

## Vom Zählen zu den Zahlen: On the Relation Between Computation and Arithmetical Structuralism<sup>†</sup>

LEON HORSTEN\*

This paper sketches an answer to the question how we, in our arithmetical practice, succeed in singling out the natural-number structure as our intended interpretation. It is argued that we bring this about by a combination of what we assert about the natural-number structure on the one hand, and our computational capacities on the other hand.

In our article ‘Computational structuralism’ [Halbach and Horsten, 2005], Volker Halbach and I argued that the intended models of arithmetic are determined, in part, by our computational practice. The position that we developed and defended was an elaboration of a suggestion made in [Horsten, 2001, p. 126]. I still stand by the main thesis of [Halbach and Horsten, 2005], when appropriately understood. But I am dissatisfied with the way in which we argued our case. The present article is an attempt to do better. The principal aim is to get the view expressed there, at least as I interpreted it, across better.

The most important weakness of [Halbach and Horsten, 2005] is that we did not always keep the question that we wanted to answer firmly in focus throughout the paper. Perhaps this was caused by a lack of confidence on the part of the authors. Anyhow, it resulted in giving undue attention to objections that really missed the mark, and failing to give due attention to aspects of matters that were central to the matter at hand.

I will not critically review the arguments in [Halbach and Horsten, 2005] here. Instead, I will try to tell the story of computational structuralism afresh. And I do not claim that Volker Halbach would agree with the theses or with the arguments that I will formulate in this paper.

<sup>†</sup> Thanks especially to Volker Halbach and to Øystein Linnebo, but also to Hannes Leitgeb, Philip Welch, Walter Dean, Peter Koellner, Felix Mühlholzer, Paula Quinon, and Dan Isaacson for invaluable comments and discussions on the subject matter of this paper. Versions of this paper were presented at the *Trends in Logic* conference in Frankfurt (September 2009), at the workshop on *Computability, Provability, and Structure* in Bristol (March 2010) and at the workshop on Tennenbaum’s Theorem (London 2011). The research for this article was sponsored by the AHRC-funded research project *Foundations of Structuralism*.

\* Department of Philosophy, University of Bristol, 43 Woodland Road, Bristol BS8 1UU, U.K. leon.horsten@bristol.ac.uk

## I

I will not have much to say about structuralism applied to disciplines other than arithmetic. The structuralist credo, applied to arithmetic, can be expressed as follows:

THESES 1. *Arithmetic is about one single structure: the structure of the natural numbers.*<sup>1</sup>

If we accept this thesis, then there are several philosophical questions that arise, among which are the following:

1. How do we know that arithmetic is about one single structure?
2. What is the nature of this single structure?
3. How do we manage to single out this structure?

These questions should not be conflated. Although they are of course related, one answer does not fit all. The first is an epistemological question. The second is a metaphysical question. The third is a reference-fixing question [Horsten, 2010, section 4].

The third question is the only one that will be addressed in this paper. It is not primarily one of term reference. The question is not simply how it comes about, for instance, that the numeral '2' refers to the number 2 rather than the number 37. The question is how we come to be able to constrain what we are talking about when we do arithmetic at least to elements of an isomorphism type. This question is an aspect of a philosophical problem that Hodes expresses eloquently:

... the phenomena of word reference and of speaker reference, and our referential abilities themselves, have a microstructure about which we can give an informative account. ... The challenge to the mathematical-object theorist is: Tell us about the microstructure of reference to, *e.g.*, cardinal numbers. In what does our ability to refer to such objects consist? What are the facts about our linguistic practice by virtue of which expressions in our language designate such objects and the concepts under which they fall or fail to fall? [Hodes, 1984, p. 126]

I will not be concerned with cardinal numbers in general, but with the natural numbers only. And I will pose this question not in the framework of mathematical-object theory, but in the context of mathematical structuralism. So for me, Hodes's question takes roughly the following form:

QUESTION 1. *How do we manage to single out the structure of the natural numbers as the interpretation of our arithmetical vocabulary?*

<sup>1</sup> Here, as well as in Question 1 below, I am using the term 'structure' in a non-committing way, as being neutral between an *ante rem* and *in rebus* interpretation.

