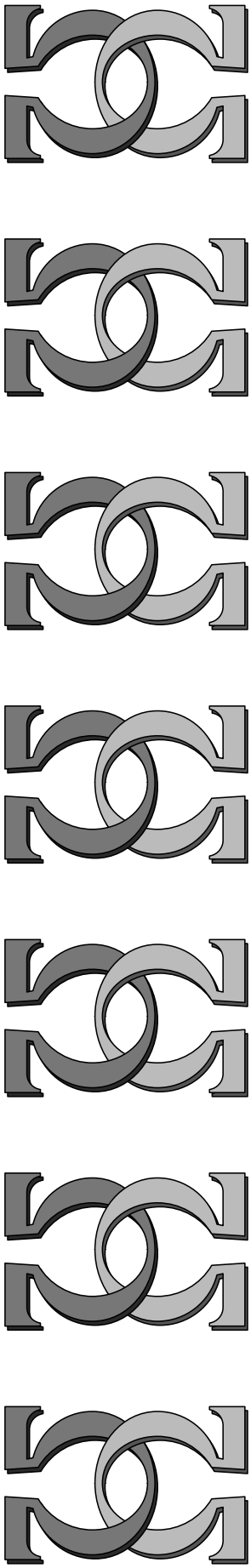

CDMTCS
Research
Report
Series

Dialogues on Quantum Computing

C. S. Calude
University of Auckland
Auckland, New Zealand

CDMTCS-219
June 2003

Centre for Discrete Mathematics and
Theoretical Computer Science

Dialogues on Quantum Computing*

Cristian S. Calude
Department of Computer Science
University of Auckland, New Zealand
Email: cristian@cs.auckland.ac.nz

What sort of machines do useful computation in a universe described by classical mechanics? The answer was provided in 1936 by the British mathematician Alan Turing, and it's known today as the *Turing machine*. But even in 1936 classical mechanics was known to be false, and so one could have asked the question: What sort of machines do useful computation in a universe described by quantum mechanics? In a trivial sense, everything is a quantum computer. A pebble is a quantum computer for calculating the constant-position function; current computers exploit quantum effects (like electrons tunneling through barriers) to control computation and to be able to run fast. But quantum computing is much more than that. In what follows we will present – in the form of a biased, informal dialogue – a few key-ideas on quantum computing,

Q: What has computing to do with physics?

A: Information, essential for any form of computing, is not a pure abstract entity. In fact, measuring, communicating and computing are *all* about exchanging information. Information is inevitably tied to a physical embodiment or representation; it can be engraved on stone tablets, represented by holes punched in a card, or by a present/absent charge or by a spin up or down. "The computer has made us aware", said Rudolf Landauer, "that information is a physical entity". And, according to Oxford physicist David Deutsch, "The reason why we find it possible to construct, say, electronic calculators, and indeed why we can perform mental arithmetic, cannot be found in mathematics or logic. *The reason is that the laws of physics "happen" to permit the existence of physical models for the operations of arithmetic* such as addition, subtraction and multiplication. If they did not, these familiar operations would be non-computable functions. We might still know *of* them and invoke them in mathematical proofs (which would be presumably called "non-constructive") but we could not perform them."

Q: Does it mean that any computation is controlled by some "physical reality"?

*To appear in C. Martin-Vide, V. Mitrana and G. Păun (eds.). *Formal Languages and Applications* to be published by Physica-Verlag, Heidelberg.

A: Yes: there can be no computation without physical support.

Q: What is the "physical reality" permitting my desktop computer to work? Does it impose any limits on computation?

A: Classical mechanics. Shannon's information theory (1948) shows the limits of information handling; Landauer's principle tells us that "the erasure of information is a dissipative process".

Q: Is there a difference between the case when my laptop works on (a) my desk, (b) on a satellite revolving rapidly around the Earth?

A: For all practical purposes, those computers will act almost identically. But as they are located in different relativistic frames, they experience relativistic effects, such as time dilatation, with respect to each other; these effects (may) make the difference. Relativistic computing deals exactly with these problems.

