When does a physical system quantum compute?

Clare Horsman, Viv Kendon, Susan Stepney
Universities of Oxford, Durham and York, UK

November 27, 2014

Paper for this abstract: "When does a physical system compute?" by Clare Horsman, Susan Stepney, Rob C. Wagner and Viv Kendon, doi:10.1098/rspa.2014.0182 *Proc. R. Soc. A* 8 September 2014 vol. 470 no. 2169 20140182 (Open Access) and also available at arχiv:1309.7979v4.

Motivation:

Which physical systems compute? In everyday life many computers are unambiguously recognisable: laptops, web servers, smart phones, etc. However, quantum computers, along with other non-standard computing devices, are now opening up the definition of a 'computer' to objects and processes whose computational abilities are much less clear-cut. Beyond deliberately engineered systems, there are also questions about computing in natural processes. Which physical entities can support a computational description? Do biological or other self-organising systems compute; is the universe a (quantum) computer? The ability to answer these questions will significantly aid us in identifying, exploring, and benchmarking new quantum computing technologies; and give important foundational insights into how the abstract concept of computation relates to the physical objects of the world around us.

Building a quantum computer that performs useful computations is coming within reach of experimental teams around the world. Determining whether the physical devices these teams construct are actually performing a *quantum* computation is thus an urgent question. The analous question also applies in the broader context of defining when some physical system is being used as a computer of any sort, and not just by humans. The lack of consensus on how to address these issues has lead to confusion over new computing technologies, the unclear use of computing analogies in biological descriptions, and even claims that every physical process is a computation.

Context:

Eight years ago, the 'first commercial quantum computer' was launched by D-Wave; its history since then is an object-lesson in the practical problems caused by an inability to define computing behaviour in physical systems. With the recent increase in support for quantum technologies, the space occupied by the D-Wave machines is about to get very crowded. A reliable way of telling which new technologies are actually computers (not simply experiments or noncomputing tools), and also which are *quantum* computers, will provide the confidence in these devices that is essential to progress towards deployable quantum computing.

Quantum computing devices have historically been analysed using the DiVincenzo criteria, a set of requirements that a quantum computer should fullfil. While a useful guide to what the experimentalists need to achieve, they are not sufficient or even appropriate to determine the computing capabilities of increasingly complex quantum devices. There is much useful

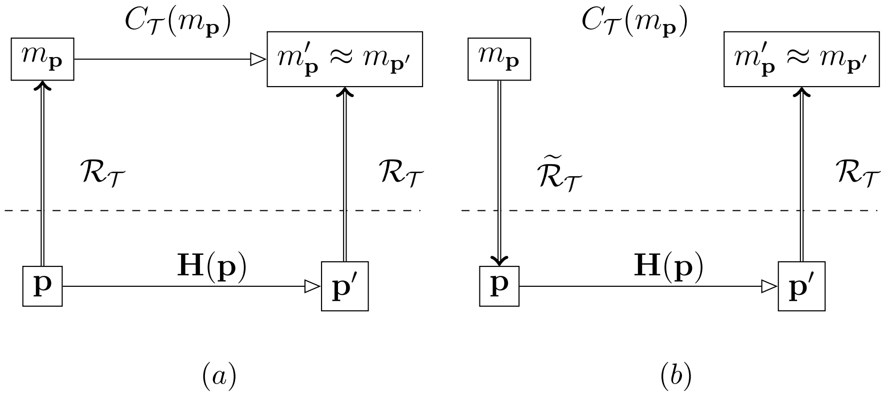

Figure 1: The framework. (a) $\mathbf{p}$ is some physical system, evolving to $\mathbf{p}'$ under the laws of physics $\mathbf{H(p)}$; $\mathcal{R}_{\mathcal{T}}$ is the representation relation that relates a physical system $\mathbf{p}$ in the real world to its model $m_{\mathbf{p}}$, relative to some theory $\mathcal{T}$, in the abstract world (this relation is not a mathemmatical relation, as it steps between the physical and abstract worlds: see paper for details); $C_{\mathcal{T}}(m_{\mathbf{p}})$ is the abstract evolution, to some model $m'_{\mathbf{p}}$. We say that the diagram *sufficiently commutes* when the two models $m'_{\mathbf{p}}$ (calculated from $C_{\mathcal{T}}$) and $m_{\mathbf{p}'}$ (represented from $\mathbf{p}'$) are sufficiently close for our purposes. This is a matter for experimental verification. (b) Using a sufficiently commuting diagram to physically compute. We use the inverse representation relation (a non-trivial inversion: see paper for details) to instantiate (encode) our desired abstract problem in a physical state, let it evolve, then represent (decode) the resultant state, and interpret this as the result of the computation $C_{\mathcal{T}}(m_{\mathbf{p}})$.

