Quantum Complexity

Ta manaria

Structural Complexity $(\mu\Pi\lambda\forall)$

May 31, 2012

Overview

Overview

1. Preliminaries
   - Qubits

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Overview

Motivation

- Ordinary computer chips: bits are physically represented by low and high voltages on wires
- There are many other ways a bit could be stored! For example, the state of a hydrogen atom
- The single electron in this atom can either be in the ground state (the lowest energy configuration) or it can be in an excited state (a high energy configuration)
- Ground state: $|0\rangle$. Excited state: $|1\rangle$

Quantum Bit

- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.

Quantum Bit

- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.
- Qubit : $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$

Quantum Bit

- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.
- Qubit : $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
- α_0, α_1: complex numbers such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$

Quantum Bit

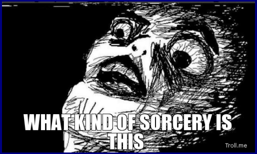
- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.
- Qubit : $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
- α_0, α_1: complex numbers such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$
- We can see a qubit as a unit length column vector in the 2-d complex space

Quantum Bit

- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.
- Qubit : $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
- α_0, α_1: complex numbers such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$
- We can see a qubit as a unit length column vector in the 2-d complex space
- For example $\frac{1}{\sqrt{5}}|0\rangle + \frac{2i}{\sqrt{5}}|1\rangle$ is a valid quantum state!

Quantum Bit

- *Superposition principle:* If a quantum state can be in one of two states, then it can be in any linear superposition of these states.
- Qubit : $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
- α_0, α_1: complex numbers such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$
- We can see a qubit as a unit length column vector in the 2-d complex space
- For example $\frac{1}{\sqrt{5}}|0\rangle + \frac{2i}{\sqrt{5}}|1\rangle$ is a valid quantum state!



Measurement

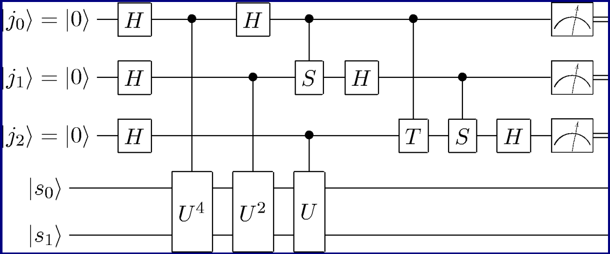
- Measurement of the qubit $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ gives 0 w.p. $\|a_0\|^2$ and 1 w.p. $\|a_1\|^2$
- Suppose two qubits: $|\phi\rangle = a_0|0\rangle + a_1|1\rangle$ and $|\psi\rangle = b_0|0\rangle + b_1|1\rangle$
- The whole state can be written as $|\phi\psi\rangle = a_0b_0|00\rangle + a_0b_1|01\rangle + a_1b_0|10\rangle + a_1b_1|11\rangle$
- Measurement of two qubits gives 00 w.p. $\|a_0b_0\|^2$, 01 w.p. $\|a_0b_1\|^2$ 10 w.p. $\|a_1b_0\|^2$ and 11 w.p. $\|a_1b_1\|^2$
- A measurement is a normalized projection onto one basis vector of the space and the probability of taking this vector is the square of the norm of this projection

- What if we have two qubits?
- Quantum state: $|\alpha\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$, such that $\Sigma_{x\in\{0,1\}^2}|a_x|^2 = 1$.
- We can see a state of 2 qubits as a unit length column vector in the 4-d complex space
- What if we have 500 qubits?
- The quantum state is a linear superposition of 2^{500} classical states! Way more than the number of elementary particles in the universe!
- Where is all this information stored?
- Can we use this to make faster computers?

Overview

Gates

- We can see Quantum gates as operators applied on one or more qubits
- Those operators are **Unitary**
- U is unitary iff $UU^{\top} = \mathbb{I}$ where $U^{\top}$ is the complex conjugate of U



Gates(2)

- **CNOT** gate:

$$|x, y\rangle \rightarrow |x, x \oplus y\rangle$$

Gates(2)

- **CNOT** gate:

$$|x, y\rangle \to |x, x \oplus y\rangle$$

- **Hadamard** gate:

$$|0\rangle \to \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$|1\rangle \to \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

$$a|0\rangle + b|1\rangle \to \frac{a+b}{\sqrt{2}}|0\rangle + \frac{a-b}{\sqrt{2}}|1\rangle$$

