Introduction to Quantum Computing | Lecturer: Tongyang, scribed by Shuo Zhou
Fall 2023, Peking University | *Date: December 20, 2023*

Lecture 20

Quantum Computational Complexity

- PSPACE
- NP, co-NP
- QMA

1 More Complexity Classes

$L \in$ PSPACE: L can be decided by poly-space deterministic classical algorithm.

$L \in$ EXP: L can be decided by an exponential-time deterministic classical algorithm.

PSPACE $\subseteq$ EXP.

$L \in$ PSPACE means L can be solved in $p(n)$ space $2^{p(n)}$.

BQP $\subseteq$ EXP : We can simulate a quantum circuit in exponential time by explicit linear algebra.

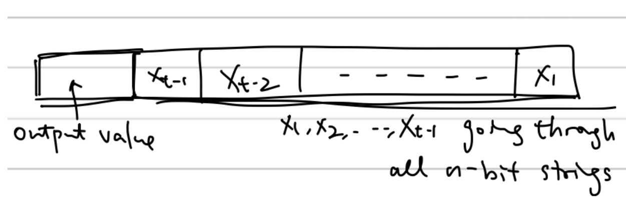
Theorem. BQP $\subseteq$ PSPACE.

Proof. Assume that the quantum algorithm is $U_t U_{t-1} \ldots U_2 U_1 |0^n\rangle$, $t = \text{poly}(n)$.

Considering $\sum_{x\in\{0,1\}^n} |x\rangle\langle x| = I_N$, we can do

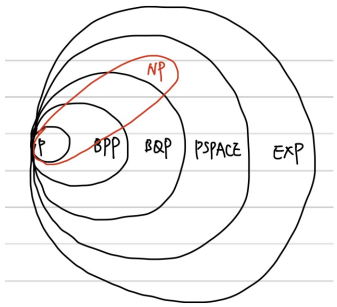
$$\langle 0^n| U_t U_{t-1} \ldots U_2 U_1 |0^n\rangle = \sum_{x_1, x_2, \ldots x_{t-1} \text{all } n\text{-bit strings}} \langle 0^n|U_t|x_{t-1}\rangle \langle x_{t-1}|U_{t-1}|x_{t-2}\rangle \cdots \langle x_1|U_1|0^n\rangle .$$

Each of these terms can be computed in poly-time for a given $x_1, \ldots, x_{t-1}$ ($\langle x_i|U_i|x_{i-1}\rangle$ gives the element in the $(x_i)^{th}$ row and $(x_{i-1})^{th}$ column of U_i). $\square$



The sum of these values can be computed in polynomial space: maintain a register at the beginning, and cumulatively add all values.

As a summary:



NP: Non-deterministic Polynomial

We say $L \in$ NP if $\exists$ poly-time classical, deterministic algorithm $A(x, y)$ such that for any $x \in \{0,1\}^n$

$x \in L \Rightarrow \exists$ storing y s.t. $A(x, y)$ accepts

(y is a efficient verifier)

$x \notin L \Rightarrow \exists$ storing y s.t. $A(x, y)$ rejects

For example: 3-SAT: Instances are 3-CNFs (conjunctive normal forms), such as

$$\varphi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_5 \vee x_7 \vee \overline{x}_3) \wedge \cdots$$

3-SAT $\in$ NP: For any $\varphi \in L$, the y is a satisfiable assignment and its correctness can be verified in linear time.

In fact 3-SAT is NP-complete (Cook-Levin Theorem)

If 3-SAT can be solved in polynomial time (i.e. $\exists$ P), then P = NP.

More on efficiently verifiable problems?

1.1 Asymmetry of NP, we only need to have short proof for yes instance.

Definition. Given a decision problem X. It's complement $\overline{X}$ is the same problem with yes and no answers reversed.

For example: Prime $X = \{2, 3, 5, 7, 11, 13, \ldots\}$

$\overline{X} = \{0, 1, 4, 6, 8, 9, 10, 12 \ldots\}$

co-NP: Complement of decision problems in NP.

Intuitively: For a problem $X \in$ NP, for yes instance there is an efficient certificate;

For a problem $X \in$ co-NP, for no instance there is an efficient disqualifier.

Fundamental question: Does NP = co-NP?

Common opinion: No.

Theorem. If NP $\neq$ co-NP

Observation. P $\subseteq$ NP $\cap$ co-NP? Mixed opinions.

Fact. Consider the FACTOR problem: Given two positive integers x and y. Does x have a nontrivial factor y?

We have FACTOR $\in$ NP $\cap$ co-NP.

Proof. FACTOR $\in$ NP: Certificate A factor p of x such that $2 \leq q \leq y$.

FACTOR $\in$ co-NP: Disqualifier: The prime factorication of x, where each prime factor $> y$.

For example: $x = 1001, y = 6$ give $1001 = 7 \times 11 \times 13$.

Determining whether a number is prime $\in$ P (AKS primarity test) □

On the other hand, currently it is not known whether FACTOR $\in$ P.

1.2 Probabilistic Versions

$$\text{P} \xrightarrow{\text{probabilistic}} \text{BPP} \xrightarrow{\text{quantum}} \text{BQP}$$

$$\text{NP} \xrightarrow{\text{probabilistic}} \text{MA} \xrightarrow{\text{quantum}} \text{QMA}$$

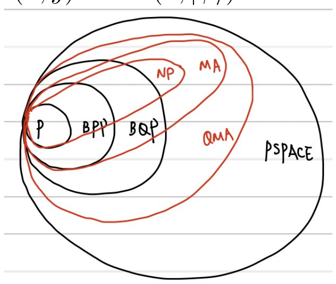
MA represents Merlin-Arthur here.

Formally, we say that a language $L \in \text{MA}$ if $\exists$ poly-time randomized algorithm $A(x, y)$ (where y is the witness from Merlin) such that for all $x \in \{0, 1\}^*$:

$x \in L \Rightarrow \exists$ string y, such that $A(x, y)$ accepts with probability $\geq \frac{2}{3}$.

$x \notin L \Rightarrow \forall$ string y, $A(x, y)$ accepts with probability $\leq \frac{1}{3}$.

QMA: Same as MA, but the algorithm A takes polynomial quantum gates, and the proof is a quantum state: $A(x, y) \longrightarrow A(x, |\psi\rangle)$.



Consider the k-local Hamiltonian systems.

Instance: Hermitian operators $H = \sum_j H_j$ where each H_j acts non-trivially on at most k out of n qubits.

Let λ be the smallest eigenvalue of H. (Since H is Hermitian, λ must be real).

Given thresholds $a < b$ with $b - a \geq \frac{1}{\text{poly}(n)}$.

Problem. Determine whether $\lambda \leq a$ or $\lambda \geq b$, under the promise that one of them is true.

k-SAT is a special case where all H_j is diagonal.

For example: Clause $x_1 \vee \overline{x}_2 \vee x_3 \longleftrightarrow$ 3-local term $|00\rangle\langle 00|$.

Eigenvalue of $|x_1 \dots x_n\rangle$ in $H = \#$ of violated constraints.

In addition k-local Hamiltonian $\in$ QMA.

Witness: Ground state $|\psi\rangle$.

Verification: Perform phase estimation on e^{-iHt} (for an appropriate t) for state $|\psi\rangle$. $\Rightarrow$ given an estimate of λt.

In addition, e^{-iHt} can be efficiently implemented by Hamiltonian simulation. Therefore, this gives a polynomial-time algorithm under the given promise of H.

In fact, k-local Hamiltonian is QMA-complete. ($k \geq 2$)

Idea: Clock construction (Kitaev).