

Computational complexity in the
philosophy of mind: unconventional
methods to solve the problem of logical
omniscience

Safal Raman Aryal

February 1st, 2022

Table of Contents

<i>i. Abstract</i>	3
<i>ii. Introduction</i>	4
<i>iii. Human Complexity Classes</i>	<i>Error! Bookmark not defined.</i>

i. Abstract

The philosophy of mind is traditionally concerned with the study of mental processes, language, the representation of knowledge and the relation of the mind shares with the body; computational complexity theory is related to the classification of computationally solvable problems (be it via execution time, storage requirements, etc...). While there are well-established links between computer science in general & the philosophy of mind, many possible solutions to traditional problems in the philosophy of mind have not yet been analyzed from the more specific lens of computational complexity theory. In his paper "Why Philosophers Should Care about Computational Complexity", Scott Aaronson argues that many conventional theories of epistemology & mind implicitly make the presupposition of omniscience (by supposing that knowing base facts means a knower necessarily understands derivative facts) - he proposes that computational complexity theory could explain why this is not the case. In this paper, I argue for a theory of mental representation & epistemology compatible with Aaronson's observations on complexity theory, overcoming that presupposition of omniscience.

ii. Introduction

The problem of logical omniscience arises fundamentally from treating *epistemic* logic (the logic explaining the acquisition & knowledge of beliefs) as a branch of *modal* logic (the branch of logic dealing with formalizing systems of *possible* truths). When the deliberation of belief is treated as a model problem, the problem of *how* exactly people can make inferences over such large systems of logic arises – how can one go through many thousands of propositions every time that is required? (“The Problem of Logical Omniscience”).

Indeed, the modal theory of belief implies that this speedy deliberation must also occur in *other* domains unrelated to constructing beliefs – this theory implies that *any* propositional knowledge rooted in a logical system implies that the knower would know *every* possible statement under that system. So, if a knower were to learn the Peano axioms, they would also possess knowledge of the solution to the Poincaré conjecture. This is an obviously problematic model of human knowledge: there is no trivial automatic deduction of inferences within larger logical systems, so what are avenues that can be taken to resolve this inconsistency?

The traditional solution to the problem of logical omniscience (as it is called, as this state would imply all people become “logically omniscient”) is one of acknowledging the idea of “implicit knowledge” – which is knowledge that does not necessarily need to be acquired but is innately present in everyone (or specific groups, depending on the logical framework in question). Thus, the problem is solved as follows: of certain frameworks, people have implicit knowledge & can deduce conclusions within them quickly & effectively. There is also a parallelism with respect to the logical frameworks themselves: sometimes *they* contain implicit knowledge which must be deduced only with reference to other systems and is thus acquired by those means. The example of mathematics, however, comes back to haunt this solution – people *do* seem to have innate knowledge of certain branches of mathematics (arithmetic being the most prominent example), and mathematics is *self-contained*, meaning that there is no *implicit* knowledge that can be uncovered through juxtaposition with other frameworks, and the only implicit knowledge is that that the propositions contained within mathematics can imply. This solution breaks down in these situations.

Scott Aaronson, professor of computer science at UT Austin, was one of the first academic scholars within computer science to note the same thing as I have here – in his paper “Why Philosophers Should Care About Computational Complexity”, he notes that the problem of logical omniscience is one that could be solved through the application of concepts from *computational complexity theory*. As a huge topic, summarizing it

concisely is difficult, but fundamentally, computational complexity theory is the study of how *many steps* and *how much space* (among many other quantitative indicators of use) certain computations take. It studies these relations in a precise & mathematical manner and is the standard way for computer scientists to analyze the effectiveness & performance of algorithms (Aaronson 10).

For example, let us say there is a function which computes the square of a number. One operation is required to square a number – either the square operation is applied directly, or it is multiplied by itself (and one could say the square operation is an abbreviation of the latter, but in some computer systems it is possible that they serve a similar function). If we were to try to measure the *time complexity* of this function (the amount of steps it takes to execute), we would say it takes 1 step. Squaring 56 numbers, on the other hand, takes 56 of these “steps”. By mapping steps to operations that a computer performs, it is possible to assess the differences between different types of algorithms – choosing the correct one to perform the correct tasks.

Computational complexity theory can also be used to calculate the amount of space that certain algorithms use up in the address space, with reference to fixed memory constraints present on computer systems. It can be used to understand even natural processes: by mapping the stages of evolution onto computational steps, it is possible to *estimate* how much computational power simulations of the phenomenon would require.

Given this flexible & interdisciplinary application, it is surprising that before Aaronson, no one thought to apply computational complexity theory to the study of philosophy, and especially the problem of logical omniscience.

iii. Complexity Classes & discussion of a possible solution

A complexity-theoretic solution to the problem of logical omniscience could take the following form given the *computational complexity* of deducing certain conclusions (for example, the Poincaré conjecture, or the fastest route through Kathmandu’s subway system), it is impossible that an “instantaneous knowledge” could arise within a person about their nature – thus, rendering the problem solved. This avoids having to draw the tenuous distinction between implicit & explicit knowledge that has been the approach in modal logic.