Properties

- Quantum gates unlike Classical gates have the same number of input and output qubits
- Quantum gates do not lose information, which means that Quantum gates...and generaly Quantum Computations are reversible
- A unitary operator preserves the length of a state and the cosine of the angle between 2 states
- So a unitary operator just rotates or mirrors the space of our states

Entanglement

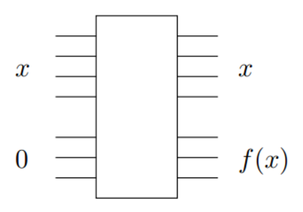
- Suppose we have the state $|\chi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$
- We cannot find states $|\phi\rangle = a_0|0\rangle + a_1|1\rangle$ and $|\psi\rangle = b_0|0\rangle + b_1|1\rangle$ such that $|\phi\rangle|\psi\rangle = |\chi\rangle$
- We say that the qubits in $|\chi\rangle$ are in entanglement
- If we measure only the first qubit we will get 0 w.p. $1/2$ and 1 w.p. $1/2$
- If we measured the first and got b then if we measure the second we will also get b immediately
- No matter the distance between the two qubits!

Parallelism

- A Quantum Computer operates in parallel
- Suppose we have the state $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$
- Let's perform the 2-qubit operator CNOT(Controlled-NOT)
- $\mathit{CNOT}|\psi\rangle = \frac{1}{\sqrt{2}}\mathit{CNOT}|00\rangle + \frac{1}{\sqrt{2}}\mathit{CNOT}|10\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$
- We performed the operator on those two states in one step!
- What a pity we don't have access to all that information $\rightarrow$ A measurement will return only 2 bits.

Computing a function

- A quantum circuit that computes a function f is a unitary operator U which takes as input:
  1. n input qubits
  2. The output qubits (in case we have a decision function we have only one output qubit) usually initialized to $|0\rangle$
- And gives as output
  1. The n qubits
  2. The answer in the output qubit
- So $|x\rangle|0\rangle \xrightarrow{U} |x\rangle|0 \oplus f(x)\rangle = |x\rangle|f(x)\rangle$
- **Measurement is always the last step of an algorithm**



- But we are in a quantum world so the input qubits can be in a superposition of many classical inputs
- And of course the output will be a superposition of all the classical outputs

Overview

Quantum Turing Machine

Definition (Quantum Turing Machine - David Deutch, 1985)

A **Quantum Turing machine** (QTM) is a 3-tuple $M = (Q, \Sigma, \delta)$, where Q is a finite set of states, Σ is the alphabet, δ is a state transition "function" and is a mapping from $Qx\Sigma$ to $Qx\Sigma x\{L, R\}xC$, where C is the set of complex numbers.

- $\delta(p, \alpha) = (q, b, d, c)$ represents the following: if M in a state p reads a symbol α (in configuration C_1), then M:
  1. writes symbol b on the square under the tape head
  2. changes the state into q
  3. moves the head on the square in the direction denoted by $d \in \{L, R\}$ (configuration C_2)
- The complex number c is called *amplitude* of this event.
- The probability that M changes its configuration from C_1 to C_2 is $|c|^2$

Overview

The Query Model

- We have an oracle for some $f : \{0,1\}^n \to \{0,1\}$ (decision problems)
- We allow our algorithm to apply arbitrary unitary transformations to its own state, as long as these are defined without reference to the values of f.
- 2 types of queries:
  1. $|x, w\rangle \to |x, w \oplus f(x)\rangle$
  2. $|x\rangle \to (-1)^{f(x)}|x\rangle$
- They can simulate each other with a single query.

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0, 1\} \to \{0, 1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0, 1\} \rightarrow \{0, 1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$
  - Apply a Hadamard $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$
  - Apply a Hadamard $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  - Apply a phase query: $|\psi\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}$

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$
  - Apply a Hadamard $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  - Apply a phase query: $|\psi\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}$
  - If $f(0) = f(1)$ ($f(0) \oplus f(1) = 0$), $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, else, $|\psi\rangle = (\pm) \cdot \frac{|0\rangle-|1\rangle}{\sqrt{2}}$

Deutsch-Jozsa Algorithm

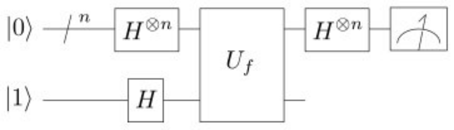
- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$
  - Apply a Hadamard $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  - Apply a phase query: $|\psi\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}$
  - If $f(0) = f(1)$ ($f(0) \oplus f(1) = 0$), $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, else, $|\psi\rangle = (\pm) \cdot \frac{|0\rangle-|1\rangle}{\sqrt{2}}$
  - Apply another Hadamard: in the first case we get $\pm|0\rangle$ and in the second case$\pm|1\rangle$

