Introduction to Quantum Computing

Quantum Complexity Theory

Marios Rozos

ECE-NTUA

June 2022

What is Quantum Computing?

Quantum Computing is a way of computation using the strangeness of Quantum Mechanics to do computation, using phenomena such as:

- Superposition
- Interference
- Entanglement

Linear Algebra

Linear Algebra is the language of QM (or at least that's an approach). A fundamental concept of Linear Algebra is vector spaces.

Vector Spaces: is a set with vectors as elements and closed under vector addition and scalar multiplication.

A **linear operator** on a vector space H is a linear transformation $T : H \; ! \; H$ of the vector space to itself.

In QM we are interested in complex vector spaces and complex linear operators.

Dirac Notation

In quantum mechanics we use the vector notation invented by Paul Dirac:

$$|i\rangle = \begin{matrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{matrix}$$

$|i\rangle$: is a vector written inside a ket.

Dual vector of $|i\rangle$ is the vector:

$$\langle i| = \begin{matrix} z_1^* & z_2^* & z_3^* & \cdots & z_n^* \end{matrix}$$

$\langle i|$: is a vector written inside a bra.

Hilbert Spaces

A **Hilbert space** H is an infinite dimensional real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product.

However, in QM we consider every complex vector space of finite dimension with inner product as a Hilbert space.

Inner Product Space

An inner product space is a vector space V over the field F together with an inner product, that is a map $h\,;\ i : V\ \ !\ \ F$ that satisfies the following properties:

1. Conjugate symmetry:
$$hy\,;x\,i = \overline{hx\,;y\,i}$$
2. Linearity in the first argument:
$$hax_1 + bx_2;y\,i = ahx_1;y\,i + bhx_2;y\,i;\ \ 8a;b\ 2\ \mathrm{C}$$
3. Positive definiteness:
$$hx;x\,i \quad 0 \text{ if } x \ 6= 0$$
$$hx;x\,i = 0 \text{ if } x = 0$$

Outer Product

Outer Product

Given two vectors of size $m \quad 1$ and $n \quad 1$ respectively,

$$u = [u_1 \; u_2 \; ::: \; u_m]^T ; v = [v_1 \; v_2 \; ::: \; v_n]^T$$

the outer product

$$u \quad v = uv^y = \begin{bmatrix} u_1v_1 & u_1v_2 & & u_1v_n \\ u_2v_1 & u_2v_2 & & u_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_mv_1 & u_mv_2 & & u_mv_n \end{bmatrix}$$

In Dirac's notation: $(j \quad ih \quad j)j \quad i = j \quad i\,(h \quad j \quad i) = (h \quad j \quad i)j \quad i$

Tensor product

Tensor product

Tensor product is defined by the following 3 properties:

- $c(j\ _1i\quad j\ _2i) = (c\,j\ _1i)\quad j\ _2i = j\ _1i\quad (c\,j\ _2i)$
- $(j\ _1i + j'\ _1i)\quad j\ _2i = j\ _1i\quad j\ _2i + j'\ _1i\quad j\ _2i$
- $j\ _1i\quad (j\ _2i + j'\ _2i) = j\ _1i\quad j\ _2i + j\ _1i\quad j'\ _2i$

for every $c\ 2\ C; j\ _1i; j'\ _1i\ 2\ H_1; j\ _2i; j\ _2i\ 2\ H_2$.

Tensor Product

$$A \otimes B = \begin{bmatrix} A_{11}B_{11} & \dots & A_{11}B_{1q} & \dots\dots & A_{1n}B_{11} & \dots & A_{1n}B_{1q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{11}B_{p1} & \dots & A_{11}B_{pq} & \dots\dots & A_{1n}B_{p1} & \dots & A_{1n}B_{pq} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{m1}B_{11} & \dots & A_{m1}B_{1q} & \dots\dots & A_{mn}B_{11} & \dots & A_{mn}B_{1q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{m1}B_{p1} & \dots & A_{m1}B_{pq} & \dots\dots & A_{mn}B_{p1} & \dots & A_{mn}B_{pq} \end{bmatrix}.$$

Example

If $u = [1 \;\; 2 \;\; 3]^T$ and $v = [4 \;\; 5]^T$, we have:

$$u \;\; {}_{\text{outer}} \; v = \begin{matrix} 2 & 3 \\ 4 & 5 \end{matrix} \begin{matrix} 4 & 5 \\ 8 & 10 \\ 12 & 15 \end{matrix}$$

and

$$u \quad v = \begin{matrix} 2 \\ 4 \end{matrix} \begin{matrix} 3 \\ 5 \end{matrix} \begin{matrix} 6 \\ 8 \\ 10 \\ 12 \end{matrix} \begin{matrix} 7 \\ 7 \\ 7 \\ 7 \\ 7 \\ 5 \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ 15 \end{matrix}$$

Unitary and Hermitean Operators

Unitary operator

An operator U is called **unitary** if $U^y U = U U^y = I$, where I is the identity operator.

