Strict Finitism's Unrequited Love for Computational Complexity

Noel Arteche

March 28, 2021

Introduction

As a philosophy of mathematics, strict finitism (or ultrafinitism) has been seen as a loose collection of radical ideas stemming from common objections to Hilbertian finitism, concerned with the notion of feasibility, and defended mostly by appealing to the physicality of mathematical practice. In particular, its formalization has influenced (and been influenced by) the field of computational complexity theory.

Historically, feasibility was already a concern for Bernays (1935), and later became a central element of Wittgenstein's middle and late philosophy of mathematics. After that, strict finitism developed on two fronts. On the technical side, Bernays' comments were made formal by Hao Wang and Alexander Yessenin-Volpin. On the philosophical side, Michael Dummett (1959; 1975) argued against Wang and Yessenin-Volpin, claiming strict finitism untenable on the grounds of sorites-type paradoxes. Since then, the ultrafinitistic ideas developed mostly technically, in the work of Parikh (1971) and Sazonov (1994), thanks to their proof-theoretic approach to almost consistent logics and feasible numbers. In parallel, bounded arithmetic developed strongly, though lacking philosophical activity. More recently, Magidor (2007; 2012) and Dean (2018) have independently argued that ultrafinitism can be saved from Dummett's arguments of inconsistency, noting that the ultrafinitists would have disagreed with the way he conceptualized their movement.

In between, Brian Rotman's semiotic approach to mathematics (1993; 1996; 2000; 2006) has put forward a view that has often been granted the name of ultrafinitist. Based on semiotic and physicality arguments along with Wittgensteinian conventionalism, Rotman has sketched an approach to feasible arithmetic named non-Euclidean arithmetic, whose model-theoretic formalization might lead to similar systems of those of Parikh, Sazonov and Dean.

Despite strict finitism being a somewhat underdefined philosophical position, what all the previously named thinkers have in common is the appeal to the vague notion of feasibility that they think is made precise by the field of computational complexity theory. In this paper, I analyse whether complexity theory is a satisfactory framework for strict finitism.

The paper is divided into two parts. The first one introduces strict finitism as a philosophy concerned with feasibility, stemming from the two most common objections to Hilbert's finitism, the type/token objection and the representability objection. In particular, I point at how the need for feasibility has been defended by appealing to physical reality.

With the ideas of feasibility and materiality in hand, the second part of the paper addresses whether computational complexity theory can act as a formal framework for ultrafinitism. I contend that, contrary to popular belief, complexity theory is not what the ultrafinitists think, and that it does not provide a theoretical framework in which to formalize their ideas—at least not while defending the material grounds for feasibility. More precisely, I present three main lines of argument as to why this is the case, and conclude that the subject matter of complexity theory is not proving physical resource bounds in computation, but rather proving the absence of exploitable properties in a search space.

Throughout the paper I assume (very basic) familiarity with computational complexity theory. An extensive philosophically-oriented introduction can be found in (Dean 2019), while a usual full-fledged contemporary reference for complexity theorists is (Arora and Barak 2009). A philosophical discussion of complexity theory can be found in (Aaronson 2011).

1 Feasibility and physicality

The development of strict finitism as a philosophy of mathematics is tortuous, partly due to the great deal of technical work in complexity theory and bounded arithmetic with little to no contact with the philosophical discussion. Tracing and comparing the different types of ultrafinitism is well beyond the scope of this paper. For this text, it will suffice to regard ultrafinitism as a philosophy concerned with feasibility and its relation to materiality. In particular, it is worth seeing how one can derive strict finitism from critiques to traditional finitism.

It is well-known that finitism faces two important objections. Firstly, regarding Hilbert's claim that the subject matter of mathematics is the sign tokens written on paper, one may raise the type/token objection: mathematics cannot be about the concrete tokens written on paper, but about the types of these tokens. Secondly, on the secure epistemological basis of finitistic mathematics, guaranteed by intuition, we may raise the representability objection: even simple finite numbers, like 32 or 124, are difficult to represent in intuition—let alone 2^{256}. It is in these two objections that we find the two pillars of a group of loose ideas often named strict finitism. Namely, the materiality of mathematical practice and the restrictions imposed by the vague notion of feasibility.