Deutsch-Jozsa Algorithm

- We are given a function $f : \{0,1\} \to \{0,1\}$ and wish to compute $f(0) \oplus f(1)$
- In the classical world we need two queries
- In the quantum world we need only one:
  - Single bit register initialized to $|0\rangle$
  - Apply a Hadamard $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  - Apply a phase query: $|\psi\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}$
  - If $f(0) = f(1)$ ($f(0) \oplus f(1) = 0$), $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, else, $|\psi\rangle = (\pm) \cdot \frac{|0\rangle-|1\rangle}{\sqrt{2}}$
  - Apply another Hadamard: in the first case we get $\pm|0\rangle$ and in the second case$\pm|1\rangle$
- Factor 2 speedup in computing the XOR of n bits

Deutsch-Jozsa Algorithm(2)

- General version: we are given a function $f : \{0,1\}^n \rightarrow \{0,1\}$, which is either constant or balanced.
- In the classical world we need (worst case) $2^{n-1} + 1$ queries
- In the quantum world, a generalization of the previous algorithm can solve the problem with 1 query!



Overview

Overview

Exact Quantum Polynomial Time

Definition (EQP)

EQP is the class of languages $L \subseteq (0,1)^*$, decidable with zero error probability by a uniform family of polynomial-size quantum circuits over some universal family of gates.

- Quantum analogue of P

Bounded Error Quantum Polynomial Time

Definition (BQP)

BQP is the class of languages $L \subseteq (0,1)^*$, decidable with bounded error probability (say $\frac{1}{3}$) by a uniform family of polynomial-size quantum circuits over some universal family of gates.

- Quantum analogue of BPP
- Factoring, DLP $\in$ BQP

Some trivial bounds

- EQP $\subseteq$ BQP

Some trivial bounds

- $\mathrm{EQP} \subseteq \mathrm{BQP}$
- $\mathrm{P} \subseteq \mathrm{EQP}$

Some trivial bounds

- $\mathrm{EQP} \subseteq \mathrm{BQP}$
- $\mathrm{P} \subseteq \mathrm{EQP}$
  - A classical circuit can be simulated by a Quantum Circuit

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)
- BPP $\subseteq$ BQP

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)
- BPP $\subseteq$ BQP
- Quantum property gives us randomness

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)
- BPP $\subseteq$ BQP
- Quantum property gives us randomness
  - Just apply a Hadamard gate on an ancilla qubit initialized to the state $|0\rangle$

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)
- BPP $\subseteq$ BQP
- Quantum property gives us randomness
  - Just apply a Hadamard gate on an ancilla qubit initialized to the state $|0\rangle$
  - H$|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Some trivial bounds

- EQP $\subseteq$ BQP
- P $\subseteq$ EQP
  - A classical circuit can be simulated by a Quantum Circuit
  - We just need to simulate the fundamental gates (for example NAND gate)
- BPP $\subseteq$ BQP
- Quantum property gives us randomness
  - Just apply a Hadamard gate on an ancilla qubit initialized to the state $|0\rangle$
  - H$|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- $\checkmark$So, a Quantum Computer is a least as powerful as a Classical Computer

Overview

BQP vs EXP

- BQP $\subseteq$ EXP

BQP vs EXP

- BQP $\subseteq$ EXP
- A classical computer can simulate the whole evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$

BQP vs EXP

- BQP $\subseteq$ EXP
- A classical computer can simulate the whole evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$
- So, a Quantum Computer can provide at most an exponential advantage over classical computers

BQP vs EXP

- BQP $\subseteq$ EXP
- A classical computer can simulate the whole evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$
- So, a Quantum Computer can provide at most an exponential advantage over classical computers
- But is that accurate?

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.
- But we just need the amplitudes of the accepting states

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.
- But we just need the amplitudes of the accepting states
- Let S be the set of all accepting states

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.
- But we just need the amplitudes of the accepting states
- Let S be the set of all accepting states
- Let α_x be the amplitude of the state $|x\rangle \in S$

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.
- But we just need the amplitudes of the accepting states
- Let S be the set of all accepting states
- Let α_x be the amplitude of the state $|x\rangle \in S$
- We can find α_x by looping over all computational paths that contribute amplitude to $|x\rangle$. This requires only polynomial space.