The unitary operators preserve inner products between vectors and the norms of them

Hermitean Operator

An operator T in a Hilbert space H is called **Hermitean** or self-adjoint if

$$T^y = T$$

Projector

A projector on a vector space H is a linear operator P that satisfies $P^2 = P$. An orthogonal projector is a projector that also satisfies $P^y = P$.

Eigenvalues

Eigenvectors and Eigenvalues

A vector $j \quad i$ is called an **eigenvector** of an operator T if

$$T \, j \quad i = c \, j \quad i$$

for some constant c. The constant c is called the **eigenvalue** of T corresponding to the eigenvector $j \quad i$.

The eigenvalues of a Hermitean operator are real.

The eigenvalues of **observables** (e.g. position, momentum, spin etc.) are real numbers and correspond to outcomes of particular measurements.

Postulates of Quantum Mechanics

Quantum Mechanics (QM) is not a theory; it is a mathematical framework of physics based on postulates

e.g. QED and QFT are theories developed in this framework

We are going to see the postulates of QM in an information theory reformulation.

Postulate 1: State Space

Postulate 1

Every isolated physical system can be associated with a **Hilbert space** known as the **state space of the system**. The system is completely described by its state vector, which is a **unit vector** in the system's state space.

The simplest QM system is the qubit (a two-dimensional state space). Suppose $j0i\,; j1i$ form an orthonormal basis. Then an arbitrary state vector in the state space can be written:

$$j \quad i = \quad j0i + \quad j1i\,; \quad ; \quad 2\ \mathrm{C};$$

Normalization Condition

$j \quad i$ is a unit vector, i.e. $h \quad j \quad i = 1$ or equivalently $j \quad j^2 + j \quad j^2 = 1$.

Postulate 1: State Space

Superposition: A quantum state can be expressed as a linear combination $\sum_i \alpha_i |i\rangle$ of states $|i\rangle$ with coefficients (amplitudes) α_i.

Example

The quantum state:

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

is the superposition of states $|0\rangle$ and $|1\rangle$ with amplitude $1/\sqrt{2}$ for $|0\rangle$ and $1/\sqrt{2}$ for the $|1\rangle$.

We have:

$$|0\rangle = \begin{matrix} 1 \\ 0 \end{matrix} \quad \text{και} \quad |1\rangle = \begin{matrix} 0 \\ 1 \end{matrix}$$

and the set $\{|0\rangle, |1\rangle\}$ is called **computational basis**.

Bloch Sphere

Every quantum state $j \quad i = \quad j0i + \quad j1i$ can be written as:

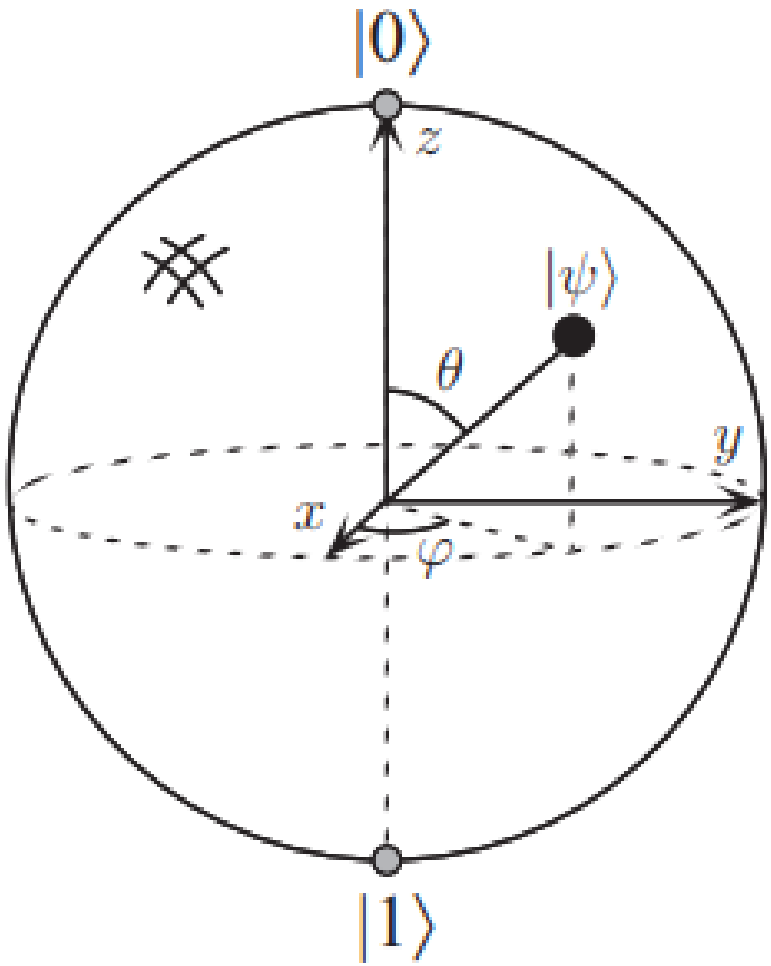
$$j \quad i = e^{i} \quad \cos \frac{}{2} j0i + e^{i\,'} \sin \frac{}{2} j1i \quad ; \quad ; \; ' ; \quad 2 \text{ R:}$$

