Discussion Notes on Physical Computation

Samuel C. Fletcher and Jason Hoelscher-Obermaier

June 15, 2010

Much has been written as of late on the status of the physical Church-Turing thesis and the relation between physics and computer science in general. The following discussion will focus on one such article [5]. The purpose of these notes is not so much to argue for a particular thesis as it is to solicit a dialog that will help clarify our own thoughts.

1 Introduction

Historically the Church-Turing thesis (CTT) has been an assertion about the limits of effective computation. The CTT states that the class of effectively computable functions coincides with the set of recursive functions. Subsequently, however, a very different interpretation according to which the CTT is a statement about the limitations of physical computers has been influential. In fact, many researchers nowadays take the CTT to be such a statement. Crucially, however, there are many different versions of this *physical CTT*, and failure to distinguish between them can easily lead to confusion. We will examine one such example here.

Nielsen [5] presents a self-adjoint operator $\hat{h} = \sum_{k=1}^{\infty} h(k)|k\rangle\langle k|$ on an infinite dimensional Hilbert space, where $h(k)$ is the *Halting function.*[1] If

[1]The Halting function is defines as follows. Fix an enumeration $\{T_n\}$ of the Turing machines. $H(n) = 1$ just when T_n halts on input n, and is zero otherwise. It is a classical

implemented as a measurement, this Halting observable would allow for the computation of a non-recursive function. From this, Nielsen concludes that there is some tension between the full formalism of quantum mechanics, in which any self-adjoint operator corresponds to a physical observable, and the CTT: either the CTT needs revision, or only a restricted class of self-adjoint operators is implementable in practice as observables.[2]

Whether or not there is in fact tension between the CTT and the "full formalism of quantum mechanics" crucially depends, of course, on what the CTT is taken to assert. Nielsen specifies the CTT to be the statement that "every function which can be computed by what we would naturally regard as an algorithm is a computable function, and vice versa." This statement makes contact with Church and Turing's original thesis on the limits of effective computation. But in order for there to be any tension with quantum mechanics, Nielsen must have some version of the physical Church-Turing thesis in mind. In fact, the character of Nielsen's argument depends crucially on which version of the physical CTT he takes to be in conflict with quantum mechanics. Let us therefore provide a brief overview and classification of different versions of the physical CTT.

2 The Physical CTT

The different versions of the physical CTT fall roughly into two classes. Some versions assert the Turing-computability (in some sense) of all physical processes; others restrict themselves to asserting the Turing-computability of only those physical processes that count (in some sense) as computations. In this paper we follow Piccinini [6] in referring to the former class as the *bold* physical CTT and the latter as the *modest* physical CTT.

result in computation that the Halting function is non-recursive.

[2]In fact it is not crucial to think about measurements. As Nielsen shows, one can just as easily write down Hamiltonian operators which, if implemented physically, would allow the computation of non-recursive functions.

Let us begin with the bold physical CTT. Piccinini lists a few (more or less) explicit, but not always obviously equivalent, versions.[3]

1. Any physical process can be simulated by some Turing machine (e.g., [2, 10, 8]).

2. Any function over denumerable domains (such as the natural numbers) that is computable by an idealized computing machine that manipulates arbitrary real-valued quantities (as defined by [1]) is Turing-computable.

3. Any system of equations describing a physical system gives rise to computable solutions (cf. [3, 9]). A solution is said to be computable just in case given computable real numbers as initial conditions, it returns computable real numbers as values.[4]

4. For any physical system S and observable W, there is a Turing-computable function f such that for all $t \in \mathbb{N}$, $f(t) = W(t)$ [7].

The bold thesis in these formulations has the character of a high-level law of nature or theoretical principle comparable to, say, the law of energy conservation which holds across various disparate physical theories. On this account of the CTT, Nielsen's argument would then be akin to showing that some scientific theory violates the law of energy conservation. The trouble for his argument in this case is that some of these bold versions of the physical CTT are likely to be untenably strong. And even if the bold physical CTT should turn out to be compatible with the all of physics except for the halting observable, it is not clear why such a non-computable physical process should

[3]We state the different versions of the bold physical CTT more or less as they are stated by Piccinini. We do not thereby vouch for their conceptual clarity and we make no attempt to do so since it will not be necessary for what follows. Note also that the references are to articles in which the corresponding version is stated. The corresponding statements are not necessarily endorsed there.