BQP vs PSPACE [Bernstein, Vazirani - 93], [Feynmann's path integral]

- BQP $\subseteq$ PSPACE
- At first it seems that we need an exponential space to simulate the evolution of the state vector $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$.
- But we just need the amplitudes of the accepting states
- Let S be the set of all accepting states
- Let α_x be the amplitude of the state $|x\rangle \in S$
- We can find α_x by looping over all computational paths that contribute amplitude to $|x\rangle$. This requires only polynomial space.
- Then we sum the probabilities of every $|x\rangle$ to take the total accepting probability.

BQP vs PP [Adleman, DeMarrais, Huang - 97]

- BQP $\subseteq$ PP

BQP vs PP [Adleman, DeMarrais, Huang - 97]

- BQP $\subseteq$ PP
- A **PP** problem involves summing up exponentially many terms and then deciding whether the sum is greater or less than some threshold, which is exactly what the Feynman Path Integral does.

BQP vs PP [Adleman, DeMarrais, Huang - 97]

- BQP $\subseteq$ PP
- A **PP** problem involves summing up exponentially many terms and then deciding whether the sum is greater or less than some threshold, which is exactly what the Feynman Path Integral does.
- $P_{accept} = \Sigma_{x \in S} |\Sigma_i a_{x,i}|^2$. This is the sum of exponentially many terms, each of which is computable in P! So we can decide in PP whether $P_{accept} \leqslant \frac{1}{3}$ or $P_{accept} \geqslant \frac{2}{3}$

BQP vs PP [Adleman, DeMarrais, Huang - 97]

- BQP $\subseteq$ PP
- A **PP** problem involves summing up exponentially many terms and then deciding whether the sum is greater or less than some threshold, which is exactly what the Feynman Path Integral does.
- $P_{accept} = \Sigma_{x \in S} |\Sigma_i a_{x,i}|^2$. This is the sum of exponentially many terms, each of which is computable in P! So we can decide in PP whether $P_{accept} \leqslant \frac{1}{3}$ or $P_{accept} \geqslant \frac{2}{3}$
- BQP is in fact low for PP, meaning that a PP machine achieves no benefit from being able to solve BQP problems instantly.

Overview

BQP is low for itself

- $BQP^{BQP} = BQP$

BQP is low for itself

- $BQP^{BQP} = BQP$
- Informally, this is true because polynomial time algorithms are closed under composition

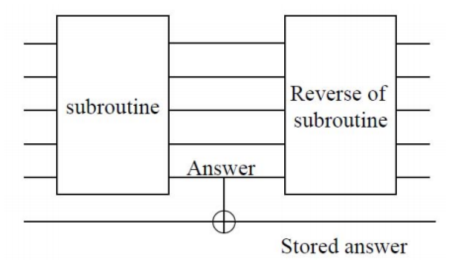
BQP is low for itself

- $BQP^{BQP} = BQP$
- Informally, this is true because polynomial time algorithms are closed under composition
- Obstacle for proving this for BQP: Entanglement (garbage)! The answer of the subroutine depends on its working qubits

BQP is low for itself

- $BQP^{BQP} = BQP$
- Informally, this is true because polynomial time algorithms are closed under composition
- Obstacle for proving this for BQP: Entanglement (garbage)! The answer of the subroutine depends on its working qubits
- Charles Bennett proposed a smart trick: Uncomputing

Uncomputing



1. Run the subroutine
2. Copy the answer qubit to a separate location
3. Run the subroutine backwards

Overview

Reminder: MA

Definition (MA)

The class of decision problems solvable by a Merlin-Arthur protocol: Merlin (unbounded computational resources) sends Arthur a polynomial-size purported proof that the answer to the problem is "yes". Arthur must verify the proof in BPP so that:

- If the answer is "yes", then there exists a proof such that Arthur accepts w.p. at least 2/3.
- If the answer is "no", then for all proofs Arthur accepts w.p. at most 1/3.

