6.841/18.405J: Advanced Complexity Theory May 8, 2002

Lecture 22

Lecturer: Madhu Sudan *Scribe: Benjamin Morse*

In this lecture we will cover Quantum Information, Manipulation, Circuits, and Computers. First we will close the topic of Average-case Complexity with some words about unanswered questions in the field.

1 Average-Case Complexity

Average-case Complexity is for the most part still not understood. Even the following simple problem, which is fundamental to the study of average-case complexity, is still unanswered.

Given a random SAT formula ϕ in n variables, with a number of 3-CNF clauses equal to Δn for some predetermined constant Δ, where the clauses are picked independantly randomly from a uniform distribution, is the formula satisfiable or not?

From experimental evidence, it's been determined that:

- if $\Delta > 6$, the probability that $\phi \in SAT$ approaches 0.
- if $\Delta < 3$, the probability that $\phi \in SAT$ approaches 1.

Between these two constants, it's harder to tell whether a given formula is satisfiable, and the threshold is around $\Delta = 4.2$. At this threshold, one would imagine that the problem is hard on average, but we have no formal way of showing this.

In practice, there are good heuristics for SAT, but no proofs - if a heuristic algorithm fails, we don't know if it's because there's no satisfying assignment, or if the heuristic simply can't cope with that particular formula. Also, for formulae that take a long time, there isn't a way of knowing if they're taking polynomial or exponential time. There's a lot of work that needs to be done to link theory and practice.

Ideally, we would want some way of relating DNP-completeness and NP-completeness. Feigenbaum and Fortnow have pointed out that one of the more obvious ways of doing this isn't going to work.

Let's assume, as we might like to, that a reduction exists from an NP-hard problem R' to a DNP problem (R, D), where D is a polynomial-time sampleable distribution, such that the randomized reduction of an instance x of R' produces (for example) four random instances x_1, x_2, x_3, x_4 of (R, D) such that all x_i are solvable. Each x_i is distributed according to D, but they don't need to be necessarily independant of each other. If this is possible, then the polynomial hierarchy collapses.

(Is it necessary that this distribution even exists, though? This is a tricky question - it does exist, but it's possible that it might not be sampleable.)

There are two ways out of this situation: We can follow a generic Turing reduction instead, or we could try using a problem that's complete for a class between P and NP (possibly BPP or 'statistical zero knowledge'), which could still give us an interesting result.

2 Physics vs. Computation

To the view of the physicist, physics is the goal, and computation is a tool that he can use to use to analyze physics and extract relationships. To the computer scientist, computers are the goal, and physics is a tool she can use to build machines that perform computation. (A fanciful example is using bent wire and a soap solution to form a surface - this solves an extremely complicated differential equation naturally.) The computer scientist comes up with a mathematical model, and tries to prove that it is both physically realizable, and that it's the strongest possible - that it can model all physical processes.

The Turing-Church hypothesis states that every physically realizable computing device can be simulated by a TM. The stronger version of this hypothesis also claims that a TM can do this with only polynomial slowdown. The hypothesis has ben challenged in the past by the introduction of randomness (which Turing

machines can't predict). Randomness is a resource that can speed up a process, and challenge the strong hypothesis (i.e. $\exists L \in BPP$ that we only know of an exponential-time deterministic algorithm for). And you can buy computer chips that will use physical processes to give you random numbers, so this could be an effictive challenge to the strong version of the hypothesis.

However, the more recent challenge to the strong Turing-Church hypothesis has come with the introduction of quantum computation.

2.1 The two-slit experiment

We have a wall that can measure the intensity of light hitting it. In front of this wall we place a screen with two slits, and in front of that, a light source. When either of the slits is covered, the wall is illuminated with a fairly predictable Gaussian-like intensity behind the open slit. However, if we uncover both slits, the intensity of the light is banded because of interference, with a node of no light at all between the two slits, instead of the straightforward superposition we would expect. We can describe the 1-screen case with differential equations, but Feynman posed the problem of n screens with n holes in series. Calculating the amount of light that hits that hits the wall at any point is in EXP. This is a case of TMs failing to simulate physical processes with polynomial slowdown, so computer scientists create a new model of computation, with Quantum Turing Machines.

