An introduction to Quantum Complexity

Peli Teloni

Advanced Topics on Algorithms and Complexity
$\mu\Pi\lambda\forall$

July 3, 2014

Outline

Outline

ECT Thesis

Goal of computational complexity:

classify problems according to *amount* of resources needed for solving them

ECT Thesis

Goal of computational complexity:

classify problems according to *amount* of resources needed for solving them

Why is this quantity well-defined?

ECT Thesis

Goal of computational complexity:
classify problems according to *amount* of resources needed for solving them

Why is this quantity well-defined?

Extended Church Turing (ECT) Thesis

Any "reasonable" model of computation can be efficiently simulated on a probabilistic Turing Machine or random access machine.

ECT Thesis

Goal of computational complexity:
classify problems according to *amount* of resources needed for solving them

Why is this quantity well-defined?

Extended Church Turing (ECT) Thesis

Any "reasonable" model of computation can be efficiently simulated on a probabilistic Turing Machine or random access machine.

However, there is evidence that ECT doesn't hold for the quantum world.
Why?
Turing Machine is based on a classical physics model of the Universe, whereas current physical theory asserts that the Universe is quantum physical.

Evidence and Meaning

Some evidence:

- **Feynman '82:** it's not clear how to simulate a quantum system on a computer without exponential penalty
- **Bernstein & Vazirani '97:** relative to an oracle, quantum poly-time properly contains probabilistic poly-time
- **Simon '97:** relative to an oracle, quantum poly-time is not contained in subexponential probabilistic time
- **Shor '97:** prime factorization and discrete logarithms solved in poly-time on a quantum computer
- **Kerenidis & Zhang '13:** players achieve correlated Nash Equilibrium unconditionally, if quantum communication is enabled

Evidence and Meaning

Some evidence:

- **Feynman '82:** it's not clear how to simulate a quantum system on a computer without exponential penalty
- **Bernstein & Vazirani '97:** relative to an oracle, quantum poly-time properly contains probabilistic poly-time
- **Simon '97:** relative to an oracle, quantum poly-time is not contained in subexponential probabilistic time
- **Shor '97:** prime factorization and discrete logarithms solved in poly-time on a quantum computer
- **Kerenidis & Zhang '13:** players achieve correlated Nash Equilibrium unconditionally, if quantum communication is enabled

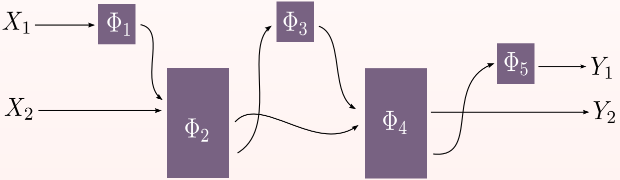
So, one of the following must hold:

- ECT thesis is false
- Quantum Physics is false
- Our picture of computational complexity theory is false

Outline

Quantum Circuit Model

A **quantum circuit** is an acyclic network of quantum gates connected by qubit wires. For example:



- convenient model when study the complexity of quantum computation
- acyclic to preserve time ordering of things
- introduced by Deutsch in '85

Qubit: intuition

Qubit is the basic unit of quantum information. Some math intuition:

An event with n possible outcomes is a vector in $\mathbb{R}^n$: $v = (p_1, \ldots, p_n)$

Qubit: intuition

Qubit is the basic unit of quantum information. Some math intuition:

An event with n possible outcomes is a vector in $\mathbb{R}^n$: $v = (p_1, \ldots, p_n)$

- $p_i \geq 0$
- $\sum p_i = 1 \Rightarrow \|v\|_1 = 1$
- e.g. bit can be seen as the vector $(p, 1-p)$
- operation: stochastic matrix

Qubit: intuition

Qubit is the basic unit of quantum information. Some math intuition:

An event with n possible outcomes is a vector in $\mathbb{R}^n$: $v = (p_1, \ldots, p_n)$

- $p_i \geq 0$
- $\sum p_i = 1 \Rightarrow \|v\|_1 = 1$
- e.g. bit can be seen as the vector $(p, 1-p)$
- operation: stochastic matrix

why not use 2-norm?

Qubit: intuition

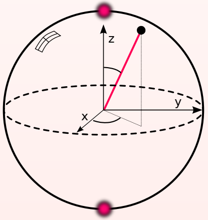
Qubit is the basic unit of quantum information. Some math intuition:

An event with n possible outcomes is a vector in $\mathbb{R}^n$: $v = (p_1, \ldots, p_n)$

- $p_i \geq 0$
- $\sum p_i = 1 \Rightarrow \|v\|_1 = 1$
- e.g. bit can be seen as the vector $(p, 1-p)$
- operation: stochastic matrix

why not use 2-norm?

- vector $v' = (a, b)$ where $a, b \in \mathbb{C}$
- $\|v'\|_2 = 1 \Rightarrow a^2 + b^2 = 1$
- operation: unitary matrix $(U^H U = I)$



Qubit

Qubit is a 2D quantum system in Hilbert Space C^2

- **basis** of C^2: $(0,1)$ and $(1,0)$
- **state** of qubit: vector in C^2
- **Dirac notation:** $\psi = (a,b) \Longrightarrow |\psi\rangle = a|0\rangle + b|1\rangle$

Properties of qubits:

- Normalization: $|a|^2 + |b|^2 = 1 = \langle\psi|\psi\rangle$
- Superposition: linear combination
- Measurement: state collapses irreversibly to one of the basis states
- Non-Clonability: cannot copy unknown quantum state
- Entanglement: see in a while

Physical implementation:

- electron spin
- photon polarization etc.

