15-459: Undergraduate Quantum Computation September 1, 2022

Lecture 2 : Rotate, compute, rotate

Lecturer: Ryan O'Donnell

Scribe: Rajeev Godse

Quantum mechanics is the simplest possible model that allows negative probabilities.

1 Probabilistic computing

A probabilistic computer consists of deterministic code along with the ability to flip a fair coin.

1.1 Questions

Is probabilistic computing more powerful than determinstic computing?

1. Well, by definition, yes. Task: print 5 truly random bits. The classical computer can't do it.
2. But also maybe not. The functions computable by a deterministic computer are the same as a probabilistic computer. If a random algorithm is always correct, we can replace all coin flips with tails and we get an equally efficient algorithm.

How can probability help with function tasks?

- If we are willing to accept a small error probability, there are some problems for which the best known randomized algorithm is more efficient than the best known deterministic algorithm (like MST finding).

Example: Primality testing.

Naive algorithm: $\approx \sqrt{2}^n$ steps.

Gary Miller, '76: There's a deterministic algorithm that uses $\approx n^4$ steps, assuming Riemann Hypothesis is true.

Solovay-Strassen '77: There is a probabilistic algorithm for primality testing using $\approx n^3$ steps using Jacobi symbols.

Rabin '80: Modifies Miller's algorithm to use randomness and tests primality in $\approx n^2$ steps without depending on the Riemann Hypothesis.

AKS '02: Deterministic algorithm that provably tests primality in n^{12}, later brought down to n^6.

When it comes to polynomial time, the best answer is **no**. Assuming strongly believed conjecture ($\mathsf{EXPTIME} \neq \mathsf{MA}$), every function problem that can be solved in polynomial time on a probabilistic computer is in P.

1.2 Summary

- It's cool!
- Classic computing + 1 new power.
- Analyzing it needs new math (probability theory).
- Engineers can build these things.

- Quintessential use: simulate something random.
- (Seems to) give speedups from one level of P to another.
- (Very likely) doesn't give speedups from exponential time to polynomial time for any function problem.

1.3 Analysis

Consider an algorithm that initializes an array with 1000 random bits and then performs some deterministic computation on it. To describe the final state of the array, we would need a random variable, which would require 2^{1000} numbers to describe.

2 Quantum computing

2.1 Summary

- It's cool!
- Classic computing + 1 new power.
- Analyzing it needs new math (linear algebra).
- Engineers are having difficulty building these things.
- Quintessential use: simulate something quantum.
- (Seems to) give exponential speedups from EXPTIME into P for at least one important problem (factoring).
- (Very likely) doesn't give speedups from exponential time to polynomial time for any NP-complete problems.

2.2 Analysis

Consider an algorithm that initializes 1000 photons with known states and then performs some deterministic computation on it. To describe the final state of the array, we require 2^{1000} numbers.