2.2 Quantum information

A Quantum bit (qubit) can be described by a vector in $\mathbb{C}^2$ of unit length. This differs from a probabilistic bit (which is a real number $p \in [0,1]$). The two components of this state represent the probabilities of the two possible outcomes - a qubit can, when observed, resolve to either a 0 or a 1.

The state of an n-qubit system can be represented by a vector in $\mathbb{C}^{2^n}$. The reason why it can't be represented as a $\mathbb{C}^{2n}$ vector is because of quantum entanglement. Each of the 2^n components represent the probabilities that the system will resolve to each of the 2^n different possible outcomes.

2.3 Quantum operations

There are two main operations we can perform on a quantum system. We can manipulate it (which is referred to as Quantum Evolution) or we can observe part of, or all of the system (which is referred to as Quantum Measurement).

Consider a 2-qubit system with state $(\frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{2}, 0)$. This state is described in standard notation as

$$\frac{1}{\sqrt{2}}|00> + \frac{1}{2}|01> + \frac{1}{2}|10> + 0|11>$$

The coefficients to the left of the states are known as the amplitudes.

Using quantum measurement, we can look at part of the system. For example, we can look at the first bit and see how it comes out. The probability we'll get a 1 is $\frac{3}{4}$, and the probability we'll get a 0 is $\frac{1}{4}$. If we get a 0, that leaves the system in the state

$$\sqrt{\frac{4}{3}} \cdot \frac{1}{\sqrt{2}}|00> + \sqrt{\frac{4}{3}} \cdot \frac{1}{2}|01>$$

If we observe a 1, that collapses the system to the state

$$1|10>$$

2.4 Quantum evolution

Consider another n-qubit quantum system with state described by $v \in \mathbb{C}^{2^n}$, and a $2^n \times 2^n$ matrix U over $\mathbb{C}$ such that U is unitary (i.e. $U \cdot U^H = I_{2^n}$. U^H is the Hermetian transpose of U, which involves transposing the matrix and taking the complex conjugate of all cells.) Quantum evolution can be modelled by taking the state v and calculating $U \cdot v$ for the new state. A helpful unitary matrix for a single qubit is one of the form

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

, which rotates a qubit according to θ. This can be used to negate or flip a single qubit at a time.

2.5 Quantum circuits

A quantum circuit consists of a n-input, m-output circuit that has has these unitary matrices as gates. These gates can take any number of qubits as inputs and output the same amount of qubits. The full generalization of quantum circuits allows measurement to take place at any point in the system, but it can be shown that for any circuit there is an equivalent one that postpones measurement until the end with a small overhead.

Quantum circuits do not compute functions per se, but essentially sample from a distribution $D : \{0,1\}^n \to \{0,1\}^m$. These distributions are not necessarily polynomial-time sampleable - there are D that are not poly-size circuits, but are poly-size Q-circuits.

2.6 QTMs

Does there exist a generalized machine that can simulate any gate arrangement? A QTM is roughly analogous to a deterministic Turing machine. Recall the formal definition of a Turing machine as the tuple

$$(Q, \Sigma, \Gamma, \delta, q_0, F)$$
$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, stay\}$$
$$q_0 \in Q, F \subseteq Q$$

A QTM has a very similar definition, except that Σ and Γ are qubits, and δ is a unitary matrix. For every quantum circuit C that samples from a distribution D, there is a QTM M that simulates it to within any arbitrary ϵ. The reason we can only attain ϵ closeness is because there can be irrational constants like $\sqrt{3}$ that appear in the gates of C, but don't show up in the QTM. We can approximate these constants, though. [Bernstein & Vazirani]