- AM (AM[2]) is the same thing, but this time Arthur goes first and the Merlin answers
- AM[k] = AM[2]

QMA, QCMA

Definition (QMA)

QMA is the class of languages $L \subseteq (0,1)^*$, for which there is a polynomial size quantum circuit A such that $\forall x$

- if $x \in L$ then there is a quantum witness $|w\rangle$ such that $A(x, |w\rangle)$ accepts with probability at least $\frac{2}{3}$
- if $x \notin L$ then for all quantum witnesses $|w\rangle$, $A(x, |w\rangle)$ accepts with probability at most $\frac{1}{3}$

- QMA is the quantum analogue of MA
- **QCMA** stands for: *Quantum Classical Merlin Arthur*.
- In QCMA the witness should be a classical string

Some Bounds

- $MA \subseteq QCMA$

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$
- $BQP \subseteq QCMA$

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$
- $BQP \subseteq QCMA$
- $QMA \subseteq PP$

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$
- $BQP \subseteq QCMA$
- $QMA \subseteq PP$
- We don't know if $QMA \neq QCMA$ (This would imply that $P \neq PSPACE$)

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$
- $BQP \subseteq QCMA$
- $QMA \subseteq PP$
- We don't know if $QMA \neq QCMA$ (This would imply that $P \neq PSPACE$)
- We don't know if there exists an oracle A s.t. $QCMA^A \neq QMA^A$

Some Bounds

- $MA \subseteq QCMA$
- $QCMA \subseteq QMA$
- $BQP \subseteq QCMA$
- $QMA \subseteq PP$
- We don't know if $QMA \neq QCMA$ (This would imply that $P \neq PSPACE$)
- We don't know if there exists an oracle A s.t. $QCMA^A \neq QMA^A$

Quantum Oracle Separation [Aaronson, Kuperberg]

There is a **quantum oracle** A (that is a black box unitary transformation) such that $QCMA^A \neq QMA^A$

Reminder

- **IP**: The class of languages $L \subseteq \{0,1\}$ for which there exists an interaction protocol between *BPP* verifier and an omnipotent prover s.t. $\forall x$:
  1. $x \in L \Rightarrow \exists$ a prover strategy that causes verifier to accept with probability $\geqslant \frac{2}{3}$
  2. $x \notin L \Rightarrow \forall$ prover strategies, verifier accepts with probability $\leqslant \frac{1}{3}$
- **IP = PSPACE** (Shamir)

Quantum Interactive proofs

- The proover and verifier can exchange quantum messages, and are limited by the laws of quantum physics. The number of gates is polynomial.
- **QIP**: The class of languages $L \subseteq \{0,1\}$ for which there exists an interaction protocol between *BQP* verifier (Arthur) and an omnipotent prover (Merlin) s.t. $\forall x$:
  1. If $x \in L$ then the prover can behave in such a way that the verifier accepts with probability at least $\frac{2}{3}$
  2. If $x \notin L$ then however the prover behaves, the verifier rejects with probability at least $\frac{2}{3}$

Quantum Interactive proofs

Theorem (Kitaev,Watrous - 2003)

Any QIP protocol can be made three-round. In other words, all QIP rounds are given by QIP$(1) =$ QMA, QAM $\subseteq$ QIP(2), and QIP$(3) =$ QIP.

Theorem (Jain,Ji,Upadhyay,Watrous - 2009)

QIP = IP = PSPACE

Overview

Overview

BPP $\stackrel{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?

BPP $\overset{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?
- This would imply that $P \neq$ PSPACE

BPP $\overset{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?
- This would imply that $P \neq$ PSPACE
- Simon's algorithm is an evidence

BPP $\overset{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?
- This would imply that $P \neq$ PSPACE
- Simon's algorithm is an evidence
- Problem: Given $f : \{0,1\}^n \rightarrow \{0,1\}^n$ s.t. $\forall x \neq y, f(x) = f(y)$ iff $x \oplus y = s$. Find s.

BPP $\overset{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?
- This would imply that $P \neq$ PSPACE
- Simon's algorithm is an evidence
- Problem: Given $f : \{0,1\}^n \to \{0,1\}^n$ s.t. $\forall x \neq y, f(x) = f(y)$ iff $x \oplus y = s$. Find s.
- Classically we need $2^{\frac{n}{2}}$ queries but Simon's algorithm needs only n queries

BPP $\overset{?}{\neq}$ BQP

- In other words: is a Quantum Computer more poweful than it's Classical counterpart?
- This would imply that $P \neq$ PSPACE
- Simon's algorithm is an evidence
- Problem: Given $f : \{0,1\}^n \rightarrow \{0,1\}^n$ s.t. $\forall x \neq y, f(x) = f(y)$ iff $x \oplus y = s$. Find s.
- Classically we need $2^{\frac{n}{2}}$ queries but Simon's algorithm needs only n queries
- It proves that there exists an oracle relative to which BPP $\neq$ BQP

Where NP sits? [Grover's Algorithm]

- Why not try every possible solution in parallel and then pick the correct one?