2 Qubits

- space now is $C^2 \otimes C^2$
- 4 basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
- 2-qubit state: $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$
- $\sum_{x \in \{0,1\}^2} |a_x|^2 = 1$

2 Qubits

- space now is $C^2 \otimes C^2$
- 4 basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
- 2-qubit state: $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$
- $\sum\limits_{x \in \{0,1\}^2} |a_x|^2 = 1$
- Measurement of 1st qubit gives 0 w.p. $p_0 = |a_{00}|^2 + |a_{01}|^2$
- If 1st qubit is 0 then system collapses to $|\psi'\rangle = \frac{a_{00}|00\rangle + a_{01}|01\rangle}{\sqrt{p_0}}$

2 Qubits

- space now is $C^2 \otimes C^2$
- 4 basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
- 2-qubit state: $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$
- $\sum_{x \in \{0,1\}^2} |a_x|^2 = 1$
- Measurement of 1st qubit gives 0 w.p. $p_0 = |a_{00}|^2 + |a_{01}|^2$
- If 1st qubit is 0 then system collapses to $|\psi'\rangle = \frac{a_{00}|00\rangle + a_{01}|01\rangle}{\sqrt{p_0}}$
- what if $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ (Bell state - EPR pair)?
- 2nd measurement gives the same with 1st -- maximally entangled state
- Entanglement: perfect (anti) correlation

2 Qubits

- space now is $C^2 \otimes C^2$
- 4 basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
- 2-qubit state: $|\psi\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$
- $\sum_{x \in \{0,1\}^2} |a_x|^2 = 1$
- Measurement of 1st qubit gives 0 w.p. $p_0 = |a_{00}|^2 + |a_{01}|^2$
- If 1st qubit is 0 then system collapses to $|\psi'\rangle = \frac{a_{00}|00\rangle + a_{01}|01\rangle}{\sqrt{p_0}}$
- what if $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ (Bell state - EPR pair)?
- 2nd measurement gives the same with 1st -- maximally entangled state
- Entanglement: perfect (anti) correlation
- n-qubit state is a linear superposition of 2^n basis states
- Huge computational power of quantum computers!

Quantum Gates

- quantum operations: unitary matrices
- search for a pattern in superposition
- rotate Hilbert space
- same number of input and output qubits
- reversible: no info is lost
- can simulate classical logic gates

Quantum Gates

- quantum operations: unitary matrices
- search for a pattern in superposition
- rotate Hilbert space
- same number of input and output qubits
- reversible: no info is lost
- can simulate classical logic gates

Need for Universal gate set

- to compare with other models
- approximate any unitary operation with arbitrary accuracy

Quantum Gates

- quantum operations: unitary matrices
- search for a pattern in superposition
- rotate Hilbert space
- same number of input and output qubits
- reversible: no info is lost
- can simulate classical logic gates

Need for Universal gate set

- to compare with other models
- approximate any unitary operation with arbitrary accuracy

Solovay-Kitaev theorem

Informally: any universal gate set can be simulated by another universal gate set with only a polynomial increase of gates.

Universal gate set

Hadamard Gate

$$|a\rangle \;—\boxed{H}—\; H\,|a\rangle$$

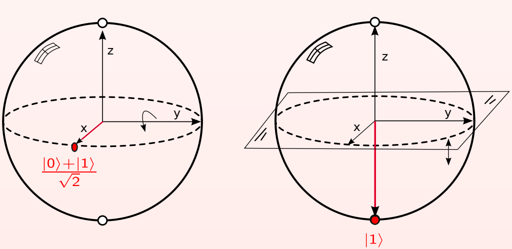
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \Longrightarrow H|a\rangle = \frac{|0\rangle + (-1)^a |1\rangle}{\sqrt{2}}$$

Universal gate set

Hadamard Gate

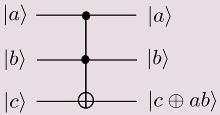
$$|a\rangle \;-\!\!\!-\; \boxed{H} \;-\!\!\!-\; H\,|a\rangle$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \Longrightarrow H|a\rangle = \frac{|0\rangle + (-1)^a|1\rangle}{\sqrt{2}}$$



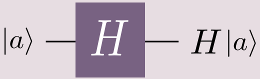
Universal gate set

Toffoli Gate



flips qubit c if a,b are 1

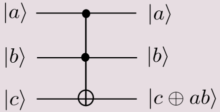
Hadamard Gate



$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \Longrightarrow H|a\rangle = \frac{|0\rangle + (-1)^a|1\rangle}{\sqrt{2}}$$

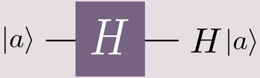
Universal gate set

Toffoli Gate



flips qubit c if a,b are 1

Hadamard Gate



$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \Longrightarrow H|a\rangle = \frac{|0\rangle + (-1)^a |1\rangle}{\sqrt{2}}$$

- proved to be quantum universal by Shi, '02
- real entries -- how approximate complex unitary matrices?
- no strict but computational universality
- can be used for fault tolerant purposes

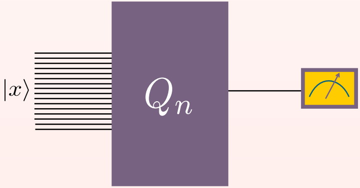
Poly-time Quantum Algorithms

Definition

In the quantum circuit model, a **quantum algorithm** Q is described by a family of quantum circuits

$$Q = \{Q_n : n \in \mathbb{N}\}$$

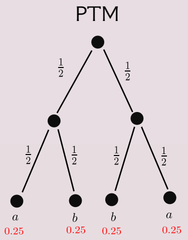
- We require that such a family is poly-time uniform
- To run this algorithm on input $x \in \{0,1\}^n$ we apply Q_n to $|x\rangle$ and measure the output in the standard basis:



- $Q(x)$ denotes the outcome, which is a random variable in general.