In addressing this question, I will adopt the ‘paradigmatic view of scientific theories’, according to which a science, so also arithmetic, is more than a list of axioms. I take arithmetic to be a human *practice*. Ordinary people have for ages been using arithmetic for all kinds of purposes. But at some point a class has emerged that is occupied with arithmetic in a professional way: the mathematicians. Lay people defer to the mathematicians as specialists about the natural numbers. So the mathematicians’ professional activity of computing on the natural numbers and proving facts about the natural numbers forms part of arithmetical practice.

This will turn out to be of some importance in what follows. I do not wish to presume that there is only one way in which the structure of the natural numbers can be singled out (as we will see). But on one traditional way of looking at the problem, it is somewhat mysterious how it can be done at all.

The traditional answer to the reference-fixing question is that we single out structures by what we assert about them. McGee formulates this as follows:

What we learn when we learn mathematical vocabulary, apart from learning how to count and measure, is a body of mathematical theory. What else could the answer be? The meaning of a mathematical term is wholly dependent upon our use of the term . . . and our practical uses of the term aren’t enough to determine the truth values; so what else is left but our use of the term in theorizing? [McGee, 1997, p. 40]

According to the received view, when we try to explicate what we assert about the intended structure of arithmetic and regiment it, we arrive at first-order Peano arithmetic ( $PA^1$ ). McGee goes further and claims that when we accept the principle of mathematical induction, we implicitly commit ourselves to accepting instances of the induction principle that belong to any possible first-order extension of the language of arithmetic. If for every set there is some such extension containing a predicate true of exactly the elements of the set, then this amounts to postulating the full second-order induction axiom. If not, if for instance only denumerably many sets are nameable in some such extension, then this strategy does not suffice to exclude nonstandard models. I will come back later in this article to the strategy of appealing to second-order logic (as is done, for example, in [Shapiro, 2000]). For now, let us go along with conventional wisdom.

Then  $PA^1$  is seen as more than a collection of strings of symbols. It is assumed that  $PA^1$  is intended to be about some domain(s) of discourse, that predicates have subsets of the domain as their extension and so on, and that the logical symbols have their usual fixed interpretation. This is the sense in which we take  $PA^1$  to be interpreted by first-order models.

In fact, this is something that needs to be argued for (and some arguments have been given), but in the dialectic, this is often simply taken for granted. It follows from the compactness theorem that  $PA^1$  has many models, even many denumerably infinite ones, that are not isomorphic with each other. So we have failed to single out the structure of the natural numbers.

Now we could, with Skolem or Putnam perhaps, acquiesce in this conclusion, and deny that arithmetic has an intended interpretation that is unique up to isomorphism [Putnam, 1980]. But I will do the opposite. I will presume that we can isolate the natural-number structure in our referential practice. So I will have no qualms about referring to the natural numbers and operations on them in my philosophical account. The question addressed in this article is how we have managed to refer to the natural-number structure: *how has our reference to the natural-number structure come about?*

## II

We referentially determine the natural-number structure in our arithmetical practice. What we assert — the finished product so to speak — plays an important role. And  $PA^1$  does a decent job of capturing what we assert in arithmetic, at least since the age of Fermat. By focussing on arithmetic since the age of Fermat, it is implied that some induction is assumed. I shall argue that before use of mathematical induction was at least implicit in arithmetical practice — whenever that was — a single structure was probably not singled out by our arithmetic practice. Less than full arithmetical induction is needed for my argumentation to go through.<sup>2</sup> But I will set this matter aside, and for simplicity's sake assume full  $PA^1$ .