The central theme of strict finitism is the latter: feasibility, an immediate consequence of the representability objection. Why should we reason beyond the limits of what is feasible for our intuition? For the ultrafinitist, this means rejecting the potential infinity of the natural numbers. It would be fallacious to believe, as intuitionists and finitists alike, that one can always count up to an ever-increasing number, as we will, sooner or later, reach the limits of our intuition. And not only of our intuition: the limits of the physical universe too. In short, there are numbers, like 2^{200}, that we cannot count up to, due to time, space and energy limits. I shall call arguments appealing to physical reality *physicality* or *materiality arguments*.

After rejecting the potential infinity of the naturals—as well as the actual, needless to say—, the ultrafinitist must address a pressing issue: What numbers are we allowed to use? What numbers are feasible? As Dummett (1975) pointed out, the problem with something being *feasible* (or *small*, or *tractable*) is that it is vague. In an attempt to circumvent vagueness, Parikh (1971) formalized the so-called *feasible numbers*. Essentially, this is a new unary predicate $F(n)$ intended to mean "n is feasible" added to the language of first-order arithmetic, supplied with the axioms

$$\begin{array}{ll} (F_0) & F(0) \\ (F_s) & \forall n(F(n) \to F(s(n))) \\ (F_\ell) & \neg F(2^{200}) \end{array}$$

where (F_ℓ) may be replaced with whatever other obviously infeasible number we may think of. Unfortunately, these are fatal: when added on top of our preferred arithmetical axioms, they render the theory inconsistent. It follows from (F_0) and (F_s) that $\forall n F(n)$, and so in particular $F(2^{200})$, contradicting (F_ℓ). This inconsistency, known after Dummett as Wang's paradox, belongs to the class of the sorites-type paradoxes, resembling the Greek paradox of the heap. On these grounds, Dummett famously claimed strict finitism untenable (Dummett 1975).

However, the question is far from settled. The ultrafinitist will point out that things are rendered feasible or infeasible with respect to something; the problem with feasibility is not that it is a vague notion, but rather a relative one. One must fix the reference relative to which things are tractable or intractable. The strict finitist sets this reference to be physical reality. For them, things are intractable with respect to human abilities or physically realizable computers—the choice does not quite matter, as both have physical constraints. Two possible developments of this view are worth outlining.

Firstly, regarding the inconsistency of the axioms, Parikh (1971) and later Sazonov (1994) named their theories "almost consistent". The key observation here is that deriving the contradiction in the system requires a proof of infeasible length. In this sense, the mathematician cannot derive a contradiction in their lifetime, making the theory consistent in practice. Furthermore, results like Haken's theorem in proof complexity (see next section) suggest that all proof systems are constrained to such feasibility limitations. This position can be attacked on the grounds of metalinguistic concerns explored in the second part of the paper.

Interestingly, there is a second approach to justify feasible numbers. The key insight is that postulating the existence of an infeasible number without naming it does not make the system inconsistent. Rather, the theory is satisfied in a nonstandard model. This is due to Kreisel, following a simple compactness argument[1].

Theorem (Kreisel's theorem)**.** *Let* PA *denote the theory of first-order Peano arithmetic, let* $(F_\exists)$ *denote the axiom* $\exists n \neg F(n)$ *and let* $(F_<)$ *be the axiom* $\forall n \forall m(F(m) \wedge n < m \to F(n))$*. Furthermore, let \$ be a new constant symbol, and let* $(F_\$)$ *denote the axiom* $\neg F(\$)$*. Then the theories*

$$\mathsf{FA}_\exists = \mathsf{PA} + (F_0) + (F_s) + (F_<) + (F_\exists)$$

1. See (Dean 2018) for discussion and a proof.

$$\mathsf{FA}_\$ = \mathsf{PA} + (F_0) + (F_s) + (F_<) + (F_\$)$$

are consistent and conservative over PA.

The upshot is that setting the limit to be a standard number trivializes the theory, but declaring its existence does not. This is noted by Rotman, who remarks that it is the process of counting up to the limit that is infeasible, not the number itself. The symbol $ used for the limit is the notation of his non-Euclidean arithmetic, the second worth-noting approach to feasible arithmetic. Like non-Euclidean geometry, non-Euclidean arithmetic is expected to behave "classically" until we reach $, the counting limit, which remains unknown. Once there, counting proceeds in a foggy partial ordering where arithmetic no longer behaves as we expect (see Figure 1).

