Randomized and Quantum Complexity Classes

Science Coffeehouse Project

Presented by: Harish Adsule
Mentor : Aditya Gulati

2021

Lecture Overview

1. Randomized Computation
   - Probabilistic Turing Machines
   - BPP Class
   - Sister Classes and their relations
2. Quantum Computation
   - Quantum Computation Model
   - BQP and QMA Classes
   - Relations with other classes
   - Probabilistic vs Quantum Computation

Randomized Computation

PTMs and BPP

Probabilistic Turing Machine

A PTM 'M' is syntactically similar to a NDTM. At each point of computation, it randomly chooses between the two trasition functions. For $T : \mathbb{N} \mapsto \mathbb{N}$, we say that a PTM M decides a language L in time $T(n)$ if M halts within $T(|x|)$ steps regardless of its random choices and $P(M(x) = L(x)) \geq 2/3$.

BPTIME and BPP

BPTIME(T(n)) denotes the set of all languages decidable by PTMs in $O(T(n))$. BPP $= \bigcup_c$ BPTIME(n^c)

Robustness

Some remarks regarding the robustness of the above definition:(See Barak-Arora 7.4)

- The constant $\frac{2}{3}$ is arbitrary. We can choose any constant greater than $\frac{1}{2}$. In fact, we can transform a given PTM into a machine with success rate exponentially close to one.
- Whether a PTM can choose between the two transition functions with equal probability or not has no effect on its power.
- Instead of *at most* $T(|x|)$ steps, we can allow for *expected* number of steps to be $T(|x|)$ which leads to an equivalent definition.*

Examples of BPP

- PRIMES
  No efficient algorithm for testing primality was known till the 1970s, when an efficient randomized algorithm was discovered hence the problem is in BPP. (It is in fact in P, a fact which was shown by Agrawal, Kayal and Saxena in 2002.)
- Polynomial Identity Testing
  Define ZEROP to be the set of all arithmetic circuits which are identically zero(Arithmetic Circuits $\rightarrow$ Polynomials). (Currently unknown if it is in P.)
- Perfect Matching in a Bipartite Graph
  A randomized algorithm was given by Lovász based on the fact that the adjacency matrix of a perfectly matched graph should have a non-zero determinant.(Deterministic algorithms are well known for this problem.)

PIT is in BPP

Schwartz-Zippel lemma

Let $p(x_1, \ldots, x_n)$ be a polynomial not identically zero and of degree d. Let S be a finite set. If we pick $a_1, \ldots, a_n$ randomly and uniformly from S, then:

$$Pr(p(a_1, \ldots, a_n) = 0) \leq \frac{d}{|S|}$$

Proof[4] is by induction on n.

Now we can easily give an algorithm.

- Choose a finite set S such that $|S| > d$.
- Choose n numbers randomly and uniformly from it.
- Plug in the numbers and accept if the output is 0 and reject otherwise.

Proof of correctness:

- If $p(x_1, \ldots, x_n)$ is identically zero, our algorithm always accepts it.
- Otherwise, by SWL, the probability of a wrong answer is at most $\frac{d}{|S|}$. This can be made arbitrarily small by performing many independent trials. i.e. if we make k independent trials, the probability reduces to $\left(\frac{d}{|S|}\right)^k$.

Sister Classes

Algorithms in BPP allow for *two-sided* errors. Now we look at classes with one-sided and zero errors.

RP and coRP

We define RTIME($T(n)$) to be the set of all languages L for which there exists a PTM running in time $T(n)$ such that:

$$P(M(x) = 1 | L(x) = 1) \geq \frac{2}{3}$$
$$P(M(x) = 1 | L(x) = 0) = 0$$

RP $= \bigcup_c$ RTIME(x^c). coRP is defined similarly, coRP $= \{$L $\mid \overline{L} \in$RP $\}$

ZP

Define ZTIME($T(n)$) to be the set of languages L for which there exists a PTM running in **expected** time $T(n)$ such that it always gives the correct answer. ZP $= \bigcup_c$ ZTIME(x^c).

