How Quantum Can a Computer Be?[1]

Elham Kashefi[2]

Prisme N°29

September 2014

[1] This text is a transcription of the presentation given by Elham Kashefi at the Cournot seminar, « Quantum Turing Testing » in April 2014, including comments by Damian Markham and questions from the audience. The video is available for viewing on the Cournot Centre website: www.centre-cournot.org.

[2] **Elham Kashefi** is a computer scientist working as a reader in the School of Informatics at the University of Edinburgh and a researcher at the French National Center for Scientific Research (CNRS) joint research lab in Information and Communication Sciences and Technologies (LTCI) at Telecom ParisTech based in Paris. Associate director of the recently established UK Networked Quantum Information Technologies (NQIT) Hub, she is in charge of developing applications to foster the link between experimentalists and the computer science community. She is also a member of the Paris Center for Quantum Computing (PCQC) and the Young Academy of Scotland (Royal Society of Edinburgh). Kashefi has trained in and pioneered a transdisciplinary research environment, investigating all aspects of quantum information processing from theoretical perspectives to actual experimental implementations.

Summary

This text briefly reviews the history of quantum computing to provide a backdrop to the new emerging field of quantum technology, which is raising new challenges. In particular, quantum computing has an acute verification and validation problem: on the one hand, since classical computations cannot scale up to the computational power of quantum mechanics, verifying the correctness of a quantum-mediated computation is challenging; on the other hand, the underlying quantum structure resists classical certification analysis. The text concludes with recent progress on how to evaluate today's quantum computing devices so that we can effectively exploit tomorrow's.

In this text I will touch on various aspects of quantum theory and include a little about how I personally got involved in it. So I'll start with my story, before briefly running through the history of quantum technology and then presenting our method of testing in more detail. Originally, I trained as a mathematician, and I was planning to do combinatorics in Canada when I left Iran. Just before leaving, an offer came to go to Imperial College, and I thought, "well, I'll learn some English…", so I went there and did a Ph.D. in mathematics and computer science. I was finding it all quite pointless and boring, and then out of the blue one of my friends said to me, "You should do quantum computing", and my reaction was "*what* computing?" She quickly introduced me to my supervisor-to-be: a physicist (not a computer scientist) who presented to me all the possibilities of quantum computing, factoring numbers, and so on. And I thought, "this is just too good to be true!" I didn't understand a single word he was saying, but everything was described as algebra, and as a student of combinatorics in Iran, I had studied algebra almost from birth. So I knew the mathematics of what he was talking about, but I had no idea about the application; it took me ten years to understand the trap I had walked into, but anyway that's how I ended up studying quantum computing.

So what is quantum technology? Everything started with a vision of Richard Feynman's in 1982: since it seemed so hard to simulate quantum systems, quantum physics and particles on classical machines, he proposed that we build a quantum machine to do it. It was a visionary approach, like some kind of crazy science fiction, but since then, there has been spectacular progress. Over the last 40 years, great strides have been made from complexity theory to cryptography, from simulation to sampling, from tomography to implementation, from the foundation to the interpretation of quantum mechanics. Each of those words could be the subject of its own paper, but what I'd like to do here is walk you through this history, up to what I call the "Feynman loop". It is an exciting time for the field, because we are at the place where we want to prove that what we are performing and observing is indeed quantum. Initially the question was: "Is it possible to build something quantum? Could it be the case?" As I will show you, it is possible, it can be built. But now the question has turned to: "Is it really quantum?" And if we don't answer this testing question at the end – this verification – then you might say that Feynman has played a joke on us: he had this vision, we realized it, but we can't prove it. We cannot really literally prove that it is quantum. So that is my big end story, but let's go through it from the beginning.

There are a lot of aspects to quantum, but I will focus on the algorithmic, computational aspect. The first day I learned about it, I thought: "I can do another algorithm; it's not such a big deal", but I haven't been able to achieve that yet! So we have this machine, which I'm not going to describe in detail yet, which allows you to obtain a speed-up over the classical machine that is beyond normal. It's not a simple case of setting up a new computer, building a model and making it go faster: the model actually breaks down. There is something that defines the foundations of computation, called the Strong Church–Turing thesis. This says that any model that anyone invents is going to be equivalent to the standard Turing Machine up to reasonable overheads. The quantum computing model, however, is breaking down that thesis. Due to quantum features, such as superposition, non-local correlations, interference, and so on, we think that there is a speed-up. Actually, we still don't really know why the speed-up is there, after 40 years, but there are things that make this exponential speed-up happen – which remains a completely unknown world. Feynman's first followers in this field were neither experimental physicists nor engineers to build it; they were actually theoreticians. They said: "What! There is a model that gives you an exponential speed-up? For what reason?" They were the first generation of scientists to get interested in quantum computation.

To give a very rough outline: if you take the normal classical computation, the normal way you think of an algorithm is to do the first step, and then the next step, and then the next step…, and so on. But then probability enters into the picture, with the idea of tossing a coin, of adding randomness. Now, instead of going through one line, you're going through a spread of lines, and things might become faster (although there is still the question of whether it's really faster or not). And then it seems that quantum probability allows you to obtain cancellation: it's as if we're still going through this normal probability distribution, but they are not real numbers; they are complex numbers, and they can cancel each other out. Because of the interference between these random lines, the completion is faster. It sounds counterintuitive, as if by exploring less space we are obtaining something greater. See Figure 1 below by Richard Cleve, which depicts this feature.

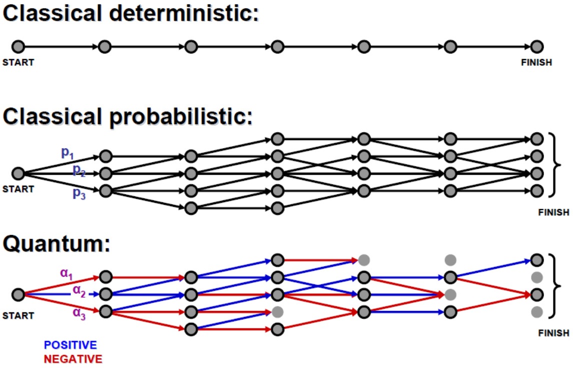


Figure 1: A comparison of deterministic, classical probabilistic and quantum computing.

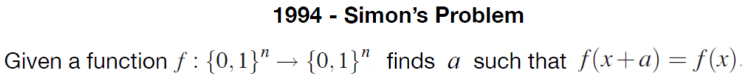
To give a very brief outline of the history, in 1985, David Deutsch, one of the founders of the field, presented the first quantum algorithm, which was then extended by Richard Jozsa in 1992, creating a definite shift in the mind-set and thinking:

$$f : \{0,1\}^n \rightarrow \{0,1\}.$$

This is actually an algorithm of no practical use: basically, you assume that you have a Boolean function that is assigning the values 0 and 1 to strings of n-digit binary values. It could be anything; some of the strings are good, some bad, but in any case, it gives you 0 and 1. Now, it is promised that this Boolean function is either constant – for all the inputs it gives you the same 0 or 1 – or it is balanced – half of them are 0 and half of them are 1. Imagine you have a black box, and every time you enter, for example, "what is the value of 00111010?" it gives you one number out. Then your task is to figure out – without looking inside this box, but by just pressing the button and asking it questions – whether it is constant or balanced. Imagine you are classical, and you have a classical algorithm, so how long will this take? What is the process to solve this problem? Deterministically, to determine with certainty whether this function is balanced or constant, you need to keep playing. We didn't ask you to evaluate all the values – we don't care about the values, we just want to know whether the function is balanced or constant – but you do not have any other solution than to compute all the values to see whether they are constant or balanced. More precisely, you need to ask this box 2^{n-1} questions, out of all the strings you can possibly create, to decide. There is no other solution, except maybe with probability. But it has been shown that in quantum theory, this problem can be solved with just one question. To me, this is one of the most beautiful aspects, because literally quantum theory allows you to perform the task you are set to do. Quantum theory will not evaluate all the values or give you those values, but it will allow you to answer precisely, in a very parsimonious way, exactly the question you're asked.

$$|f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle$$

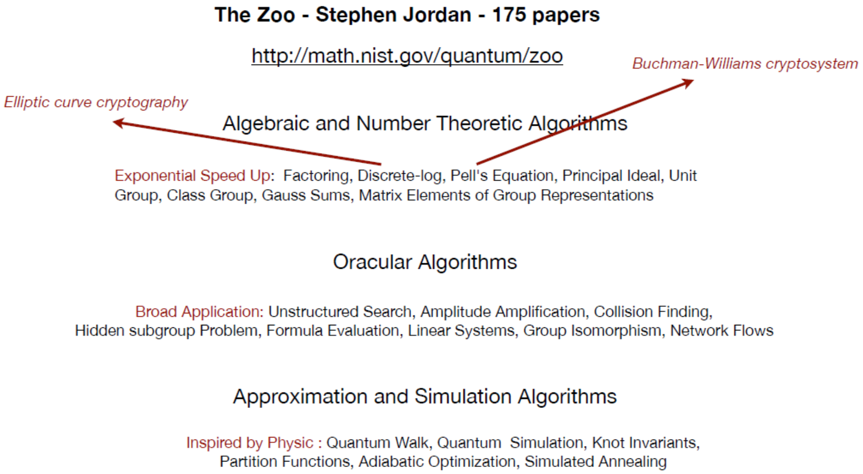
The detail is that with only one query (due to superposition, and so on), you can create a physical state – a quantum state as formally written above – that if the function is constant, there will be one particular element in the vector space, and if it is balanced, it will be the other one, the orthogonal. We also know in quantum theory that the orthogonal state can be determined. The details do not matter, but what is important to grasp is that with quantum tools, we can solve this problem exponentially faster. And this simple observation has become the foundation of a few other more useful quantum algorithms, demonstrating quantum speed-ups. You might have heard of Shor's algorithm (named after the mathematician Peter Shor) that allows you to factor numbers – exponentially faster than any known classical algorithms – so that the cryptography scheme collapses and the whole security system has to be changed. This is how it came about. But in between there was another algorithm conceived by Daniel Simon in 1994 to find the periodicity of a function where it is promised to have such a structure:



1994 - Simon's Problem

Given a function $f:\{0,1\}^n \rightarrow \{0,1\}^n$ finds a such that $f(x+a)=f(x)$.