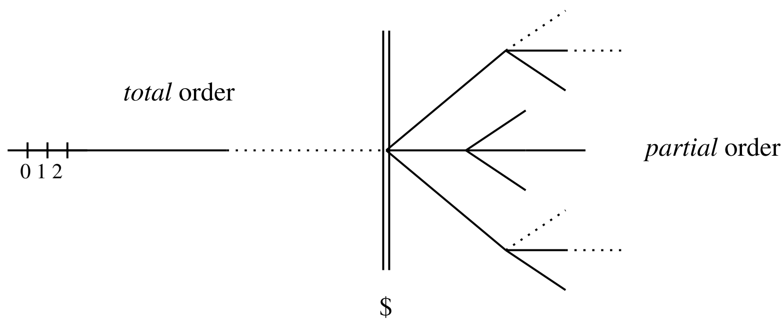

Figure 1: Total order holds until the $-limit, where arithmetic is "locally Euclidean."

Due to Kreisel's theorem, non-Euclidean arithmetic could be satisfied under some nonstandard model of arithmetic. Nevertheless, Rotman's motivation is not model-theoretical, but purely material (what amounts, in Rotman's words, to *semiotic corporeality*[2]):

> Imagine counting by an ideal computer. Thermodynamics and information theory provide formulas for how much energy such a computer would consume to count up to any number *n*. In theory, the computer could keep counting until it had consumed all the energy in the universe. No one knows how much energy that is, but physicists have estimated *U*, the mass-energy of the visible universe, at around 10^{75} joules. On the basis of that figure, an ideal computer could count to about 10^{96} before running out of energy. [...] Even under such extravagant conditions, the computer couldn't get beyond $10^{10^{98}}$ —which, you could say, is the outer horizon of all counting in this universe. [...]
>
> The crucial thing about a limit to counting though is not where the limit lies but that it exists. If you tried to count that far from inside the universe,

2. Materiality arguments are a central part of Rotman's general philosophy of mathematics. In particular, he needs the material aspect of mathematical production to explain in Marxist terms how the mathematician is alienated from the theorems he produces and hence thinks they do not belong to him but have existed in some outer Platonic reality.

> using a real computer with real energy requirements, you would use up more and more of the fabric of the universe trying to get there. Classical arithmetic doesn't reflect that reality. Non-Euclidean arithmetic does.
>
> (Rotman 2000, 134-135)

As showcased by almost consistent theories and non-Euclidean arithmetic, ultrafinitists seem unable to avoid the appeal to materiality, and it is also clear why they see complexity theory as the suitable framework for their claims. Precisely, it contains the notion of polynomial time, which seems to capture feasibility with respect to physical reality. The complexity class **P** (or a functional variant of it) containing precisely those problems that can be solved in deterministic polynomial time would represent our intuitive idea of feasibility, and certainly, complexity theory defines its fundamental classes in terms of physical resource bounds on Turing machines (logarithmic/polynomial/exponential time and space). Besides, the subfield of proof complexity, concerned with proving exponential lower bounds on the length of proofs, conjectures that no proof system has short proofs for every theorem—a key postulate for almost consistent theories to work.

In the next section I study this in more detail and argue that, contrary to this initial impression, complexity theory is not what ultrafinitists think it is.

2 Complexity theory against strict finitism

As a philosophy concerned with feasibility and grounded in materiality, computational complexity theory seems the perfect formalization for strict finitism. This is especially clear in that complexity theory provides the versatile and tried and tested polynomial versus super-polynomial time and space constraints. Early researchers in complexity, such as Cook and Buss, were motivated by ultrafinitism when they began working on bounded arithmetic. However, the development of complexity theory, with its rich interactions with other branches of theoretical computer science and mathematics, soon abstracted away from the material concerns of the philosophy. I now contend that, contrary to popular belief, complexity theory is not what the ultrafinitists think it is and it does not provide a theoretical framework in which to formalize their ideas. For this purpose, I present three main arguments: the ontological argument, the metalinguistic argument and the subject matter argument.

Before that, however, it is worth noting an important preliminary concern. Complexity theory studies which computational problems are tractable for Turing machines and seemingly equivalent computational models. Hence, claiming that polynomial-time restrictions also apply to human minds seems to require the controversial assumption that the human mind is a mechanical computational device. Furthermore, one seems to need the Church-Turing thesis, and even part of its strong variant too up to some extent[3]. Surveying the classical philosophical problem of minds and machines

3. The strong Church-Turing thesis postulates that not only are all computable functions precisely the Turing-computable ones, but that efficiency is also invariant between physically realizable computational models. That is, if a problem is efficiently solvable in some model, then it is also efficiently solvable (up to

is well beyond the scope of this paper, but it is worth keeping in mind that this is a substantial hurdle for ultrafinitism[4].