Relationship with other classes

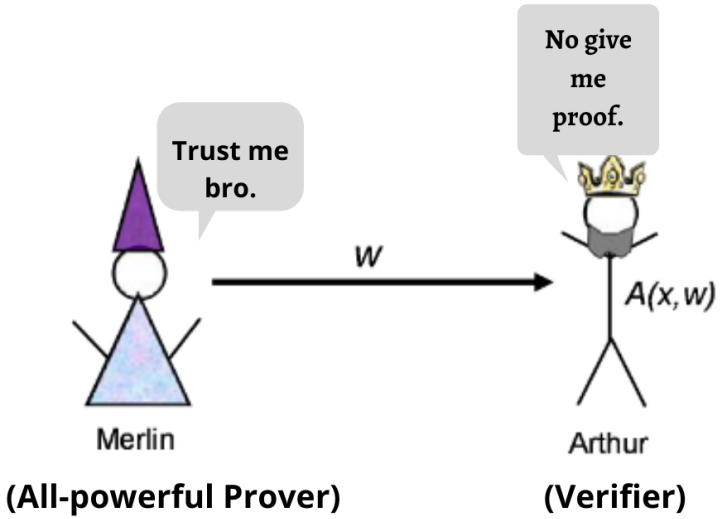
Known relations

- $P \subseteq BPP \subseteq EXP$
- $BPP = coBPP$
- $ZPP = RP \bigcap coRP$
- $RP, coRP \subseteq BPP$
- $RP \subseteq NP$

Open questions

- BPP vs NP ?
  Is non-determinism more powerful than randomization?
- BPP vs P
  i.e. Does randomization really give any advantage over determinism? Can be proved otherwise under certain assumptions. Essentially, the result necessitates the existence of "suitable" pseudo-random number generators. (Impagliazzo-Wigderson,97)

Glimpse of Interactive Protocols



- Interactive protocols involve a verifier and one or more provers which can interact with one another and the verifier can then decide whether to accept the input or not.
- We can define a probabilistic analog of NP, the MA (Merlin Arthur) class.
- For languages in MA, the verifier(Arthur) can run a probabilistic poly time algorithm after the prover(Merlin) gives it the certificate/proof.

Quantum Computation

Review of notation

Here is a quick review of the Dirac notation:

- $|\psi\rangle$ represents a column vector in matrix notation.
- $\langle\psi|$ represents the adjoint of $|\psi\rangle$ i.e. it is a row vector.
- $\langle\phi|\psi\rangle$ represents the inner product of the two vectors.
- $|\phi\rangle\langle\psi|$ represents the outer product of two vectors each of size n. It is a matrix of size $n \times n$.
- Completeness Relation
  For a given set of orthonormal basis vectors $|i\rangle$, we have:

  $$\Sigma_i |i\rangle\langle i| = I$$

  Can be shown easily by expressing an arbitrary vector in the given basis.

Quantum Computation Model

We now formally define Quantum Computation:

Quantum Computation

Let $f : \{0,1\}^* \mapsto \{0,1\}^*$ and $T : \mathbb{N} \mapsto \mathbb{N}$ be some functions. f is said to be computable by a Quantum Turing Machine in time $T(n)$ if it can be computed by the following process:

1. Initialize an m qubit quantum register to the state $|x0^{m-n}\rangle$ (i.e., x padded with zeroes).
2. Apply one after the other $T(n)$ elementary quantum operations $F_1, \ldots, F_T$ to the register (where we require that there is a polynomial-time TM that on input $(1^n, 1^{T(n)})$ outputs the descriptions of $F_1, \ldots, F_T$.
3. Measure the register and let Y denote the obtained value.

The first $|f(x)|$ bits of Y must be equal to $f(x)$ with a probability greater than $\frac{2}{3}$.

BQP and QMA Classes

BQP

A language L is said to be in BQP if there exists a polynomial $p(n)$ such that L is computable in Quantum $p(n)$ time.
(L corresponds to the boolean function f(x) = 1 if $x \in L$ and f(x) = 0 otherwise.)

We can also define a quantum analog of NP.

QMA (Quantum Merlin Arthur)

A language L is said to be in QMA if there exists a poly time Quantum verifier V such that:
for $x \in L$, $\exists |\psi\rangle$ such that V accepts $|x\rangle, |\psi\rangle$ with a probability greater than $\frac{2}{3}$.
for $x \notin L$, $\forall |\psi\rangle$, V accepts $|x\rangle, |\psi\rangle$ with a probability less than $\frac{1}{3}$.