Computational speed-up compared to any classical solution (similar to the Deutsch–Josza algorithm) is obtained, as there is no need for the actual evaluation of all the function values. Simon's algorithm shows another speed-up, and the person who was refereeing his paper – remember this is at the beginning of the field – was Peter Shor. He looked at this proof and saw that the same idea could be extended, not to solve this problem, which is rather useless, but to solve the factoring problem. This is known as Shor's period-finding problem: given a n-bit integer, find the prime factorization, which breaks the RSA cryptosystem.[3] And that was the breakthrough, the big bang in the field, because suddenly everybody – all the governments, the National Security Agency (NSA), and so on – said "Wait a second. If they can factor numbers and break the RSA code, we'd better keep a close watch on what's happening in this field!" And now there even exists an Algorithms Zoo.



The Zoo - Stephen Jordan - 175 papers

http://math.nist.gov/quantum/zoo

Buchman-Williams cryptosystem

Elliptic curve cryptography

Algebraic and Number Theoretic Algorithms

Exponential Speed Up: Factoring, Discrete-log, Pell's Equation, Principal Ideal, Unit Group, Class Group, Gauss Sums, Matrix Elements of Group Representations

Oracular Algorithms

Broad Application: Unstructured Search, Amplitude Amplification, Collision Finding, Hidden subgroup Problem, Formula Evaluation, Linear Systems, Group Isomorphism, Network Flows

Approximation and Simulation Algorithms

Inspired by Physic : Quantum Walk, Quantum Simulation, Knot Invariants, Partition Functions, Adiabatic Optimization, Simulated Annealing

Figure 2: Algorithms Zoo

[3] RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key, which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem.

This zoo contains hundreds of papers and lots of other interesting quantum algorithms that demonstrate various types of speed-ups. But the real claim to fame in this algorithmic direction is that a lot of number theory problems, which appear to be very hard to solve in the classical world (and our whole cryptosystem is based on that difficulty), are precisely the ones that quantum computing is going to be able to solve. Hence the need for a new crypto system, as I will mention later. But to finish off the discussion on the algorithmic direction of the field, we must recall that Feynman's original idea and motivation for building a quantum system was to simulate physics, because that seemed to be the best application. And that is what is really happening now. While we are able to build small physical systems to implement Shor's factoring algorithm in principle, the real excitement is in the field of quantum simulation: finally we have a device to simulate systems that we were never able to simulate before.

While proving a quantum computer is truly more powerful than any classical computing device is a challenging task (equivalent to one of the major unsolved problems in computer science: P versus NP), there is still strong evidence that for certain samplings, a problem quantum computer proves more powerful taking into account some complexity assumption. And through this exploration we might succeed in refuting the Strong Church–Turing thesis – the very foundations of computer science. The other fun thing about this story is that if quantum computing could be built, then it is breaking most of the commonly used cryptosystems. That would also mean that the current technology we have for security is really only secure as long as we cannot build a large-scale quantum computer. It's difficult not to get excited about that – literally right now we are speaking in terms of "post-quantum cryptography". Meetings are being held to discuss what the world will be like in order to build a framework for analysis of security in the event that tomorrow, in 50 or 100 years, quantum computing really exists.

There is a saying that what quantum computing technology and information takes with one hand, it gives back with the other. So we have quantum cryptography to replace the classical model. While classical cryptography depends on the difficulty of certain problems, quantum cryptography is based on the axioms of physics. If what everybody believes about quantum mechanics is correct – and for 100 years nobody has refuted quantum theory – then we can build a cryptosystem that takes quantum theory as its basis of security, rather than difficulty. So we have one cryptosystem, which is based on the fact that so far, mathematicians, who are so smart, have not been able to solve this problem, and another cryptosystem based on a theory that physicists, who are also really smart, have not been able to refute! The term used for quantum cryptography is "unconditional security". In quantum cryptography, we are no longer working on the basis of number theory. Instead, we are using quantum features such as no-cloning, meaning that you cannot take a physical system and make a perfect clone of it. Another feature is that every time you ask a measure of a quantum system, the state changes. It is impossible to make an observation without changing what you are trying to observe. That principle, for example, can be used against eavesdroppers. The method couldn't be better, because every time an eavesdropper wants to look at something, to see what's going on, they're going to change it. For the users, the red flag goes up, telling them that something's happened. Classical cryptography cannot provide that kind of guarantee of informing you when non-secure communication has been used. So that is the promise of quantum cryptography, which is the other jewel of our field.

Does the definition of security change with the global framework? There has been a lot of effort in this direction. The mathematical framework that defines the notion of security is ultimately probability theory; it involves the probability that the variable describing the knowledge of the eavesdropper is independent from the random variable describing the secret information of the parties. Ultimate security means making sure of that independence. That is how the security of classical cryptography is defined, and that is precisely what quantum cryptography has done. So ultimately, it's all about the independence of these random variables, the amount of knowledge that has been leaked. The same toolkit from Shannon's theory has been successfully extended to the quantum setting to derive a unified framework for describing security. Similar to the classical scenario, however, there is a huge problem in proving the actual quantum devices fulfil the claim of unconditional security as there remains a gap between theoretical proof and experimental demonstrations. Nevertheless, we believe that with intensive joint effort, we will soon resolve this issue.

Quantum cryptography started with Stephen Wiesner, who published a paper in 1983[4] proposing the idea that when someone prepares a random photon (a

[4] S.J. Wiesner (1983), "Conjugate Coding", *SIGACT News* 15:1, pp. 78–88.

photon being a physical system in a random state), nobody else has any knowledge of it. Anyone who tries to observe this photon is going to change it, and this change can be detected. This was a beautiful observation, but he had trouble getting the paper published. It took him more than 10 years as the field of quantum cryptography did not even exist at the time. He was too far ahead of his time, and only now is the idea he proposed of unforgeable secure quantum money coming back. So this is another example of a very simple, but visionary idea. And the story goes that some scientists used his crazy quantum idea and adapted it to classical cryptography, which built the foundations of classical computer cryptography. Then later on, Charles Bennett and Gilles Brassard (1984)[5] along with Artur Ekert (1991)[6] worked on the public key distribution problem. This opened up a completely new direction. Governments and security agencies are all interested for obvious reasons, but there is also the fundamental idea that security based on an axiom of physics rather than an axiom of mathematics is a different world, and in fact it touches on the very foundation of theoretical physics today. Since then there has been many other quantum protocols introduced and various impossibility results have been demonstrated too. The field has rapidly expanded within the last decades with even a few commercial companies offering Quantum key distribution devices.

What are the perspectives for quantum cryptography? We are now at the stage of widely setting up quantum key distribution networks. In 2008, the Secure Communication based on Quantum Cryptography project (SECOQC) used 200 km of standard fibre optic cable to interconnect six locations across Vienna and St. Pölten. Telecom ParisTech is one of the pioneers in the implementation of these networks, and there are many other European countries involved. The DARPA 10-node quantum network has been running since 2004 in Massachusetts, between BBN Technologies, Harvard University, Boston University and QinetiQ. The Tokyo QKD Network involves seven partners: NEC, Mitsubishi Electric, NTT, NICT, Toshiba Research Europe Ltd. (UK), Id Quantique (Switzerland) and All Vienna. In particular,

[5] C. H. Bennett and G. Brassard (1984), "Quantum cryptography: Public key distribution and coin tossing", in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing,* volume 175, p. 8, New York.

[6] Ekert, Artur (1991), D.Phil. Thesis, Oxford, Physical Review Letters, 67, pp. 661-663.

recently Europe and China have been gaining the upper-hand in the race to develop earth-to-satellite quantum communication.

So to summarize, quantum information and computation theory – and more specifically formalizing the quantum Turing machine and the quantum model – was initially developed by theoreticians in the 1980s. At that time, experimentalists and engineers were extremely dubious, considering it to be science fiction. They became involved during the 1990s as they began to observe photons, free electrons, and so on. It was at that point that the theory left the realm of science fiction and became possibility; they saw that the ideas of Deutsch, Bennett and Brassard could be implemented. A new wave of applications came in the 2000s with early start-up companies, moving even further from the original idea that changed our perspective. Now there could be real applications, real simulation, and not just observations of whether a function is constant or balanced, or whether there is secrecy. The effect, the excitement of quantum, has not lessoned: we can still mesmerize any innocent undergraduate student with the story of quantum; once they've learned about it, they don't want to do anything else. We are still struggling to fully understand what is going on, and yet the field has matured, and new challenges, questions and directions continue to be created as we progress.