2.1 The ontological argument

The ontological argument stems from the subtle fear that, under a reasonable ultrafinitistic ontology of mathematics, most mathematical results in general and complexity-theoretic ones in particular might be rendered meaningless. What a reasonable ontology is for the strict finitist is an open question that should be tackled elsewhere, but one idea seems essential: a full-blooded rejection of the potential infinity of the naturals.

Based on Kreisel's theorem, the ultrafinitist will claim that feasible arithmetic can be consistent... but there are big caveats. The theorem holds because the theories $\mathsf{FA}_\exists$ and $\mathsf{FA}_\$$ are satisfied in some nonstandard model. The countable nonstandard models all have a copy of the full actual infinity of the naturals, followed by nonstandard numbers. Hence, if the strict finitist accepts these models, they will be accepting a built-in copy of $\mathbb{N}$ too. Even if one gives an ultrafinitistic reading of such a nonstandard model, there are methodological problems. The proof of the theorem relies on compactness, which in turn relies on completeness of first-order logic, which in turn relies on Lindenbaum's lemma, proven via Henkin extensions requiring a potential infinity of new constant symbols to add to the language. Surely, one can try and rebuild the metatheory of first-order logic ultrafinitistically from scratch, but this will require a rework of all the relevant notions of set theory too.

Even if the ultrafinitist was capable of such a feat and provided a meaningful formalization of, say, non-Euclidean arithmetic, and gave convincing arguments for its consistency, that theory would likely not be conservative over PA. Yet, complexity theory relies on the full power of Peano arithmetic for even its most fundamental claims, and thus the notion of polynomial-time may no longer be meaningful under the new formalization. Notably, we need potentially infinite natural numbers to serve as descriptors for Turing machines.

Furthermore, things like the Time Hierarchy Theorem[5] (very much at the core of the notion of polynomial time) would collapse. In complexity theory, it follows from this result that

$$\mathbf{DTIME}(n) \subsetneq \mathbf{DTIME}(n^2) \subsetneq \mathbf{DTIME}(n^3) \subsetneq \cdots \subsetneq \mathbf{DTIME}(n^{100}) \subsetneq \dots$$

but one might claim that these inclusions stop being strict as soon as we reach the class $\mathbf{DTIME}(n^\$)$. This implies, for instance, that the class $\mathbf{P}$ would not have problems of ever-increasing complexity. Surely, the ultrafinitist will argue that problems strictly contained in, say, $\mathbf{DTIME}(n^{78})$ are already impossible to conceive for the human mind. In any case, they will be forced to take action to save the idea of polynomial time and its properties. This requires either accepting potential infinity as a useful fiction (becoming

---

polynomial factors) on a Turing machine, and vice versa. Quantum computers are a strong candidate against this variant of the thesis.

4. A complexity-theoretic perspective on the minds and machines problem has been recently given in (Aaronson 2011).

5. The Time Hierarchy Theorem states that *if f and g are time-constructible functions such that $f(n)\log_2 f(n) \in o(g(n))$, then* $\mathbf{DTIME}(f(n)) \subsetneq \mathbf{DTIME}(g(n))$. See (Arora and Barak 2009, 69) for a proof.

almost indistinguishable from the traditional finitist) or rebuilding complexity from scratch as well, thus accepting that complexity theory as it is does not represent their views in the first place.

2.2 The metalinguistic argument

Earlier, the idea of almost consistent theories was briefly presented. In an almost consistent theory, there is an axiom $\neg F(\ell)$, where ℓ is some explicit "obviously infeasible" number; say, $\neg F(2^{200})$. Then, a derivation of Wang's paradox is supposed to require ℓ steps or more inside the system, which is infeasible.

The key observation is, indeed, that formal proofs belong to the object language, while the number of steps is a property of the metalanguage. Hence, the strict finitist cannot say that the number of steps of the proof is infeasible, as in the metalanguage this is still a vague notion. If they disregard this, then one is perfectly entitled to show that the system is inconsistent by reasoning shortly in the metalanguage[6], with a proof that is equally valid and succinct. Thus, far from solving Dummett's problem, the ultrafinitist pushes it elsewhere.

