Title: No Arithmetical Determinacy From Supertask Computations

Long Abstract:

Call ARITHMETICAL DETERMINACY the thesis that any statement in arithmetic is determinately true or determinately false. Benacerraf and Putnam (1984) argue the thesis can be defended by appealing to so-called ‘supertask computations’.

To explain. Consider first Goldbach’s Conjecture: the claim that every even number greater than two is the sum of two primes. The conjecture has a simple form being expressible in arithmetic. And the statement ‘if x is an even number greater than two then x is the sum of two prime numbers’ is also expressible. To fix some terminology, let $\mathsf{GB}(\mathsf{x})$ be a formula in the language of arithmetic abbreviating the latter statement and let $\mathsf{GB}(\overline{\mathsf{n}})$ be the obvious substitution. Goldbach’s Conjecture is then the sentence $\forall \mathsf{x}\ \mathsf{GB}(\mathsf{x})$.

It is easy to set up a computable procedure deciding the truth-value of each $\mathsf{GB}(\overline{\mathsf{n}})$ in a finite time. But suppose. further, it is possible for a Turing-like machine to decide in a finite time *every* $\mathsf{GB}(\overline{\mathsf{n}})$ by, for instance, performing each step faster and faster. It takes half a minute to check $n = 0$, a quarter to check $n = 1$ and so on. It is clear the computation will describe a supertask. And, by running this ‘supertask computation’, we could check in one minute if Goldbach’s Conjecture is either true or false.

Now, suppose Goldbach’s Conjecture is true. Then there is a fact of the matter – to wit, the result of the computation – determining the statement’s truth-value. If this is the case then we should take the result of the computation as a constraint on our arithmetical practice, meaning that in any intended model of arithmetic the Goldbach’s Conjecture is true. This is, Goldbach’s Conjecture is determinately true. (And a parallel argument if the Conjecture is false). As a result, the computation secures the determinacy of Goldbach’s Conjecture. Furthermore, since the procedure can be generalized to formulas of arbitrary complexity, Benacerraf and Putnam argue that supertask computations secure ARITHMETICAL DETERMINACY.

In the talk, I have two main goals. First, I discuss an objection presented by Warren and Waxman (2020) against Benacerraf and Putnam’s argument. Warren and Waxman argue that even if we assume that the supertask computation does decide that every $\mathsf{GB}(\overline{\mathsf{n}})$ is true – this is, the computation establishes that $\mathsf{GB}(\overline{\mathsf{n}})$ is true for every standard numeral $\overline{\mathsf{n}}$ – we still cannot infer that Goldbach’s Conjecture is also true. On the contrary, the Conjecture is the universal sentence $\forall \mathsf{x}\ \mathsf{GB}(\mathsf{x})$; and to pass from every instance of $\mathsf{GB}(\overline{\mathsf{n}})$ to this universal sentence we should, they argue, assume the ω-rule:

$$\omega\text{-rule:}\quad \frac{\varphi(0) \qquad \varphi(\overline{1}) \qquad \varphi(\overline{2}) \qquad \varphi(\overline{3}) \qquad \dots}{\forall x \varphi(x)}$$

But Warren & Waxman observe that nothing in the supertask computation justifies this last step. The ω-rule does not follow from the results of the computation but is instead an extra assumption required to infer the truth of the universal claim. Worse: augmenting Peano Arithmetic (PA) or any other theory extending Robinson’s Arithmetic

with the ω-rule generates the theory of True Arithmetic (TA). And all models of TA have the same first-order theory. Therefore, adding the ω-rule to our arithmetical system generates a stronger theory whose models decide any sentence, of whatever complexity, in the exact same manner. As a consequence, no indeterminacy will arise: any sentence will already be determinately true or determinately false. In sum, for Benacerraf and Putnam's argument to work, Warren and Waxman argue that we must assume the ω-rule which already secures ARITHMETICAL DETERMINACY all by itself, quite independently of any supertask computation. This renders Benacerraf and Putnam's argument redundant.

Though I am sympathetic to their objection, I explain contra Warren and Waxman that the problematic ω-rule is dispensable and therefore does not pose a challenge to Benacerraf and Putnam. In particular, I argue for the following two claims:

(i) I explain how inferring the universal $\forall \mathsf{x}\ \mathsf{GB}(\mathsf{x})$ from the many $\mathsf{GB}(0)$, $\mathsf{GB}(\overline{1})$, ..., can be done inside arithmetical systems weaker than TA by suitably weakening the ω-rule. This shows that Warren and Waxman's objection is conditional on the gratuitous assumption that $\forall \mathsf{x}\ \mathsf{GB}(\mathsf{x})$ is *only* inferred given the introduction of the strong ω-rule:

(ii) More important, I explain how meta-theoretic rules governing the semantics of the universal quantifier already allow us to infer the universal $\forall \mathsf{x}\ \mathsf{GB}(\mathsf{x})$ from the many $\mathsf{GB}(0)$, $\mathsf{GB}(\overline{1})$, ... This again shows that the ω-rule is dispensable blocking Warren and Waxman's objection.