As we have seen, asserting  $PA^1$  does not suffice. It merely restricts us to  $PA^1$ -models  $M$  of signature  $\langle D, s_M, +_M, \times_M \rangle$ , say. The extra ingredient whereby we fix the intended structures is related to the fact that numbers are entities that mathematicians calculate on or with. We can calculate scores of arithmetical functions. Addition is one of the most elementary functions that we can calculate. We have learned a simple algorithm that allows us in principle to calculate arbitrarily large sums of natural numbers.

The sense in which addition is algorithmically computable is not to be understood as Turing-computability or  $\mu$ -recursiveness. What is operative here is the pre-theoretical practical sense of computability which has motivated the mathematical definitions of computability. It is a practical notion of computing on strings of symbols. This capacity of algorithmically

<sup>2</sup> See [Halbach and Horsten, 2005, p. 185].

calculating on strings of symbols is itself rooted in a more primitive capacity of engaging in rule-governed behaviour.

It may sound trite that we calculate with or on the natural numbers — I certainly hope it does. But on standard *ante rem* structuralist accounts, this is strictly speaking a category mistake. We (and our abstract *alter egos*: the Turing machines) calculate with or on representations, on numerals, but not on the numbers themselves. After all, in a computation use is made of the internal structure of the objects on which we compute. On an *ante rem* structuralist picture, numbers do not have any internal structure. So there is no room for the structuralist to bring in any constraint of calculability — at least not directly.

It is not enough for the objects of the domain of a model to have internal structure for computation to be possible. Consider the following unintended model  $O$ . The domain of  $O$  consists of a denumerably infinite set of physical objects chosen in an arbitrary way from the universe: a shoe, a cat's hair, a star from the crab nebula, some mud, the mereological sum of the star and the cat's hair, . . . . Every object in the domain plays the role of a standard number, but the successor function is random: the star is the zero, the mud is the number 1, . . . . The operations  $+$  and  $\times$  are defined from the successor function in the usual way. Now even though  $O$  is in fact isomorphic to  $\mathbb{N}$ , the successor function, addition, and multiplication are not computable in any recognisable sense.

For any of the standard *in rebus* structuralist accounts, the system  $O$  is as much an intended system as the system of the standard arabic numerals with the successor, addition, and multiplication relations defined on them in the expected way. Indeed, it would seem that the systems isomorphic to  $\mathbb{N}$  that we can compute on (in any recognisable sense of computation) are quite exceptional. Therefore we may conclude that on standard *in rebus* structuralist accounts, it is impossible to calculate with or on most systems of objects that play the role of the natural numbers.

The objects that we do compute with are structured expressions that belong to notation systems: we call them *numerals*. The numerals that we use (the Hindu-Arabic numerals, the Roman numerals, the binary numerals) are recursively generated. Not all recursive notation systems are intended either. Consider our ordinary Arabic notation system. Now permute the numerals in some highly non-recursive way, so that, e.g., '12' becomes the successor of '231', and so on. Then the successor function and the addition function on this randomised notation system are not computable anymore. The numerical systems that we do successfully use are not only recursively generated; they are ordered in such a way that elementary arithmetical functions are computable on them.

When actual computers calculate, they use token numerals rather than types. This is also what we humans do when we compute on a piece of paper. Presumably there are only finitely many numeral tokens in the

universe; so not every multiplication of natural numbers can in fact be computed. When we idealise from actual computers to Turing machines, we move from token numerals to types. A similar thing happens when we idealise away from actual human computation, leaving the boundedness of our memory behind. But the important thing is that the algorithms that actual and idealised computers and computers follow, exploit the internal structure of the numerals.

Computability can be defined number-theoretically, using Kleene's  $T$ -predicate and  $U$  function symbol, for instance. This allows us to bypass the complication that strictly speaking humans and abstract computing devices compute on strings of symbols. But I propose to take folk wisdom more literally. If we really calculate on numbers, then the domains of the models that are singled out in our arithmetical practice are composed of expressions: symbolic entities that have internal structure, and that belong to a recursively generated notation system with a finite basis.