Furthermore, the metalinguistic argument thwarts the faith of the ultrafinitist in proof complexity, the subfield of complexity theory concerned with the fundamental question of whether proof systems exist that can provide short proofs for all propositional tautologies[7]. The ultrafinitist will invoke here something like Haken's theorem, the first exponential lower bound for the Resolution proof system[8]. The theorem states that any derivation of the Pigeonhole Principle (the fact that there is no bijection between $\{1,\ldots,n\}$ and $\{1,\ldots,n-1\}$) in the Resolution proof-system has length exponential in n. The ultrafinitist can claim that Haken's theorem shows the limits of feasible proofs, but this misses the metalinguistic point once again. The Pigeonhole Principle does not state anything infeasible: if one has ten pigeons and nine holes, they cannot fit every pigeon in a pigeonhole, and the same follows for larger numbers by straightforward induction... a power that Resolution does not possess. As a result, it gets stuck in the locality of the formulas: it tries one of the 2^n possible assignments, sees that it does not work, and tries again.

The ultrafinitist could claim that this is the case for us too: human cognition has limits imposed by physical reality, and there will be cases in which we do not know better than trying all possible cases. In the same way that Resolution cannot do induction, there might be theorems and proof methods that humans cannot grasp. I reply to this criticism later, invoking a variant of the ontological argument.

6. Remarkably, using further insights of first-order logic and with the help of the arithmetical theories supplied next to the feasibility axioms, Boolos observed that we can derive a contradiction in the system in $\log_2 \ell$ steps, feasible in complexity-theoretic terms. Furthermore, using the cut shortening technique devised by Solovay one can obtain up to super-exponential speed-ups; see (Dean 2018) for a contextualized discussion of these techniques. Either way, it is clear that if the ultrafinitist is entitled to appeal to feasibility in metalinguistic proof-theoretic arguments, then so is everyone else.

7. A result known as the Cook-Reckhow theorem relates this issue to the classes **NP** and **coNP**. More concretely, proving that no polynomially-bounded proof system exists for all propositional tautologies amounts to showing $\mathbf{NP} \neq \mathbf{coNP}$, which would imply the famous $\mathbf{P} \neq \mathbf{NP}$. The conjecture is that $\mathbf{NP} \neq \mathbf{coNP}$.

8. The original argument, known as Haken's *bottleneck method* appeared in (Haken 1985), and was later simplified in (Beame and Pitassi 1996). A modern reproduction of the proof can be found in (Arora and Barak 2009, 310-311).

In short, complexity theory does show exponential lower bounds like those the strict finitist is after, but they ignore the metalinguistic perspective used to prove those lower bounds in the first place.

2.3 The subject matter argument

The metalinguistic argument's relation to proof complexity already points to the fact that complexity does not talk about physical constraints. This might come as a surprise, given that, at first glance, complexity theory phrases its results in terms of asymptotic bounds on the physical resources (space and time) needed to solve a problem. I present two directions in which to argue that these resource bounds are not at all the subject matter of complexity.

The first reasoning is that, following Rotman (1993; 2000), what is infeasible is the process of counting, and not the big numbers themselves. This coincides with how we measure complexity growing asymptotically as the problem instances grow larger. Interestingly, this has unacceptable consequences for the ultrafinitist. For instance, they will agree that deciding whether whites have a winning-strategy for chess is something infeasible in general—there are too many games to explore. Yet, for complexity theory, this problem is "easy": the number of possible chess games can be approximated by Shannon's number: 10^{120}, significantly more than the number of atoms in the visible universe... yet constant. One can evaluate all possible games with a constant-time algorithm—complexity $O(1)$. Chess is in **P**.

It turns out that physical constraints are blurred under the magnifying glass of structural complexity, as it is measured in terms of how problems scale. A game of chess does not scale at all; the number of games is constant if we fix the size of the board. Hence, deciding chess on an 8×8 board is "easy"; what is not easy is, for example, deciding whether whites have a winning-strategy in k moves given some starting state of an $n \times n$ board, as n increases[9]. In short, complexity theory is not built to cater for the physical constraints that the ultrafinitist cherishes. Rather, it abstracts away from the physical nitty-gritty, and sometimes even from the problems. This ultimately showcases that the primary concern is in showing the absence of structure in a search space.