We ignore e^{i} (global phase), so we have:

$$j \quad i = \cos \frac{}{2} j0i + e^{i\,'} \sin \frac{}{2} j1i$$

Using and $'$ we can define a 3D-Sphere, called Block Sphere, which helps us in qubit visualization.

Bloch Sphere



Other computational bases

We define:

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

and

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

$|+\rangle$ and $|-\rangle$ define Hadamard computational basis.

We can also have:

$$|R\rangle \quad \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$$

and

$$|L\rangle \quad \frac{1}{\sqrt{2}}|0\rangle \quad \frac{i}{\sqrt{2}}|1\rangle$$

Computational bases

- The state basis around z axis z is: $fj0i\,;j1ig$
- The state basis around z axis x is: $fj+i\,;j\quad ig$
- The state basis around z axis y is: $fjR\,i\,;jLig$

Postulate 2: Time-Evolution

Postulate 2

The time-evolution of the state of a **closed quantum system** is described by a **unitary operator**. That is, for any evolution of the closed system there exists a unitary operator U such that if the initial state of the system is $|\psi_1\rangle$, then after the evolution the state of the system will be

$$|\psi_2\rangle = U|\psi_1\rangle$$

We will see that the quantum gates are unitary operators.

Postulate 3: Composition of Systems

Postulate 3

When two physical systems are treated **as one combined system**, the state space of the **combined physical system is the tensor product space** $H_1 \quad H_2$ of the state spaces $H_1; H_2$ of the component subsystems. If the first system is in the state $j \ _1 i$ and the second system in the state $j \ _2 i$, then the state of the combined system is:

$$j \ _1 i \quad j \ _2 i$$

Postulate 4: Measurement

Postulate 4

The measurement of quantum states are described by a set $fM_m g$ of measurement operators, where m corresponds to one possible outcome. The probability of measuring the mth state is

$$p(m) = h \quad j M_m^y M_m j \quad i$$

and after the measurement the system collapses into the state

$$\frac{M_m j \quad i}{h \quad j M_m^y M_m j \quad i}$$

The measurement operators satisfy the following relationship.

$$\sum_m M_m^y M_m = I$$

Postulate 4: Measurement

Measuring a qubit

Suppose we have a qubit $j\ \ i = \ \ j0i + \ \ j1i$. Measurement operators are $M_0 = j0i\,h0j$ and $M_1 = j1i\,h1j$.

The probability of measuring 0 is:

$$p(0) = h\ \ j\,M_0^y M_0\,j\ \ i = h\ \ j\,M_0\,j\ \ i = j\ \ j^2$$

and measuring 1: $p(1) = j\ \ j^2$.

Notice that the probability of each outcome in the measurement is the **square** of the corresponding coefficient in the linear combination. After the measurement the quantum state **collapses into the state that we measured**.

Figure: Some of the distinguished attendees of IBM and MIT's conference on The Physics of Computation, held at MIT's Endicott House in Dedham, Massachusetts, May 6 to 8, 1981. [Photo: courtesy of Charlie Bennett]

History of Quantum Computing

- 1980: Paul Benioff proposed a quantum mechanical model of the Turing machine
- 1981: IBM and MIT's conference in Endicott House
- Richard Feynman suggested that quantum computers could be used for simulation that classical computers could not do
- 1985: Universal Quantum Computer by David Deutsch
- 1992: Superdense coding by Charles Bennett and Stephen Wiesner
- 1994: Factoring and Discrete Logarithm in **BQP** by Peter Shor
- 1994: BBBV paper
- 1995: Grover's Algorithm
- 1997: 'Quantum Complexity Theory' by Bernstein and Vazirani

