7th IEEE Conference on Structure in Complexity Theory, June 1992

The Quantum Challenge to Structural Complexity Theory

André Berthiaume * and Gilles Brassard †
Département IRO
Université de Montréal.
C.P. 6128, Succursale "A"
Montréal (Québec) Canada H3C 3J7

Abstract

This is a non-technical survey paper of recent quantum-mechanical discoveries that challenge generally accepted complexity-theoretic versions of the Church–Turing thesis. In particular, building on pioneering work of David Deutsch and Richard Jozsa, we construct an oracle relative to which there exists a set that can be recognized in Quantum Polynomial Time (**QP**), *yet any Turing machine that recognizes it would require exponential time even if allowed to be probabilistic, provided that errors are not tolerated. In particular,* $\mathbf{QP} \not\subseteq \mathbf{ZPP}$ *relative to this oracle. Furthermore, there are cryptographic tasks that are demonstrably impossible to implement with unlimited computing power probabilistic interactive Turing machines, yet they can be implemented even in practice by quantum mechanical apparatus.*

1 Deutsch's Quantum Computer

In a bold paper published in the *Proceedings of the Royal Society*, David Deutsch put forth in 1985 the *quantum computer* [7] (see also [8]). Even though this may come one day to be recognized as one of the most important papers of that decade in theoretical computer science, it was written primarily for the benefit of physicists, in a language somewhat foreign to computer scientists. It is not our purpose here to describe the principles of quantum computing; rather, we wish to explore some of its implications on structural complexity theory. Nevertheless, a short review of quantum computing is in order.

What makes the quantum computer so different from a Turing machine is the possibility it offers for massive parallelism within a single piece of hardware. Let f be a computable function. If you have to compute it on two different inputs x and y, you may have to compute $f(x)$ and then start afresh to compute $f(y)$, thus requiring about twice as long as if you had needed only one of these values. The quantum computer allows you to prepare an input that encodes both x and y in a so-called *quantum superposition.* If you run your program that computes f on that input, it produces the quantum superposition of $f(x)$ and $f(y)$ in the time needed to compute the function once only. Even better, you can prepare in linear time the quantum superposition of an *exponential* number of inputs, after which a single call to the program computes the quantum superposition of all the corresponding outputs! If this sounds too good to be true, *it is* in a sense. The catch is that even though you have produced exponentially many outputs with a single call to the program, quantum mechanics severely restricts the way in which you can look at the result. Moreover, any measurement of the output state disturbs it by virtue of Heisenberg's uncertainty principle, making further measurements less informative or even completely useless.

The beauty of quantum computing is that some functions of these outputs (such as the exclusive-or of any two of them) are *computable by quantum parallelism.* This means that there exists a measurement so that the correct result is obtained with nonzero probability when the output state is subjected to it; otherwise, the measurement yields "bad luck". Thus, such a measurement acts as a Las Vegas algorithms: it never gives a wrong answer, but sometimes it fails to answer at all. Regardless of success, the output state is irreversibly randomized by the measurement process, so that it cannot undergo additional measurements. Therefore, the entire effort was in vain when the answer is "bad luck". Nevertheless, this outcome is given probabilistically according to the laws of quantum mechanics, and it may well not reoccur if

* Supported in part by an NSERC postgraduate fellowship.
† Supported in part by the E. W. R. Steacie Memorial Fellowship (NSERC).

you start the whole process all over again. Unfortunately, the probability of success is never better than 2^{-n} in the worst case[1] when there are 2^n outputs in equally weighted superposition.[2] The simplest example of such behaviour is useless but instructive: any single one of the 2^n outputs can be obtained with probability 2^{-n} of success.[3]

But hang on! Deutsch gives a situation in which quantum parallelism is definitely interesting [7]. Consider the case of only two inputs: 0 and 1. Let us say that $f(0) \oplus f(1)$ represents a crucial decision that must be made within 24 hours, such as buying or selling on the stock market, but that computing f takes almost one day on either input. With a classical computer, nothing can ever be achieved: it is too late to make any use of the result by the time both $f(0)$ and $f(1)$ have been computed. In sharp contrast, the quantum computer allows one to obtain this exclusive-or before deadline with probability 50%. Even though either approach requires two days on average to compute one answer, the quantum approach gives a useful answer every other day, while the classical computer never does.