One thing we have learned at our mother's knee is how to use an algorithm for computing arbitrary sums of integers. The algorithm we use will be dependent in part on which notation system for the natural numbers that we use (Roman, Arabic, binary, ...). But we all use an algorithm for computing sums.

This aspect of our arithmetical practice gives us a handle on the reference fixing problem. Here is my thesis about the way in which we have narrowed down the reference of our arithmetical vocabulary:

THESES 2. *The reference of our arithmetical vocabulary is determined by our principles of arithmetic together with our use of an algorithm for computing sums.*

For one thing, this means that intended models have to be denumerable, otherwise it would not make sense to say that we can compute on them in the ordinary sense of the word. So intended models  $M$  of arithmetic are denumerable models of signature  $\langle D, s_M, +_M, \times_M \rangle$ , say, satisfying the principles of Peano arithmetic and such that there is at least an algorithm for carrying out the operation  $+_M$ .

Tennenbaum's theorem says that all models of Peano arithmetic in which the addition function is Turing-computable, are isomorphic to the standard natural-numbers structure [Tennenbaum, 1959]. So if we assume Church's Thesis, then we can appeal to Tennenbaum's theorem to claim that the models that are intended according to Thesis 2 are all isomorphic to the natural numbers.<sup>3</sup> (Indeed, it is not hard to see that these intended models are pairwise recursively isomorphic.) So there are two factors,

<sup>3</sup> I am here only sketching the argument on the basis of a rough statement of Tennenbaum's theorem. For a more detailed discussion, see [Halbach and Horsten, 2005, §§ 2 and 3].

inherent in arithmetical practice, that conspire so as to enable us to single out the natural numbers structure in our referential practice. Firstly, number candidates form symbolic systems that make our numerical assertions true. Secondly, number candidates are entities that we can calculate with.

Incidentally, settling on an effective method for deciding the successor relation on the domain of notations is not enough to fix the reference of our arithmetical vocabulary, for there exist nonstandard models with a recursive successor relation [Kaye, 1991, p. 157, Exercise 11.10].

Over and above using an algorithm for computing sums, most of us *know* that we are using an effective method for computing sums. But such a reflexive awareness plays no role in the account of reference fixing that was given above. Knowing that we have an algorithm for computing sums is not needed to fix on the structure of the natural numbers: it suffices simply to adopt the algorithm.

### III

If one is of an *in rebus* persuasion, and one is sympathetic to what I have said so far, then the story pretty much ends here. All the appropriate notation systems are on a par. Arithmetic is about any one of these models as much as about any other of them. Or, alternatively, one can say that an arithmetical assertion involves tacit quantifications over all systems that satisfy the relevant structural characterisation.

Since the domain of every intended model is a notation system, the structuralist view that I am proposing has a formalist air to it. Nevertheless, it certainly cannot be identified with standard formalism, for the number one is no more identified with the Arabic numeral ‘1’ than with the Roman numeral ‘I’. Indeed, there is no single ‘real’ number one.

Many *in rebus* structuralists have nominalistic leanings. Nominalistic structuralists will insist that the domains of the intended models invariably consist of concrete entities. Those nominalists who do not eschew modal notions, need not restrict themselves to actual concrete objects: they may throw in possible objects to boot.

But a nominalist line is by no means forced upon the *in rebus* computational structuralist. She may be happy to allow notations to be ‘light’ abstract objects, *i.e.*, abstract objects that are close to concrete entities, such as the expression types of which concrete inscriptions are instantiations. Of course it is then incumbent on such an *in rebus* structuralist to tell a credible story about what it means for humans and machines to compute on or with types of inscriptions.