Q: What is quantum computing?

A: Quantum computing is the quest to understand what sort of machines do useful computation in a universe described by quantum mechanics. Today the subject is mostly theoretical, but tentatively, slowly and hesitantly groping towards some practical applications.

Q: What is quantum mechanics?

A: Quantum mechanics tries to describe the behaviour of very small objects, the size of atoms or smaller, in contrast with relativity theory which describes the laws of larger everyday objects. Interestingly, particles do not behave in the same way as larger everyday objects, such as billiard balls. If we strike a billiard ball in a very precise way and we know its exact initial position, then we can predict with (theoretical) certainty where it will go. The same is not true for particles.

Q: Why?

A: Here is a classical illustration. Imagine a brick wall with two holes in it, each the same size and large enough to fire bullets through; then put a second wall behind where the bullets will strike. After firing a few rounds you will see two clusters of hits in line with the two holes. This is what is going to happen if you consider that light photons travel as particles.

Now imagine that light travels as a wave, and think of it as a water tank. As the wave spreads out from its source it would reach both holes at the same time. Holes would act as new sources and waves would then spread out again (in step or in phase), and, moving forward, they would eventually interfere with one another. Two types of interference are possible: constructive and destructive. If both waves are lifting the water surface upward, we get a more pronounced crest, a constructive interference; if one wave is trying to create a crest but the other is trying to create a trough, the two cancel out

and the water level is undisturbed, a destructive interference.

Q: What happens if we try to do the above experiment with light?

A: If we carried out this procedure with light instead of water, then (a) if light travels as waves, then the pattern on the second wall would appear as an interference pattern of alternate dark and light bands across the wall, but (b) if light travels as particles, then it would produce two separate areas of light. This experiment has in fact been carried out many times always with the same results.

Q: This is rather confusing, isn't it?

A: According to the American physicist John Wheeler, "The quantum is the greatest mystery we've got. Never in my life was I more up a tree than today." And if this is not enough, let's cite the Nobel laureate Richard Feynman, who said that "nobody understands quantum mechanics" ...

Quantum mechanics, considered the deepest theory of physics, is used every day; it may not offer the ultimate explanation of how the universe works, but quantum mechanics just works. In Born's words, "The theory produces a great deal but hardly brings us closer to the secrets of the old one". Physicists are almost unanimous in their conviction that quantum weirdness needs no apologies. Citing Feynman again, "we have to accept Nature as she is, fundamentally absurd".

Q: If classical mechanics is wrong, why do we still use it?

A: Classical mechanics is flawed *only* when dealing with the very small (atomic size) or the very fast (near the speed of light). For everyday things, classical physics does an excellent job.

Q: What are the main features of quantum mechanics?

A: Here are four:

- *Quantisation*: observable quantities do not vary continuously, but come in discrete chunks called quanta.
- *Randomness*: physical reality is irreducibly random.
- *Interference*: the outcome of a quantum process depends on all the possible histories of that process.
- *Superposition*: the ability of carrying out computations with "blends" of states, superpositions.
- *Entanglement*: two spatially separated and non-interacting quantum systems, that have interacted in the past could have some locally inaccessible information in common – information which cannot be accessed in any experiment performed on either of them alone.

Q: Are these features useful for quantum computing?

A: Quantisation makes quantum computing possible at all. Randomness, superposition and interference make quantum computers more powerful than classical ones. Entanglement is useful in quantum cryptography.

Q: Computers are constantly becoming smaller, faster, cheaper and more potent. Why should we be concerned with quantum computing?

A: The classical computer is indeed the only commodity ever to become exponentially better as it gets cheaper. Its information handling capacity has grown at a rate ten million times faster than the information handling capacity of our nervous systems during the 4 billion years since life began on Earth. Yet, this exponential race will not guarantee solutions to the many intractable/undecidable problems challenging computer science. Even worse, it has been predicted that this trend of conventional technology will hit the wall in less than 10 to 15 years because information is carried by discrete quanta. Conventional computing is approaching a critical phase, where new technologies will be required to provide significant progress.