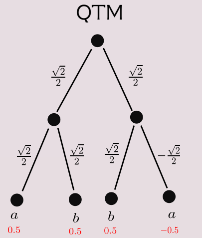
QTM: informal

- Reminder: internal state of PTM changes in a probabilistic way
- **description** of configurations: a probability vector $\vec{p}$
- step of computation: $M \cdot \vec{p} = \vec{q}$ where M is a stochastic matrix.
- QTM is the same
- just change M to be unitary and $\vec{p}$ to be 2-norm unit vector



$\Pr(a) = 0.5$

$\Pr(b) = 0.5$



$$\Pr(a) = |\sum_{i:a} c_i|^2 = |\tfrac{1}{2} - \tfrac{1}{2}|^2 = 0$$

$$\Pr(b) = |\sum_{i:b} c_i|^2 = |\tfrac{1}{2} + \tfrac{1}{2}|^2 = 1$$

QTM: formal

Quantum Turing Machine [Deutsch, '85]

A QTM is defined by a triplet (Σ, Q, δ), where Σ is the alphabet, Q is a finite set of states and δ is the quantum transition function

$$\delta : Q \times \Sigma \longrightarrow \tilde{C}^{\Sigma \times Q \times D}$$

with $D = \{L, R\}$ and $\tilde{C}$ the set of "efficiently computable" complex numbers.

- each state of QTM is a linear combination $\sum_c a_c|c\rangle$ of all classical configurations $c = |a, q, m\rangle$ (tape content, state, head position)
- $\delta(p, \sigma)$ gives a superposition of all possible (finite) configs which the machine will take when in state p reading a σ.
- so δ is like a unitary matrix

Query Complexity Model

- Almost all quantum algorithms operate in the **query complexity model**.
- In this model, input is not a bit-string but a "black box" computing some function $f : \{0,1\}^n \to \{0,1\}$ which returns $f(x)$ when x is passed in.
    - put it in quantum words:

Query Complexity Model

- Almost all quantum algorithms operate in the **query complexity model**.
- In this model, input is not a bit-string but a "black box" computing some function $f : \{0,1\}^n \to \{0,1\}$ which returns $f(x)$ when x is passed in.
  - put it in quantum words: we have access to a unitary oracle $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ where $f(x) \in \{0,1\}$ y is the *target* bit
  - one call to U_f is called a **query**

Query Complexity Model

- Almost all quantum algorithms operate in the **query complexity model**.
- In this model, input is not a bit-string but a "black box" computing some function $f : \{0,1\}^n \to \{0,1\}$ which returns $f(x)$ when x is passed in.
  - put it in quantum words: we have access to a unitary oracle $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ where $f(x) \in \{0,1\}$ y is the *target* bit
  - one call to U_f is called a **query**
  - another type of query that puts the output variable in the phase of the state: $U_{f,\pm} : |x\rangle \to (-1)^{f(x)}|x\rangle$ just set target bit to $H|1\rangle$
  - both types of queries simulate each other with only one query
  - goal: compute some property of f using the minimum worst case number of queries

Query Complexity Model

- Almost all quantum algorithms operate in the **query complexity model**.
- In this model, input is not a bit-string but a "black box" computing some function $f : \{0,1\}^n \to \{0,1\}$ which returns $f(x)$ when x is passed in.
  - put it in quantum words: we have access to a unitary oracle $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ where $f(x) \in \{0,1\}$ y is the *target* bit
  - one call to U_f is called a **query**
  - another type of query that puts the output variable in the phase of the state: $U_{f,\pm} : |x\rangle \to (-1)^{f(x)}|x\rangle$ just set target bit to $H|1\rangle$
  - both types of queries simulate each other with only one query
  - goal: compute some property of f using the minimum worst case number of queries
- Algorithm can also apply arbitrary unitary transformations as long as values of f are not involved in their definitions.
- Pros: if there is a circuit simulating U_f just plug it in and return to computational complexity model.
- Cons: quantum-classical separations are relative to an oracle.

Deutsch's Algorithm

- initially proposed by David Deutsch in '85 - improved by Cleve, Ekert, Macchiavello, and Mosca in '92
- combines quantum parallelism with *interference*

Deutsch's Problem

given $f : \{0,1\} \to \{0,1\}$ we wish to compute $f(0) \oplus f(1)$

- classical query complexity is 2
- quantum query complexity is 1

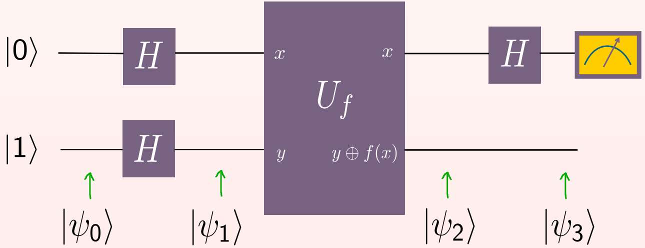
Deutsch's Algorithm

- initially proposed by David Deutsch in '85 - improved by Cleve, Ekert, Macchiavello, and Mosca in '92
- combines quantum parallelism with *interference*

Deutsch's Problem

given $f : \{0,1\} \to \{0,1\}$ we wish to compute $f(0) \oplus f(1)$

- classical query complexity is 2
- quantum query complexity is 1



Analysis of Deutsch's Algorithm

- initialization: $|\psi_0\rangle = |0\rangle|1\rangle$

Analysis of Deutsch's Algorithm

- initialization: $|\psi_0\rangle = |0\rangle|1\rangle$
- unpack: $|\psi_1\rangle = H|0\rangle H|1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$

Analysis of Deutsch's Algorithm

- initialization: $|\psi_0\rangle = |0\rangle|1\rangle$
- unpack: $|\psi_1\rangle = H|0\rangle H|1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$
- quantum parallelism:
  - Observe that: $U_f|x\rangle H|1\rangle = (-1)^{f(x)}|x\rangle H|1\rangle$ (remember phase oracle)
  - So: $U_f|\psi_1\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}} H|1\rangle$
  - Therefore: $|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] H|1\rangle & \text{if } f(0) = f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] H|1\rangle & \text{if } f(0) \neq f(1) \end{cases}$

