Lecture Outline: Quantum Complexity Theory

5.1 What is (Computational) Complexity Theory?

Complexity theory and quantum computing

Computational complexity theory focuses on classifying **computational problems** according to their *resource usage*, and relating classes of problems to each other. (Wikipedia)

5.1.1 Examples of computational problems

1. **Multiplication**. Given a pair of integers (m, n) as input, compute their product

2. **Factoring**. Given a composite number n as input, decompose n into a product of smaller integers

3. **Satisfaction** (SAT). Given a set of constraints on a set of Boolean variables, decide if the constraint system is satisfiable or not

   $(x_1 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge \cdots$

4. **Counting** (#SAT). Given a set of constraints on a set of Boolean variables, compute the number of different solutions to the constraint system

5. **True Quantified Boolean Formula**. Given a formula in quantified propositional logic where every variable is quantified by either existential or universal quantifiers at the beginning of the formula, decide if the formula evaluates to true

   $\forall x_1 \exists x_2 \forall x_3 \forall x_4 ((x_1 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge \cdots)$

Remarks

1. A *family* of problem instances, indexed by the input size

2. Decision problems, search problems, promise problems

   Decision problem: a set $A \subseteq \Sigma^*$ where $\Sigma = \{0, 1\}$

   Promise problem: disjoint sets $A_{yes}, A_{no} \subseteq \Sigma^*$

5.1.2 Computational resources

Time, space, depth, query, . . .

Computational models

- Turing machines

  A brief discussion of TMs

  Transition rule $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

  Configuration of a TM

  1. Current state q
  2. Current tape content
  3. Current head position

- Classical and quantum *circuit models*

  Classical circuits are not necessarily reversible

Efficient computation Polynomial-time (mathematically simple, model-independent)

5.1.3 P vs NP

- P: A promise problem A is in P if and only if there exists a polynomial-time *deterministic* Turing machine that accepts every string $x \in A_{yes}$ and rejects every string $x \in A_{no}$

- NP: *Non-deterministic* polynomial-time

  Non-deterministic transition rule $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$

  A non-deterministic Turing machine accepts if there exists an accepting computational path

  Configurations of a non-deterministic TM

  A proof-verification perspective of NP

5.1.4 Church-Turing thesis

A problem is (*efficiently*) computable by an **effective method** if and only if it is (*efficiently*) computable by a Turing machine.

Quantum computing is a candidate that disproves the *extended* Church-Turing thesis.

5.1.5 Exercise 1

Let n be a composite number. It is then easy to see that n has a factor at most $\sqrt{n}$. Does this fact give us an efficient algorithm for factoring?

5.1.6 Exercise 2

The decision version of the factoring problem asks if n has a factor less than k when given (n, k) as input. Show that the factoring problem is efficiently reducible to this decision version. That is, if there is an efficient algorithm for the decision version, there is also an efficient algorithm for the standard version.

5.2 BQP: Efficient Quantum Computation

5.2.1 Definition

Let $A = (A_{yes}, A_{no})$ be a promise problem and let $c, s : \mathbb{N} \to [0,1]$ be functions. Then $A \in \mathrm{BQP}(c,s)$ if and only if there exists a *polynomial-time uniform family of quantum circuits* $\{Q_n : n \in \mathbb{N}\}$, where Q_n takes n qubits as input and outputs 1 bit, such that

- if $x \in A_{yes}$ then $\Pr[Q_{|x|}(x) = 1] \geq c(|x|)$, and
- if $x \in A_{no}$ then $\Pr[Q_{|x|}(x) = 1] \leq s(|x|)$.

The class BQP is defined as $\mathrm{BQP} = \mathrm{BQP}(2/3, 1/3)$.

5.2.2 Error reduction for BQP

Theorem. Let $p : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function satisfying $p(n) \geq 2$ for all n. Then it holds that $\mathrm{BQP} = \mathrm{BQP}(1 - 2^{-p}, 2^{-p})$.

5.2.3 BQP subroutine theorem

Theorem. $\mathrm{BQP}^{\mathrm{BQP}} = \mathrm{BQP}$.

5.2.4 Complexity classes of oracle machines

An oracle is a subset $B \subseteq \Sigma^*$. An oracle Turing machine with oracle B attached is a Turing machine which may call the oracle B at intermediate computational steps and the call counts as a **single** step.

$\mathrm{P}^B, \mathrm{NP}^B, \ldots$

Oracles in the circuit model: in addition to the usual gates, we have a family of big gates $\{O_m\}$ such that

$$O_{|y|}(y) = \begin{cases} 1 & y \in B, \\ 0 & y \notin B. \end{cases}$$

For a complexity class C, we define

$$\mathrm{P}^{\mathrm{C}} = \bigcup_{B \in \mathrm{C}} \mathrm{P}^B.$$

$\mathrm{NP}^{\mathrm{NP}}$ and the polynomial hierarchy

In the quantum case, we adopt the form of the oracle access as $O_m : |y, a\rangle = |y, a \oplus O_m(y)\rangle$

5.2.5 Proof

What do we need to prove?

Two difficulties:

1. The output of a BQP circuit is probabilistic
2. We need to simulate the behaviour of the O_m gate on all qubits

5.2.6 Optimality of Grover search

Proof outline Hybrid method

Phase-kickback oracle: $O|i\rangle = (-1)^{x_i}|i\rangle$.

Assume the algorithm has the form $U_T O_T U_{T-1} \cdots U_1 O_1 U_0$.