There are, however, some issues with this solution to the problem: for one, the mind does *not* perceive the complexity of tasks in necessarily the same way that computers would. For example, the proof to Fermat’s Last Theorem took centuries to discover and it took an odd association made by Andrew Wiles within the field of elliptic curves to arrive at it, but the proof itself would be rather simple. Given that the basic rules of mathematics can be arranged logically in a formal & straightforward manner to arrive at it, to a computer, finding Fermat’s Last Theorem would *theoretically* be a rather simple task if the statements could be phrased that way (because of non-computability,

I doubt this is an actual example, but there are probably other instances, like the Pythagorean theorem). This is the fundamental observation that a *computer's method of classifying complexity is not the same as a human being*. The second issue that it is unknown whether human minds *operate* as computers. It will be argued that this is irrelevant to constructing a complexity-theoretic solution to the problem.

Computational complexity classes are grouped together by *aggregating* the number of steps that certain operations take even when scaled up, most commonly in the form of finding functions which fit their models. For example, a function which calculates the n th term of a quadratic sequence might involve squaring a number, then adding its product with the number 5, and then adding the constant 7. Another function might deal with a sequence that has *more* terms, but because in both cases the number of operations does *not* grow with the size of the input, they belong to the complexity *class* $O(1)$ (which means that they take a multiple of 1 step to complete).

Contrast this to a function which operates on lists. Since the lengths of lists vary considerably from one another, a function which adds a number to every member of a list or multiplies it by a constant might be considerably complex for some lists and certainly *less* complex for others. In this case, the number of steps the function takes *depends* on the number of members in the list, and the relation is one of direct proportionality – which is called $O(n)$. Other types of growth (and decrease, though this

is unlikely) relations are also possible – the point is that precise categorizations of the number of steps operations take are *possible* in the first place.

This is not so simple when dealing with the case of human reasoning. In human reasoning, it is often *not possible* to track the number of steps a certain deduction took to make because they are not made in the same procedural manner as they are for computers. For example, returning to Andrew Wiles' example, it took him years to reach his insights about elliptic curves – and this clearly required a lot of mental deliberation. Obviously, it did not take long for reviewers to *read* through them & understand the logic by which he employed them, but it did take him time. A computational representation of his proof *does not* allow for an accurate representation of his labor.

A part of this problem comes from the fact that the inner workings of individual minds are often obscure – as much as we are *conscious*, many judgements of ours are made *preconsciously* and outside of our explicit deliberation. These clearly involve our basic biological cognitive faculties, but it does not seem necessary for them to invoke our conscious cognitive faculties because they have concurrent operations with our bodies.

One way around this through the lens of complexity theory is formulate new complexity classes: ones specifically fit to explain the limits of human cognition by imposing complexity limits through understanding the average *biological* limitations that people face.

Consider the work in standardizing batteries for intelligence tests, for example. The task of an intelligence test designer is to formulate a battery of tests which assesses human cognitive ability as broadly as possible and reduces them to *basic* skills. Often, these basic skills are put on percentile scales to understand how measurements of them are interpreted statistically – and this percentile understanding could be used to calculate an “upper limit” for the inferences that a person is able to make within their long-term and short-term memories, and thus classify their thoughts into a “complexity class” relative to the rest of the sample on which the test is normed.

While this is not a direct analogy to assessing the computational complexity of a given idea or train of thought with reference to human limitations *directly*, it does manage to accomplish one task – quantify the degree to which a person *may* be able to reason. Once this is done, assessments can be made of whether it may be out of reach for a person to perform certain tasks. For example, a child could not prove the Poincaré conjecture, because their ideal state of *logical omniscience* is limited by biological cognitive factors, and this is represented by a numerical value based on cognitive abilities.

Instruments assessing cognitive abilities have obvious flaws, and it is not intended to promote them blindly as tools to do this with. Rather, what is being prompted is an approach which *breaks* down cognitive abilities in understanding logical propositions to

assess a given individual's likelihood of being able to make certain deductions which employ skills *associated with* that – and through calculating “complexity classes” on this basis, the basic logic behind Aaronson's solution is preserved.

This also eliminates the problem of having to deliberate on whether or not computational reasoning maps *directly* onto human reasoning: insofar as an appropriate model is developed to make the quantitative classification of cognitive abilities, the inaccessibility of the mind simply means that the fact of why the brain performs with complexity is considered *outside* of the scope of this brand of philosophy: the job of the philosopher here is simply to resolve the epistemological conundrum that arrives from having to consider the problem of logical omniscience.

iv. Conclusion

Scott Aaronson's proposition to use aspects of computational complexity theory to solve the problem of logical omniscience has considerable promise – it provides a neat solution to the problem of *how* exactly deductions are created and allows for flexible ambiguity in committing to a computationalist theory of mind. Empirical concerns remain about the *efficacy* of how cognitive ability assessments of adequate quality could be designed to assess them accurately enough to allow for the consideration of complicated problems – the understanding of “cognition” here is limited to the logical type.

v. Works Cited

“The Problem of Logical Omniscience.” *The Problem of Logical Omniscience*, 5 Apr. 2001, <https://www.informatik.uni-leipzig.de/~duc/Thesis/node15.html>.

Aaronson, Scott. “Why Philosophers Should Care About Computational Complexity.” *ArXiv.org*, 14 Aug. 2011, <https://arxiv.org/abs/1108.1791>.