Take the problems of deciding whether an undirected graph has an Eulerian cycle versus deciding whether it has a Hamiltonian cycle. Both are quite similar: in the Eulerian case, the cycle has to cover every *edge* exactly once, whilst the Hamiltonian cycle needs to go through every *vertex* exactly once. Surprisingly, the former is easy (is in **P**), while the latter is likely not (is **NP**-complete). Deciding the existence of Eulerian cycles is easy, due to an easy-to-check property known as Euler's theorem: a graph has an Eulerian cycle if and only if the degree of every vertex is even. On the other hand, no polynomial-time algorithm is known for finding Hamiltonian cycles, and no such

9. Under a reasonable generalization of chess, this problem is **PSPACE**-complete (Storer 1983), which means it can be solved in polynomial space, and, likely, not in polynomial time. Playing generalized chess is not something humans do, but things like this are common in complexity. It happens often that an asymptotically worse algorithm is prefered because the hidden constants of the better one make it intractable in practice. Take the case of testing primality, where probabilistic procedures are much faster in practice and prefered over the deterministic polynomial-time algorithms, even at the expense of error.

thing is expected to exist. Remarkably, this is not about physical resource bounds: it is about the graphs and their intrinsic properties, independent of time and space. Note that not even a quantum computer, a model conjectured to break the strong Church-Turing thesis, is expected to solve this efficiently, despite its physical superpowers[10].

Yes, complexity theory phrases its questions in terms of asymptotic bounds on physical resources, but this is mainly a consequence of the historical development of the field. In fact, big open problems such as $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ are mostly phrased in terms of the certificate definitions (e.g., "whenever it is easy to check a solution, is it also easy to come up with it?"). Approaches like descriptive complexity show that it is perfectly possible to rephrase everything in purely formal linguistic terms, machine independently and without reference to physicality. For example, **P** is also the class of problems expressible in first-order logic with least fixed-points, and **NP** corresponds to second-order existential logic[11].

As a last resort, like in the case of Haken's theorem, the ultrafinitist can claim that an easy-to-check characterization for Hamiltonian graphs might exist, yet be ungraspable for human cognition due to its limitations. Unfortunately, they would be betraying their creed in claiming the existence of some abstract property that works for all graphs and that exists beyond the physical limits of human cognition, as this contradicts the essential conventionalism of the ultrafinitist, who emphasizes mathematics being something done by humans and constrained by their limitations.

Brief, complexity theory is primarily concerned with proving the lack of exploitable structural properties in the search space of a decision problem, and its formalization as asymptotic bounds on physical resources is just a consequence and reconceptualization of this insight motivated by its historical development, and not an intrinsic characteristic of the subject matter of complexity.

Conclusion

In the light of this discussion, I hope to have made clear two points: (i) that strict finitism should be understood as a philosophy primarily concerned with feasibility and

10. On this matter, Nielsen and Chuang write that

> the essential reason for the difficulty of **NP**-complete problems is that their search space has essentially no structure, and (up to polynomial factors) the best possible method for solving such a problem is to adopt a search method. If one takes this point of view, then it is bad news for quantum computing, indicating that the class of problems efficiently soluble on a quantum computer, **BQP**, does not contain the **NP**-complete problems [due to optimality of Grover's search algorithm].
>
> A nice example to illustrate this is the problem of factoring, widely believed to be in the class of problems intermediate in difficulty between **P** and the **NP**-complete. The key to the efficient quantum mechanical solution of the factoring problem was the exploitation of a structure "hidden" within the problem [...]. Even with this amazing structure revealed, it has not been found possible to exploit the structure to develop an efficient classical algorithm for factoring [...].
>
> (Nielsen and Chuang 2010, 271)

11. These characterizations, due respectively to Immerman and Fagin, as well as the entire program of descriptive complexity, are covered at length in (Immerman 1999).

materiality; and (ii) that computational complexity theory is not what the ultrafinitist has in mind and cannot be used to support their philosophical position.

Whether strict finitism is a viable philosophy of mathematics is a question that remains open. Certainly, a comprehensive study of all the different variants and positions involved should be carried out before further analysis. If anything, this paper has only argued that the strict finitist cannot defend themselves by appealing to the soundness and success of computational complexity theory. Surely, seminal ideas were originally influenced by ultrafinitistic concerns, but these were soon refined, adapted and eventually forgotten in favour of mathematical rigour, applicability and scope.

Possibly, the further study of these questions might lead the ultrafinitist to reconsider their position and sharpen their views, there where complexity has shown deficiencies. Finally, I believe to have contributed to the yet small but growing philosophical discussion about complexity theory, a domain often forgotten by philosophers of mathematics.