Analysis of Deutsch's Algorithm

- initialization: $|\psi_0\rangle = |0\rangle|1\rangle$
- unpack: $|\psi_1\rangle = H|0\rangle H|1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$
- quantum parallelism:
  - Observe that: $U_f|x\rangle H|1\rangle = (-1)^{f(x)}|x\rangle H|1\rangle$ (remember phase oracle)
  - So: $U_f|\psi_1\rangle = \frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}} H|1\rangle$
  - Therefore: $|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] H|1\rangle & \text{if } f(0)=f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] H|1\rangle & \text{if } f(0)\neq f(1) \end{cases}$
- interference: just apply Hadamard gate to first qubit
  - $|\psi_3\rangle = \begin{cases} \pm|0\rangle H|1\rangle & \text{if } f(0)=f(1) \\ \pm|1\rangle H|1\rangle & \text{if } f(0)\neq f(1) \end{cases}$
  - notice that if $f(0)=f(1)$ then $f(0)\oplus f(1) = 0$
  - finally: $|\psi_3\rangle = |f(0)\oplus f(1)\rangle\, H|1\rangle \Rightarrow$ just measure first qubit!

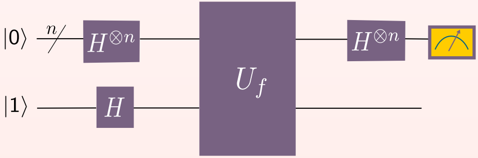
Some query complexity separations (1)

- We've seen only a 2-speedup factor in computing the XOR of n qubits
- Is there a bigger quantum-classical gap?

Deutsch-Jozsa Problem

We have a function $f : \{0,1\}^n \to \{0,1\}$ which is either constant or balanced (0 for half the inputs, 1 for the other half). The goal is to find out what it is.

- in classical world, we need $2^{n-1}+1$ queries (error prob. is not allowed)
- in quantum world, a generalization of prev. algorithm uses only 1 query



Some query complexity separations (2)

Simon's Problem

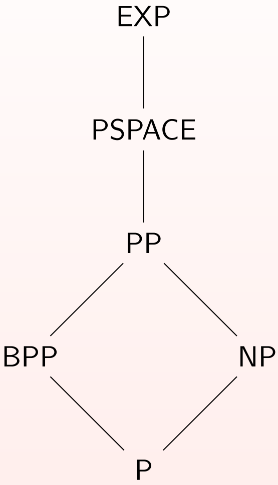
We have a function $f:\{0,1\}^n \to \{0,1\}^n$ and we are promised that there exists a "secret XOR mask" $s \in \{0,1\}^n$ s.t. $f(x) = f(y) \Leftrightarrow y = x \oplus s$ for all distinct (x,y) pairs. The goal is to find out the identity of s.

- Deutsch's Problem is a special case for $n = 1$.
- Classically, we know that any algorithm in the query model (even with error probability at most ϵ) will make $\Omega(\sqrt{2^n \log \frac{1}{\epsilon}})$ queries.
- Quantumly, it can be solved with $O(n \log \frac{1}{\epsilon})$ queries.

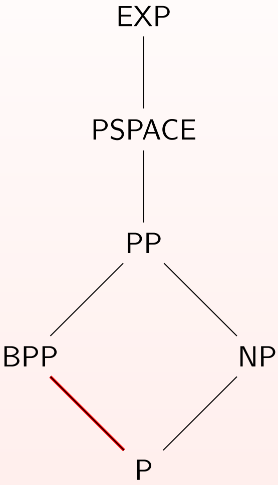
So, in the query complexity model, there are quantum algorithms which *do* achieve an exponential separation between quantum and classical.

Outline

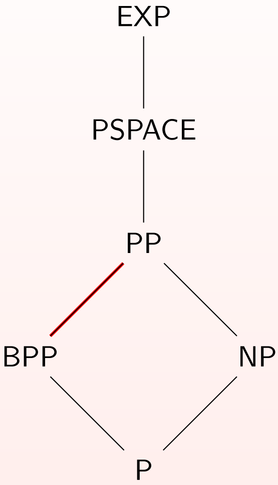
Recap



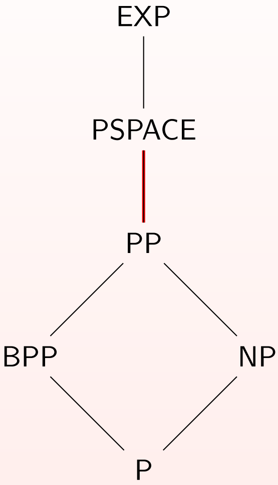
Recap



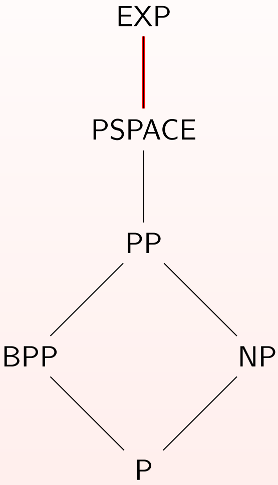
Recap



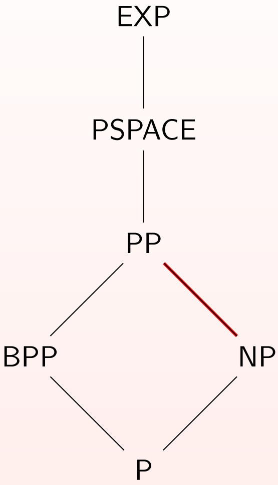
Recap



Recap



Recap



BQP

- One of the fundamental classes in quantum complexity.
- It describes what we can efficiently solve with a quantum computer.