More to the point for this paper, the logical OR function of the outputs is *not* computable by quantum parallelism in the sense of the above template: no measurement can give the OR with nonzero probability of being certain of the answer [14]. Nevertheless, Deutsch and Jozsa have discovered that it *can* be computed efficiently in the context of some *promise problems*[4] [9], where the promise is that either all the outputs are 0 or that exactly half of them are 1. Moreover, this can be done *with certainty* in the sense that the result of the measurement is always correct and never "bad luck", provided that the promise is honoured (an unpredictable answer may otherwise result). In other words, consider any function $f : \Sigma^* \rightarrow \Sigma$, where Σ denote the binary alphabet $\{0, 1\}$. Assume that the function f has the property that, for each integer n, either $f(x) = 0$ for all x in Σ^n or else $f(x) = 1$ for exactly 2^{n-1} strings x in Σ^n. Consider a classical algorithm A capable of computing $f(x)$ in worst-case time in $O(t(n))$ for all x of length n, where $t(n) \geq n$ for all n. The result of Deutsch and Jozsa implies that the quantum computer can decide in worst-case time in $O(t(n))$ whether or not $f(x)$ is identically zero on inputs of length n.

Note that this could also be decided just as quickly by a one-sided-error probabilistic algorithm (it is thus in **RP** provided that $t(n)$ is a polynomial and the input n is given in unary). However, any correct deterministic uniform black-box classical algorithm *must* take time at least exponential in n for *every* value of n when in fact $f(x)$ is the identically zero function. This is true even of any *probabilistic* uniform black-box classical algorithm if we insist that the correct answer be obtained with certainty (zero error probability), and even considering the luckiest flips of the coin. In other words, any 100%–correct uniform black-box classical algorithm, deterministic or probabilistic, must take at least exponential time even in the *best* case (best input and best flips of the coin)! [By "black box" we mean that the algorithm has no access to the "code" of f: it is restricted to accessing it as if it were an oracle; by "uniform" we mean that the algorithm does not depend on the actual function f.]

2 Quantum Complexity Classes

The most natural quantum complexity class is obviously **QP**, the class of decision problems that can be solved with certainty in worst-case polynomial time by the quantum computer. Note that even though the quantum computer is allowed randomization by the very nature of quantum physics, it is allowed (in the context of **QP**) neither a non-zero error probability, nor a non-zero probability of exceeding its polynomial time bound. The same requirement for a probabilistic Turing machine would define nothing but the class **P** since coin-tossing is useless in such an austere setting. Obviously $\mathbf{P} \subseteq \mathbf{QP}$. The main point of this paper is to construct an oracle relative to which this inclusion is proper (Theorem 1). It must be pointed out that the class **QP** was introduced under this very name by Deutsch and Jozsa [9], showing that these physicists are far from ignorant of computational complexity theory. Thus we probably have far more to learn from them than they have to learn from us.

It is equally natural to define quantum versions of just about any classical complexity class. For instance, **QZPP** and **QBPP** are the classes of decision problems that can be solved by the quantum com-

[1] Worst case in the sense that the probability of success may depend on the actual values of the 2^n outputs.

[2] The only exception is that the result can be obtained with certainty if you wish to compute a constant function of the outputs, *i.e.* if the outputs are not used...

[3] This is useless since one can achieve the same goal without quantum mechanics: secretly choose one input r at random and compute $f(r)$. Then, if you need the value of $f(x)$ for some x chosen independently from r, just check whether $x = r$ and say "success" (with probability 2^{-n}) if that is so.