Hybrid t for $t = 0, 1, \ldots T$: $O_s = \begin{cases} O & s > T - t, \\ I & \text{o.w.} \end{cases}$

Let $|\xi_t\rangle$ be the final state of the Hybrid t circuit.

Consider Hybrid 0 and let $|\psi_t\rangle = \sum_{i=1}^{n} \alpha_i^{(t)} |i\rangle$ be the state of this hybrid before the t-th query.

We have

$$\sum_{i=1}^{n} \sum_{t=1}^{T} \left|\alpha_i^{(t)}\right|^2 = T, \tag{1}$$

and there exists i_0 such that $\sum_{t=1}^{T} \left|\alpha_{i_0}^{(t)}\right|^2 \leq T/n$. By Cauchy-Schwarz inequality,

$$\sum_{t} \left|\alpha_{i_0}^{(t)}\right| \leq \sqrt{T \sum_{t} \left|\alpha_{i_0}^{(t)}\right|^2} = T/\sqrt{n}. \tag{2}$$

Then it is easy to show

$$\begin{aligned} \||\xi_t\rangle - |\xi_{t-1}\rangle\| &\leq 2\left|\alpha_{i_0}^{(t)}\right|, \\ \||\xi_T\rangle - |\xi_0\rangle\| &\leq 2\sum_{t} \left|\alpha_{i_0}^{(t)}\right| \leq 2T/\sqrt{n}. \end{aligned} \tag{3}$$

$|\xi_T\rangle$ is the final state of the algorithm, and it has a small probability of giving outcome i_0 as it is very close to $|\xi_0\rangle$.

Quantum query complexity theory

5.2.7 Relation with classical friends

- BPP: Same as BQP, but uses (random) classical circuits
- PP: Same as BPP, but with $c > 1/2$ and $s \leq 1/2$
- PSPACE: A promise problem A is in PSPACE if and only if there exists a deterministic Turing machine running in polynomial space that accepts every string $x \in A_{yes}$ and rejects every string $x \in A_{no}$
- PH: Polynomial hierarchy

Meet more complexity animals at Complexity Zoo!

P $\subseteq$ BPP $\subseteq$ BQP $\subseteq$ QMA $\subseteq$ PP $\subseteq$ PSPACE

Conjecture. BQP is not contained in NP and vice versa.

5.2.8 BQP vs PP

Theorem. BQP $\subseteq$ PP.

- GapP functions

  A function $g : \Sigma^* \to \mathbb{Z}$ is a *GapP function* if there exists a polynomial p and a polynomial-time computable funcion f such that

$$g(x) = \#\{y \in \Sigma^{p(|x|)} : f(x,y) = 0\} - \#\{y \in \Sigma^{p(|x|)} : f(x,y) = 1\} = \sum_{y \in \Sigma^{p(|x|)}} (-1)^{f(x,y)}.$$

- Lemma: A promise problem is in PP if and only if there is a GapP function g such that

  1. if $x \in A_{yes}$ then $g(x) > 0$, and
  2. if $x \in A_{no}$ then $g(x) \leq 0$.

- Fact: quantum computational universality of H and Toffoli
- Quantum computing is all about estimating the first entry of unitary circuits
- Encode amplitudes as GapP functions

5.2.9 Beyond BQP: Quantum Merlin Arthur

Definition Let $A = (A_{yes}, A_{no})$ be a promise problem and let $c, s : \mathbb{N} \to [0,1]$ be functions. Then $A \in$ QMA(c,s) if and only if there exists a *polynomial-time uniform family of quantum circuits* $\{Q_n : n \in \mathbb{N}\}$, where Q_n takes $n + m(n)$ qubits as input for some polynomial m and outputs 1 bit, such that

- *(Completeness)* if $x \in A_{yes} \cap \Sigma^n$ then there exists an $m(n)$-qubit state $|\psi\rangle$ such that $\Pr[Q_n(x, |\psi\rangle) = 1] \geq c(n)$, and
- *(Soundness)* if $x \in A_{no} \cap \Sigma^n$ then for all $m(n)$-qubit state $|\psi\rangle$, $\Pr[Q_n(x, |\psi\rangle) = 1] \leq s(n)$.

The class QMA is defined as QMA$(2/3, 1/3)$.
Quantization of both the witness and the circuit.

Satisfiability and local Hamiltonian problem **Definition**. A k-local Hamiltonian H is a summation $H = \sum_{j=1}^{m} H_j$ of local terms H_j acting on at most k qubits (out of n qubits). The k-local Hamiltonian problem is the promise problem with

- *Input*: (H, a, b) where H is a k-local Hamiltonian, a, b are real numbers such that $b - a \geq 1/poly(n)$,
- *Yes instances*: The smallest eigenvalue of H is at most a,
- *No instances*: The smallest eigenvalue of H is at least b.

A 3-SAT clause as a Hamiltonian term

- **Quantum Cook-Levin Theorem**. The k-local Hamiltonian problem is QMA-complete for all $k \geq 2$.

Circuit to Hamiltonian construction Aharonov and Naveh, Quantum NP: A Survey

Error reduction for QMA Marriott and Watrous, Quantum Arthur-Merlin Games

5.2.10 Exercise 3

Write down a definition of BQP without looking at any reference. Compare it with the definition given above and see if you have missed anything.

5.2.11 Exercise 4

Prove the error reduction theorem for BQP.

5.2.12 Exercise 5

Prove that BQP is in PSPACE