History of Quantum Computing

- 1998: first two-qubit quantum computer by Isaac Chuang, Neil Gershenfeld and Mark Kubinec
- 2017: the first commercially usable quantum computer by IBM
- 2019: Google is the first to claim to have achieved quantum supremacy by performing calculations on the Sycamore quantum computer
- 2020: USTC group claims to have demonstrated quantum supremacy with Boson sampling on 76 photons with a photonic quantum computer
- 2021: 127-qubit microprocessor named IBM Eagle

Figure: IBM Eagle

Quantum Supremacy or Quantum Advantage

A term coined by John Preskill referring to the feat of demonstrating that a quantum computer can solve a problem beyond the capabilities of state-of-the-art classical computers.

The problem need not be useful, it may be a potential future benchmark.

The most important proposal: **boson sampling** by Aaronson and Arkhipov

Sending identical photons through a linear-optical network we try to calculate the **permanent** of Gaussian matrices (#P-Hard), which can be simulated efficiently with in a system with large enough loss and noise

Future applications of QC

- Mostly for fault-tolerant devices: Simulation of quantum systems, search problems, factoring and quantum cryptography, computer-aided drug design, *...*
- Even in NISQ-devices: optimization, quantum machine learnign, quantum chemistry,

NISQ: Noisy intermediate-scale quantum era

Church-Turing Thesis

Church - Turing Thesis

A computing problem can be solved on any computer that we could hope to build, if and only if it can be solved on a Turing Machine.

The original Church–Turing Thesis says nothing about the efficiency of computation.

Strong Church - Turing Thesis

Any algorithmic process can be simulated **e ciently** using a Turing machine.

Church-Turing Thesis

mid 1970s: Solovay-Strassen test for primality

$\Rightarrow$ Randomized algorithms pose a challenge

We can do a modification, but it is ad hoc

Extended Church - Turing Thesis

Any algorithmic process can be simulated **e ciently** using a probabilistic Turing machine.

'Efficiently' means up to **polynomial reductions**.

Church - Turing Thesis

In 1985 David Deutsch: Is there an even stronger version of the Church–Turing thesis based on the laws of physics

$$) \text{ Quantum Mechanics}$$

Physical Church - Turing Thesis

Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means.

E. Bernstein and U. Vazirani (1997) stated that:

Computational Complexity-Theoretic Church–Turing thesis

All 'reasonable' models of computation yield the same class of problems that can be computed in polynomial time

Quantum Circuits

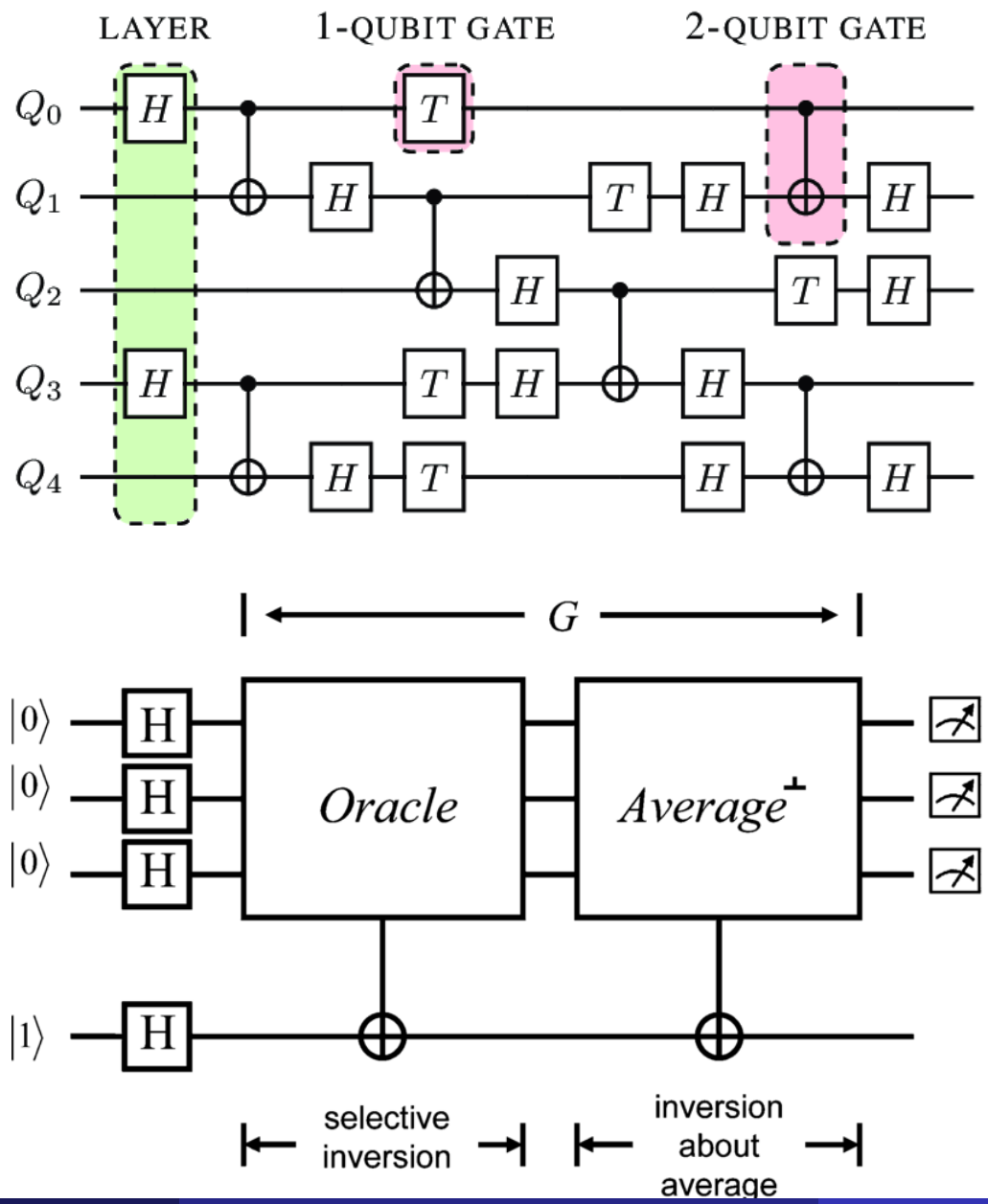
The most used computational model in QC and in quantum algorithms quantum circuits. We use **uniform quantum circuit families**.