Q: Can you give a simple example of a problem and a quantum-like solution?

A: A famous example is the Merchant's Problem: "A merchant learns than one of his five stacks of $\Gamma = 1$ gram coins contains only false coins, $\gamma = 0.001$ grams heavier than normal ones. Can he find the odd stack by a single "weighing"?

The well-known solution of this problem is the following: we take one coin from the first stack, two coins from the second stack, ..., five coins from the last stack. Then by weighing the combination of coins described above we obtain the number $Q = 15 + \gamma \times n$ grams, which tells us that the n-th stack contains false coins.

Q: In contrast with classical computers which use bits, quantum computers process quantum bits. What's the difference?

A: To represent a classical bit you need a system described by one or more continuous parameters. For example, the position of gear teeth in Babbage's difference engine or the voltage between the plates in a capacitor represent a bit of information. In the last example, a charged capacitor denotes 1 and an uncharged capacitor 0. The parameter is used to separate the space into two well-defined regions chosen to represent 0 and 1. Manufacturing imperfections and local perturbations may affect the signal, so signals are periodically restored near these regions to prevent them from drifting away. The American physicist David Mermin has argued for the distinction between the abstract bit (0 or 1) and a classical system representing a bit, which could be called Cbit. The term Qbit would denote the quantum generalisation of the Cbit, that is a quantum system (event) in which we have two possible mutually exclusive outcomes.

All knowledge of the quantum system is based upon acts of observation. The information derived from an elementary act of observation is no more than a single bit, but *there is*

more to it than that. To mark this difference, the physicist Bill Schumaker has coined the name quantum bit, qubit or better still Qbit. Examples of Qbits are: an atom, a nuclear spin or a polarised photon. For example, the state of a spin-$\frac{1}{2}$ particle, when measured, is always found to be in one of two possible states, represented as

$$|+\frac{1}{2}\rangle \text{ (spin-up) or } |-\frac{1}{2}\rangle \text{ (spin-down).}$$

Unlike the intermediate states of a Cbit (for example, any voltages between the "standard" representations of 0 and 1) which can be distinguished from 0 and 1, but do not exist from an informational point of view, quantum intermediate states cannot be reliably distinguished, even in principle, from the basis states, but do have an informational "existence". Before measurement, the system can be in any intermediate quantum state, that is in a superposition of 0 and 1, in a (sort of) mixture of 0 and 1 containing both classical (contradicting) states at once; after observation, we get either 0 or 1 with some probability. So, an observation is simultaneously like a coin-toss and not like a coin-toss.

For example, a Cbit can be realised using two different polarisations of light or two different electronic states of an atom. However, if we choose a polarised photon as a Cbit, then quantum mechanics tells us that apart from the two distinct states the photon can be also prepared in a coherent superposition of the two states, that is in both state 0 and state 1.

Q: Are Qbits responsible for the famous "exponential explosion"?

A: Yes. Any classical register composed of three Cbits can store in a given moment of time only one out of eight different numbers because the register can be in only *one* out of eight possible configurations: 000, 001, 010, 011, 100, 101, 111. A quantum register composed of three Qbits can store in a given moment of time *all* eight numbers in a quantum superposition. If we increase the number of Qbits to the register, then we increase its storage capacity exponentially: three Qbits can store eight different numbers at once, four Qbits can store sixteen different numbers at once, in general n Qbits can store 2^n numbers at once.

Q: What can you do with superpositions?

A: We can perform operations on them. During such an operation each number in the superposition is affected and as the result we obtain a massive parallel computation albeit in just one piece of quantum hardware. As in the solution of the Merchant's Problem, we can act at once on all stacks of coins. A quantum computer offers an enormous gain in time and memory.

Q: Where is the catch?