Suppose one feels comfortable with reifying abstract signs from concrete inscriptions. Then it is but a small step to abstract an *ante rem* — or would it be better to say ‘*post rem*’? — natural-number structure consisting of positions without internal structure and which feature only relations



to other positions in the same structure. Thus we can widen the class of intended models from the recursive isomorphism type generated by the Arabic numerals to the full class of all  $\omega$ -sequences, as Benacerraf urges us to do [1996]. We may even jettison the formerly intended models of arithmetic and insist that arithmetic is exclusively about this single *ante rem* structure. The trade-off is clear. The *in rebus* structuralist faces a choice between two options neither of which is very attractive: she can opt for a slightly dodgy ‘multiple reference’ doctrine of arithmetic, or commit herself to the claim that the logical form of arithmetical statements differs from its surface grammatical form. If arithmetic is about a single *ante rem* structure, then this choice is avoided. But if we say that arithmetic is about the single *ante rem* structure that all systems (abstract and concrete) have in common, then we lose the ability to say that we compute on the natural numbers, for this structure consists of positions without internal structure. We can only compute on the instantiations of the real natural numbers, for only these shadowy creatures sport internal structure.

#### IV

In conversation, Øystein Linnebo posed the following problem for the computational structuralist. If we are allowed to make free use of what we say about the natural numbers and what we are able to do with the natural numbers, then there may be a simpler thesis about how the reference of our arithmetical language is determined:

THESES 3. *The reference of our arithmetical vocabulary is determined by our principles of arithmetic together with our ability to count up to every natural number.*

If Thesis 2 is acceptable, then, it appears, Thesis 3 is acceptable, too. After all, we are able in principle to count up to every natural number. We learn to count out indefinitely long initial segments of the natural numbers at a very early age.

Moreover, Thesis 3 gets the models right, at least up to isomorphism. There presumably is a minimum finite amount of time that it takes to count out a natural number. And the structure of time is that of the reals. So we could not possibly count up to a nonstandard number. (Some sort of caveat may have to be made here for weird space-times with ordinary finite computers spiraling around black holes,<sup>4</sup> that is not clear to me, but let us leave those aside.)

An advantage of Thesis 3 is that it leaves out this business about Tennenbaum’s theorem altogether. In that sense, it seems a much simpler

<sup>4</sup> This scenario was first entertained in [Pitowski, 1990]. For a good overview of the literature to which this gave rise, see [Welch, 2011].



and more straightforward proposal. If anything, it seems to make nailing down the intended models an altogether trivial affair. Also, if matters are as Thesis 3 has them, then we need not assume that we are asserting any induction principles at all in arithmetical practice. It suffices that we accept the recursive axioms of addition and multiplication (and the successor axioms).

We do indeed learn at a very tender age how effectively to count out indefinitely many numbers. We learn that the natural numbers are  $0, 1, 2, \dots, 9, 10, 11, \dots, 100, 101$ , and so on. The question is how the ‘and so on’ should be understood. What is understood by the child is that you can go on as long as you like (using the recursive notation system): you always get natural numbers.

If the teacher is of the strict and rigorous type, then she might even want to go further. In an attempt to make absolutely sure that unintended interpretations of natural-number talk are excluded, she may add, after teaching the children to count out indefinitely many natural numbers:

... and that is all the natural numbers there are. All the natural numbers can be counted up to in this way.

By this assertion the teacher implicitly conveys to her pupils the information that her desk, for instance, is not a natural number; it is not the sort of thing that will ever be counted out in this way.

One may wonder if the teacher’s assertion is not sufficient to exclude all nonstandard interpretations of the child’s idiolect. To repeat, if it is reasonable to suppose that future-directed time is structured as the half-line  $[0, \infty)$ , and if there is a fixed minimal amount of time that is required to count out a number, then only standard numerals can in principle be counted out.<sup>5</sup>

In order for the teacher’s assertion to constrain the admissible interpretations of the child’s idiolect to an isomorphism type, nature must play its part (by having a time dimension that is structured in the right way, and so on). But the child also has to do something: she has to take on board what the teacher says. And for her to do this, she has to understand fully the content of the teacher’s assertion. But the child can only take this content fully on board if she already has the standard natural-number concept. She must already understand that the moments at which numerals are counted out are ordered like the standard natural numbers. She must know what it means that for every natural number there is a finite linearly ordered set of moments such that at the last of these, the number is counted out. But the child does not have the required conceptual machinery yet.