(Uniform means that there is a classical algorithm of polynomial time that generates a (classical) description of the circuit.)

Every classical circuit can be converted into an equivalent quantum circuit.

Quantum circuits are helpful in time complexity classes and in describing algorithms, while quantum TM are useful in space complexity classes.

Examples of quantum circuits



Pauli Gates

$$X = \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} ; Y = \begin{matrix} 0 & i \\ i & 0 \end{matrix} ; Z = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} ; I = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Hadamard Gate

$$H = \frac{1}{\sqrt{2}} \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

S Gate and T gate

$$S = \begin{matrix} 1 & 0 \\ 0 & i \end{matrix} ; T = \begin{matrix} 1 & 0 \\ 0 & e^{i} \end{matrix} =4$$

Pauli gates

X-gate: bit-shift (NOT gate)

$$X\,|0\rangle = |1\rangle\,;\;\; X\,|1\rangle = |0\rangle\,;$$

Z-gate: phase - flip

$$Z\,|0\rangle = |0\rangle\,;\;\; Z\,|1\rangle = \;\; |1\rangle$$

Y-gate:bit - phase - flip

$$Y\,|0\rangle = i\,|1\rangle\,;\;\; Y\,|1\rangle = \;\; i\,|0\rangle$$

Hadamard Gate

Hadamard gate turns a qubit into superposition of states (notice that $H^2 = I$)

$$H = \frac{1}{\sqrt{2}} \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}$$

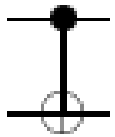
then:

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

and

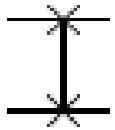
$$H|1\rangle \quad |-\rangle \quad \frac{1}{\sqrt{2}}|0\rangle \quad \frac{1}{\sqrt{2}}|1\rangle$$

controlled-NOT



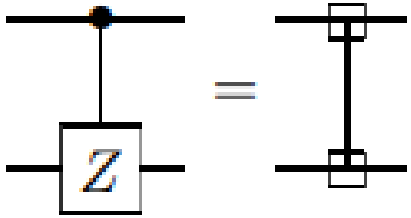
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

swap



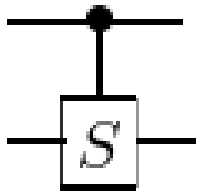
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