Where NP sits? [Grover's Algorithm]

- Why not try every possible solution in parallel and then pick the correct one?
- It has not been proved that NP $\nsubseteq$ BQP

Where NP sits? [Grover's Algorithm]

- Why not try every possible solution in parallel and then pick the correct one?
- It has not been proved that NP $\nsubseteq$ BQP
- Classically we need on average 2^{n-1} queries to find a valid solution over a space of 2^n possible solutions

Where NP sits? [Grover's Algorithm]

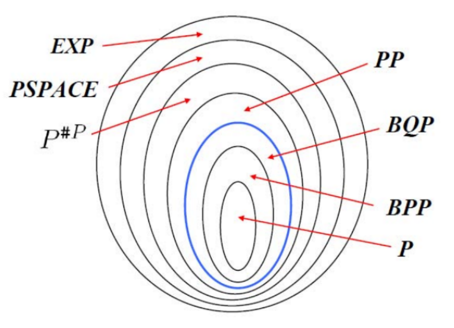
- Why not try every possible solution in parallel and then pick the correct one?
- It has not been proved that NP $\nsubseteq$ BQP
- Classically we need on average 2^{n-1} queries to find a valid solution over a space of 2^n possible solutions
- Quantumly, Grover's algorithm needs only $2^{\frac{n}{2}}$ on average

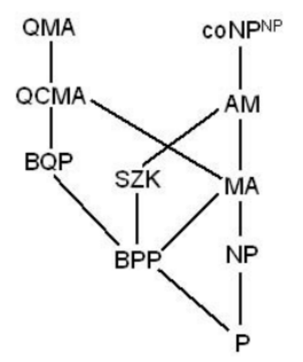
Where NP sits? [Grover's Algorithm]

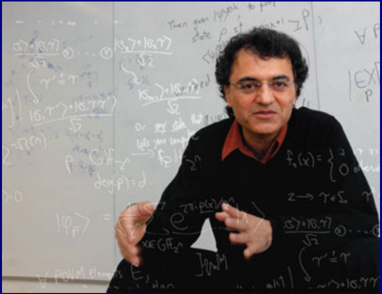
- Why not try every possible solution in parallel and then pick the correct one?
- It has not been proved that NP $\nsubseteq$ BQP
- Classically we need on average 2^{n-1} queries to find a valid solution over a space of 2^n possible solutions
- Quantumly, Grover's algorithm needs only $2^{\frac{n}{2}}$ on average
- Quantum Computers give quadratic (not exponential) speedup!

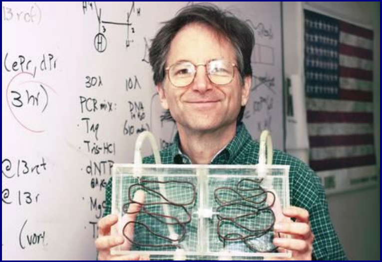
Overview

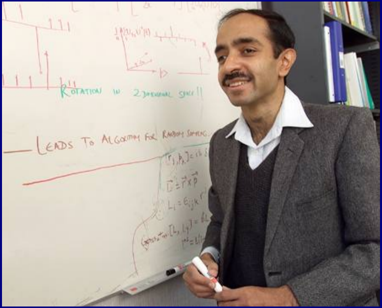
Quantum Complexity Relations

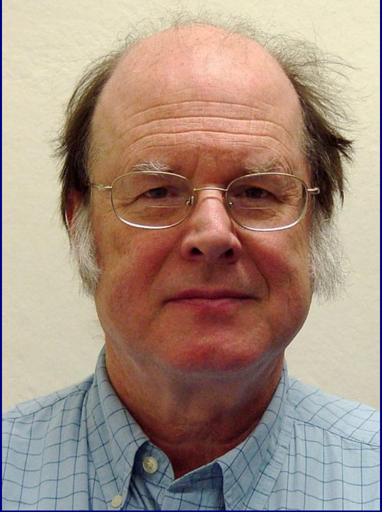







PCR mix:
Tris-HCl


LEADS TO



References

- Quantum Complexity Theory, Vazirani - Bernstein
- An Introduction to Quantum Complexity, Tetsuro Nishino
- An Introduction to Quantum Complexity Theory, Richard Cleve
- Quantum Computational Complexity, John Watrous
- Lecture Notes On Quantum Complexity- MIT, Scott Aaronson
- Lecture Notes on Quantum Computations, Iordanis Kerenidis
- Algorithms, Papadimitriou - Vazirani