<sup>5</sup> I do not here consider the possibility in principle, if it is one, to count faster and faster, so that before a fixed moment in time all the standard numerals have been counted out.

Suppose that the child can in principle count out arbitrarily long initial segments of the natural numbers. And suppose that for any two numerals, the child is able to tell whether the second of them ‘gives us’ the successor of the first one. Then even if she assents to the standard axioms for addition and multiplication, non-standard models are still possible interpretations of her idiolect. But if we are also convinced that the child has mastered the algorithm for addition, then we must conclude that she has in her language singled out the natural-number structure.

Mastery of the enumeration procedure for counting out the natural numbers is not sufficient in the context of Peano arithmetic to single out the natural-number structure. It does not guarantee that the domain of discourse exhausts the standard natural numbers. How is it different for addition? How does our arithmetical practice guarantee that our addition algorithm is total? The answer is that it is part of the child’s elementary instruction. Her teacher says: ‘In order to find the sum of *any* two natural numbers, this is what you do: ...’. So the mastery of the algorithm for addition entails that the addition operation singled out in the idiolect of the child is total and computable. If it is not total, then the child has not completely mastered the algorithm. If it is not computable, then the child has not mastered an algorithm for it.

The child will never be able to present *conclusive* evidence that she is able to count out arbitrarily long initial segments of the natural numbers, nor that she is capable in principle of adding arbitrary natural numbers. And in this respect, the mathematical community as a whole is no better off than the child of five. Thus a Kripkensteinian sceptic may refuse to admit that we are ever able to single out the natural-numbers structure by anything that we do [Kripke, 1982]. I do not have anything to add to the vast literature on the tenability of this radically sceptical stance. Here, I presume that it is reasonable to attribute to a person the ability to count out arbitrarily large initial segments of the natural numbers and to have an algorithm for addition.

The child need not have any reflective knowledge about her algorithmic addition powers. She just has to know what to do in response to the teacher’s instructions (such as ‘28, 23: add!’). When the child has acquired the right disposition, the admissible interpretations of her natural-number talk are restricted to an isomorphism type.<sup>6</sup> Here, too, there is a part that the world has to do, and a part that the child has to do. The world has to ensure that the algorithm terminates in a finite number of steps; the child has to master the algorithm. But, unlike fully taking on board what the teacher says when she says ‘and that is all the numbers there are’, the child *can*

<sup>6</sup> It is not easy to give an exhaustive philosophical account of what it means to have a disposition to respond to input according to a rule. But that is a philosophical problem that I am not addressing here.

do this without first having to come to understand somewhat sophisticated concepts (such as the concept of finiteness). The child does not need to know that her method always works; she may even not yet possess the concept of a method or the notion of finiteness that is involved in it.

Of course the teacher could, instead of teaching the child how to add, explain from the outset that the natural numbers are given by the smallest collection closed under the recursive operations for generating numerals. I assume that the child can take this on board.<sup>7</sup> And when she does, the admissible interpretations of her idiolect are appropriately constrained. It also allows the child to grasp fully the content of the assertion ‘... and that is all the natural numbers there are’. However, this is not the way things are typically done. For such an explication to have a sufficiently determinate meaning for the child, some elements of set theory have to be in place. Something like the operation of taking intersections of infinite sets needs to be grasped. The teacher can indeed explain this if she so chooses. But I maintain that in our world, the interpretation of the child’s arithmetical vocabulary is not narrowed down in this way.