Robustness

- As before, the choice of $\frac{2}{3}$ is arbitrary, any constant greater than $\frac{1}{2}$ can be chosen. (The proof is slightly different from the classical case since unknown quantum states cannot be cloned.)
- Whether QMA is equal to the QMA_1 (one-sided error QMA) is an open question. (While the probabilistic classes MA and MA_1 have been shown to be equal.)

Relationship between classes

Known

- $P \subseteq BPP \subseteq BQP \subseteq QMA$
  i.e. Quantum computation is at least as powerful as classical probabilistic one.
- $BQP \subseteq PSPACE$ This can be proved using the sum over histories method introduced by Feynman for his Path Integral Formulation of Quantum Mechanics.

Open Questions

- $BQP \stackrel{?}{=} BPP$
  Believed to be false. If it does turn out to be false, it would invalidate the Church-Turing hypothesis.
- BQP vs NP ?

BQP $\subseteq$ PSPACE

- We want a PSPACE algorithm to calculate the probability of all the basis states after the quantum computation is completed.
- Consider an m-qubit Quantum Circuit. The basis states are $|i\rangle \in \{0,1\}^m$.
- Let the initial state be $|\psi_0\rangle$ and the gates be applied in the sequence $U_1, U_2, \ldots, U_T$. Hence the final state $|\psi_f\rangle$ can be given by:

$$|\psi_f\rangle = U_T \ldots U_2 U_1 |\psi_0\rangle$$

- The co-eficient of the $|i\rangle$ in the final state can be given by:

$$\begin{aligned} v_i &= \langle i | \psi_f \rangle \\ &= \langle i | U_T \ldots U_2 U_1 |\psi_0\rangle \end{aligned}$$

BQP $\subseteq$ PSPACE

- Using the completeness relation, we can write:

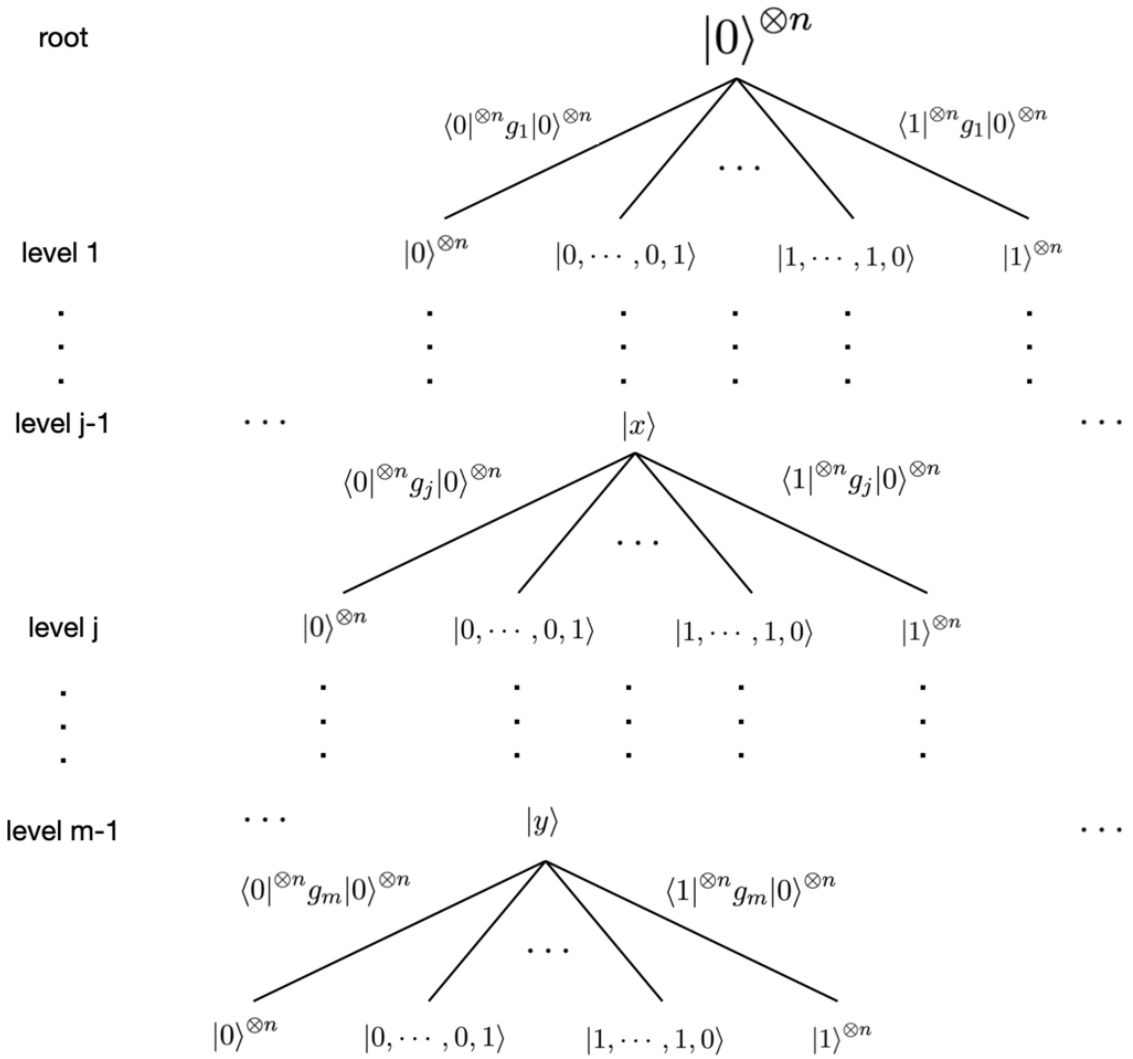
$$v_i = \langle i| U_T \left(\Sigma_{i_{T-1}} |i_{T-1}\rangle \langle i_{T-1}|\right) .. U_2 \left(\Sigma_{i_1} |i_1\rangle \langle i_1|\right) U_1 |\psi_0\rangle$$