work focused on aspects of this problem, such as verification of quantum computations [1] and methods for compressed tomography [2] to treat larger quantum experiments, but no way to integrate these into a full picture of physical computation. The current lack of standardisation, and problems with scaling and integrating proposed components, frequently cloud the issue of whether a system is performing a useful quantum computational function, or even if it is computing at all. As quantum devices leave the proof-of-principle stage, a new and more sophisticated framework for computation is required.

Our framework:

Questions of when a physical system is computing are questions about the relationship between an abstract object (a computation) and a physical object (a computer). The standard approach to representing a computer in theoretical computer science is as a mathematical object, for example the concrete semantics of Abstract Interpretation [3]. This entirely mathematical approach does not tell us about the relationship to the *physical* computer: the assumption from the beginning is that the concrete semantics are somehow a truthful representation of the physical system. However, this begs the question and tells us nothing about what that relation consists of, and how it can be determined. What is needed is a formal language of relations, not from mathematical objects to mathematical objects (as is usual in theoretical computer science), but between physical objects and those in the abstract domain.

In this work, we provide just such a rigorous treatment of the abstract-physical interface. The core of our framework is the representation relation [4], taking physical objects to abstract objects. The power of this framework comes from its ability to deal formally with this relation (which must exist for science to be possible) without needing to resolve the long-standing foundational problems of the relationship of the abstract to the physical. Experimental science, engineering, and computing all require the interplay of abstract and physical objects via representation in such a way that formal diagrams such as figure 1 commute: the same result is gained through either physical or abstract evolutions. Our key result is given in terms of such commuting diagrams and their relationship to theoretical prediction in experimental science: *computing is the use of a physical system to predict abstract evolution.*

The framework allows us to distinguish the distinct cases of when a physical system is being experimented upon (where the theory under which it evolves is being characterised), when is is merely evolving under the laws of physics, and when it is being exploited as a sufficiently well-engineered device to perform computation. It also allows us to determine where the computation is happening: in the device itself, or in the essential encoding and decoding of the relevant input and output information. For example, with our framework, it is possible to show that some alleged super-Turing and hypercomputers, and apparently paradoxical 'stone' computers [5], perform their alleged computations outwith the physical device itself.

In Summary:

We have produced a rigorous framework in which the question of whether, and what, a physical system is computing (*in addition to* evolving under physical laws) can be posed, and answered. As quantum computation comes of age in an era of ubiquitous computing, our framework will allow us to specify the boundaries of, and relationships between, computation and the physical world with illuminating precision.

References

[1] Stefanie Barz, Joseph F. Fitzsimons, Elham Kashefi, Philip Walther Experimental verification of quantum computations ArXiv:1309.0005 [quant-ph].

[2] David Gross, Yi-Kai Liu, Steven T. Flammia, Stephen Becker, Jens Eisert Quantum state tomography via compressed sensing Phys. Rev. Lett. **105**, 150401, 2010.

[3] Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252. ACM Press, New York, 1977.

[4] Bas C. Van Fraassen. *Scientific Representation.* Oxford University Press, 2008.

[5] David J. Chalmers. Does a rock implement every finite-state automaton? *Synthese*, 108(3):309–333, 1996.

Paper for this abstract: "When does a physical system compute?" by Clare Horsman, Susan Stepney, Rob C. Wagner and Viv Kendon, doi:10.1098/rspa.2014.0182 *Proc. R. Soc. A* 8 September 2014 vol. 470 no. 2169 20140182 (Open Access) and also available at arχiv:1309.7979v4.