[4] Of course, "promise problems" is *our* jargon [12], not theirs!

puter with zero error probability in *expected* polynomial time, and bounded error probability in worst case polynomial time, respectively. Not only is there an oracle under which the inclusion **ZPP** $\subset$ **QZPP** is proper, but the language used for this separation result is in fact in **QP** relative to the same oracle (Theorem 3). We have not yet investigated the relative powers of **QBPP** versus **BPP**, which is likely to be a difficult problem. (The reader not familiar with classical probabilistic classes should refer to [11] for **BPP** and **ZPP**, and to [1] for **R**, which we call **RP** in this paper.)

3 Relativized Separation of P from QP

Let Σ denote the binary alphabet $\{0,1\}$ and let $f : \Sigma^* \to \Sigma$ be an arbitrary function. Let $S_f \subseteq \Sigma^*$ be the (sparse) set defined as

$$S_f = \{1^n \mid (\exists x \in \Sigma^n)\,[f(x) = 1]\,\}\,.$$

Let B be a predicate on functions which is `true` of f if and only if

$$(\forall n)\,[\,\#\{x \in \Sigma^n \mid f(x) = 1\} \in \{0, 2^{n-1}\}\,]\,.$$

The result of Deutsch and Jozsa described at the end of Section 1 says precisely that $S_f \in \mathbf{QP}^f$ provided that $B(f)$ holds, where $\mathbf{QP}^f$ denotes the oracle version of **QP** with an oracle for computing function f. The desired separation result is obtained by constructing an oracle f such that $B(f)$ holds and $S_f \notin \mathbf{P}^f$. Even better: any deterministic Turing machine that recognizes S_f correctly on all inputs, using f as oracle, must take *exponential time* on infinitely many inputs.

Notation: If M is a deterministic Turing machine, g is a function oracle and $x \in \Sigma^n$ for some integer $n \geq 2$, let $\mathcal{Q}(M, g, x)$ stand for the set of oracle questions that M^g asks of its oracle g when run on input x. However, if $M^g(x)$ asks more than 2^{n-2} different questions, $\mathcal{Q}(M, g, x)$ denotes the set of the first 2^{n-2} questions only. (Note that $\mathcal{Q}$ is not computable in general, but the following construction for f can be made effective with very little additional effort.)

Notation: Let M_2, M_3, M_4, … stand for a list of all deterministic oracle Turing machines. (We start at M_2 for an inessential technical reason.)

Theorem 1. There exists a function $f : \Sigma^* \to \Sigma$ such that $\mathbf{P}^f \subset \mathbf{QP}^f$ (strict inclusion).

Proof. We shall construct function f in stages with the help of sets $X_1 \subseteq X_2 \subseteq X_3 \subseteq \cdots$. At the end of stage $i \geq 1$, $f_i : \Sigma^* \to \Sigma$ will be defined such that $f_i(x) = 0$ for all $x \notin X_i$ (which is not to say that $f_i(x) = 1$ for all $x \in X_i$). Think of X_i as the set of those inputs on which the final function f has already been defined: once $x \in X_i$, we have for all $j > i$ that $x \in X_j$ and $f_j(x) = f_i(x)$. The key invariant in the proof (whose easy verification is left to the reader) is that $\#(X_m \cap \Sigma^n) < 2^{m-1}$ for all $m < n$.

Stage 1: Set $X_1 = \{x \in \Sigma^* \mid \text{length}(x) \leq 1\}$ and $f_0(x) = f_1(x) = 0$ for all $x \in \Sigma^*$.

Stage $n \geq 2$: If $\#\mathcal{Q}(M_n, f_{n-1}, 1^n) = 2^{n-2}$ or if $M_n^{f_{n-1}}(1^n) \neq 0$ then set $f_n = f_{n-1}$. Otherwise, consider $Y_n = \Sigma^n \cap (X_{n-1} \cup \mathcal{Q}(M_n, f_{n-1}, 1^n))$. By the invariant, $\#(X_{n-1} \cap \Sigma^n) < 2^{n-2}$ and, by definition of $\mathcal{Q}$, $\#\mathcal{Q}(M_n, f_{n-1}, 1^n) \leq 2^{n-2}$. Therefore, $\#Y_n < 2^{n-1}$. Let Z_n stand for an arbitrary subset of Σ^n that has empty intersection with Y_n and whose size is exactly 2^{n-1}. Define

$$f_n(x) = \begin{cases} 1 & \text{if } x \in Z_n \\ f_{n-1}(x) & \text{otherwise.} \end{cases}$$

