumed that PA didn't entail $\forall x GB(x)$, and assuming that PA is ω-consistent and that the Goldbach computation does not halt, it doesn't entail $\neg\forall x GB(x)$ either. In this situation, PA will have models in which Goldbach's conjecture is true and models in which it is false. The Goldbach computation itself does *nothing whatsoever* to rule inadmissible one of these classes of models. Once again, the instances don't lead to the generalization. They are ruled inadmissible either by using the omega rule or a strong induction principle as constraints on our actual arithmetical practice.[15]

This is why we say that these principles are doing all of the explanatory work with respect to determinacy, even with the Goldbach computation in hand. The computation is a deductive aid, helping us to see, quickly and easily, which basic arithmetical sentences are true, but it offers no help at all in explaining why either the omega rule or strong induction principles are valid. In other words:

[13] Typically by way of a second-order categoricity theorem; see Shapiro (1991) for details. An analogous result is available in the open-ended, first-order setting; for this approach, see Lavine (n.d.), McGee (1997), and Parsons (2008), chapter 8.

[14] We use this notion rather than categoricity, which is stronger. True arithmetic has nonstandard models, for example.

[15] Our discussion concerns only attempts to exploit the computational power of supertask computers to establish determinacy. But, as a referee for this journal has noted, an alternative strategy would be to argue that the existence of Malament–Hogarth machines of the kind envisioned here directly explains the truth of universally quantified claims over infinite physical structures. In essence, this approach appeals to the structure that must be physically realized in order for supertask computation to be possible, as well as the presumed determinacy of the physical vocabulary used to refer to it. Considered as a strategy for securing determinacy this is closely related to the cosmological approach of Field (1994). Accordingly, supertask *computations* aren't playing an essential role in this type of approach.

epistemologically, for poor creatures like us, the supertask computation may be revelatory, but *metasemantically*, it gives us nothing that is not already given to us by PA.

Someone might object that for Π_1-sentences like Goldbach's conjecture, we can prove in set theory or some other metatheory that if $\mathsf{PA} \nvdash \neg\forall x GB(x)$, then $\forall x GB(x)$ is *true*. But this is to assume the ω-completeness of arithmetic, and thus to implicitly appeal to a determinate notion of arithmetical truth. Since this is the very thing we are trying to explain, we cannot use it or related notions in our explanations. Once again: in trying to reason from the Goldbach computation's failure to halt to the truth of Goldbach's conjecture, we find ourselves appealing to general principles which are (*a*) not guaranteed by the computation itself and (*b*) in themselves, sufficient to explain determinacy.

Supertask computers, by themselves, are no help at all in explaining arithmetical determinacy. They are at best useful illustrations that we have dispositions to accept generalizations upon accepting every instance, or to accept a strong type of mathematical induction that applies even to predicates about supertask computers. But if these dispositions are supposed to explain the *validity* of these principles, then *that* is what we need a detailed accounting of in order to explain determinacy. In every case: supertask computations themselves only "explain" arithmetical determinacy by building it in on the ground floor.

Acknowledgements Thanks to Sharon Berry, Hartry Field, Tomi Francis, Casper Storm Hansen, Beau Madison Mount, James Studd, Jack Woods, and two (possibly identical) referees for this journal.

References

Benacerraf, P., & Putnam, H. (1983). Introduction. In P. Benacerraf & H. Putnam (Eds.), *Philosophy of mathematics: Selected readings* (2nd ed.). Cambridge: Cambridge University Press.

Berry, S. (2014). Malament–Hogarth machines and Tait's axiomatic conception of mathematics. *Erkenntnis*, *79*(4), 893–907.

Bishop, E. (1967). *Foundations of constructive analysis*. New York: McGraw-Hill.

Button, T. (2009). SAD computers and two versions of the Church–Turing thesis. *British Journal for the Philosophy of Science*, *60*(4), 765–792.

Davies, E. B. (2001). Building infinite machines. *British Journal for the Philosophy of Science*, *52*(4), 671–682.

Dummett, M. (1977). *Elements of intuitionism*. Oxford: Clarendon Press.

Earman, J., & Norton, J. D. (1993). Forever is a day: Supertasks in Pitowsky and Malament–Hogarth spacetimes. *Philosophy of Science*, *60*(1), 22–42.

Earman, J., & Norton, J. D. (1996). Infinite pains: The trouble with supertasks. In A. Morton & S. P. Stich (Eds.), *Benacerraf and his critics* (pp. 231–259). Oxford: Blackwell.

Field, H. (1994). Are our mathematical and logical concepts highly indeterminate? *Midwest Studies in Philosophy*, *19*(1), 391–429. P. French, T. Uehling, & H. Wettstein (Eds.).

Hogarth, M. (2004). Deciding arithmetic using SAD computers. *British Journal for the Philosophy of Science*, *55*(4), 681–691.

Kreisel, G. (1967). Informal rigour and completeness proofs. In I. Lakatos (Ed.), *Problems in the philosophy of mathematics* (pp. 138–157). Amsterdam: North-Holland.

Lavine, S. (n.d.). *Skolem was wrong*.

McGee, V. (1997). How We Learn Mathematical Language. *The Philosophical Review*, *106*(1), 35.

Parsons, C. (2008). *Mathematical Thought and its Objects*. Cambridge: Cambridge University Press.

Putnam, H. (1980). Models and reality. *Journal of Symbolic Logic*, *45*(3), 464–482.

Salmon, W. C. (Ed.). (1970). *Zeno's paradoxes*. Indianapolis and New York: The Bobbs–Merrill Company Inc.

Shapiro, S. (1991). *Foundations without foundationalism: A case for second-order logic*. Oxford: Oxford University Press.

Tennant, N. (1978). *Natural logic*. Edinburgh: Edinburgh University Press.

Warren, J., & Waxman, D. (forthcoming). A metasemantic challenge for mathematical determinacy. *Synthese*. https://doi.org/10.1007/s11229-016-1266-y.

Warren, J., & Waxman, D. (2016). A metasemantic challenge for mathematical determinacy. *Synthese*. https://doi.org/10.1007/s11229-016-1266-y

Weyl, H. (1949). *Philosophy of mathematics and natural science*. Princeton: Princeton University Press.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.