In fact, the field reinvents itself every 10 years. A new era is emerging with a new buzzword – "quantum technology". Let me give an example from the UK, on a project that I have been involved in. In December 2013, the UK government announced the UK National Quantum Technologies Programme: £270m of investment in quantum technology. This didn't come out of nowhere as the UK has been pioneering the field of quantum information since the very beginning, and more recently a huge lobbying effort was put into effect to strengthen further the leading role of the UK. Nobody, however, expected a sum of that magnitude, and £150m of it is going directly to research, in the form of grants. The timeline is tight: the announcement of £270m in December; submission of interest on 1 February; submission of outline proposals, including the hub, partnerships with industry, and so on in March; deadline for full proposals in June; interviews and decisions in September, and the centre should be up and running in December 2014. The initiative's level of speed and impatience, has been very intense. This is money delivered against a business plan – "to develop emerging technologies; a multi-stakeholder, technology-focused initiative to last for an initial period of five years", covering the particular topics of interest in sensors, metrology, secure commerce, computation, simulation, and so on: the UK National Quantum Technologies programme is focused not on quantum science, but on the exploitation of that science for technological benefit. Usually, most theoretical scientists would be up in arms, saying "you must be kidding me!", and not have anything to do with it. It is hard, however, to resist £270m, and it is a very exciting, unique opportunity. To some extent, we have to change direction, but the change of direction is an interesting question: the focus is now on "let's build it!" It's no longer a question of proof of principle. They want to see a prototype at the end of five years that they can convince industry to take over. The effect it will have on quantum science, I cannot say. Everybody who has been working on theoretical topics in this field probably now has a new focus on quantum technology. The central objective of my research that I will describe next is an end-to-end exploration through all these themes. I will be leading a unique approach to Verification of Quantum Technology that is currently the key challenge for making the above transition from theory to practice possible.

When our field began in the 1980s, the question was: "What is a quantum computer?" After 10 years, as Feynman said, you don't understand quantum mechanics, you just get used to it. But from wanting to know what a quantum computer is – the definition, structure, applications, and so on – the question has now changed to: "Is it a quantum computer?" Even more interesting: is it a quantum box? Forget the word computer – the universal machine that has become a laptop used for everything. And this is a realistic question. There is a company in Vancouver that claims to have built a quantum annealing device. It's a box, a gigantic box, but is it really a quantum computer? You want to know, because they are not playing with two or three photons; they are selling a 1000 qubit quantum computer. But is it a quantum box? They don't claim it to be a universal machine: it is purpose-built for particular simulations. I should explain why this question is not trivial. It is not enough simply to say: "Well, we have a factoring algorithm that works better than any classical machines, so it's probably quantum". That is not proof that something quantum is happening in this box. We can even forget about the word quantum, and simply ask: "Is this box doing anything correct?"

That is the question that has suddenly become central now, because whatever this quantum technology produces in the end, researchers will want to answer this question. With all the money being spent and all these research centres working together to build this machine, it is essential to know the following. Is it quantum? Is it correct? Is it doing what the specification asks for? If you look at classical computer science over the last 30 years, there have been Turing award winners for the very same questions of verification, model checking and testing. Here, however, that's not possible, because what makes something quantum makes its verification quantum too. We believe that quantum computing is exponentially more powerful than classical computing, which means we cannot run some sort of simulation and plug in the classical testing that we used to do and say: "Yes, the result is correct". The very reason that this thing is supposed to be a quantum box means that everything that we have done so far does not apply; we have to start all over again. The problem that we are looking at now, which makes sense in the context of this quantum technology, is how to formalize this question. What structure, what methodology can we use to formalize the question of what it means to be a quantum box? And that is what I am going to go through now.

For me, the answer involves going back to the Turing test. In 1950, Alan Turing envisioned a test of artificial intelligence where you are in a room interacting with a person and a machine in another room. You want to know whether you can tell the difference between the machine and the human, having defined all the rules of the conversation beforehand.

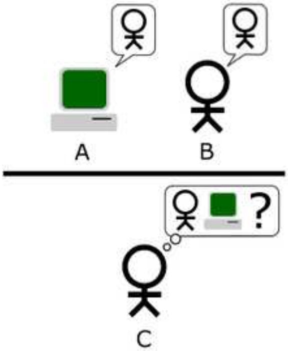


Figure 3: An illustration of the classical Turing test, where player C interrogates A and B and tries to determine which is human and which is a computer.

For quantum computing, it is almost the same situation: we are interacting with a machine, but now we want to know if it's a classical machine or a quantum machine; it's as simple as that. But do we need to use quantum communication to test "quantumness", or do we stay classical? And the other question is: "Is it efficient?" Since we believe that the quantum computer is more powerful than the classical one, does that mean that we cannot efficiently test the correctness of the outcome? Or is testing the correctness of the outcome another problem, calling for a super-powerful computer? Such questions are only three or four years old.

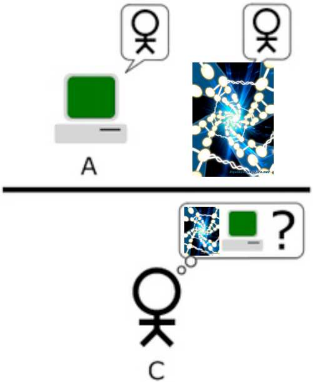


Figure 4: Quantum Turing testing; © EQUINOX GRAPHICS

To sum up, what makes quantum not classical makes its verification not classical either. But in a sense, it's not that "unclassical". I said earlier that we have to throw out everything that's been done in the theoretical context and start over, but that's not really true. The best we can do is to go back, look at what's been done, find the most similar question in the classical context that scientists have tried to answer, and then adapt it to quantum. Interactive proof systems are one of the jewels of the theory of computer science in complexity, verification and testing. Again, this was an attempt to formalize the question of how you prove something. One way is to write a mathematical proof and then go away and test it. That proof will be very limited in terms of progress, however: if you ask me a question, and I'm supposed to give you a proof, and then there's no further communication, you are restricting yourself to a very particular class, like factoring. If I want to prove to you that I know how to factor, you give me a gigantic number; I give you two numbers; you take those numbers and multiply them, and you can see for yourself. The beauty of the interactive proof is that if you bring randomness and communication (interaction) into the picture, then you can do much more than verify the factoring results.

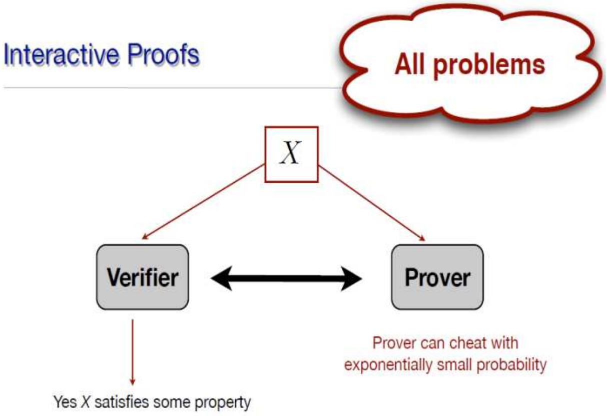
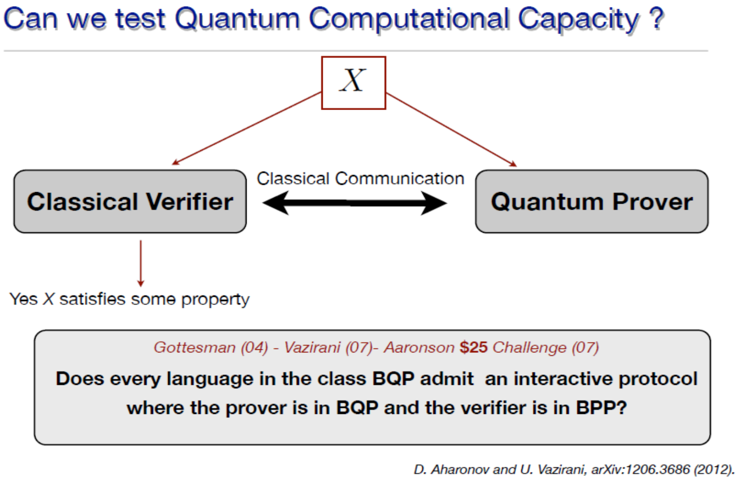

Interactive Proofs
All problems
X
Verifier
Prover
Prover can cheat with exponentially small probability
Yes X satisfies some property

Figure 5: Interactive proofs

Here you have the prover, which is the untrusted server, nature, the device, quantum technology or whatever – somebody who can do a really hard task, but who is untrusted – and you have the verifier, who is very limited, only able to do some very simple types of calculations. The prover can use any sort of complexity class, while on the verifier side, you want to restrict yourself to bounded polynomial-time computing – anything your laptop can do, basically. The verifier asks the prover a question, sees what answer the prover gives, does some calculations, then asks the next tricky question, sees what it answers, and so on. This conversation needs to be limited, and the verifier's computer needs to be limited. In theoretical computer science, the whole class of theorems can be verified like that. It's really powerful. There is a small element of randomness, so you allow the prover to cheat, with a very small probability. Sometimes the prover might get lucky and prove something that is not correct, but in allowing for that little bit of randomness, the world is yours! So the most natural way to formalize the question of whether we can test quantum computational capacity would be by adapting this technique to the quantum setting.