- We can rewrite the above expression as follows to show that it can be computed in poly space:

$$v_i = \sum_{i_{T-1},\ldots,i_2,i_1} \prod_{j=1}^{T} \langle i_j| U_j |i_{j-1}\rangle$$

- $\langle i_j| U_j |i_{j-1}\rangle$ simply represents the (i_j, i_{j-1}) entry of U_j. Hence, $\prod_{j=1}^{T} \langle i_j| U_j |i_{j-1}\rangle$ can easily be computed in poly space (T(n) is a polynomial).
- Since we can reuse space, the sum of $\prod_{j=1}^{T} \langle i_j| U_j |i_{j-1}\rangle$ can also be computed in polynomial space. In terms of sum over histories, we are storing only one history of the state at a time which requires only poly space.

The sum over histories tree



Examples of problems in BQP and QMA

BQP

- Integer Factorization (Shor's Algorithm)
- Discrete Logarithm Problem (Also by Shor)
- Simulating Quantum systems

QMA

- The Local Hamiltonian Problem
  Given a k-local Hamiltonian (basically a Hermitian Matrix), find its smallest eigenvalue i.e. the ground-state energy.
- QCSAT
  Quantum version of the circuit satisfiability problem.
- Q5SAT
  Quantum analog of 3SAT.

Classical Probabilistic Computers vs Quantum Computers

"The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative."

-Richard Feynman

"Quantum mechanics is a beautiful generalization of the laws of probability: a generalization based on the 2-norm rather than the 1-norm, and on complex numbers rather than nonnegative real numbers."

-Scott Aronson

Classical Probabilistic Computers can also be defined in a way similar to the Quantum Circuit model. If we modify the Quantum Circuit model by restricting the co-effcients of the basis states to be real, non-negative, and their sum adding up to one (as opposed to the sum of their absolute value squared), the model becomes essentially equivalent to a Probabilistic Turing Machine. But those differences are precisely what seem to give Quantum Computers their power.

Thank You!

Feynman playing the Bongo Drum

References

1. Sanjeev Arora, Boaz Barak. Computational Complexity, A Modern Approach
2. Ronald de Wolf, Quantum Complexity Theory
3. Scott Aaronson. 6.845 Quantum Complexity Theory. Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, `https://ocw.mit.edu`.
4. Nick Harvey, Randomized Algorithms
5. Scott Aronson, Quantum Computing Since Democritus