Noel Arteche
Amsterdam
March 28, 2021

References

Aaronson, Scott. 2011. "Why philosophers should care about computational complexity." *arXiv preprint arXiv:1108.1791.*

Arora, Sanjeev, and Boaz Barak. 2009. *Computational Complexity: A Modern Approach.* Cambridge University Press.

Beame, Paul, and Toniann Pitassi. 1996. "Simplified and improved resolution lower bounds." In *Proceedings of 37th Conference on Foundations of Computer Science,* 274–282. IEEE.

Bernays, Paul. 1935. "On Platonism in Mathematics." *Philosophy of Mathematics: Selected Readings,* 274–286.

Claude, E. Shannon. 1950. "Programming a Computer for Playing Chess." *Philosophical Magazine, Ser* 7 (41): 314.

Dean, Walter. 2018. "Strict Finitism, Feasibility, and the Sorites." *Review of Symbolic Logic* 11 (2): 295–346. https://doi.org/10.1017/S1755020318000163.

———. 2019. "Computational Complexity Theory and the Philosophy of Mathematics†." *Philosophia Mathematica* 27 (3): 381–439. https://doi.org/10.1093/philmat/nkz021.

Dummett, Michael. 1959. "Wittgenstein's philosophy of mathematics." *The Philosophical Review* 68 (3): 324–348.

———. 1975. "Wang's Paradox." *Synthese* 30 (3-4): 201–32. https://doi.org/10.1007/BF00485048.

Haken, Armin. 1985. "The intractability of resolution." *Theoretical Computer Science* 39:297–308.

Hilbert, David. 1984. "On the infinite." In *Philosophy of Mathematics: Selected Readings,* 2nd ed., edited by Paul Benacerraf and HilaryEditors Putnam, 183–201. Cambridge University Press. https://doi.org/10.1017/CBO9781139171519.010.

Immerman, Neil. 1999. *Descriptive Complexity.* Graduate Texts in Computer Science. New York: Springer. ISBN: 0387986006.

Kielkopf, Charles F. 1970. *Strict Finitism: An Examination of Ludwig Wittgenstein's Remarks on the Foundation of Mathematics.* Mouton.

Kornai, András. 2003. "Explicit finitism." *International Journal of Theoretical Physics* 42 (2): 301–307.

Krajıček, Jan. 1995. *Bounded arithmetic, propositional logic and complexity theory.* Cambridge University Press.

———. 2004. "Proof complexity." In *European congress of mathematics (ECM), Stockholm, Sweden,* 221–231.

Magidor, Ofra. 2007. "Strict Finitism Refuted?" *Proceedings of the Aristotelian Society* 107 (1pt3): 403–411. https://doi.org/10.1111/j.1467-9264.2007.00230.x.

———. 2012. "Strict finitism and the happy sorites." *Journal of philosophical logic* 41 (2): 471–491.

Nielsen, Michael A., and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press. https://doi.org/10.1017/CBO9780511976667.

Parikh, Rohit. 1971. "Existence and feasibility in arithmetic." *The Journal of Symbolic Logic* 36 (3): 494–508.

Rodych, Victor. 2018. "Wittgenstein's Philosophy of Mathematics." In *The Stanford Encyclopedia of Philosophy,* Spring 2018, edited by Edward N. Zalta. Metaphysics Research Lab, Stanford University.

Rotman, Brian. 1993. *Ad Infinitum... The Ghost in Turing's Machine: Taking God Out of Mathematics and Putting the Body Back In. An Essay in Corporeal Semiotics.* Stanford University Press.

———. 1996. "Counting Information: A Note on Physicalized Numbers." *Minds and Machines* 6 (2): 229–238. https://doi.org/10.1007/BF00391287.

———. 2000. *Mathematics as sign: Writing, imagining, counting.* Stanford University Press.

———. 2006. "Towards a semiotics of mathematics." In *18 Unconventional Essays on the Nature of Mathematics,* 97–127. Springer.

Sazonov, Vladimir Yu. 1994. "On feasible numbers." In *International Workshop on Logic and Computational Complexity,* 30–51. Springer.

Storer, James A. 1983. “On the complexity of chess.” *Journal of computer and system sciences* 27 (1): 77–100.

Tait, William W. 1981. “Finitism.” *The Journal of Philosophy* 78 (9): 524–546.