So we are the classical verifier with access to only the classical machine and classical devices. Then there is this "box" (which is going to appear in some magical UK centre after the £50m investment has built it), and we want to ask it questions and get answers, back and forth, and prove the correctness of this box. We have to be a bit careful here, and I'm not yet talking about quantumness: the first step is the correctness of the computation. This was a question that was first addressed in 2004: is quantum theory classically falsifiable or not? Can I use classical logic, classical theory, and test the correctness of the outcome? The question is simple: you will say, well, just run the experiment. Alain Aspect is one of the founders of the field, who has run a magical experiment showing the quantum effect, but there remain loopholes. So how do we go about this?



Can we test Quantum Computational Capacity ?

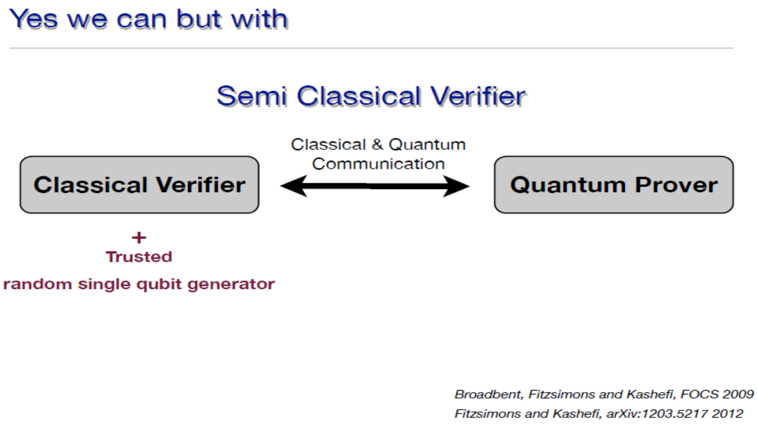
Gottesman (04) - Vazirani (07)- Aaronson **$25** Challenge (07)

Does every language in the class BQP admit an interactive protocol where the prover is in BQP and the verifier is in BPP?

D. Aharonov and U. Vazirani, arXiv:1206.3686 (2012).

Figure 6: Testing quantum computational capacity

We want to check anything that a quantum computer claims to do efficiently. In Figure 6, BQP stands for "bounded-error quantum polynomial" time, something that quantum computers seem to be good at doing. BPP stands for "bounded-error probabilistic polynomial" time, meaning anything that our laptop with a little bit of the classical randomness can do. So if I am in BPP, meaning I toss a coin and do some calculation, we want to see whether I can test the correctness of prover in BQP using the classical verifier. We believe BQP to be bigger than BPP, and that is the whole reason we are doing this, because the quantum computer can do things that we cannot do classically. And the answer is yes, we can:



Yes we can but with

Semi Classical Verifier

Classical Verifier

Classical & Quantum Communication

Quantum Prover

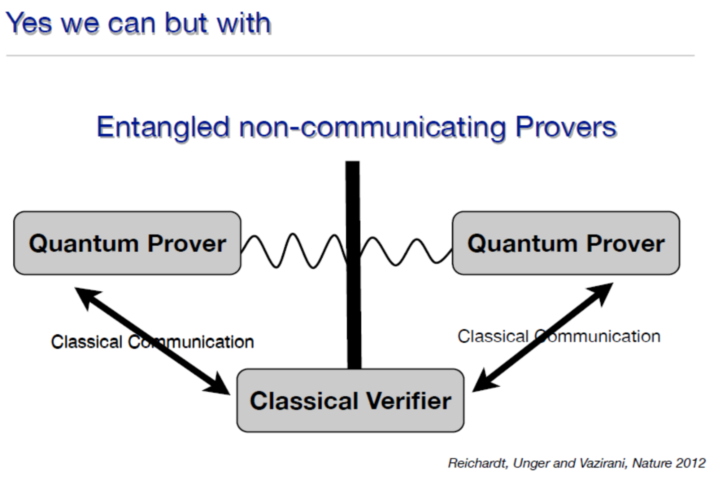
+ Trusted random single qubit generator

Broadbent, Fitzsimons and Kashefi, FOCS 2009
Fitzsimons and Kashefi, arXiv:1203.5217 2012

Figure 7: Semi-classical Verifier

Figure 7 shows the work I did a few years back with Anne Broadbent in Canada and Joseph Fitzsimons in Singapore. So the answer is that you can, but you can't have a completely classical reference. It's not completely BPP – that question is still open. Anyway, we won $10, partly by answering a challenge set by a prominent researcher from Harvard, Scott Aaronson, so that was a big motivation. So how does this work? The verifier's computer is classical, but we give it a little bit of quantum randomness by means of a device that gives a random photon in some direction every time you press it. This is something a physicist can build: it is not a quantum computer; it is a very simple machine with photons (as we demonstrated with the Vienna experimental group). It's like a mini "quantum coin" if you like, and with that quantum coin, the answer is "yes, we can test correctness in an efficient way". Another group in the United States (Reichardt *et al.*) have published an article last year, explaining that there is another way to introduce probability[7]:

[7] Reichardt, Ben, Falk Unger and Umesh Vazirani (2013), "Classical command of quantum systems", *Nature* 496, pp. 456–460, 25 April.



Yes we can but with

Entangled non-communicating Provers

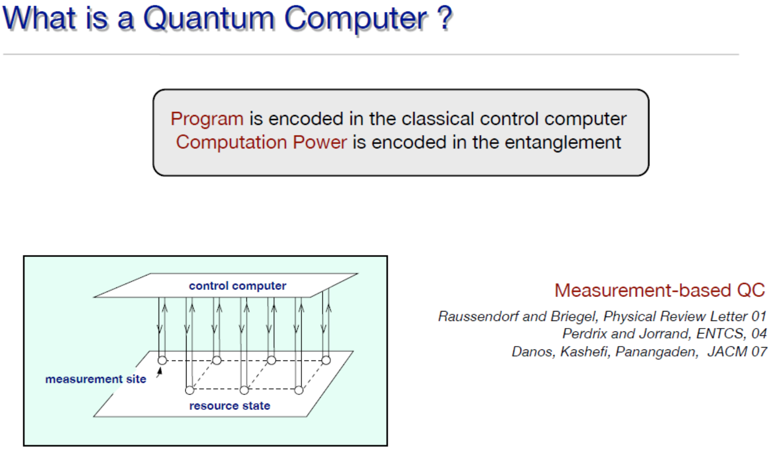
Reichardt, Unger and Vazirani, Nature 2012

Figure 8: Entangled non-communicating provers

Here we have two quantum servers, and the verifier is completely classical, so we don't have the quantum coin any longer. These two quantum servers are not able to talk to each other, so they cannot "cheat". But there is something called "quantum entanglement", which is even stronger than quantum randomness, but actually has exactly the same effect. If two observers share these entangled particles – two particles that communicate together and are then separated – every time one observer looks at this thing, tosses a coin to give a random bit, the other observer will get the same random bit. It's a sort of correlated randomness, which doesn't exist classically. So if you allow the two servers to have these things, again it is possible to test efficiently for correctness. So the randomness (in the form of entanglement) shared between the provers, the untrusted parties, and the randomness (in the form of the "quantum coin") given to the verifier – the trusted party – seem to be somewhat equivalent, and both solve the problem.

We are only just starting to look at the link between these two approaches, but this is the current state of things, and the question is: "Can I get rid of this weird entanglement in testing? Can I get rid of my quantum coin?" In other words, can we move on to achieve completely classical testing? For me, that is the most exciting question to solve. The other reason it is so important is because we need quantum technology verification (especially with £270m investment going into it), and this protocol I'm talking about is still a theoretician's protocol. Twenty years ago, once we had got to this stage, we would have said: "That's nice; the question is solved, let's find the next question." But because the drive is now on to build more and more quantum machines, we need to make these things so optimal that they can actually be used for the practical devices that may soon emerge from it. A lot of other work needs to be done, because as we have shown, there is a possibility. We must now, however, work out how to make this verification technique implementable for the industrial partners who are looking for a way to test.

Let me explain a bit more in detail how our protocol works. First, we need to go a few years back where the question of "what is a quantum computer" was a common line of my joint research with Damian Markham (the Commentator of this text) for a long time. One way of formalizing this is via the so-called measurement-based quantum computing model:



What is a Quantum Computer ?

Program is encoded in the classical control computer
Computation Power is encoded in the entanglement

Measurement-based QC

Raussendorf and Briegel, Physical Review Letter 01
Perdrix and Jorrand, ENTCS, 04
Danos, Kashefi, Panangaden, JACM 07

Figure 9: What is a quantum computer?

It's a bizarre device: the programme is a sort of classical control that sits on top of this quantum entanglement, and then in order to get the full power of the quantum system, we do some clever controlling. This is the "assembly" language of quantum computers, if you like, without the quantum jargon – a sort of quantum punch card operating on the particles. We have spent 10 years formalizing and understanding this, the flow of information, and all sorts of interesting things. But then comes the next question: is it really a quantum computer? As I said earlier, I need to translate the question from interactive proof theory. Now remember, what was supposed to be the case? The verifier must be as simple and as classical as possible. So can I put my verifier – this trusted woman who has no super powers – as the classical control, and my prover – the untrusted superman – as the quantum device? This is the most natural way to go.