Definition

BQP: is the class containing all languages $L \subset \{0,1\}^*$ for which there exists a poly-time uniform family $Q = \{Q_n : n \in \mathbb{N}\}$ of quantum circuits s.t. for all inputs x it holds that:

$$x \in L \Rightarrow \Pr[Q(x) = 1] \geq 2/3$$

$$x \notin L \Rightarrow \Pr[Q(x) = 0] \geq 2/3$$

- Error reduction: just like BPP, repeat computation and take majority vote
- Assumption: circuits use gates form a universal gate set
- Auxiliary qubits are bounded by some polynomial q:

$$Q(x) = Q(|x\rangle|0\rangle^{\otimes q(n)})$$

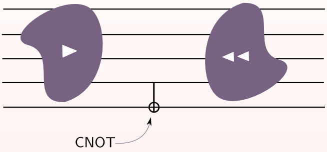
Some Structural Properties of BQP

1. BQP is closed under complement
2. BQP is closed under intersection (and union)
3. BQP is low for itself, meaning $\mathrm{BQP}^{\mathrm{BQP}} = \mathrm{BQP}$

- If you can't prove 1. and 2. by now, then I completely failed to attract your interest ☹
- The proof about 3. is like that of BPP with one exception:
  - when a quantum algo terminates, we measure only the output qubit
  - all other qubits are considered as garbage
  - so when we replace BQP oracle with a BQP subroutine, we have some subroutine garbage left
  - in case of pure states, we just throw them away
  - but in case of mixed states, they may annoy interference
  - what can we do to avoid this?

Uncomputing

1. play subroutine
2. copy answer qubit to separate location
3. rewind subroutine



- solution proposed by Bennett in the '80s
- quantum mechanics cleans its mess
- if subroutine has some error probability, it won't erase everything
  - solution: apply probability amplification in the subroutine part

BPP $\subseteq$ BQP

How can we simulate randomness?

BPP $\subseteq$ BQP

How can we simulate randomness?

- whenever a BQP machine wants to flip a coin, just apply a Hadamard gate on input $|0\rangle$ and you'll have a random source for 0 and 1.

How can we simulate a classical circuit with a quantum one?

BPP $\subseteq$ BQP

How can we simulate randomness?

- whenever a BQP machine wants to flip a coin, just apply a Hadamard gate on input $|0\rangle$ and you'll have a random source for 0 and 1.

How can we simulate a classical circuit with a quantum one?

- make classical logic gates reversible: e.g. a Toffoli gate can simulate a NAND gate, which is universal in the classical set
- By **Solovay-Kitaev** theorem, with a universal quantum gate set we can approximate efficiently any other unitary transformation: simulating arbitrary gates up to exponentially small error, costs only a polynomial overhead

BPP $\subseteq$ BQP

How can we simulate randomness?

- whenever a BQP machine wants to flip a coin, just apply a Hadamard gate on input $|0\rangle$ and you'll have a random source for 0 and 1.

How can we simulate a classical circuit with a quantum one?

- make classical logic gates reversible: e.g. a Toffoli gate can simulate a NAND gate, which is universal in the classical set
- By **Solovay-Kitaev** theorem, with a universal quantum gate set we can approximate efficiently any other unitary transformation: simulating arbitrary gates up to exponentially small error, costs only a polynomial overhead

So, a quantum computer is at least as powerful as a classical one.

BQP $\subseteq$ EXP

- We've seen that a quantum state is $|\psi\rangle = \sum_i a_i |i\rangle$ where $i \in \{0,1\}^n$
- so, this state vector moves inside an exponential space
- to simulate with a classical computer the evolution of this vector, exponential time should suffice
- conclusion: quantum computers can offer no more than an exponential advantage over classical ones.
- can we find better lower bound?

BQP $\subseteq$ PSPACE [Bernstein & Vazirani '93]

Basic Idea: integrating over computational paths

- We have a language $L \in$ BQP.
- So, there exists a BQP machine $\mathcal{M}$ that decides L within time $p(n)$, for some polynomial p and input $x \in \{0,1\}^n$.
- The tree of the computation has depth $p(n)$.
- For now, let the transition amplitudes be computed in polynomial time (and therefore in polynomial space).
- For each path on the tree:
  - If path ends up accepting, add its amplitude to a running total.
  - Reuse space an repeat process for all paths ($2^{p(n)}$).
- We conclude that the total amplitude needs poly-space to be stored.
- If we square it, we get the probability that $\mathcal{M}$ accepts.
- So $L \in$ PSPACE.

BQP $\subseteq$ PSPACE [Bernstein & Vazirani '93]

How to remove the assumption?

BQP $\subseteq$ PSPACE [Bernstein & Vazirani '93]

How to remove the assumption?

- Given an arbitrary $\mathcal{M}' \in \mathrm{BQP}$, Bernstein & Vazirani showed it suffices to use a similar machine $\mathcal{M}''$ that its transition amplitudes can be exactly calculated.
- If the amplitude of $\mathcal{M}''$ is at least $\frac{7}{12}$ we accept, otherwise we reject.
- They proved that this simulation requires polynomial space.

BQP $\subseteq$ PSPACE [Bernstein & Vazirani '93]

How to remove the assumption?

- Given an arbitrary $\mathcal{M}' \in \mathsf{BQP}$, Bernstein & Vazirani showed it suffices to use a similar machine $\mathcal{M}''$ that its transition amplitudes can be exactly calculated.
- If the amplitude of $\mathcal{M}''$ is at least $\frac{7}{12}$ we accept, otherwise we reject.
- They proved that this simulation requires polynomial space.

Some backstage notes:

- In a universal gate set, each gate operates in a bounded number of qubits.
- a complex number is represented by two integers (one for the real, and one for the imaginary part) with some accuracy they fix.