Note that $\mathcal{Q}(M_n, f_n, 1^n) = \mathcal{Q}(M_n, f_{n-1}, 1^n)$ and $M_n^{f_n}(1^n) = M_n^{f_{n-1}}(1^n) = 0$ since $M_n^{f_n}(1^n)$ never queries $f_n(x)$ on those points $x \in Z_n$ on which f_n and f_{n-1} differ.

In all cases, set $X_n = X_{n-1} \cup \Sigma^n \cup \mathcal{Q}(M_n, f_n, 1^n)$.

Finally, let $f : \Sigma^* \to \Sigma$ be defined by $f(x) = f_n(x)$ for each $x \in \Sigma^n$. (Equivalently, $f(x) = \lim_{n\to\infty} f_n(x)$.) By construction, promise $B(f)$ holds and therefore $S_f \in \mathbf{QP}^f$. Consider now an arbitrary oracle deterministic Turing machine M_n so that M_n^f on 1^n asks less than 2^{n-2} questions of its oracle. By construction, $M_n^f(1^n) = 0$ if and only if $1^n \in S_f$, and thus M_n^f fails to recognize S_f. Therefore, any deterministic Turing machine that *does* recognize S_f given oracle f must ask an exponential number of questions, and thus spend exponential time, on at least one input. It is easy to modify the above construction to force each accepting Turing machine to take exponential time on infinitely many inputs. □

Note that the interest of this theorem is somewhat marred by the fact that $S_f \in \mathbf{RP}^f$ whatever the function f, provided that $B(f)$ holds. It is an open question whether there is an oracle relative to which there is a set in **QP**, or even in **QBPP**, that is not in **BPP**. Nevertheless, the next section states a relativized separation result between **QP** and **ZPP** (Theorem 3).

4 Extensions

The following theorems can be proved with similar techniques.

Theorem 2. Consider any strictly increasing time-constructible function t of the natural numbers. There exists an oracle relative to which there is a set that can be recognized in quantum time $O(t(n))$, yet any deterministic Turing machine that recognizes it requires $\Omega(2^{t(n)})$ time on infinitely many inputs. In particular, there is an oracle that separates **EXP** from **QEXP**. $\square$

Theorem 3. There exists an oracle relative to which there is a set that can be recognized in Quantum linear time, yet any zero-error-probability (Las Vegas) probabilistic Turing machine that recognizes it requires exponential time in the *best* case (and therefore also in the expected case) on infinitely many inputs. As a weak consequence, there is an oracle f and a set S such that $S \in \mathbf{QP}^f$, yet $S \notin \mathbf{ZPP}^f$. $\square$

5 Toward an Unrelativized Theory

A major open question is to separate **P** from **QP** in the real world. This is likely to be very difficult since no set is known to belong to **QP** that does not obviously belong to **BPP** (even in relativized settings). Therefore a major breakthrough either in quantum computing or in proving that $\mathbf{P} \neq \mathbf{BPP}$ would be required. Nevertheless, it is possible to formulate a classical (as opposed to quantum) computational complexity assumption under which (unrelativized) $\mathbf{P} \subset \mathbf{QP}$ follows immediately.

This involves a variant on the class $\mathbf{C_{==}P}$ (also known as **WPP**, see [10]), which is defined as the class of sets X for which there exists a nondeterministic polynomial-time Turing machine M and two polynomial-time computable functions f and g from Σ^* to the natural numbers such that $\#M(x) = f(x)$ for all $x \notin X$ whereas $\#M(x) = g(x)$ for all $x \in X$, where $\#M(x)$ denotes the number of accepting paths of machine M on input x. Valiant's better-known class **UP**, for instance, is a subclass of $\mathbf{C_{==}P}$ (with $f(x) = 0$ and $g(x) = 1$ for all x).