[4]A real number is computable iff there is a Turing machine which computes arbitrarily good approximations to it.

not simply be taken as empirical evidence against the bold physical CTT. In the words of Chris Timpson: "[I]f [...] a well-confirmed and detailed physical theory [...] says that some process is possible, then that theory holds the trump card over a less specific generalization covering the same domain" (where the less specific generalization is the physical Church-Turing thesis).[5]

There is a different way of understanding the physical CTT, in which we only require anything *computable* by a physical process to be Turing computable. This is the modest version of the thesis. Of course, an exact formulation requires a precisification of "computable." On a first pass, we might think of a physical computation as any physical process that allows agents to gain knowledge about the values of some known function. Hence, one might say that while the bold thesis is a metaphysical thesis, the modest thesis is epistemological.

In the context of the modest physical CTT, it seems that a definition of physical computation must therefore presuppose a definition of knowledge. Given the historical intractability of the latter problem we will not attempt to advance a necessary and sufficient set of conditions for physical computation. Instead, we can consider some proposals for what might amount to necessary conditions — plausible candidates, at least. Here are Piccinini's:[6]

1. *Readable input and output.* Most importantly, having to measure a real-valued quantity exactly must not be an essential part of a physical computation.

2. *Process-independent definition.* The function computed must be defined outside of the context of the computation, that is, one needs to have a sufficiently precise characterization of what is purportedly being computed before a process is to count. Process-independence rules out machines such as true random number generators which would compute *some* non-recursive function, without specifying which.

[5]Personal communication.

[6]Note that also other authors (e.g. [4]) have proposed different "packages" of conditions.

3. *Repeatability.* On a reasonably weak version of repeatability one might require merely that it be possible to implement the same process again.

4. *Resettability.* One must be able to reset the physical system enacting the computation to some initial state.[7]

5. *Physical constructibility.* The process implementing the computation must in some sense be possible, at least in principle. (We will have much more to say about this below.)[8]

6. *Reliability.* For example, it should not be the case that it is only reasonable to use the process for the purpose of computation under the assumption of perfect control over the system and its environment.

Does Nielsen have the bold or the modest physical CTT in mind? It seems overwhelmingly likely that he has some version of the modest CTT in mind since he explicitly requires that the process be constructible in a laboratory, stable under perturbations, and in some sense verifiable. But in order to demonstrate a tension between the modest physical CTT and QM, Nielsen at the very least needs to explicate which criteria he is using, and show that his examples meet these necessary criteria in order for them to "count" as computations. And among the necessary criteria for physical computations, it seems that physical constructibility is the critical point: Nielsen's dilemma is whether or not it is "possible in principle" to construct the halting observable. But there is an interesting ambiguity in the force of "possible in principle," here:

[7]Of course, there are many different versions of different strength for both the repeatability as well as the resettability requirement.

[8]Piccinini also explains why the fact that Turing machines are themselves unlikely to be physically constructible is not an argument against this requirement. The physical version of the CTT is supposed to give merely an "upper bound" on what is physically computable. The fact that it might not even be possible to physically realize all Turing computations does not invalidate it. The physical CTT does not assert that we can actually (physically) build a Turing machine, but rather that we could not build something more powerful.

1. Does "possible in principle" mean merely that it does not contradict any known natural laws (i.e. is merely consistent with them)?

2. Slightly stronger: Assuming that the known laws are *complete*, is it consistent with them?

3. Slightly stronger again: Assuming that the known laws are complete, is it *likely* (according to some specified measure, for example on the set of initial conditions)?

4. Does it mean that we could realize it if we had access to arbitrary resources such as non-recursive functions?

5. Or is it a requirement with respect to current (or near future) human capabilities, that is: Does it mean that there is an effective procedure to build such a machine?