BQP $\subseteq$ PP [Adleman, Demarrais & Huang '97]

- Like before, proof is based on Feynman path integral.
- Let S be the set of basis states where the output qubit will be $|1\rangle$ (accepting states)
- for each $|x\rangle \in S$ loop over all paths that contribute amplitude to it:
  - the total amplitude of $|x\rangle$ is $a_x = \sum_i a_{x,i}$
  - each $a_{x,i}$ is the amplitude of a path that has $|x\rangle$ as its leaf.
- So $\mathsf{P}_{\text{accept}} = \sum_{x \in S} |\sum_i a_{x,i}|^2 = \sum_{x \in S} \sum_{i,j} a_{x,i} \cdot a^*_{x,j}$
- This is a sum of exponentially many terms, where each term can be computed in poly-time.
- Recall the definition of PP: in order to decide a language, such a machine take the sum of exponentially many terms and decides if it's above or below some threshold.

BQP $\subseteq$ PP [Adleman, Demarrais & Huang '97]

- Let $L \in$ BQP.
- Non deterministically guess x, i, j.
  - If $a_{x,i} \cdot a^*_{x,j} > 0$ then make $|$accepting paths$| \sim |a_{x,i} \cdot a^*_{x,j}|$.
  - If $a_{x,i} \cdot a^*_{x,j} < 0$ then make $|$rejecting paths$| \sim |a_{x,i} \cdot a^*_{x,j}|$.
  - If $a_{x,i} \cdot a^*_{x,j} = 0$ then $|$accepting paths$| \sim |$rejecting paths$|$.
- Notice $x \in L \Rightarrow \mathsf{P}_{\text{accept}} \geq \frac{2}{3} > \frac{1}{2}$ and $x \notin L \Rightarrow \mathsf{P}_{\text{accept}} \leq \frac{1}{3} < \frac{1}{2}$
- So, $L \in PP$.

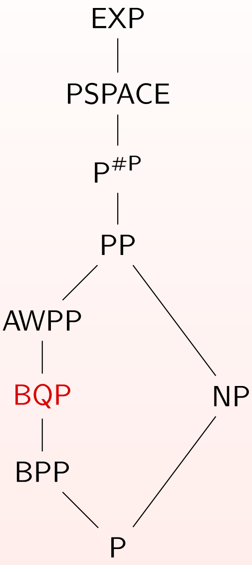
BQP ⊆ PP [Adleman, Demarrais & Huang '97]

- Let $L \in$ BQP.
- Non deterministically guess x, i, j.
  - If $a_{x,i} \cdot a^*_{x,j} > 0$ then make |accepting paths| $\sim |a_{x,i} \cdot a^*_{x,j}|$.
  - If $a_{x,i} \cdot a^*_{x,j} < 0$ then make |rejecting paths| $\sim |a_{x,i} \cdot a^*_{x,j}|$.
  - If $a_{x,i} \cdot a^*_{x,j} = 0$ then |accepting paths| $\sim$ |rejecting paths|.
- Notice $x \in L \Rightarrow \mathrm{P}_{\text{accept}} \geq \frac{2}{3} > \frac{1}{2}$ and $x \notin L \Rightarrow \mathrm{P}_{\text{accept}} \leq \frac{1}{3} < \frac{1}{2}$
- So, $L \in PP$.

Further Notes:

- Best classical upper bound: BQP ⊆ AWPP [Fortnow & Rogers '99].
- They also showed that BQP is low for PP.
- Scott Aaronson, via "post-selection", proved that PostBQP=PP.

What we've seen so far



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

- If $\mathrm{BQP} \neq \mathrm{BPP}$ then a quantum computer would be more powerful than a classical one.
- Furthermore, that would imply that $\mathrm{P} \neq \mathrm{PSPACE}$.

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

- If BQP $\neq$ BPP then a quantum computer would be more powerful than a classical one.
- Furthermore, that would imply that P $\neq$ PSPACE.

- Recall Simon's algorithm (find hidden XOR mask s): does it prove that BQP $\neq$ BPP?

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

- If $\mathrm{BQP} \neq \mathrm{BPP}$ then a quantum computer would be more powerful than a classical one.
- Furthermore, that would imply that $\mathrm{P} \neq \mathrm{PSPACE}$.

- Recall Simon's algorithm (find hidden XOR mask s): does it prove that $\mathrm{BQP} \neq \mathrm{BPP}$?
- No! Due to its black box formulation, it only proves that there is an oracle A for which it holds $\mathrm{BQP}^A \neq \mathrm{BPP}^A$.
- still lack of formal evidence..

what about NP?

- Let's say we have a space of 2^n possible solutions and we are looking for the right one.
- Assume we are in the query model, where we feed a black box oracle with a solution and it replies if it's correct.
- Classically, we need $\sim 2^{n-1}$ queries on average.
- Quantumly, Grover's algorithm makes $2^{n/2}$ queries.
- Actually, Bennett et al. proved that this result is optimal.
- So, for "unstructured" search problems, quantum computers give only quadratic speedup!

- We don't know if $\mathsf{NP} \not\subseteq \mathsf{BQP}$ (unrelativised)
- We don't even know $\mathsf{P} \neq \mathsf{NP} \Rightarrow \mathsf{NP} \not\subseteq \mathsf{BQP}$.
- Abrams & Lloyd in '98 proved that if we remove linearity from quantum mechanics then quantum computers can solve NP-complete problems.

Outline

Recap

NP: A promise problem A is in NP iff there exists:

1. a polynomial p
2. a poly-time deterministic V s.t.

Completeness: if $x \in A_{\text{yes}}$, then $\exists y \; |y| = p(|x|)$ s.t. $V(x,y) = 1$

Soundness: if $x \in A_{\text{no}}$, then $\forall y \; |y| = p(|x|)$ it holds that $V(x,y) = 0$

MA: A promise problem A is in MA iff there exists:

1. a polynomial p
2. a poly-time probabilistic V s.t.