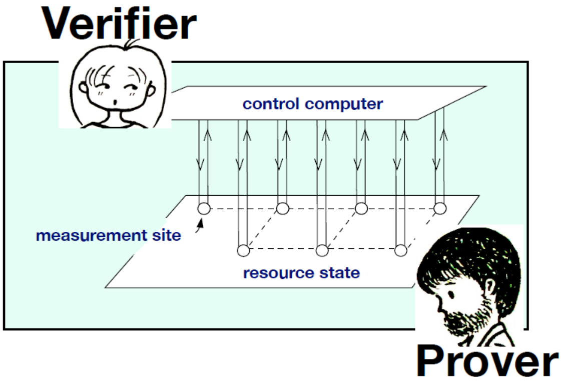


Figure 10: Verifier and Prover

As already mentioned, this model is where everything becomes coincidence. We were just playing with it, and the interactive proof matched so well with it. Probably a different model of quantum computing wouldn't answer the question so well. So the classical control is supposed to act as the verifier – asking the questions and sending the interactions, and so on – and the person who is running this quantum device – getting hold of this entanglement – will be my quantum prover. In this assembly language, the lines going from the control to the resource state are the classical bits, which are describing the angle of measurement. Each of these single qubits is a unit vector in the two-dimensional complex vector space (C2). You send a signal indicating in which direction you're going to measure it, and you can fix this so that they are a discrete set, choosing between eight possible angles on the plane. You then make an observation, and this observation is a probabilistic quantum measurement. So you have a single vector in C2, and you are projecting it with some probability that it will go in one direction, and some probability that it will go in the orthogonal direction. If it goes one way, the value is 0; if it goes the other way, it is 1. That value is then sent back to the classical control, and based on that bit, 0 or 1, the classical control decides that the next measurement should go in a given direction, and you keep repeating this operation. So this is the model: you set up this cluster of entangled states, do the first level of measurement, get the classical probabilistic data, do a little bit of magical calculation, and then decide where the next one is going. And this is universal; you can do any quantum computation you want - factoring, whatever.

Now, if I want to test this thing, and I don't trust this superman, the easiest way to test is by overkilling it. I want to make sure he doesn't even understand what he's doing – he literally becomes my slave. So I make him run the computation, but I want to introduce a little bit of cryptography, the usual cryptography, so he makes the measurement and calculates the data for me, but he doesn't know what measurement he is performing; he doesn't even know which result he's getting. The only thing added is that I make him blind. Despite the fact that I make him do the correct measurements for me, the angle of measurement – this magical number – is hidden. And despite the fact that he is getting the results of the measurement, they are also randomized, using a one-time pad – a classical one-time programme, which is the most secure way of doing cryptography. You just get your message and apply a random bit to it, so that it is completely randomized. You then send that, and the result becomes randomized. This is the classical one-time pad.

Remember that our protocol of verification needs to have this quantum randomness. I want to make sure that the prover is doing the correct computation, but that everything to his eye is totally mixed. So the security proof, the bottom line, is that the random variable describing the information that the prover gets is completely independent of the random variable describing the secret information.

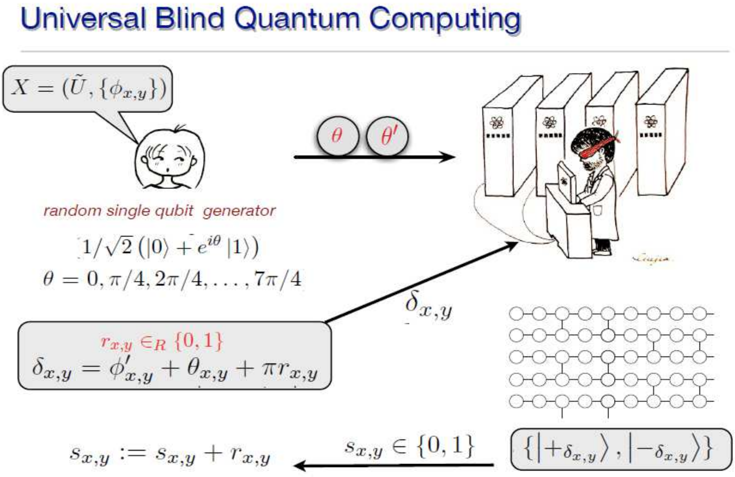
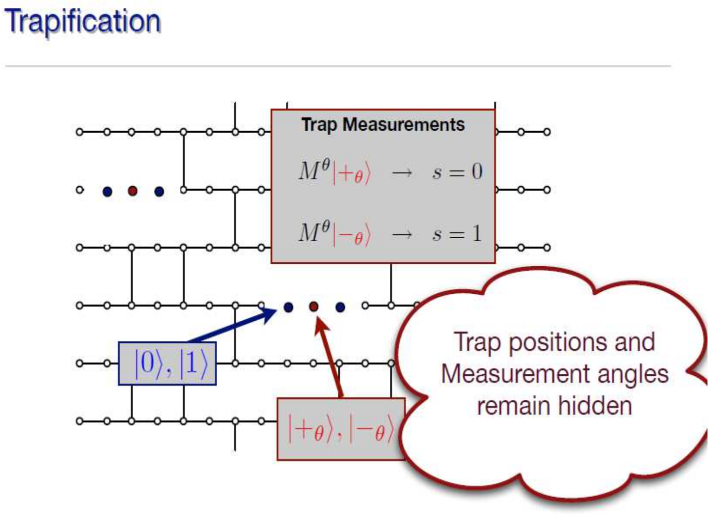

Universal Blind Quantum Computing
$X = (\tilde{U}, \{\phi_{x,y}\})$
θ
θ'
random single qubit generator
$1/\sqrt{2}\,(|0\rangle + e^{i\theta}\,|1\rangle)$
$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$
$\delta_{x,y}$
$r_{x,y} \in_R \{0,1\}$
$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$
$s_{x,y} := s_{x,y} + r_{x,y}$
$s_{x,y} \in \{0,1\}$
$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$

Figure 11: Universal Blind Quantum Computing

The equation at the top-left is the random variable describing the secret in the hands of the verifier, with a one-time pad: this random variable is the key to the crypto-message sent to the prover. Below that, we have the quantum random variable, a uniformly random quantum coin. So you have a single qubit, which is a vector in the two-dimensional variable space, and you are just choosing a particular regular structure. θ becomes a parameter in this, but the point is, quantum theory tells us that when I give you this quantum state, and encoded inside it is my random θ parameter, you can't learn anything (almost) about it, because you don't know how I've prepared it. So this is the strongest element (recall the very original idea of quantum cryptography by Wiesner). So when I give this physical system to the prover – this magical box which has a secret key – it's so powerful that all I need later on is for the computation I want to do to match that key. So in this one-time padding, first the key is sent, and then you do the encoding of the message. Usually, you pick the message, add the random things to it, and then you send the key. But here, you send the key first, and then put the message on top of it afterwards, and everything is encoded, of course. So we are sending these qubits; all of them are randomized, and the prover has no idea what they are. With the magical formula – my secret – I make sure he learns nothing about it. The message is totally classical: you take your secret message and a random string, put them together, and when you give it to the prover, from his point of view, it is totally mixed; it reveals no information whatsoever. But the funny part is that when you take this state, and you put it on this physical system that has the key, it decodes it: the verifier's key and the prover's key cancel each other out, so my computation is correct. So I send my secret randomness, which seems to be not the computation I had in mind; the key's already there; they cancel each other out in this physical machine, and the prover learns nothing.

So the first step takes place before I even know what I want to compute. There is this quantum company that says: "We are here ready to serve you", so I buy my investments from this company: I buy 100 random qubits. I have this little device that someone has built for me, and every time I press it, it creates a sort of random coin. This device tells me "your θ is $\pi/4$, and this is the qubit which is encoding $\pi/4$", and I send it to this company. Then I press it again, and it says "your θ is $\pi/2$, and here is a qubit with $\pi/2$ encryption", and so on. I send 100 of these things, and I classically remember the sequence of $\pi/4$, $\pi/2$, $\pi/8$, and so on. And then, maybe 10 years later, these qubits are sitting in the freezer, nicely entangled within this structure. I haven't told the prover anything; he is just keeping these coins there for me waiting – and then I decide, "actually, I want to factor". "In order to factor, the angles are supposed to be $3\pi/8$, $5\pi/8$, and $3\pi/4\ldots$", so I take $3\pi/8$, add it to $\pi/4$ (to get $5\pi/8$), and I say to the company, "OK, could you please measure for me that first qubit with the angle $5\pi/8$?" Now $5\pi/8$ means nothing, because it's uncorrelated to my initial thing. So here is the next step: I keep sending this information to him. Every time I send a message, the previous rotation is cancelled out, and it implements the correct rotation. So there is a sort of decoding: I give the message to him, he measures and gives the result back to me, and I repeat the process.

How about verification? Once I have this little gadget that allows me to prove that the server doesn't learn anything (apart from the dimension: he learns the size of the computation, but nothing else), it's very easy to test him if I am able to ensure that he's running a computation but has no clue what he's doing. The rest is not so surprising. As he's doing something but doesn't know what it is, I can start setting traps – a little trap here, a little trap there. As I know where the traps are and what the computation is, I can put everything together. So to perform the verification as well as the computation part, I continue inserting the traps, which are designed so that they do not mess up the computation; they are literally part of it. It's just like the way my mum used to test me when I was in school. She would know about some things that happened at school on a given day, and when I came home, she would ask me those questions to see if I was giving the right answer to the things she knew about. That would show that I was probably answering all the questions correctly, because I didn't know which questions she actually knew the answer to. So, there is a high probability that everything is correct.