It is sometimes argued that a mastery of the concept of set is not necessary for something like the above explanation to succeed. Understanding of plural talk in English may suffice to obtain the required result; we may be allowed the benefits of a ‘second-order’ account without committing ourselves to sets [Boolos, 1984]. This line of thought has given rise to an extensive debate in the literature, to which I have little to add. But even if plural talk is not merely talk about collections in disguise, the plural story will have to be structurally similar to the set-theoretic story given above [Linnebo, 2003]. And I do maintain that a story with this structure is usually given to a child only after she has been taught a method for adding natural numbers.

In sum, at the level of speaker’s reference, the intended models typically are restricted to an isomorphism type by first learning to count out the numerals, then to learn how to execute elementary operations algorithmically, and subsequently to accept a degree of mathematical induction. Even though learning how to count out the natural numbers in the natural order comes even earlier in a child’s development than learning how to add and accepting the principle of mathematical induction, it is not this ability to count out the natural numbers that fixes the natural-numbers structure as the reference of the child’s arithmetical vocabulary.

At the level of word meaning, the account that I have been defending has implications for the historical development of the meaning of our arithmetical words. After all, reference fixing is a historical process. The

<sup>7</sup> Of course a philosophical account has to be given of how the child acquires a sufficiently standard concept of set and of operations on (infinite) sets. But the onus is not on me to give this explanation.

descriptive claims entailed by my account are the following. At some very early stage in history, humans constructed recursive notation systems for natural numbers. At some later point, elementary algorithmic operations were introduced (addition, multiplication) and it was realised that they satisfy certain principles. Even at that point the interpretation of arithmetical vocabulary had not been narrowed down to an isomorphism type. This happened only when (relatively modest) mathematical induction principles became accepted principles of mathematical practice. But this does mean that non-standard models were excluded well before Dedekind and others asserted that the full second-order induction axiom governs the natural numbers.

Our determination of the interpretation of the language of mathematical analysis is a wholly different matter. Modern mathematical analysis is a much more abstract discipline than arithmetic. Concerning our determination of the interpretation of our analytical vocabulary, a story involving second-order logic or set theory may well have to be told. But the interpretation of our arithmetical language was already determined before the mathematical community had a passable understanding of set-theoretic notions.

## V

If we did not have an algorithm for computing sums, then we could never have learned how in principle to add arbitrary natural numbers. But we have learned how to compute sums.

How do we *know* that we can algorithmically calculate arbitrary sums? How do we know that all our additions of natural numbers are (finitely) terminating calculations?

The immediate answer is that we can actually prove that they do [Halbach and Horsten, 2005, p. 179]. We take our algorithm for computing sums, translate it in (or regiment it as) a Turing machine, and prove in arithmetic that its computations always finitely terminate.

As noted in Section 2, Church's Thesis is assumed in this line of reasoning. Church's Thesis is widely regarded as correct. Even though it cannot be proved mathematically, it is strongly supported by Turing's analysis of the notion of algorithm [Soare, 1996; Sieg, 1994]. It is generally accepted that Turing's idealisations about the structure of time, memory, . . . , are the appropriate ones. And this means that alternative notions of computation, such as Infinite Time Turing Machine computation, do not model our notion of algorithmic computation correctly.

It is sometimes objected that non-standard models would judge all Turing machine additions to be terminating as well. But the length of some of their 'computations' would be given by a non-standard number. So what value does our 'proof' that all additions finitely terminate really have?

This is a sceptical challenge, albeit a less radical one than the Kripkensteinian scenario that was briefly discussed earlier. It is based on the so-called ‘just more theory’ move in [Putnam, 1980], for it rests on the observation that the theory consisting of first-order Peano arithmetic plus the assertion that addition is recursive, also has non-standard models. It is clear that with respect to this challenge, second-order logic is in the same boat as computational structuralism.