One might think that the force of "in principle" is to abstract away from human-specific considerations. But Nielsen urges us to construct (literally) the measuring apparatus and prepare the system to be measured if it is "possible in principle" to perform a measurement of the halting observable. This locates his "possible in principle" relatively far down on the above list (number five); for Nielsen, according to the physical CTT it is impossible in principle (in the sense of number five) to perform a measurement of the halting observable. But this sense of "impossible in principle" is not necessarily in conflict with physical law. In a word, quantum mechanics allows that the measurement of the halting observable is possible in principle in senses one and two, at least, but this is not in conflict with the impossibility of measuring the halting observable in sense five.[9]

[9]Contingent upon the conflict between the CTT and QM, Nielsen requires that we accept one and modify the other. But why, say, if we accept QM must we *modify* the CTT? Why can't we simply reject it? His insistence on modifying the CTT so as to be consistent with QM might arise from a confusion of the physical CTT with the traditional CTT. Insofar as we are in need of a definition of "computable," we need a definition that

As a final point, we should note that, independent of its validity, Nielsen's argument does not depend on any distinguishing features of QM. One could equally well construct classical observables and classical Hamiltonians that one could use to compute the Halting function in the same sense. The air of plausibility to his argument may derive in part from the many extraordinary and unintuitive consequences of the quantum theory, especially with regard to the theory of computing; we are much less susceptible to believing his argument in the context of classical mechanics.

3 The Intersection of Physics and Computation

At this point we wish to zoom out a bit and pose some general questions for consideration and discussion.

- We have seen above a case wherein the application of concepts from computer science to physics has been misguided. Are there truly justified cases where such a process could fruitfully occur?
- Although in the above the particular usage of QM turned out to be irrelevant, there are certainly distinctions between classical and quantum computation that are philosophically interesting. For example, it is well known that there exist quantum algorithms — e.g. Shor's and Grover's algorithms for factoring and searching, respectively — that outperform their classical counterparts. What does the statement "quantum algorithms outperform their classical counterparts" mean, particularly since these algorithms are formulated in different computational models? Can we identify what aspects of QM explain this

is not only compatible with but also useful in the context of our (best physical) theories. But the physical CTT provides no such definition. So we will pass over his discussion of emendations in silence.

speedup?

- We measure the complexity of some particular algorithm by the way the number of basic computational steps increases with the size of the input. What justifies the use of any such particular basic steps? Are they justified merely from pragmatic concerns, i.e., is the label "basic" shorthand for "easily implementable"?

- Quantum algorithms are built from unitary operators, and the product of unitary operators is a unitary operator. Thus, any quantum algorithm can always be written as a single unitary transformation.[10] Why do we take any particular set of unitary transformations to be the "basic" building blocks of a quantum algorithm?

- Given some version of the physical CTT, are there functions which are recursive but not physically computable? Given any function, recursive or not, is it an empirical question whether that function is computable?

- Given any specific version of the physical CTT are there counterexamples to it?

References

[1] Blum, L. et al. *Complexity and Real Computation.* New York: Springer.

[2] Deutsch, David. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer." *Proceedings of the Royal Society of London A* (1985) 400: 97-117.

[3] Earman, John. *A Primer on Determinism.* Dordtrecht: D. Reidel.

[10] Presuming that one performs a measurement at the end of the algorithm, even a unitary transformation is not necessary if we instead change the measurement basis in the appropriate way.

[4] Geroch, R. and J. B Hartle. “Computability and physical theories.” *Foundations of Physics* 16 (1986): 533550.

[5] Nielsen, Michael A. “Computable Functions, Quantum Measurements, and Quantum Dynamics.” *Physical Review Letters* 79 (1997): 2915-2918.

[6] Piccinini, G. “The Physical Church-Turing Thesis: Modest or Bold?” *Unpublished Draft* (2010).
`http://www.umsl.edu/~piccininig/CT_Modest_or_Bold.pdf`

[7] Pitowsky, I. “The Physical Church Thesis and Physical Computational Complexity.” *Iyyun* 39 (1990): 81–99.

[8] Pitowsky, I. “Quantum Speedup of Computations.” *Philosophy of Science* 69 (2002): S168–S177.

[9] Pour-El, M. B. “Abstract Computability and its Relation to the General Purpose Analog Computer (Some Connections Between Logic, Differential Equations, and Analog Computers)” *Transactions of the American Mathematical Society* 199 (1974): 1–28.

[10] Wolfram, S. “Undecidability and Intractability in Theoretical Physics.” *Physical Review Letters* 54 (1985): 735–738.