Computability, finiteness and the standard model of arithmetic

Horsten (2010, p. 276) considers the following problem:

(Question) How do we manage to single out the structure of the natural numbers as the interpretation of our arithmetical vocabulary?

Horsten submits the following answer:

(TH) The reference of our arithmetical vocabulary is determined by our principles of arithmetic together with our use of an algorithm of computation.

As Horsten stresses, it is important to understand that (TH) is put forward as an answer to the question about HOW we manage to single out the standard model as the intended interpretation of our arithmetical vocabulary. It is not an answer to the question WHETHER we do succeed in doing so. Horsten's proposal is not a reply to sceptical challenges (2010, p. 278) that contend that there is no way to single out the alleged intended structure of natural numbers. He gives for granted that, as a matter of fact, we are able to grasp the structure of standard natural numbers and puts the question of HOW that is possible.

We want to argue that (TH) is overwhelming: it requires more than it is needed.

Indeed, (TH) presupposes the knowledge of the principles of arithmetic, which presupposes, in turn, the knowledge of the arithmetical language *L* (including terms and formulas). Horsten says explicitly that the language *L* at issue is that of first order Peano arithmetic *PA*. The problem arises: how can one understand the inductive definition of the arithmetical language? We will argue that the understanding of *L,* in particular of numerals, rests on an *absolute* (well determined) notion of FINITENESS.

The notion of finiteness that we are talking about here is not a "sophisticated concept" (Horsten 2010, p.285). It is not a set-theoretic notion, nor a notion to be defined in second order logic. It is a basic, not formalizable, primitive notion. Through this, we can regard the usual inductive definition of the formal system of arithmetic as a description of a process generating all the intended syntactical entities. Once these entities have been grasped, the standard model of arithmetic is exemplified by the structure of numerals. It follows that the requirement of "the use of an algorithm of computation" in (TH) is an unnecessary *surplus*. Our answer to (*Question*) is the following:

(TH) We manage to single out the structure of natural numbers through the intuitive notion of FINITENESS, in virtue of which we can understand what the syntactical entities of the formal language of arithmetic are.*

Perhaps Horsten could object that the notion of finiteness can be grasped, in turn, through a training of practical computation. We don't think so. It seems to us that an implicit grasping of the proper notion of computation presupposes an implicit grasping of the notion of finiteness.

First of all, it is part of the notion of "computable procedure" that it consists of finitely many steps. Moreover, computation is performed working on FINITE sequences of signs, in particular on numerals, which can be taken to be FINITE sequences of signs (Field 2001). Anyway, if one concedes that the grasp of finiteness rests on the "use of an algorithm of computation", then it follows that such a use is a ground for understanding the principles of arithmetic so that, again, it is not a *surplus* beyond the understanding of arithmetical principles.

Button and Smith (2012) argue that neither computability nor finiteness can answer sceptical challenges about our ability to isolate the standard model of arithmetic. As there are non-standard interpretations of our arithmetical vocabulary, they argue, there are non-standard interpretations of the theories formalizing our notion of computability and finiteness.

Observe, however, that if the sceptic takes for granted that we have a well-determined arithmetical language and denies that this has a privileged interpretation, then we can reject her position since, as we saw, the standard model is exemplified by the numerals. On the other hand, if the sceptic maintains that the inductive definition of formulas and terms is to be regarded not as a description of a generating process but – in turn – as an axiomatic system that fails to single out the intended structure of the syntactical entities, then the very same notion of *arithmetical language* is undetermined and there is no room for speaking of the interpretations (standard or not) of THE arithmetical language.

References

Button, T. & Smith, P. (2012). The Philosophical Significance of Tennenbaum's Theorem. Philosophia Mathematica 20 (1):114-121.

H. Field, *Truth and the Absence of Fact*, OUP.

L. Horsten (2012). Vom Zahlen zu den Zahlen: On the Relation Between Computation and Arithmetical Structuralism. Philosophia Mathematica 20 (3):275-288.