Trapification

Trap Measurements

$M^\theta|+_\theta\rangle \rightarrow s = 0$

$M^\theta|-_\theta\rangle \rightarrow s = 1$

$|0\rangle, |1\rangle$

$|+_\theta\rangle, |-_\theta\rangle$

Trap positions and Measurement angles remain hidden

Figure 12: Trapification

The middle dots in the above diagram are single isolated traps, particular states that I have prepared that are not entangled with the rest of the planet. Their measurement results are deterministic. In general, measuring a quantum state is in fact probabilistic, but if I measure a quantum state corresponding to the eigenstates of the measurement observable by the operator, the outcome is deterministic. Hence all measurements performed as part of the general computation are probabilistic in my model, and I have no ways of checking their correctness. But then I insert the traps, which are deterministic, to use as my minefield, and if the untrusted machine gives me the right answer for all those minefields, then I know that the whole probabilistic computation is correct. This is the bottom line. If you want to know the framework to prove it, it is the usual probability theory: writing random variables, density operators, calculating the statistical differences, and so on.

To wrap up, I will share one of the things I am working on. With the protocol that I have just shown, for computing a 3-qubit computation (which is pathetic), the required number of qubits to use is of the order of 300. So in reality, to give an exponential bound to make sure that it is correct would require enormous resources. So this is where we are: in theory, we have solved the problem, but the question we face is – do we really need to have the exponential bound in proving the correctness, or can we just compromise on this?

What can we do with 4-qubits

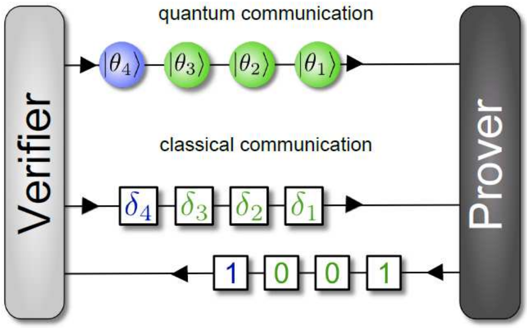


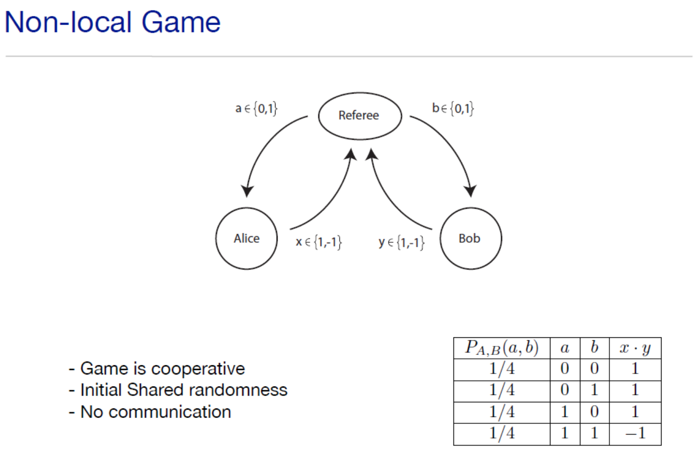
Figure 13: What can be done with 4 qubits?

Here I will present some work I did a year ago with an experimental lab, the Vienna group led by Philip Walther. All he offered was 4 qubits and nothing more, and not all the rotations either. So what can you do with that? This is also the story not just of quantum technology, but also of the fact that theory and experiments need to go together: we're not living in a different world, and that's unique to our field of quantum theory (except for maybe biology). So we talk to the experimentalists, and they say, "OK, nice, you have a protocol; you're happy, well done, but come back when you can do something with our set up". And then when you want to do it again for experiments, every proof has to be adjusted. The proofs need to come from the beginning, and more importantly, the question is, "can we demonstrate anything interesting at all?" I need to give you some background to explain what we did next.

There is a fundamental theorem called the "Bell inequality", which radically changed the field in the 1960s. [8] Its author John Stewart Bell gave a mathematical recipe for checking whether something is quantum and not classical. For me, this is probably the most beautiful fundamental theorem of all time. The idea of quantum entanglement is hard to accept: how could two people toss two different coins independently and happen to obtain the same result all the time, as achieved via quantum entanglement? For some thinkers (see the "Einstein Podolsky Rosen paradox", 1935[9]), the solution was that physicists and mathematicians were simply not clever enough, not capable of giving a complete theory. There must be some kind of "local hidden variable" – something hidden in my hand and something hidden in the other person's hand that our quantum theory does not describe – that produces this correlation. To refute this criticism, Bell proposed a game where the referee gives a random bit, either 0 or 1, to each of the two players, Alice and Bob, and they have to give an answer, either 1 or -1. They can use any local hidden variable they like, any probabilistic theory they want, to choose the answer they give, but without communicating with each other.

[8] Bell, John (1964), "On the Einstein Podlsky Rosen Paradox", Physics 1(3): 195–200.

[9] Einstein, Albert, Boris Podolsky, and Nathan Rosen (1935), *Can quantum-mechanical description of physical reality be considered complete? Phys. Rev*: **47** 777.



Non-local Game

- Game is cooperative
- Initial Shared randomness
- No communication

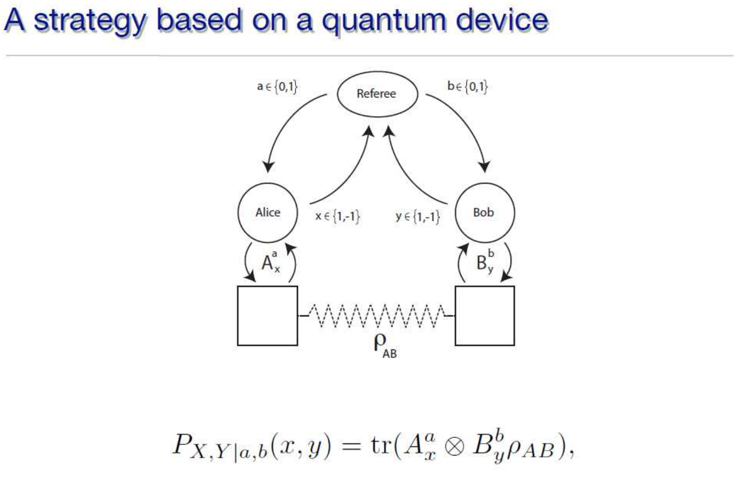
$P_{A,B}(a,b)$	a	b	$x \cdot y$
$1/4$	0	0	1
$1/4$	0	1	1
$1/4$	1	0	1
$1/4$	1	1	-1

Figure 14: Non-local game

The box in the figure above shows the recipe for winning. If the questions asked of Alice and Bob are 0 and 0, the result (x times y) should be 1, and so on. So it's the simplest game you can define – this is the famous non-local game. Using probability theory on the chances of winning, what is the best correlated strategy for this game to make sure that the answers x and y that the players give to these bits of information 0 and 1 match the winning answers in this table (without communication between the players)? What is the probability of winning this cooperative game? If the players are both thinking optimally, the probability of winning is ¾, or 75 per cent. That's because in the optimal deterministic strategy, you can always satisfy three of these equations, but not the fourth:

$$\begin{array}{l} x = f(a) \\ y = g(b) \end{array} \qquad \left.\begin{array}{r} f(0)g(0) = 1 \\ f(0)g(1) = 1 \\ f(1)g(0) = 1 \\ f(1)g(1) = -1 \end{array}\right\} \Rightarrow \left\{\begin{array}{r} f(0) = g(0) \\ f(0) = g(1) \\ f(1) = g(0) \\ f(1) = -g(1) \end{array}\right.$$

And whatever probabilistic strategy you use, whatever theory of local hidden variables or whatever, the probability is always 75 per cent. Now, what Bell proved was that actually, there is a quantum strategy, if the players share entanglement:

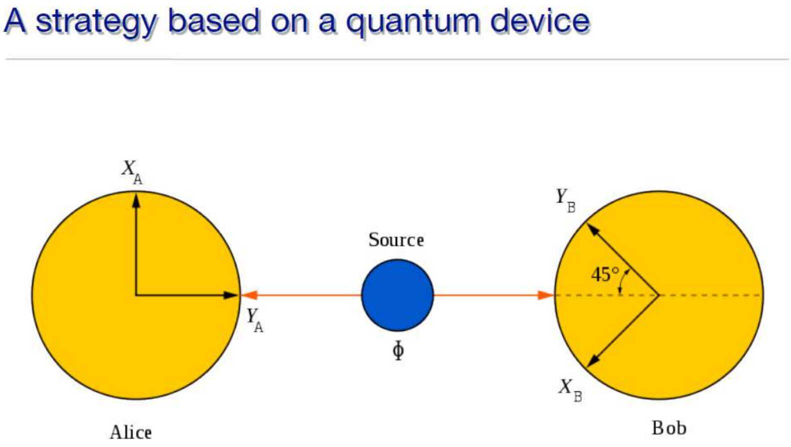


A strategy based on a quantum device

$$P_{X,Y|a,b}(x,y) = \mathrm{tr}(A^a_x \otimes B^b_y \rho_{AB}),$$

Figure 15: A strategy based on a quantum device

where if you do the calculations (see below), the probability of winning is 85 per cent. It's just crazy. Classically, the probability of winning is 75 per cent, but "quantumly" it's 85 per cent, just because of the quantum correlation.