Let us define $\mathbf{C_{==}P}[\mathsf{half}]$ as the subclass of $\mathbf{C_{==}P}$ in which $f(x) = 0$ for all x and $g(x)$ is exactly half the number of accepting paths of M on input x (the number of paths can be made easily computable on a suitable universal enumeration of nondeterministic Turing machines, hence the function g is polynomial time). The result of Deutsch and Jozsa implies that $\mathbf{C_{==}P}[\mathsf{half}] \subseteq \mathbf{QP}$. Therefore, the somewhat reasonable assumption that $\mathbf{P} \subset \mathbf{C_{==}P}[\mathsf{half}]$ suffices to conclude that $\mathbf{P} \subset \mathbf{QP}$. Similarly, we get an unrelativized separation result between **ZPP** and **QP** if we assume the existence of a set $X \in \mathbf{C_{==}P}[\mathsf{half}]$ such that $X \notin \mathbf{ZPP}$. Note that once again, there is no hope to go further along this line without a novel approach to quantum computing since clearly $\mathbf{C_{==}P}[\mathsf{half}] \subseteq \mathbf{RP}$.

6 Quantum Cryptography

The results presented so far combine science-fiction with even more science-fiction: Deutsch's quantum computer, although perfectly fine in theory, could not be implemented with current or foreseeable technology, *and* we require the presence of an oracle or the proof of an assumption that would imply among other things that $\mathbf{P} \neq \mathbf{NP}$. In contrast, *quantum cryptography* provides a technologically feasible oracle-free approach to harnessing quantum physics for computational purposes. However, quantum cryptography bears no direct relevance to computational complexity theory: instead, it mildly challenges the Church–Turing thesis itself. Indeed, it allows one to achieve tasks that are demonstrably impossible when Turing machines are allowed unlimited computing power (but curiously, these tasks may well be possible if all Turing machines under consideration are restricted to probabilistic polynomial time!). Please refer to [2] for background and history about quantum cryptography.

Quantum cryptography allows the following tasks to be carried out by polynomial-time participants in a way that would-be cheaters cannot hope to gain any advantage from access to unlimited computing power. The list below has been purposely simplified, sometimes to the point of not being exactly true. Consult the references for details on the exact claims and protocols.

Key distribution [3]: Two parties that have no secret information initially end up with a secret shared key. To this effect, they exchange both quantum and classical messages. The eavesdropper can eavesdrop without restriction on the classical channel, but she cannot tamper with its transmissions, and she can do (almost) whatever is allowed by the laws of quantum physics with the quantum transmission. The claim is that the eavesdropper cannot fool the legitimate users into believing that they have succeeded in exchanging a secret shared key if in fact their key is either not secret or not shared or both. A working apparatus that implements this protocol is described in [2].

Bit commitment [3, 6]: One party can commit to the value of a bit b in such a way that the other party has no information about which bit it is, yet the first party can only open the commitment to show the original bit b. (Note that coin-tossing and zero-knowledge protocols are easy to achieve from a primitive for bit commitment.)

Oblivious transfer [4]: One party can send a bit b to another party in such a way that the bit is received with probability 50%. Neither party can influence the probability that the bit is received. The sending party obtains no information as to whether the transfer was successful, but the receiving party finds out with certainty.

Secure two party computation [4]: Consider a two-parameter polynomial-time computable function f. One party knows input x and the other party knows input y. The protocol allows both parties to compute $f(x, y)$ without disclosing any information to either party about the other party's secret input, except of course for what can be inferred from knowledge of one's secret input and of the final output.

Unfortunately, it has not yet been proven that the quantum cryptographic protocols proposed so far for these tasks can withstand *all* attacks consistent with the laws of quantum physics (although significant progress was recently achieved in this direction [5]). Nevertheless, it *is* proven that the only potentially damaging attacks would involve technologically infeasible feats, such as the ability to perform so-called coherent measurement on a large number of photons. (This was the meaning of "(almost)" in the description of the key distribution achievement.) Moreover, the following result holds without any assumptions other than the validity of quantum mechanics.