Completeness: if $x \in A_{\text{yes}}$, then $\exists y \; |y| = p(|x|)$ s.t. $\Pr[V(x,y) = 1] \geq \frac{2}{3}$

Soundness: if $x \in A_{\text{no}}$, then $\forall y \; |y| = p(|x|)$ it holds that $\Pr[V(x,y) = 0] \geq \frac{2}{3}$

QMA

- The natural quantum analogue of NP is actually the quantum analogue of MA
- name QMA was coined by Watrous
- briefly: make V quantum and allow proof to be a quantum state

QMA$_p$: A promise problem A is in QMA$_p$ iff there exists:

1. a polynomial p
2. a family $Q = \{Q_n : n \in \mathbb{N}\}$ of quantum circuits s.t.

Completeness: if $x \in A_{\text{yes}}$, then $\exists$ state ρ on $p(|x|)$ qubits s.t. $\Pr[Q(x,y) = 1] \geq \frac{2}{3}$

Soundness: if $x \in A_{\text{no}}$, then $\forall$ state ρ on $p(|x|)$ qubits $\Pr[Q(x,y) = 0] \geq \frac{2}{3}$

- QMA $= \bigcup_p$ QMA$_p$
- QMA is unrealistic because ρ may be difficult to prepare
- but the point of QMA is quantum verification
- QCMA is like QMA but Merlin is classical.

Some Bounds

- QMA $\subseteq$ NEXP: Arthur simulates all witness states that Merlin could send

Some Bounds

- QMA $\subseteq$ NEXP: Arthur simulates all witness states that Merlin could send
- MA $\subseteq$ QCMA: we know that BPP $\subseteq$ BQP

Some Bounds

- QMA $\subseteq$ NEXP: Arthur simulates all witness states that Merlin could send
- MA $\subseteq$ QCMA: we know that BPP $\subseteq$ BQP
- BQP $\subseteq$ QCMA: Merlin sends nothing

Some Bounds

- QMA $\subseteq$ NEXP: Arthur simulates all witness states that Merlin could send
- MA $\subseteq$ QCMA: we know that BPP $\subseteq$ BQP
- BQP $\subseteq$ QCMA: Merlin sends nothing
- NP $\subseteq$ QMA: trivially by Completeness and Soundness conditions

Some Bounds

- $\mathsf{QMA} \subseteq \mathsf{NEXP}$: Arthur simulates all witness states that Merlin could send
- $\mathsf{MA} \subseteq \mathsf{QCMA}$: we know that $\mathsf{BPP} \subseteq \mathsf{BQP}$
- $\mathsf{BQP} \subseteq \mathsf{QCMA}$: Merlin sends nothing
- $\mathsf{NP} \subseteq \mathsf{QMA}$: trivially by Completeness and Soundness conditions
- $\mathsf{QCMA} \subseteq \mathsf{QMA}$: a classical Merlin can be simulated by a quantum one
  - We don't know if $\mathsf{QCMA} \neq \mathsf{QMA}$ (not even relativized)

Some Bounds

- QMA $\subseteq$ NEXP: Arthur simulates all witness states that Merlin could send
- MA $\subseteq$ QCMA: we know that BPP $\subseteq$ BQP
- BQP $\subseteq$ QCMA: Merlin sends nothing
- NP $\subseteq$ QMA: trivially by Completeness and Soundness conditions
- QCMA $\subseteq$ QMA: a classical Merlin can be simulated by a quantum one
  - We don't know if QCMA $\neq$ QMA (not even relativized)

Quantum Oracle Separation [Aaronson & Kuperberg, '07]

There exists a quantum oracle A s.t. $\mathrm{QCMA}^A \neq \mathrm{QMA}^A$

Play with QMA's conditions

Perfect Completeness

- $\mathsf{MA} = \mathsf{MA}_1$ (Zachos & Fürer, '87)
- $\mathsf{QMA} \stackrel{?}{=} \mathsf{QMA}_1$
  - $\exists$ quantum oracle A s.t. $\mathsf{QMA}_1^A \subset \mathsf{QMA}^A$ [Aaronson, '09]
  - we need a quantumly nonrelativizing proof
- $\mathsf{QCMA} = \mathsf{QCMA}_1$ [Jordan, Kobayashi, Nagaj & Nishimura, '12]

Play with QMA's conditions

Perfect Completeness

- $\mathsf{MA} = \mathsf{MA}_1$ (Zachos & Fürer, '87)
- $\mathsf{QMA} \stackrel{?}{=} \mathsf{QMA}_1$
  - $\exists$ quantum oracle A s.t. $\mathsf{QMA}_1^A \subset \mathsf{QMA}^A$ [Aaronson, '09]
  - we need a quantumly nonrelativizing proof
- $\mathsf{QCMA} = \mathsf{QCMA}_1$ [Jordan, Kobayashi, Nagaj & Nishimura, '12]

Perfect Soundness

- if perfect soundness, then we have NQP [Kobayashi, Matsumoto & Yamakami, '08]
- NQP is the quantum analogue of probabilistic characterization of NP
- QMA is the quantum analogue of quantifier characterization of NP
- $\mathsf{NQP} = co\mathsf{C}_{=}\mathsf{P}$ [Yamakami & Yao, '99]

Error reduction of QMA

- if we copy the quantum proof it will be damaged
- no need for a fresh copy each time - find another Verifier

Strong error reduction of QMA [Marriott & Watrous, '04]

For any choice of p and completeness and soundness probabilities a and b with $a(n) - b(n) \geq \frac{1}{q(n)}$ for some polynomial q, it holds that $\forall$ polynomial r

$$\mathrm{QMA}_p(a,b) = \mathrm{QMA}_p(1-2^{-r}, 2^{-r})$$

- we can make r bigger than p
- error will be smaller than the reciprocal of Hilbert space dimension