A: Qbits suffer from a major limitation which doesn't affect Cbits: given a superposition of Qbits in some state, there is nothing one can do to the Qbits to be able to extract what that state is in.

Q: Is this the only limitation?

A: No. There are also limited possibilities to extract the information contained in a Qbit. Learning the value of a combination of Cbits is so easy (you print it out) that it is not even explicitly regarded as a part of the computation. More importantly, Cbits are not altered by "reading" them. Not anymore with Qbits: we can extract the information from a Qbit *only* by measurement, a process which: (a) is probabilistic (recall the intrinsic randomness of quantum mechanics), and (b) affects the state of the Qbit. As a consequence, simple operations taken for granted in classical computing, like copying a Cbit into another Cbit, are simply not available in quantum computing.

Q: So, what are Qbits good for?

A: The art is to produce a superposition in which the useful information has a high probability of being indicated by measurement and the unimportant information can be expected to appear with probability close to zero. To make the result safe, one has to be able to easily *confirm* the result of the computation ...

Q: Can you give an example?

A: Peter Shor, a mathematician from Bell Labs, has shown in 1994 that quantum factoring integers is dramatically faster than any *known* classical algorithm. The obvious method of factoring N into primes is to attempt to divide N by all numbers from 2 to the square root of N; if N has n bits, then we need to go through about $2^{n/2}$ trials. A much smarter algorithm (based on sophisticated mathematical results) does the job in approximately $2^{c\sqrt[3]{n}}$ steps, where c is a constant; still, factoring a number of a million of bits would require a time larger than the age of the Universe.

Shor has observed that the factoring problem can be rephrased in terms of a search for how often some "period" of a finite sequence is repeating itself within the sequence. For example, the sequence 123412341234 has 1234 as period which repeats itself three times. Periods may be seen as waves, undulating streams. Shor's idea was to analyse periods in such a "number wave": factors will come out naturally. Even under this reduction, the computation involves an enormous number of steps: the beauty is that most of these steps can be performed simultaneously via an appropriate quantum superposition. The quantum algorithm is polynomial-time in the number of bits necessary to represent the number to be factored. Confirming the result is easy.

Q: Is the notion of "quantum polynomial-time" algorithm the same as the classical notion of polynomial-time algorithm?

A: No, they are different as they involve machines with different hardware in which the time is computed in different ways.

Q: Can we say that Shor's algorithm is faster than any classical algorithm?

A: No, for a number of reasons. First and foremost, the problem whether there is a

classical polynomial-time algorithm for factoring is open – today nobody knows the answer. Recently, an easier, but closely-related problem, the primality problem, has been shown to be classically polynomial-time computable[1] – a piece of news reported in the *New York Times* on 13 August 2002. Secondly, Shor's algorithm is probabilistic, not deterministic, as any quantum algorithm.

Q: What about Grover's quantum algorithm?

A: Start with an example. Searching a telephone directory containing n names in alphabetic ordering[2] requires about $\log_2 n$ steps. Searching the name in the telephone directory, when the telephone number is known, is much more difficult because the list is unsorted with respect to telephone numbers. We need about $n/2$ steps on average and n steps in the worst case. Looking up a name given a number is exponentially more difficult than looking up a number given a name. Grover's quantum algorithm searches an unsorted list very fast; it uses an equally distributed superposition of all possible indices of the entries in the telephone directory containing the target index, then creates a special "amplitude amplification" operator capable of boosting the target index. This procedure needs roughly $\pi/4\sqrt{n}$ quantum steps.

Q: So Grover's quantum algorithm is faster than any classical algorithm searching an unsorted list ...

A: Yes.

Q: Does this mean that it is provably faster than any other algorithm?

A: No. It might be possible to design a "natural computing" algorithm running as fast as Grover's quantum algorithm. In fact, there is one, a modification of the bead-sort algorithm.

Q: What will quantum computers be good at?

A: These are the most important applications currently known:

- *Cryptography*: RSA code breaking, perfectly secure communication.
- *Searching*: fast searching (Grover's algorithm).
- *Simulating*: efficient simulation of quantum-mechanical systems.

