6.896 Quantum Complexity Theory September 18, 2008

Lecture 5

Lecturer: Scott Aaronson

Last time we looked at what's known about quantum computation as it relates to classical complexity classes. Today we talk somewhat more 'concretely' about the prospects for quantum computing.

1 Recap

1.1 BQP and the Rest

We've informally shown the following complexity class inclusions: $P \subseteq BPP \subseteq BQP \subseteq PP \subseteq P^{\sharp P} \subseteq PSPACE \subseteq EXP$. We know $P \neq EXP$, so at least one of the inclusions in the chain must be strict (and we suspect that all except the first one are strict), but we can't even show $P \neq PSPACE$. This shows that separating BPP from BQP is at least as hard as the older, notorious open problem of P vs $PSPACE$.

1.2 Operations on Subsystems, Tensor Products

We've identified quantum transformations with unitary matrices acting on states given as vectors of complex amplitudes. As in classical circuit complexity, it is natural to restrict ourselves to applying 'local' transformations on our data. In the quantum case, we realize this as follows. Suppose we conceptualize our quantum state as having two 'registers' each taking multiple possible values (the partition we use may change from gate to gate, as in the circuit setting, depending which parts of the system we are operating upon). We write a quantum state as

$$|\psi\rangle = \sum_{i \leq m, j \leq n} \alpha_{i,j} |i\rangle |j\rangle,$$

where the first register takes on values identified in a manner of our choosing with $\{1, \ldots m\}$ and the second takes values in $\{1, \ldots n\}$.

A unitary transformation T on the system is considered 'local' to the first register if there exists some m-by-m matrix $U = (u_{i,k})_{i,k \leq m}$ such that T's action on basis vectors can be summarized as

$$|i\rangle |j\rangle \rightarrow (\sum_{k \leq m} u_{k,i} |k\rangle) |j\rangle,$$

for all $i \leq m, j \leq n$. We note that for T to be unitary, U must itself be unitary.

It can be verified that if we write the amplitudes of a state vector $|\psi\rangle$ in a column in the order $\alpha^T = (\alpha_{1,1}, \alpha_{2,1}, \ldots \alpha_{m,1}, \alpha_{1,2}, \ldots \alpha_{m,2} \ldots \alpha_{1,n}, \ldots \alpha_{m,n})$, then the resulting state after the transformation T is given by $(U \otimes I_n) \cdot \alpha$. Here the tensor product of an m-by-m matrix $A = (a_{i,j})$, with an n-by-n matrix B, is an mn-by-mn matrix, defined in block form by

$$(A \otimes B) = (a_{i,k} B)_{i,k \leq m}.$$

Similarly, if T is local to the 2nd register, summarized by an n-by-n unitary V, the action of T on the global state α is to produce $(I_m \otimes V)\alpha$.

This representation allows us to reason about quantum computations with tools from linear algebra. For instance, it allows us to show that operations applied to separate subsystems commute; see Pset 1.

2 Prospects for Quantum Algorithmics

2.1 Subroutines

In classical programming, we're used to the idea of designing a small algorithm to solve a simple problem, then calling this algorithm repeatedly to help solve a bigger problem. In our analyses we like to rely only on the fact that this 'subroutine' gives the correct answer, and think of it otherwise as a 'black box'. This is called *modularity*. The fact that this idea works correctly, without an unacceptable increase in running time, is expressed in the oracle result $P^P = P$.

It is less obvious that a similar idea works in the world of quantum circuits, but it does. A trick called 'uncomputing', described in the previous lecture, allows us to call subroutines as we'd like. This yields the complexity result $BQP^{BQP} = BQP$, which gives us confidence that BQP is a 'robust' class of languages, and a world in which we can design algorithms in modular fashion.

2.2 Controlling Error

In the early days of quantum computing (the '80s, say), skeptics said quantum computation was 'just' another form of analog computation (a machine manipulating real-valued quantities), and subject to the same limitation: perturbative noise that could erase information and throw off a computation by successively accumulating errors.

This criticism ignored or glossed over an important insight (explained in Bernstein-Vazirani '93): unlike other types of transformations used in classical analog computers, quantum operations *cannot* amplify error except by directly introducing more of it! To see what is meant, suppose the 'true' input to a unitary U in a computation is the state $|\psi\rangle$, but noise has intruded so that we have instead some $|\psi'\rangle$ at an l_2 distance at most ϵ away from $|\psi\rangle$. Then, using linearity and unitarity, $||U|\psi'\rangle - U|\psi\rangle||_2 = ||U(|\psi'\rangle - |\psi\rangle)||_2 = |||\psi'\rangle - |\psi\rangle|| \leq \epsilon$. Thus if all unitaries in the computation perform as expected, an ϵ error in the initial prepared state will yield an output state ϵ away from correct. This implies that the acceptance probability will be close to what it should've been.

Similarly, suppose that a unitary operation performs not exactly to our specifications, but is off by a 'small' amount. Let's model this as follows: we want to apply U, but instead apply $(U + \epsilon V)$, where V is a matrix with induced norm at most 1, i.e. $||Vx||_2 \leq ||x||_2$ for all x. In this case, inputting some value $|\psi\rangle$ yields $(U + \epsilon V)|\psi\rangle = U|\psi\rangle + \epsilon V|\psi\rangle$, a state whose l_2 distance from our desired state is at most $||\epsilon V|\psi\rangle||_2 \leq \epsilon ||V||_2 |||\psi\rangle|| = \epsilon$.