controlled-Z



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

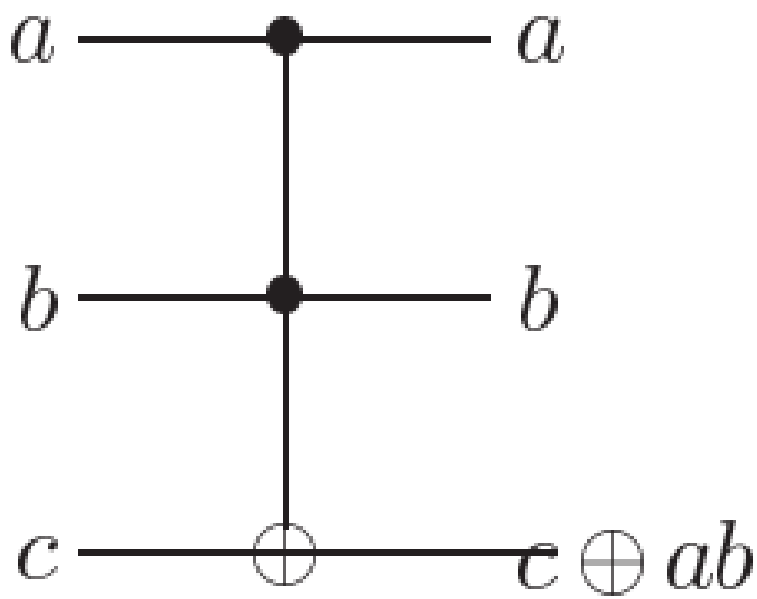
controlled-phase



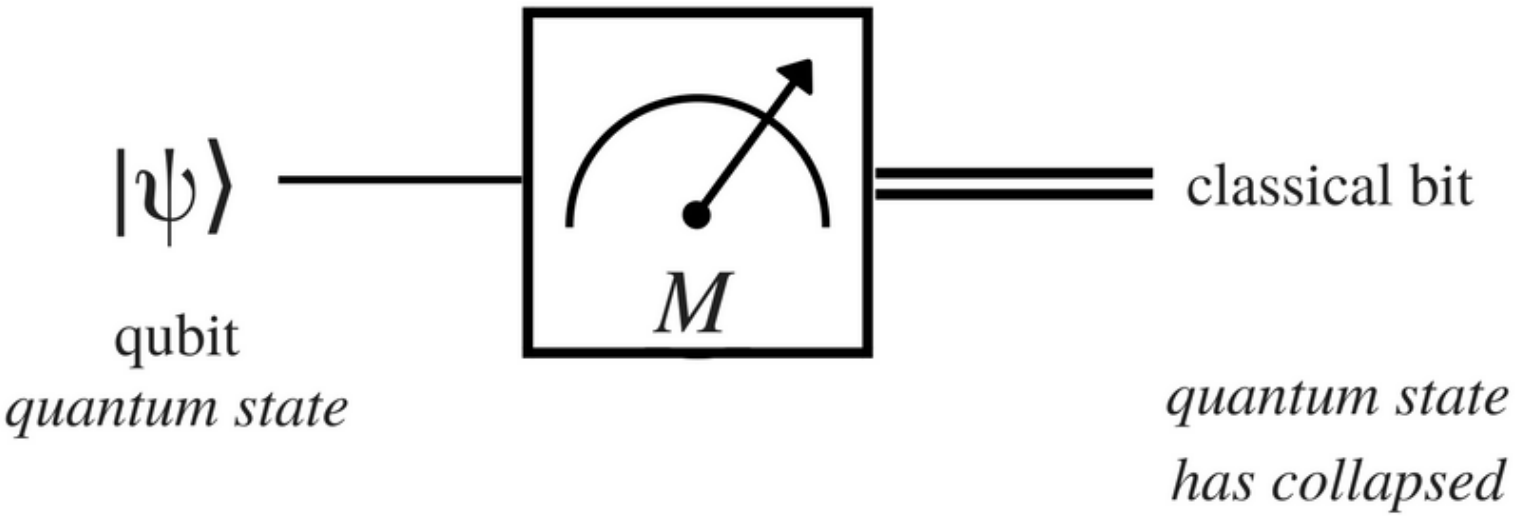
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

Toffoli Gate

Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



Measurement Gate



The No-Cloning Theorem

Can we copy an unknown quantum state? The answer is no!
This is called **no-cloning theorem**.

Proof

Let's say we have two slots, A and B. Slot A starts in an unknown but pure quantum state $j\ \ i$ and we want to copy it into slot B, while B starts in some pure state $js\,i$.

The initial state is: $j\ \ i\ \ \ \ js\,i$.
Some unitary evolution U now effects the copying procedure

$$j\ \ i\ \ \ \ js\,i \overset{U}{!}\ U(j\ \ i\ \ \ \ js\,i) = j\ \ i\ \ \ \ j\ \ i$$

The No-Cloning Theorem

Can we copy an unknown quantum state? The answer is no!
This is called **no-cloning theorem**.