Q: Can I learn quantum computing without understanding quantum mechanics?

A: Yes, you can. Recently, Lance Fortnow, a mathematician at NEC Institute, has published a nice paper titled "One complexity theorist's view of quantum computing" in which he shows that a large part of quantum computing can be understood without

[1] The authors are M. Agrawal and his students N. Kayal and N. Saxena from the Indian Institute of Technology.

[2] Not for Chinese directories.

any knowledge of quantum mechanics. David Mermin, already cited in this article, has argued that the amount of quantum mechanics required for the mainstream quantum computing is limited and can be taught in four lectures. This parallels the situation of classical computing, where computer scientists need not know much about transistors and the way they work. Of course, knowing quantum mechanics doesn't hurt, on the contrary ...

Q: Is randomness a "bad feature" of quantum computing?

A: Yes and no. Randomness limits the computer capability of producing 100% true results, hence, of course, it's a bad feature. But this is only half of the story. It is well known that classical computing cannot produce "true random bits". The American mathematician John von Neumann once said that "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." What about quantum computing? Randomness of quantum mechanics allows an affirmative answer. Take, for example, a quantum coin-toss experiment, in which the spins of single electrons $\#1, 2, 3, \ldots$ are prepared in a particular polarization direction. Suppose that their spin state is measured along a perpendicular direction. If the spin state is "up", we associate the code 0 and if the spin state is "down" we associate the code 1. Assume that the experiment is lossless, that is no electron passes the device without being detected in either the "up" or "down" detector. By the postulates of quantum mechanics, the resulting sequence is purely random. Any classical computer supplied with a source of random bits is a provably more powerful machine. The problem whether there are Monte Carlo algorithms – fast algorithms (using truly random bits) which are probably correct – that are faster than any deterministic algorithm for the same problem is still *open*.[3]

Q: Is randomness just an academic issue?

A: Certainly not. Randomness is essential for the security of a nation's secrets, and much more. For example, Soviet codes have been broken during the World War II because the one-time pad cryptosystem (the Vernam cipher) used for diplomatic communication re-used random keys.

Q: You have convinced me. Can I buy quantum random numbers?

A: Yes, for example from GAP-Optique, a company associated with the University of Geneva, Switzerland, `http://www.gapoptic.unige.ch/Prototypes/QRNG/.` See `http://www.cs.berkeley.edu/ daw/rnd/index.html` for other sources.

Q: We discussed physical limitations placed on various computing devices. What about mathematical limitations?

A: The *Church-Turing Thesis*, a prevailing paradigm in classical computing, states that no realizable computing device can be "globally" more powerful, that is, aside from

[3]'Las Vegas algorithms are always correct and probably fast.

relative speedups, than a universal Turing machine. The original intent of Church and Turing was to relate computations performed with paper and pencil by a human clerk (a 'computor') and Turing machine computations, hence the form: "What is humanly computable is computable by a universal Turing machine". The modern form of the Church-Turing Thesis states that "any 'reasonable' model of computation can be effectively simulated by a (probabilistic) Turing machine." There is also a physical variant discussed by David Deutsch, sometimes referred to as the *Church-Turing Principle*, stating that "every effectively realisable system can be defined by a Turing machine", or a bit looser, "it is possible to perfectly simulate the physical reality on a Turing machine". If the Church-Turing principle is true, then it's a fundamental statement about our Universe; if it's false, then a bomb that explodes in cyberspace may be in reality a dud.

Q: Are the Church-Turing Thesis or the Church-Turing Principle provable?

A: They are both not provable; the only possibility might be to disprove them. For example, by showing that one can build a realisable system which cannot be simulated by any Turing machine.

Q: Do you believe in the Church-Turing Thesis? But in the Church-Turing Principle?

A: I believe in the Church-Turing Thesis in its original intent, but I don't believe in the Church-Turing Principle and I have tried to disprove it ...

