The state of quantum computer science

Chris Heunen

Computer devices using the laws of quantum mechanics are coming on to the market. Such quantum hardware can solve certain problems much more efficiently, but by nature they are also very hard to program or reason about. To use their full potential, we need to go back to the basics first.

Quantum devices are imminent. Proofs of concept, with a handful of bits of memory, have existed for over a decade, and laboratories the world over are racing to scale these up. The Canadian company D-Wave will in fact sell you a machine, that it claims can perform quantum computations with 512 bits of memory, which the likes of IBM, Google, and NASA, have bought. Such full-blown quantum computers can quickly answer questions that are very hard for ordinary computers. A famous example is factoring large numbers into primes. The complexity of this question underlies many cryptographical techniques, whose security is therefore threatened by quantum algorithms. Luckily, companies such as ID Quantique and MagiQ sell quantum communicators. These devices, which are more like phones than computers, guarantee communication that is secure by the theory of physics, rather than by open computational problems.

For all their benefits, however, quantum devices are very difficult to program. Currently, quantum protocols are designed by wiring together basic components "by hand", rather like electrical circuits. There is nothing like the programming languages and development environments modern programmers are used to, nor anything like the high-level concepts from computer science like recursion. Worse still, once you have successfully programmed a quantum protocol, it is exceedingly difficult to *prove* anything about it, including proving that it does what it should do! These are fundamental problems, that run straight to the heart of the interpretational difficulties with quantum mechanics. We have learned to use quantum mechanics to great effect, but don't *really* understand it. We have stumbled on some fascinating quantum protocols, but they are hard to come by. To see why, let's go back to the basic logic of (quantum) computer programs.

State spaces and logic

A computer is a physical object, and is therefore governed by the laws of physics. We ordinarily think of physical systems as being in some state, that evolves over time as the system undergoes transformations and interactions. Together with dynamical behaviour, this *state space* completely determines the system, and can ordinarily be any set. In this regard classical mechanics perfectly matches Turing's view of a computer as a state-based machine.

Reasoning about computations in this perspective just comes down to answering questions such as "Is the outcome of this computation 37 on input 1?" That is, *propositions* correspond to subsets of the state space, namely the collection of those states in which the output variable has the right value. Propositions can easily be manipulated, for example by disjunction ("Is the outcome either 37 or 42?") or conjunction ("Do these two inputs lead to the same outcome?"). This ordinarily follows the rules of classical logic à la Boole.

Quantum logic

In a quantum setting, however, logic rapidly becomes very counterintuitive. Quantum mechanics dictates that the state space is now no longer just a set. Instead, it is a *Euclidean space*, where the individual states are vectors that you can add, and whose angles you can measure. Propositions no longer corre-

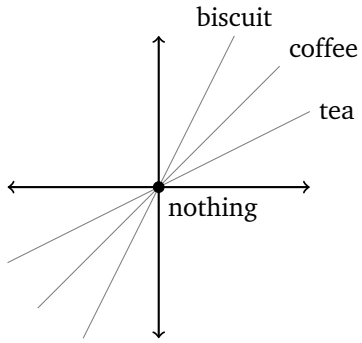

Figure 1: Quantum logic is not distributive

spond to subsets, but have to be Euclidean subspaces in their own right. That is, a disjunction of two propositions is no longer just the union of both sets of vectors, but rather the smallest Euclidean subspace containing those vectors. For example, the state space of a single quantum bit is the two-dimensional plane. Imagine two propositions, corresponding to one-dimensional lines not at a right angle, that we'll label "tea" and "coffee" (see Figure 1). Their disjunction is the whole plane, whereas their conjunction is just the origin, or "nothing". It follows that quantum logic is not distributive:

$$\begin{aligned} &\text{(tea or coffee) and biscuit} \\ \neq\ &\text{(tea and biscuit) or (coffee and biscuit)}, \end{aligned}$$

because the former proposition equals "biscuit", whereas the latter is "nothing".

Needless to say, this spells disaster for quantum programmers, who have to take watchful care to play by these strange rules rather than let their basic intuition creep in. It is much more desirable to have a programming language that does not require such counterintuitive expertise on the part of the programmer.

Quantum state spaces

The counterintuitiveness of quantum logic arises because propositions can make angles other than 90°. As long as we restrict ourselves to considering only propositions at right angles, say those aligned with the axes, the laws of Boolean logic prevail. By rotating our axes, every proposition can still be considered this way. Hence we can safely think of a quantum system as a collection of ordinary state spaces, namely those states aligned to some choice of axes, or *classical viewpoint*. The problem crops up when different classical viewpoints interact.

In this sense, Euclidean space is hardly a good model for states of a quantum system. In fact, famous and deep results by Bell, Kochen, and Specker, rigorously prove that it is *impossible* to conceive of any state space that completely determines the quantum system, at least, if you want it to be consistent across classical viewpoints.

Ambition

This no-go theorem seems to be the end of the story. What good is a state space if it predicts different results depending on the way you look at a system? But we can turn this caveat on its head! Let's take an "active" notion of state space, that incorporates all classical viewpoints as a primitive ingredient, as well as how they are related to one another. Even though it is no longer "spatial" in our ordinary geometric sense, this information *does* turn out to completely determine a quantum system in a consistent way, and is moreover intuitively understandable.[1] Developing this new, "active", notion of state space for quantum systems, and its logic, could greatly ease the development of protocols for quantum hardware.

[1] As a bonus, this turns the spotlight on relationships between systems. Understanding these is key to making intuitive sense of *entanglement*. It would take us too far afield to discuss it here, but this phenomenon lies at the root of many of the benefits of quantum computer science.