Proof (cont'd)

Suppose this copying procedure works for two particular pure states, $j\ \ i$ and $j\ \ i$.
Then:

$$U(j\ \ i\ \ \ \ js\,i) = j\ \ i\ \ \ \ j\ \ i$$

$$U(j\ \ i\ \ \ \ js\,i) = j\ \ i\ \ \ \ j\ \ i$$

Taking the inner product of these two equations gives

$$h\ \ j\ \ i = (h\ \ j\ \ i)^2$$

The No-Cloning Theorem

Can we copy an unknown quantum state? The answer is no!
This is called **no-cloning theorem**.

Proof (cont'd)

$$h \;\; j \;\; i = (h \;\; j \;\; i)^2$$

This means either $j \;\; i = j \;\; i$ or $j \;\; i \,; j \;\; i$ are orthogonal.

Thus a cloning device can only clone states which are orthogonal to one another, and therefore a general quantum cloning device is impossible.

Why Quantum Complexity Theory is important

- It gives insight about Computation, QM and Mathematics
- It gives new and elegant proofs about classical results
- It is fun!

BPP

Let's remember the **BPP** complexity class:

Bounded-error probabilistic polynomial time (BPP)

A language L is in **BPP** if and only if there exists a **randomized classical algorithm** A running with worst-case polynomial time such that for any input $x \in \Sigma$ we have

- if $x \in L$, then the probability that A accepts x is at least $\frac{2}{3}$
- if $x \notin L$, then the probability that A accepts x is at most $\frac{1}{3}$

BQP

BQP: the quantum analogue to the complexity class BPP
contains the problems that can be solved efficiently by a quantum computer

Bounded-error Quantum Polynomial Time (BQP)

A language L is in **BQP** if and only if there exists a **quantum algorithm** A running with worst-case polynomial time such that for any input $x \in \Sigma$ we have

- if $x \in L$ then the probability that A accepts x is at least $\frac{2}{3}$.
- if $x \notin L$ then the probability that A accepts x is at most $\frac{1}{3}$.

BQP

BQP [Bernstein - Vazirani 1997]

A language L is in **BQP** if and only if there exists a polynomial-time uniform family of quantum circuits $fQ_n : n\ 2\ \mathrm{N}g$ such that for all $n\ 2$ N takes n qubits as input and outputs 1 bit and:

- for all $x\ 2\ L$, $\mathrm{P}[Q_{jxj}(x) = 1]$ $\frac{2}{3}$
- for all x 62 L, $\mathrm{P}[Q_{jxj}(x) = 0]$ $\frac{2}{3}$

Nothing's special about $\frac{2}{3}$

The same holds if the algorithm gives the correct answer w.p. 1=2 + and the wrong answer w.p. 1=2 .

Chernoff Bounds

The Chernoff bound

Suppose $X_1; \quad ; X_n$ are iid random variables, each taking the value 1 with probability 1=2 , and the value 0 with probability 1=2 . Then

$$p \left[\sum_{i=1}^{n} X_i \quad n=2 \right] \quad e^{\ 2\ ^2 n}:$$

For fixed the probability of decreases exponentially quickly in the number of repetitions of the algorithm.

For $= 1=4$ it takes 100 repetitions for error $10^{\ 20}$.

The main problems with BQP

Since 1993 a central concern has been how **BQP** relates to classical complexity classes, such as **P**, **NP**, and **PH**.

- Can quantum computers efficiently solve any problems that classical computers cannot? In other words, does **BPP** $=$ **BQP**?
- Can quantum computers solve **NP**-complete problems in polynomial time? In other words, is **NP** **BQP**?
- What is the best classical upper bound on the power of quantum computation? Is **BQP** **NP**?Is **BQP** **PH**?

Three decades later, all three of these still stand as defining questions of the field!

What do we know about BQP?

- Bernstein and Vazirani, 1997: $\mathbf{BPP} \quad \mathbf{BQP} \quad \mathbf{P}^{\#P}$
- Adleman, DeMarrais, and Huang, 1997: $\mathbf{BQP} \quad \mathbf{PP}$

  so we get:

  $\mathbf{P} \quad \mathbf{BPP} \quad \mathbf{BQP} \quad \mathbf{PP} \quad \mathbf{P}^{\#P} \quad \mathbf{PSPACE} \quad \mathbf{EXP}$

- Bennett, Bernstein, Brassard, and Vazirani, 1994: $\mathbf{BQP}^{BQP} = \mathbf{BQP}$
- Fortnow and Rogers, 1998: $\mathbf{PP}^{BQP} = \mathbf{PP}$

Bernstein - Vazirani Problem

Bernstein - Vazirani Problem [BV97]

Given an oracle that implements a function $f : f0;1g^n \; ! \; f0;1g$ in which f(x) is promised to be the dot product between x and a secret string $s \; 2 \; f0;1g^n$ mod 2,

$$f(x) = x \quad s = x_1 s_1 \quad x_2 s_2 \qquad x_n s_n;$$

find s.