A strategy based on a quantum device

Figure 16: A strategy based on a quantum device

$$\rho_{AB} = |\psi\rangle\langle\psi|, \quad |\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

$$\begin{aligned}
|\phi_1(\theta)\rangle =& \cos(\theta)|0\rangle + \sin(\theta)|1\rangle, \\
|\phi_{-1}(\theta)\rangle =& -\sin(\theta)|0\rangle + \cos(\theta)|1\rangle.
\end{aligned}$$

$$\begin{aligned}
A_1^0 =& |\phi_1(0)\rangle\langle\phi_1(0)|, & A_{-1}^0 =& |\phi_{-1}(0)\rangle\langle\phi_{-1}(0)|, \\
A_1^1 =& |\phi_1(\pi/4)\rangle\langle\phi_1(\pi/4)|, & A_{-1}^1 =& |\phi_{-1}(\pi/4)\rangle\langle\phi_{-1}(\pi/4)|.
\end{aligned}$$

$$\begin{aligned}
B_1^0 =& |\phi_1(\pi/8)\rangle\langle\phi_1(\pi/8)|, & B_{-1}^0 =& |\phi_{-1}(\pi/8)\rangle\langle\phi_{-1}(\pi/8)|, \\
B_1^1 =& |\phi_1(-\pi/8)\rangle\langle\phi_1(-\pi/8)|, & B_{-1}^1 =& |\phi_{-1}(-\pi/8)\rangle\langle\phi_{-1}(-\pi/8)|.
\end{aligned}$$

$$\begin{aligned}
P_{\text{win}} =& \frac{1}{4} \sum_{(a,b,x,y)\in\mathcal{W}} \operatorname{tr}(A_x^a \otimes B_y^b \rho_{AB}) \\
=& \frac{1}{8} \sum_{(0,0,x,y)\in\mathcal{W}} \left| \langle\phi_x(0)|0\rangle\langle\phi_y(\pi/8)|0\rangle + \langle\phi_x(0)|1\rangle\langle\phi_y(\pi/8)|1\rangle \right|^2 \\
&+ \frac{1}{8} \sum_{(0,1,x,y)\in\mathcal{W}} \left| \langle\phi_x(0)|0\rangle\langle\phi_y(-\pi/8)|0\rangle + \langle\phi_x(0)|1\rangle\langle\phi_y(-\pi/8)|1\rangle \right|^2 \\
&+ \frac{1}{8} \sum_{(1,0,x,y)\in\mathcal{W}} \left| \langle\phi_x(\phi/4)|0\rangle\langle\phi_y(\pi/8)|0\rangle + \langle\phi_x(\pi/4)|1\rangle\langle\phi_y(\phi/8)|1\rangle \right|^2 \\
&+ \frac{1}{8} \sum_{(1,1,x,y)\in\mathcal{W}} \left| \langle\phi_x(\pi/4)|0\rangle\langle\phi_y(-\pi/8)|0\rangle + \langle\phi_x(\pi/4)|1\rangle\langle\phi_y(-\pi/8)|1\rangle \right|^2 \\
=& \frac{1}{2}(1 + \frac{1}{\sqrt{2}}) \\
\approx& 0.85
\end{aligned}$$

Alain Aspect was the first person to implement this; he showed that it is not just theory. If you have entanglement and play this game, the data produced show that it cannot be local hidden variable theory. So quantum theory is complete in terms of describing this correlation – maybe not complete in terms of capturing gravity, and so on, but it is proven in this respect. It is as simple and beautiful as that.

Now I can return to our test. Winning this game with this magical probability that is better than 75 per cent is an indication that something quantum is happening; it is no longer classical. So that's the beginning of the test. As I told you, the experimenters gave us 4 qubits: so we implemented this Bell test blindly. We said if we have a box that some company has built and brought to us, we'll use our technique of verification adapted to the setting, so that we can measure the winning percentage that this company produces. If it's above a threshold, there may be something quantum; it's not proven that it's quantum theory and quantum entanglement, but it's proven that it's no longer classical. So this is the beginning: we have a test to tell us whether it's a classical magical machine that has secretly been built to simulate quantum theory or whether it's really a quantum machine.

To conclude, universal quantum computing is not going to happen anytime soon, but a lot of intermediate models – purpose-built models – are emerging. They are not as powerful as a quantum computer, but they are doing something that is not possible classically, such as trace estimations with the one-pure qubit model or the boson sampling with the linear optics model, or any "Quantum Simulators" in general: they may not do all Hamiltonian of quantum theory, but they do something else. What I would really like to see is whether we can adapt our quantum verification machinery to these settings. Hopefully, when we adapt it to these intermediate models, we won't need the 300 qubits that we cannot have anyway. Checking that the data coming out is correct is one thing, but the other question is whether this data really comes from *quantumness*. Bell inequality is only one aspect of quantumness: there are several others that I should mention, such as non-classical correlation, contextuality, dimensionality and superposition. So one day we hope to certify which aspect of quantumness has been demonstrated in the box.

Comments by Damian Markham[10] and questions from the audience

I'm not really experimental; I'm a link between what Elham is doing and the experimenters. So we weren't involved in these experiments that Elham was talking about, although there is a group here in Paris and one in Bristol who are doing similar experiments. My story of how we came to this is a bit different from that of Elham. But the question we are both working on, from the perspectives of both physics and computer sciences, is: "Is this box quantum or not?" As Elham said, this question is very broad, because on the one hand, from physics, this is a question about physics, because if this box is the universe, is the inside of this box quantum? Is the universe quantum? And from computer science, it has all these applications that Elham talked about, and the question of what else you can do with a quantum box compared with a classical box. In terms of these millions of euros that we can ask for in grants, it's also an engineering question: if I want to sell this thing, if I want to say I'm doing this thing, this is the right question. So my trajectory met that of Elham's in the same office where she met quantum computing, with our supervisor Vlatko Vedral. I was completely a physicist. I got interested in physics, like most people who study physics, by watching science fiction and wondering how the universe works. I was not interested in how a computer works; it was just a detail of what I liked. So out of the term "quantum computing", Elham and I each took the word that we were interested in. But through the years of arguing and fighting, we have both realized that these questions are intertwined. For me, before, computer science was just programming, not so interesting, and now of course, it's a much, much deeper field than that. I'm actually in the computer science section at the CNRS, so now I'm totally merged, as is the title of our field.

So finally, these questions of what makes a model of computation and how do we ask questions about how powerful a computer is, and so on – these merge into

[10] **Damian Markham** is a quantum physicist and computer scientist working for the CNRS in the theoretical computer science section based in the LTCI lab at Telecom ParisTech. Together with computer scientists and theoretical and experimental physicists in Telecom ParisTech, they have built an interdisciplinary group working on many aspects of quantum information, from theoretical conception to practical implementation. This approach has recently been boosted by jointly forming the Paris Centre for Quantum Computing with other computer science and physics departments around Paris.

the questions of physics, of the description of what's going on inside. The kind of questions that Elham presented: "is this box quantum?" or even "is this box correct?" in terms of what computational model you're talking about, merge with physics questions. So for a physicist, all of these questions split exactly into these ways, like the question of randomness in quantum mechanics that Elham talked about in the non-local game. Generally in physics, before quantum mechanics, we thought probabilities arose because we were essentially stupid and didn't know what was going on. This view gave rise to the EPR paradox. So if you think about statistical physics, you talk about a Boltzmann distribution of particles in a box. That is because you don't know the position and momentum of every particle in the box: if you did, you wouldn't have a probability distribution any more; you'd have the force on the wall that you could calculate. In quantum mechanics, on the other hand, these probabilities seem to be genuinely real probabilities. One interpretation of this Bell inequality is to say that locally, quantum mechanics is actually probabilistic. It's not because of our ignorance; it's just because that's the way nature is. So another way of understanding what's happening with quantum computation and quantum cryptography is that we are using these genuine probabilistic effects that come from nature. And in particular, these effects – these probabilities that you can share through quantum mechanics – get even weirder when you share them over distance. So these words – "contextuality", "dimensionality" and "superposition" – are all about this question: "What is a real description of what's going on?" From a physics point of view, we're lucky that all these questions are the same in a sense. They seem to be very different questions: "is the universe random or not?", and "how do I test if that's a quantum device or not?", but in the end they are closely related. So for me, the story of Elham's talk is the story of how these two things, these two trajectories meet.

Q: Cournot, who died in 1877, was profoundly, ontologically probabilistic, and that is also what the Cournot Centre is about. So this is, as the French say, "honey to our ears". Now, if I could kick off the questions, I think you could be extremely practical in telling us what Google is running after in terms of building its own quantum device?

Elham: If you are referring to their recent purchase of the D-Wave machine, we do not know if it is really quantum or not, and even if it is quantum, we don't know if it's giving us any speed-up… But for Google, buying a device for 20 million euros – well it's not a lot for them, so they bought it. They have a lab where they're doing machine learning and trying to use it for the optimization problems, and Google's idea I guess is to play with this device, among all the other devices they are working on, to see whether there is any advantage or not. But in the scientific community, there is huge controversy about this quantum device, because there are models based on quantum annealing, which is matching the statistics of the strings, and there are models based on classical simulated annealing, which matches the data. So the question is whether it is a classical model that is doing this or a quantum model. This is a 1000 qubits machine (they claim), but what they have shown is that the unit cells of 8 qubits are really quantum.

