Lecture 1

Wed., 15 JAN 2014

- everyone should introduce themselves
- go over administrative aspects

changing course time to be

MW 12:30pm - 1:50pm

grading is pass | fail

expectation is for you to do all of the homeworks on time

Final presentation:

similar to last semester

20-minute presentation

- new developments w/ quantum algorithms

"solving" linear equations

estimating the gradient of a function

google "quantum algorithm zoo" maintained by Stephen Jordan of NIST

collision finding — given access to a function f find inputs x ≠ y for which f(x) = f(y)

breaking some digital signature schemes

②

quantum simulation:

simulating a physically realistic Hamiltonian H, the time evolution operator e^{-iHt} can be implemented using $\text{poly}(n,t)$ q. gates.

We will learn that finding ground state energies of an arbitrary physically realistic Hamiltonian is QMA-complete (hard for a q. computer), there are q. algorithms for approximating the ground states (or thermal states) of some Hamiltonians.

evaluating partition functions (from which any thermodynamic quantity can be derived) can be done w/ a speedup for a q. computer q: computational complexity theory

We will learn about standard classes such as BQP & QMA

Investigations into more exotic classes is welcome.

QMA(2), QIP(2), QSZK, multiprover interactive proofs, quantum games.

Course Overview:

goal: What are the ultimate physical limits on computation?

several aspects to this question

begin by reviewing classical theory of computation

- Turing machines - model

Church-Turing thesis
(any function which is "effectively calculable" by some mechanical / physical process is computable by a Turing machine)

halting problem — some functions are not computable

clever self-referential trick

brings up the idea of reductions

can show that it is impossible to solve a given problem (a function is uncomputable if there is a method to solve the halting problem given a method to solve it. this notion is essential in complexity theory.

complexity theory is a scaled down version of computability theory in which we add qualifiers on resources such as time + space. many of the conceptual aspects remain.

complexity theory is a way of classifying problems into easy + hard ones (but then many refinements of this notion) develop the classical circuit model of computation, which might seem more closely connected to a physical device such as a computer chip.

- a "circuit" is really a directed, acyclic graph composed of logic gates & wires.

a reasonable variation of this model is equivalent to the Turing machine model.

most well known fused model for quantum computation is similar to this model.

---

discuss basic complexity classes

P, NP, ~~[illegible]~~, PSPACE, EXP

BPP, MA

"P ≠ NP" law

if hierarchy developed by physicists, then these ~~relations~~ conjectures would be called "physical laws" review q. mechanics (Jonny Olson)

---

basics of q. computation

skip q. Turing machines

define universal gate sets

not all q. operations will be realizable in poly-time.

can then define uniform quantum circuit families

can then define BQP.

---

basic q. algorithms

Deutsch-Josza

Grover search

Shor algorithm, more generally phase estimation

Quantum Complexity Classes

BQP

QMA – canonical problem for it

is local Hamiltonian

(uses ideas of Feynman)

– properties of the class

(robustness under deviations)

~~mixed state q. mechanics~~

q. interactive proofs

QIP(2)

QIP(3) collapse of the

QIP hierarchy

QSZK

q. games

multiprover scenarios