My second aim in the talk is to elaborate a new objection against Benacerraf and Putnam. To explain my argument consider, this time, some Gödel-sentence for PA, like $\mathsf{G\ddot{o}del_{PA}} := \neg\exists \mathsf{x}\ \mathsf{Proof}\ (\mathsf{x}, \overline{\ulcorner \mathsf{G\ddot{o}del_{PA}} \urcorner})$. We know that, on the assumption that PA is consistent, $\mathsf{PA} \nvdash \mathsf{G\ddot{o}del_{PA}}$. This means that for every standard numeral $\overline{\mathsf{n}}$: $\mathsf{PA} \vdash \neg\mathsf{Proof}\ (\overline{\mathsf{n}}, \overline{\ulcorner \mathsf{G\ddot{o}del_{PA}} \urcorner})$. Hence, there is no $n \in \mathbb{N}$ such that n codes a proof of $\mathsf{G\ddot{o}del_{PA}}$.

Nonetheless, consider a supertask computation e which checks each number and looks for (a code of) a proof for the Gödel-sentence. By the preceding, we know that if e runs inside the standard model the computation will never find a proof of the Gödel-sentence. As a result, the supertask computation will say that the Gödel-sentence is true and, by Benacerraf and Putnam's argument, the sentence is determinately true.

Yet, why does the computation never find a proof? This, I take it, is explained by the fact that there is no standard number which codes a proof of $\mathsf{G\ddot{o}del_{PA}}$. After all, the sentence is constructed in such a way that there can be no relevant PA-proof on pain of PA's own inconsistency. But this only passes the buck: why is the computation looking only at standard numbers or, more precisely, why is e running only in the standard model?

Note this question is key. For it is easy to show that there is a non-standard model $\mathfrak{M}$ satisfying $\mathsf{PA} + \neg\mathsf{G\ddot{o}del_{PA}}$ with some non-standard $m \in M - \mathbb{N}$ which is a witness for $\mathfrak{M} \models \exists \mathsf{x}\ \mathsf{Proof}\ (\mathsf{x}, \overline{\ulcorner \mathsf{G\ddot{o}del_{PA}} \urcorner})$. Now, suppose we run e inside $\mathfrak{M}$ instead. The computation goes through each element of the model's domain searching for a number encoding a proof of $\mathsf{Gödel}_{\mathsf{PA}}$. If the computation only looks at the standard part of $\mathfrak{M}$ it never finds such a proof; but, of course, if it 'waits' a non-standard amount of time e will eventually find a non-standard instance falsifying the sentence. And observe that if, again as per Benacerraf and Putnam's argument, the result of a supertask computation is enough to secure truth-determinacy we must conclude that the Gödel-sentence is determinately false.

As a result, I conclude that Benacerraf and Putnam's argument commits us to the existence of arithmetical sentences which are *both* determinately true and determinately false, and hence to contradictions.

Time permitting, I discuss (what I think to be) an intuitive objection to my argument based on Tennenbaum's Theorem and prove it wanting. Somewhat loosely, the objection is the following. The upholder of Benacerraf and Putnam's argument may require that only decision procedures running inside computable models be intended and, therefore, that only these can play a role in matters of truth-determinacy. The upholder will then appeal to Tennenbaum's Theorem, according to which the isomorphism-type containing the standard model is the sole computable. And since the model $\mathfrak{M}$ of $\mathsf{PA} + \neg\mathsf{Gödel}_{\mathsf{PA}}$, inside which the computation e runs, is necessarily non-standard it follows that the upholder can, by using Tennenbaum's Theorem, dismiss this model has having little bearing on truth-determinacy.

Based on a recent result by Hamkins (2016), I argue that appealing to Tennebaum's Theorem is in this case circular. In particular, I explain that in order to use Tennenbaum's Theorem, the upholder must already assume a privileged notion of 'computability' where decisions procedures running inside non-standard models are automatically non-computable. Therefore the upholder is already assuming what she wants to prove: that the model $\mathfrak{M}$ of $\mathsf{PA} + \neg\mathsf{Gödel}_{\mathsf{PA}}$ is not computable, and so she begs the question.

I conclude that supertask computations and Benacerraf and Putnam's argument do not establish ARITHMETICAL DETERMINACY.

References

Benacerraf, P. & Putnam, H. (1984) 'Introduction' In P. Benacerraf & H. Putnam (eds), *Philosophy of Mathematics: Selected Readings* (2nd ed) (Cambridge: Cambridge University Press), 1-38.

Hamkins, J. (2016) 'Every function can be computable!'. In: `http://jdh.hamkins.org/every-function-can-be-computable/`

Warren, J. & Waxman, D. (2020) 'Supertasks and arithmetical truth', *Philosophical Studies*, 177(5), 1275-1282.