Theorem 4. If both parties have the availability of a fool-proof black box that implements perfect bit commitment, they can use it, together with quantum physics, to implement perfect oblivious transfer. The resulting protocol is secure against unlimited computing power and unrestricted technology, provided that the laws of quantum physics are not violated. $\square$

This theorem is particularly interesting in view of the result of Impagliazzo and Rudich to the effect that *any* proof of the above theorem in the classical (non quantum) setting could be translated into a proof that $\mathbf{P} \neq \mathbf{NP}$ in the real world [13].

7 Conclusion

The impact of quantum physics on structural complexity theory has been demonstrated. However, this is only the beginning of the story: no doubt that quantum physics has in store much more amazing potential.

Acknowledgements

This work has benefited from discussions with Charles H. Bennett, Claude Crépeau, David Deutsch, Artur Ekert, Lane Hemachandra, Richard Jozsa and Seinosuke Toda. The idea of considering oracle worlds was sparkled by the need to give a talk at the first New Mexico Theory Day, organized by Tim Long, which took place after this *Structures* program committee meeting. The seed for Theorem 2 was planted by a question asked by Lane Hemachandra during that talk, and he also deserves our thanks for suggesting that we investigate the relevance of the class $\mathbf{C}_{==}\mathbf{P}$. Quantum cryptography is joint work with Charles H. Bennett and Claude Crépeau, with occasional help from others.

References

[1] Adleman, L. and K. Manders, "Reducibility, randomness, and intractibility", *Proceedings of the 9th Annual ACM Symposium on Theory of Computing*, 1977, pp. 151 – 163.

[2] Bennett, C.H., F. Bessette, G. Brassard, L. Salvail and J. Smolin, "Experimental quantum cryptography", *Journal of Cryptology*, Vol. 5, no. 1, pp. 3 – 28.

[3] Bennett, C.H. and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing", *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, Bangalore, India, December 1984, pp. 175 – 179.

[4] Bennett, C.H., G. Brassard, C. Crépeau and M.–H. Skubiszewska, "Practical quantum oblivious transfer", *Advances in Cryptology — Crypto '91 Proceedings*, August 1991, pp. 351 – 366.

[5] Bennett, C.H., G. Brassard and N.D. Mermin, "Quantum cryptography without Bell's theorem", *Physical Review Letters*, Vol. 68, no. 5, 1992, pp. 557 – 559.

[6] Brassard, G. and C. Crépeau, "Quantum bit commitment and coin tossing protocols", *Advances in Cryptology — Crypto '90 Proceedings*, August 1990, pp. 49 – 61.

[7] Deutsch, D., "Quantum theory, the Church–Turing principle and the universal quantum computer", *Proceedings of the Royal Society*, London, Vol. A400, 1985, pp. 97 – 117.

[8] Deutsch, D., in *Quantum Concepts in Space and Time* (R. Penrose and C. Isham, editors), Oxford: Clarendon Press, 1986, pp. 215 – 225.

[9] Deutsch, D. and R. Jozsa, "Rapid solution of problems by quantum computation", manuscript, December 1991.

[10] S.A. Fenner, L. Fortnow and S. Kurtz, "Gap-definable counting classes", *Proceedings of 6th Annual IEEE Conference on Structure in Complexity Theory*, 1991, pp. 30 – 42.

[11] Gill, J., "Computational complexity of probabilistic Turing machines", *SIAM Journal on Computing*, Vol. 6, 1977, pp. 675 – 695.

[12] Grollmann, J. and A.L. Selman, "Complexity measures for public-key cryptosystems", *SIAM Journal on Computing*, Vol. 17, 1988, pp. 309 – 335.

[13] Impagliazzo, R. and S. Rudich, "Limits on the provable consequences of one-way permutations", *Proceedings of 21st Annual ACM Symposium on the Theory of Computing*, 1989, pp. 44 – 61.

[14] Jozsa, R., "Characterizing classes of functions computable by quantum parallelism", *Proceedings of the Royal Society*, London, Vol. A435, 1991, pp. 563 – 574.

[Apologies to Nature for this wasted page…]