QMA $\subseteq$ PP

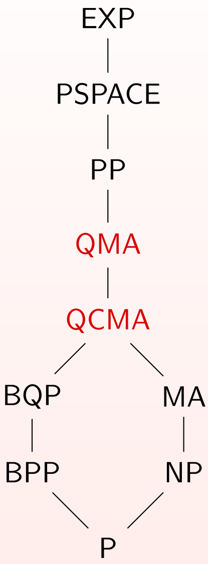
- Let $L \in \mathsf{QMA}$
- So for some p we have $L \in \mathsf{QMA}_p(\frac{2}{3}, \frac{1}{3})$
- by strong reduction $L \in \mathsf{QMA}_p(1 - \frac{1}{2^{p+2}}, \frac{1}{2^{p+2}})$

We consider the following algorithm:

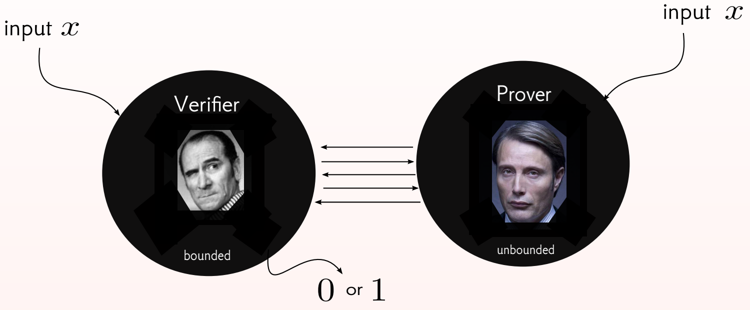
1. randomly guess a quantum proof on p qubits
2. feed this proof to a Verifier $V \in \mathsf{BQP}(1 - \frac{1}{2^{p+2}}, \frac{1}{2^{p+2}})$

- $\forall x \in L$ V accepts w.p. $\geq \frac{1}{2^{p(|x|)+1}}$
- $\forall x \notin L$ V accepts w.p. $\leq \frac{1}{2^{p(|x|)+2}}$
- V is not good but gives tiny amount of info about the correct answer
- $V \in \mathsf{PQP}$ (quantum analogue of PP)
- $\mathsf{PQP} = \mathsf{PP}$ [Watrous, '09]
- $\mathsf{QMA} = \mathsf{PP} \Rightarrow \mathsf{PH} \subseteq \mathsf{PP}$ [Vyali, '03]

What we've seen so far



Last Recap



- extend the notion of verification to interactive setting
- replace proof with an entity that answers questions

A language $L \subset \{0,1\}^*$ has an **interactive proof system** if:

Completeness: $\forall x \in L$, $\exists$ prover-strategy s.t. Verifier accepts with high prob.

Soundness: $\forall x \notin L$, for every prover-strategy, Verifier rejects with high prob.

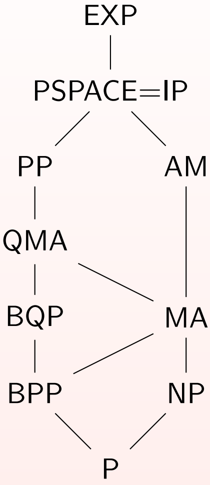
AM: class of languages that have classical interactive proof systems with constant number of rounds

- AM(m) =AM(2)

IP: class of languages that have classical interactive proof systems with polynomial number of rounds

IP=PSPACE [Shamir, '90]

quantum interactive proof systems: the same, just allow Prover and Verifier to be quantum



QIP

- QIP is the same as IP but with quantum interactive proof systems
- QIP(m): at most m rounds, where $m \in \mathbb{Z}^+$

[Jain, Ji, Upadhyay & Watrous '09]

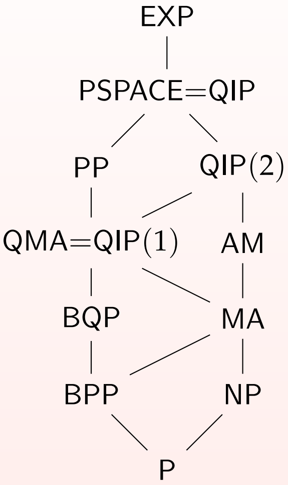
QIP(3) =QIP=PSPACE

- So quantum int. proof systems no more powerful than classical ones.
- with only 3 rounds, you get full power of QIP, even for polynomial number of rounds
- it's not believed that AM=PSPACE
- quantumly, there is a significant reduction in the number of rounds
- problems in PSPACE probably need polynomial number of rounds

[Kitaev & Watrous, '03]

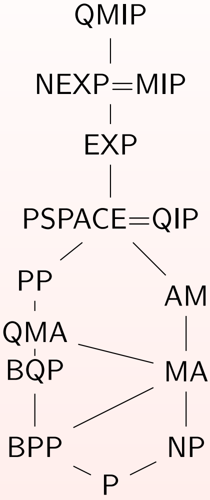
QIP(1) =QMA
QAM$\subseteq$QIP(2)

What we've finally seen so far



(More) Open Problems

- BQP ? PH
- what if we limit quantum models?
  - linear optical quantum computers
  - one-clean-qubit model
  - matchgate circuits
- Upper bounds on entangled provers?
  - we know MIP=NEXP
  - NEXP$\subseteq$ QMIP [Ito & Vidick, '12]



Outline

References

Scott Aaronson.
Quantum Computing Since Democritus: lecture notes.
University of Waterloo, 2006.

Ethan Bernstein and Umesh Vazirani.
Quantum complexity theory.
STOC '93. ACM, 1993.

Richard Cleve.
An introduction to quantum complexity theory.
Technical report, arXiv, 1999.

Ronald de Wolf.
Quantum Computing: lecture Notes.
University of Amsterdam, 2014.

Michael A. Nielsen and Isaac L. Chuang.
Quantum Computation and Quantum Information: 10th Anniversary Edition.
Cambridge University Press, 2011.

John Watrous.
Quantum Computational Complexity.
University of Waterloo, 2008.

Thank you!