Q: Are there any connections between automata theory and quantum computing?

A: There are. First, automata theory has been used in modelling various quantum effects, like complementarity or the famous Einstein, Podolsky and Rosen (EPR) effect (used in quantum teleportation). Secondly, "quantum automata" are objects of increasing theoretical and practical interest.

Q: How soon a quantum computer might be built?

A: Lab experiments show that the basic principles of quantum computing are sound. To realistically compete with classical computing, quantum computing must be carried out on significantly larger scales ... It is unreasonable to make predictions; however, it is reasonable to expect that small milestones will continue to appear.

Q: A quantum computer would be "a spy master's dream", isn't it?

A: Certainly new technologies, quantum computing (possibly) among them, tend to empower some groups relative to others. With a quantum computer, any hacker (working for Osama Bin Laden) could crack all codes that now protect digital data, could scan Pentagon databases and could steal funds from banks to pay for the attacks. Of course, the Pentagon and the banks would take all possible countermeasures; the CIA might use quantum computers to break al-Qaida's codes as well ...

Q: Sure, quantum computing is not the only unconventional type of computing. What

are other approaches?

A: Quantum computing is just one unconventional paradigm. DNA computing is another one; membrane computing is close but still different. You will find more about these topics in this book.

Q: Do you recommend any papers or books on quantum computing?

A: Here are some titles (but not a complete list):

1. Books on quantum computing:

   - M. Brooks (ed.). *Quantum Computing and Communications*, Springer-Verlag, Berlin, 1999.
   - J. Gruska. *Quantum Computing,* McGraw-Hill, London, 1999.
   - M. Hirvensalo. *Quantum Computing*, Springer-Verlag, Berlin, 2001.
   - A. Yu. Kitaev, A. H. Shen, M. N. Vylalyi. *Classical and Quantum Computation*, American Mathematical Society, Providence, Rhode-Island, 2002.
   - M. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
   - C. P. Williams, S. H. Clearwater. *Explorations in Quantum Computing,* Springer-Verlag, New York, 1997.
   - C. P. Williams, S. H. Clearwater. *Ultimate Zero and One: Computing at the Quantum Frontier,* Springer-Verlag, Heidelberg, 2000.

2. Books containing chapters on quantum computing:

   - C. S. Calude, G. Păun. *Computing with Cells and Atoms*, Taylor & Francis Publishers, London, 2001.
   - J. G. Hey and R. W. Allen, (eds.). *Feynman Lectures on Computation,* Addison-Wesley, Reading, Massachusetts, 1996.
   - J. G. Hey (ed.). *Feynman and Computation. Exploring the Limits of Computers,* Perseus Books, Reading, Massachusetts, 1999.
   - K. Svozil. *Randomness & Undecidability in Physics*, World Scientific, Singapore, 1993.

3. Non-technical books on classical and quantum computing:

   - J. Barrow. *Impossibility – The Limits of Science and the Science of Limits*, Oxford University Press, Oxford, 1998.
   - D. Deutsch. *The Fabric of Reality*, Allen Lane, Penguin Press, 1997.
   - G. Johnson. *A Shortcut Through Time: The Path to a Quantum Computer*, Alfred A. Knopf, New York, 2003.

- G. Milburn. *The Feynman Processor. An Introduction to Quantum Computation*, Allen & Unwin, St. Leonards, 1998.
- T. Siegfried. *The Bit and the Pendulum: How the New Physics of Information is Revolutionizing Science*, John Wiley & Sons, New York, 1999.