Note in this context that the full proof of the finite termination of calculations of sums is actually not so easily reproduced in first-order logic, because of the fact that informal talk about algorithms is not readily expressed in classical first-order logic. Also, it is not clear how formalising talk about algorithms in first-order logic would be helpful: our informal talk about algorithms is more than uninterpreted first-order theory anyway. And this points us to what is in my view the correct response to the sceptical charge. Our already interpreted mother tongue, of which the language of mathematics is part and parcel, is and must necessarily be rock bottom. We have no alternative but to express our arguments in it, and can only hope to be understood by others.

Be this as it may: the reference-fixing question is what I set out to address, not sceptical challenges. The question was: *given that most of what we think we know about the natural numbers is correct*, how have we managed to fix the reference of our arithmetical vocabulary? To assume that this requires us to answer sceptical challenges would be a mistake.

An analogy may be instructive here. As a philosopher of science you may be interested in unobservable entities. Then it is a legitimate philosophical question how we manage to refer to unobservables. Perhaps some sort of stipulative act of initial baptism (‘water molecules are what *this* stuff consists of’) did the job? Or perhaps we have managed it in some other way. It is not immediately clear that answering this question requires us to answer the question how we know that there are unobservables in the first place. And even if we know that there are unobservables, we may be puzzled about how we have managed to fix reference to them.

## REFERENCES

- BENACERRAF, P. [1996]: ‘Recantation or Any old  $\omega$ -sequence would do after all’, *Philosophia Mathematica* (3) **4**, 184–189.
- BOLOS, G. [1984]: ‘To be is to be a value of a variable (or to be some values of some variables)’, *Journal of Philosophy* **81**, 430–449.
- HALBACH, V., and L. HORSTEN [2005]: ‘Computational structuralism’, *Philosophia Mathematica* (3) **13**, 174–186.
- HODES, H. [1984]: ‘Logicism and the ontological commitments of arithmetic’, *Journal of Philosophy* **81**, 123–149.

- HORSTEN, L. [2001]: 'On our ability to fix intended structures', in: T. Derksen, ed., *Moving Ahead: Philosophy of Mind and Realism*, pp. 121–134. Studies in General Philosophy of Science. Tilburg: Tilburg University Press.
- [2010]: 'Having an interpretation', *Philosophical Studies* **150**, 449–459.
- KAYE, R. [1991]: *Models of Peano Arithmetic*. Oxford Logic Guides. Oxford: Oxford University Press.
- KRIPKE, S. [1982]: *Wittgenstein on Rules and Private Language: An Elementary Exposition*. Oxford: Basil Blackwell.
- LINNEBO, Ø. [2003]: 'Plural quantification exposed', *Noûs* **37**, 71–92.
- MCGEE, V. [1997]: 'How we learn mathematical language', *Philosophical Review* **106**, 35–68.
- PITOWSKY, I. [1990]: 'The physical Church-Turing thesis and physical computational complexity', *Iyyun* **39**, 81–99.
- PUTNAM, H. [1980]: 'Models and reality', *Journal of Symbolic Logic* **45**, 464–482.
- SHAPIRO, S. [2000]: *Foundations without Foundationalism: A Case for Second-Order Logic*. Oxford: Oxford University Press.
- SIEG, W. [1994]: 'Mechanical methods and mathematical experience', in A. George, ed., *Mathematics and Mind*, pp. 71–117. Oxford: Oxford University Press.
- SOARE, R. [1996]: 'Computability and recursion', *Bulletin of Symbolic Logic* **2**, 284–321.
- TENNENBAUM, S. [1959]: 'Non-archimedean models of arithmetic', *Notices of the American Mathematical Society* **6**, 270.
- WELCH, P. [2011]: 'Discrete transfinite computation models', in S.B. Cooper and A. Sorbi, eds, *Computability in Context: Computation and Logic in the Real World*, pp. 375–414. London: Imperial College Press; Singapore: World Scientific.