COL863: Quantum Computation and Information

Ragesh Jaiswal, CSE, IIT Delhi

Quantum Computation: Complexity class BQP

- Complexity class BPP: The class of all problems (or languages) that can be solved probabilistic polynomial time. That is, a randomized algorithm that runs in time polynomial in the input length and has a bounded error probability (this can be assumed to be $1/4$).
- Exercise: Argue that $\mathsf{P} \subseteq \mathsf{BPP}$.

BQP (Bounded Quantum Polynomial)

A language is in BQP if there is a family of polynomial size quantum circuits which decides the language with probabilistic error of at most $1/4$. Also, the circuits should be uniformly generated.

- Complexity class BPP: The class of all problems (or languages) that can be solved probabilistic polynomial time. That is, a randomized algorithm that runs in time polynomial in the input length and has a bounded error probability (this can be assumed to be $1/4$).
- Exercise: Argue that $P \subseteq BPP$.

BQP (Bounded Quantum Polynomial)

A language is in BQP if there is a family of polynomial size quantum circuits which decides the language with probabilistic error of at most $1/4$. Also, the circuits should be uniformly generated.

- Exercise: Argue that $P \subseteq BPP \subseteq BQP$.

- Complexity class BPP: The class of all problems (or languages) that can be solved probabilistic polynomial time. That is, a randomized algorithm that runs in time polynomial in the input length and has a bounded error probability (this can be assumed to be $1/4$).
- Exercise: Argue that $P \subseteq BPP$.

BQP (Bounded Quantum Polynomial)

A language is in BQP if there is a family of polynomial size quantum circuits which decides the language with probabilistic error of at most $1/4$. Also, the circuits should be uniformly generated.

- Exercise: Argue that $P \subseteq BPP \subseteq BQP$.
- Complexity class PSPACE: A language is in PSPACE if there is a polynomial space Turing Machine (algorithm) that decides the language.

- Complexity class BPP: The class of all problems (or languages) that can be solved probabilistic polynomial time. That is, a randomized algorithm that runs in time polynomial in the input length and has a bounded error probability (this can be assumed to be $1/4$).
- Exercise: Argue that $P \subseteq BPP$.

BQP (Bounded Quantum Polynomial)

A language is in BQP if there is a family of polynomial size quantum circuits which decides the language with probabilistic error of at most $1/4$. Also, the circuits should be uniformly generated.

- Exercise: Argue that $P \subseteq BPP \subseteq BQP$.
- Complexity class PSPACE: A language is in PSPACE if there is a polynomial space Turing Machine (algorithm) that decides the language.

Theorem

$BQP \subseteq PSPACE$.

Theorem

BQP $\subseteq$ PSPACE.

Proof sketch

- For any language L, consider the quantum computer that decides L.
- Let the quantum circuit corresponding to inputs of length n contain $p(n)$ gates for some polynomial p.
- Suppose the quantum circuit starts in state $|0\rangle$ and uses a sequence of gates $U_1, ..., U_{p(n)}$.
- Question: Can we find the probability of this circuit ending in state $|y\rangle$ on final measurement in polynomial space?

Theorem

BQP $\subseteq$ PSPACE.

Proof sketch

- For any language L, consider the quantum computer that decides L.
- Let the quantum circuit corresponding to inputs of length n contain $p(n)$ gates for some polynomial p.
- Suppose the quantum circuit starts in state $|0\rangle$ and uses a sequence of gates $U_1, ..., U_{p(n)}$.
- Question: Can we find the probability of this circuit ending in state $|y\rangle$ on final measurement in polynomial space? Yes
  - The probability of measuring state $|y\rangle$ is modulus squared of:

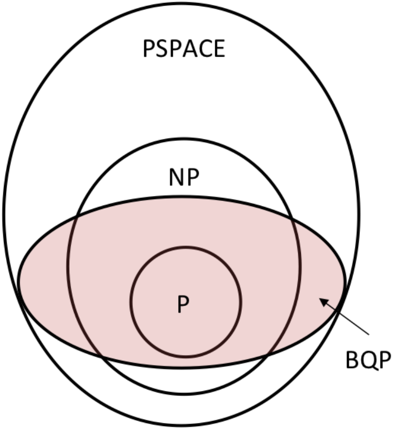
$$\langle y| U_{p(n)} ... U_1 |0\rangle .$$

  - We note that

$$\langle y| U_{p(n)} ... U_1 |0\rangle = \sum_{x_1, \ldots, x_{p(n)-1}} \langle y| U_{p(n)} \left|x_{p(n)-1}\right\rangle \left\langle x_{p(n)-1}\right| U_{p(n)-2} ... U_2 |x_1\rangle \langle x_1| U_1 |0\rangle .$$

  - Claim: The above sum can be computed in polynomial space.

- Complexity picture:



End