In my opinion, however, this is still exciting: there is a device coming from a commercial company that has been built (initially, there was going to be a lot of IP behind it, so they were not revealing any data about what's inside it, but they have changed their minds). It is very fascinating from an engineering point of view: superconducting qubits at almost zero temperatures – it's very impressive – but the question remains, is it quantum or is it classical? And if it is quantum, is it performing faster or not? Even if it's turned out to be not quantum, because of the private investment behind the company, they have brought the state-of-the-art to a level that the scientific labs would not be able to attain. So it seems that now there is some sort of emerging partnership with some scientists, seeking to understand what they have done, where it could be improved, where the transition between classical and quantum lies, and so on. And I think the idea of Google, or anybody else who can afford it, is to just buy it and play it: one way of exploring the new technology is to have the box and just plug and play. You can literally think about this as a gigantic black box in which you simply put the data in and get the data out. Given that this is an analogue machine rather than a digital one, and there are all sorts of different models, what is the field for evaluating and describing it?

Q: My first perception of the imminence of quantum computing was linked to Moore's law and the fact that under a nanometre, physics laws change…

Elham: I think it's a very technological issue. The better computers they build, the better the chips are inside. It's sort of the same battle: in one sense, we want to build a quantum effect and protect it, and the classical architect wants to make sure that the classical design is protected against the quantum. Something in-between happens: we want to make sure this quantum effect is coming and is protected, whereas the classical want to make sure the quantum effect is not coming – they want to get rid of it. I think there will be a crossing point; it's a matter of size, the machines are getting smaller and smaller. At present, we are at a point where parallel computing is becoming the hot topic. And this is true for the design: our school in Edinburgh has five floors; I'm on the fifth floor doing useless things, theoretical things, and the architects are on the first floor, and so we talk to these people. They know that there is no longer much space left in terms of clever miniaturization, so there has been a huge shift towards parallel computation. That's going to give you a bigger space for a long time, but in the parallelization, again there is the question of size. But we have already moved to the place where technology has driven the interest in parallel programming, parallel design, parallel over-the-internet, parallel web…, and everything is emerging there. Meanwhile, they are aware that quantum is also there…, but in order to really exist in that space, we need a quantum computer to be built. So if you ask people to invest in parallel computation because it will solve our problems for the next 10 or 20 years, that can be built; but whether quantum power will replace parallel power…, that is difficult to say.

Q: Why should quantum be faster, conceptually?

Elham: Because of superposition, entanglement, all of the things that don't exist in classical computing…. A very bad question is the following: do you believe in a parallel universe? Well, we do believe in it! If you run a computation, there's no doubt that the parallel computation is faster than the classical one. That's a bad answer, but it's the beginning of the explanation. So if you have a question and give it to a million computers to run it, it's faster because of the parallel power of the computation. Quantum computing seems to have a bit of that – because we're living in the parallel universe: you bring every measurement in the distribution into the parallel universe, and this computation is done under massive parallelism, and then somehow you can bring this universe back together with the classical world. So it's as if one quantum was processing 1000 quantums – it's not really the case, but it's as if it were.

Damian: Actually I think formally you can say that for your first example. If you write down this algorithm, you're really doing this black box check on each of the inputs; your input is a superposition of each of these inputs. And then at the end you just do the measurement to check that they're the same.

Q: My understanding was that given that each qubit is in a superposition of states, if you have n qubits, with 2 states each, then you get 2^n states, and you have an exponential power of representation. So with 10 qubits you can do what you would with 1000 classical bits.

Elham: Well, the world is not as perfect as you would expect; you cannot do everything you want to do with these qubits: they are sitting in their quantum world, which is not accessible to us. All this magical computation is done, but the moment you measure, you get only one of these strings back. So you have the 1000 strings, but you're only measuring one of them. But if you cleverly correlate – and that's where the amplification and cancellation (due to negative complex number) comes in – and correctly combine those strings together, it is possible for a specific problem to get that speed-up. A specific problem is the property of those values; the relation between them might be evaluated quantumly faster than classically it is done.

Q: What is the most complex computation that has been done to date, experimentally?

Damian: I believe factoring 21 is the best that's been done.

Elham: More impressive is quantum simulation. With factoring, the story is the usual thing: if you want to do something useful, factoring is a practical problem, but because of lots of other elements, you're losing part of it. But if you just say: "for the time being, I'm not interested in giving you something sensible, some useful application, I just want to prove a principle by showing you some real quantum application" – then I think the state of the art is about 10 qubits. So you can get the 10 qubits successfully, the full tomography of part of it, and do some particular Hamiltonian simulation. But then, is it doing anything interesting? Well we are getting there, so there is a little bit of the two sides of the story. The things that the physicists are pushing – and I think that if the physicists promise that in five years' time there will be 50 or 100 qubits, that's not unrealistic, because there is no barrier – are that once you do the 10 qubits, it is just a matter of money to take you to 50 or 100. Maybe taking it to 1000 would require better performance, but the question of going from 10 to 100 is a matter of the coherence time, better detection, and so on. It's not that easy, but there are ideas, like putting units of 10 and 10 and 10 that we plan to do in one of the UK quantum technology hubs led by Oxford. You connect them and can improve things, but there are things that we can do in this intermediate role that don't need the full connectivity, the full coherence of factoring 10-digit numbers, for example. Let's put it this way: showing something that cannot be done classically – we have already seen it, but we are going to see more and more of it. The performance and experimental demonstration of computation that cannot be performed classically: we are already in that domain. Whether that computation is a useful one – that's a different story. But we have gone through this basic "here it is, something that your best classical simulator cannot simulate" – we are there.

Damian: These examples of the game, interpreting Bell's inequality as a game: even if you can't do a big computer with 10 qubits, you can still play all sorts of different games. So if you're thinking about communications over a network, there are lots of things that aren't computation, but are very powerful, like the "leader election" protocol and things like this that we are working on at Telecom ParisTech.

Elham: There is a reason why the UK lobbying of the government succeeded in bringing this huge amount of money – it's not completely out of the blue. As the economists know, no matter how wonderfully you are lobbying, at the bottom line, it was a bunch of scientists who brought their case to the politicians. There were a lot of case studies, built up on lots of things, done in the UK and on the other side of the Channel and wherever: it convinced them that now is the time.

Q: Are there any patents?

Elham: Yes, a lot of patents! My guess is that the real capital return on D-Wave will be from their patents. It's not going to be through the sale of quantum computers. I have not seen their business plan, of course, but they have patented so many engineering things…. The very first quantum company was called Quantum Magic; I think it was in New York… almost twenty years ago, way before anything concrete had been done, and their business plan was based on helping scientists to get patents. We were just writing papers and putting out all this stuff that clearly, no one believed, but this company had a vision. Well they did build an encrypted key distribution as well, but their business plan really was to form some kind of partnership with the scientists and make some money out of it, being the bridge for the patents.

Q: And what aspects of quantum can you patent?

Elham: I've just done a patent for fun perhaps, so I can answer that! Initially we had no idea how we were going to do it, and we figured out security is probably the only place where there is something to sell in quantum, so that is our business strategy direction. The way we did it at the University of Edinburgh is that we patented a prototype. The bottom line is a protocol: the BB84 protocol for the key distribution is already patented, but there is a variation of this protocol that we have patented, if it's accepted. It's a prototype for a particular purpose, for running a computation securely on some sort of framework device, in simple terms. You can't just patent an algorithm; you need to patent the machine. But we're not experimentalists, so we drew figures, envisioned the way that it could be built, and then we worked with the lawyers. We needed to envision that it could be with the atom, with the photon, that it could be this prototype – our circuitry. So we drew 20 different things. It was funny the difference between writing a paper and writing a patent: anything useless that you would not put in a paper, you put in a patent, all the different possibilities. There's nothing quantum about the patent; it's exactly the classical story, the prototype. Then we had to bring our case to a company in Edinburgh that was hired by the University of Edinburgh, and which looks at the potential; they do not agree that any random idea that comes into your mind is worth the investment of hiring a lawyer and paying £10,000 to patent it, so you bring your case to them, for free. There's a sort of hierarchy, so you present your case to a business representative, and he/she presents it to this commercial company, which will say whether there seems to be a potential market. Then the university gives you money to hire the lawyer to do the patent, and so on. It's fun; we'll see where it goes, but that's the process.

Q: Mathematically, what is the difference between the classical machine and the quantum machine?

Elham: Mathematically, the only thing that makes quantum different from classical is that the numbers are complex. If I have a vector that has this complex number, the square of that complex number is the probability. So I can dig the probability out of this complex number, and then it becomes probabilistic evolution. But when the strings go in the parallel universe, they're still complex numbers, and they are negative and positive. For example $-\alpha$ and α cancel each other out, whereas you can't have negative probabilities. So in the terminology of physics, there is this interference (OK, well in classic waves there is also interference), this cancellation, which is the quantumness of the computing.

Q: Can you prove whether or not a computer or box is quantum using a purely deterministic process, or do you always have to have that randomness?

Elham: I think this is something that will be established one way or another, and it's a very interesting question: to test quantumness do you need a bit of quantumness? But the quantumness you use for testing is sort of "deterministic" quantumness. I give you this Bell pair (a quantum particle); there's no randomness there. I give it to you deterministically, and then I can test it. But even in that protocol, you need the classical probability to hide it – so no, even if I quantumly give this resource, which is producing randomness later, you still need to use the classical randomness within your protocol…. But that's BPP versus P, so there is still randomness. You can't get rid of randomness; we can't get rid of the Cournot Centre yet!