Applying this idea repeatedly to an entire computation, Bernstein and Vazirani observe that the total error is (in an appropriate sense) at most the 'sum' of the errors in all preparation and gate components. Thus, if engineers can produce circuit elements with error $\frac{1}{t}$, we expect to be able to faithfully execute computations on quantum circuits with size on the order of t.

This idea was improved upon substantially. In a sequence of papers, work by Aharanov, Ben-Or, Knill, Laflange, Zurek and others culminated in the 1996 'Threshold Theorem', which showed how

to use 'hierarchical coding' ideas to build quantum circuits for arbitrary BQP computations, which would remain reliable even if each gate was subjected to a sufficiently small (but *constant*) rate of error. This result was analogous to (but more complicated than) earlier work by Von Neumann on fault-tolerant classical computation.

There are certain requirements for the Threshold Theorem to be valid. These are: a constant supply of 'fresh' qubits; 'parallel processing'; and an extremely low error rate (on the order of 10^5 to .03, depending on your model). Razborov and others have shown limits to fault-tolerant quantum computation, showing that for particular circuit models there exist absolute constants of allowed error beyond which computational power appears to decrease. In Razborov's model, at an error rate .5 (since improved by others), quantum circuits provably degenerate to the computational strength of bounded-depth quantum circuits, the class $BQNC$.

However, (a) the true computational power of circuits facing so much error may be much, much lower, and (b) $BQNC$ already can perform most if not all of the quantum algorithms we now know of.

It was asked how the ideas of the Threshold Theorem can seem to suppress error in a computation, when we know unitaries are reversible and cannot erase error. This apparent inconsistency is resolved as follows: the Theorem's constructions do not suppress 'error' in an absolute sense by returning very nearly the state that would result from an error-free circuit; it is only the 'logical structure' of the computation that is preserved. In particular, the final acceptance probability is near what we'd expect, although the distribution on registers other than the final measured one may be wildly different from the error-free case.

3 Introducing Quantum Query Complexity

3.1 The Query Model

We have already discussed how proving complexity lower bounds on quantum computation seems very difficult. Just as with classical complexity, then, we turn to simplified, idealized models of computation in the hopes of proving something and gaining intuitions. We try to model, in the quantum world, the classical idea of a 'bounded-query' algorithm, one which has access to a very large (binary) input and tries to compute some predicate of the input without looking at too many input bits. We think of this input as a function $f(x) : \{0,1\}^n \to \{0,1\}$.

In this model, our algorithm maintains a quantum state over some unrestricted number of qubits. We allow the algorithm to apply *arbitrary* unitary transformations to its own state, as long as these are defined without reference to the values of f. We also allow the algorithm to perform a unitar update whose action on a basis state $|x\rangle$ is specified with reference to a single output value of f, possibly depending on x.

These 'query' unitaries are frequently restricted to be one of two types. The first type (a 'controlled-not query') maps basis state $|x, w\rangle$ to $|x, w \oplus f(x)\rangle$. Here x is the main register and w an auxiliary or 'ancilla' qubit.

The second type of query unitary frequently considered (the 'textitphase query) maps $|x\rangle \to (-1)^{f(x)}|x\rangle$. Note how in either case we take care to make the transformation reversible, and we check that the action is unitary.

It turns out that each of the two types above can simulate the other type in a single query (as long as ancillas are allowed). Thus we have a reasonably robust notion of a quantum query. We

are interested in the *number* of quantum queries required for various computations.

3.2 Example: The Deutsch-Jozsa Algorithm

Now we look at our first example of a quantum query algorithm, one which gives us a taste of how quantum parallelism can lead to faster algorithms than in the classical world. Suppose we are given a function $f : \{0,1\} \to \{0,1\}$, and wish to compute $f(0) \oplus f(1)$. In the classical world, of course, it would take us two queries to f to determine this value. We now show how to find it in only *one* quantum query.

Use a single-qubit register, initialized to $|0\rangle$. Apply a Hadamard operation, yielding $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$. (We don't charge for this, since the update rule doesn't refer to f.)

Next apply a phase query to our superposition, yielding (by linearity) the state

$$|\psi'\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}.$$

Note that if $f(0) = f(1)$ (equivalent to $f(0) \oplus f(1) = 0$), we have $|\psi'\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ (at least, modulo an irrelevant phase shift ± 1), while if $f(0) \neq f(1)$ we have $|\psi'\rangle = (\pm 1) \cdot \frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

Now apply another Hadamard. In the first case we get $\pm|0\rangle$, in the second case $\pm|1\rangle$. Thus a final, standard measurement to the register yields 0 with certainty if $f(0) \oplus f(1) = 0$, and 1 with certainty otherwise. This is the Deutsch-Jozsa algorithm, which yields a factor-2 speedup in computing the XOR of n bits. In future lectures, we will see quantum query algorithms for other problems which achieve much more dramatic speedups over classical query algorithms.

6.845 Quantum Complexity Theory
Fall 2010