4. Influential papers in quantum computing:

   - A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margoluous, P. W. Schnor, T. Sleator, J. A. Smolin, H. Weinfurter. Elementary gates of quantum computation, *Physical Review*, A 52 (1995), 3457 – 3467.
   - C. H. Bennett. The thermodynamics of computation, *International Journal of Theoretical Physics*, 21 (1982), 905 – 940.
   - C. H. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, *Phys. Rev. Lett.* 70 (1993), 1895 – 1898.
   - D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer, *Proceedings of the Royal Society of London* A 400 (1985), 97 – 119.
   - D. Deutsch, Quantum computation, *Physics World* 5 (1992), 57 – 61.
   - L. K. Grover. A fast quantum mechanical algorithm for database search, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, 1996, 212 – 219.
   - P. W. Shor. Algorithms for quantum computation: discrete log and factoring, *Proceedings of the 35th IEEE Annual Symposium on Foundations of Computer Science*, 1994, 124 – 134.

5. Papers on automata and quantum computing:

   - D. Finkelstein, S. R. Finkelstein, Computational complementarity, *International Journal of Theoretical Physics*, 22, 8 (1983), 753 – 779.
   - C. S. Calude, E. Calude, K. Svozil. Quantum correlations conundrum: an automaton-theoretic approach, in C. Martin-Vide, Gh. Păun (eds.) *Recent Topics in Mathematical and Computational Linguistics*, The Publishing House of the Romanian Academy, Bucharest, 2000, 55 – 67.
   - C. S. Calude, Elena Calude, K. Svozil. Computational complementarity for probabilistic automata, in C. Martin-Vide, V. Mitrana (eds.). *Where Mathematics, Computer Science, Linguistics and Biology Meet*, Kluwer, Amsterdam, 2001, 99 – 113.
   - C. S. Calude, E. Calude, K. Svozil, S. Yu. Physical versus computational complementarity I, *International Journal of Theoretical Physics*, 36, 7 (1997), 1495 – 1523.

6. Papers challenging the Church-Turing Principle:

   - V. A. Adamyan, C. S. Calude, B. S. Pavlov. Transcending the Limits of Turing Computability, Los Alamos preprint archive, `http://quant-ph/0304128`, 16 April 2003.
   - C. S. Calude, M. J. Dinneen, K. Svozil. Reflections on quantum computing, *Complexity*, 6, 1 (2000), 35 – 37.
   - C. S. Calude, B. Pavlov. Coins, quantum measurements, and Turing's barrier, *Quantum Information Processing* 1, 1 – 2 (2002), 107 – 127.
   - G. Etesi, I. Németi. Non-Turing computations via Malament-Hogarth spacetimes, *International Journal of Theoretical Physics* 41 (2002), 341 – 370.
   - T. D. Kieu. Quantum hypercomputation, *Minds and Machines: Journal for Artificial Intelligence, Philosophy and Cognitive Science*, 12, 4 (2002), 541 – 561.
   - T. D. Kieu. Computing the noncomputable, Los Alamos preprint archive `http://arXiv:quant-ph/0203034`, v1, 7 March 2002.
   - H. T. Siegelmann. Computation beyond the Turing limit, *Science*, 268 (April 1995), 545 – 548.
   - K. Svozil. The Church-Turing Thesis as a guiding principle for physics, in C. S. Calude, J. Casti, M. J. Dinneen (eds.). *Unconventional Models of Computation*, Springer Verlag, Singapore, 1998, 371 – 385.
   - K. Svozil. Computational universes. *Chaos, Solitons & Fractals*, to appear.

Q: What about web references?

A:There are many. Here are some important sites:

- Quantum Computing at IBM Research Yorktown, `http://www.research.ibm.com/quantuminfo/`.
- Oxford Centre for Quantum Computation, `http://www.qubit.org/`.
- John Preskill Course (Physics of Computation), `http://www.theory.caltech.edu/people/preskill/ph229/`.
- The Home of the Home Pages Page (in Quantum Computing), `http://www.cs.berkeley.edu/~vandam/homes.html`.

Q: I noticed that quite a few answers were "yes and no" ...

A: Yes.

Q: Anything else?

A: I would like to thank Carlos Martin-Vide, Victor Mitrana and Gheorghe Păun for inviting me to contribute to this volume, Elena Calude, Pulkit Grover, Karl Svozil Garry Tee for comments and various suggestions; the quantum coin-toss experiment was described by Karl.