This problem was designed to probe an oracle separation between **BQP** and **BPP**, i.e. there exists an oracle A s.t. $\mathbf{BPP}^A \not= \mathbf{BQP}^A$.

Search problem

Bennett, Bernstein, Brassard, and Vazirani [BBBV97]

There exists an oracle relative to which **NP** *6* **BQP**. Relative to this oracle there are problems that take n time for an **NP** machine but $\Omega(2^{n=2})$ time for a **BQP** machine.

Grover's Algorithm

We can search any list of N items quantumly in $O(\sqrt{N})$ queries.

Using [BBBV97] we can show that Grover's Algorithm is optimal.

In other words, any quantum algorithm for NP-complete problems that gets more than the square-root speedup of Grover's algorithm must be "non-black-box." It must exploit the structure of a particular NP-complete problem much like a classical algorithm would have to, rather than treating the problem as just an abstract space of 2n possible solution.

Shor's Algorithm

In 1994 Peter Shor developed a polynomial time quantum computer algorithm for finding prime factors.

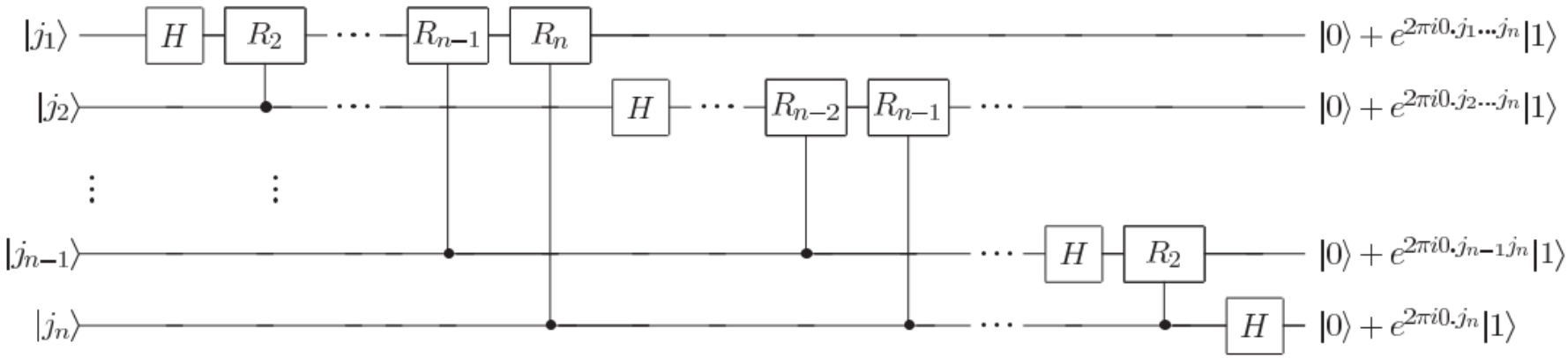
So he showed that factoring and discrete logarithm problems are in **BQP**, giving also a stronger separation between **BQP** and **BPP**.

Shor's algorithm: $O(\ (\log N)^2(\log\log N)(\log\log\log N)\ )$ quantum gates

Most efficient classical algorithm: $O\ \ e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}}$

Shor's Algorithm

The efficiency of Shor's algorithm is due to the efficiency of the **quantum Fourier transform**: it uses $O(n^2)$ gates instead of $O(n2^n)$.



2001: IBM factored 15 = 3 5

2012: Factorization of 21 was achieved

2019: IBM Q System One failed to factor 35

QMA

Quantum Merlin-Arthur (QMA)

A language L is in QMA(c,s) if there exists a polynomial time quantum verifier V and a polynomial p(x) such that:

- $\forall x \in L$, there exists a quantum state $|\psi\rangle$ such that the probability that V accepts the input $(|x\rangle, |\psi\rangle)$ is greater than c.
- $\forall x \notin L$, for all quantum states $|\psi\rangle$ such that the probability that V accepts the input $(|x\rangle, |\psi\rangle)$ is less than s.

where $|\psi\rangle$ ranges over all quantum states with at most $p(|x|)$ qubits.

We define $\mathbf{QMA} = \mathbf{QMA}\left(\frac{2}{3}, \frac{1}{3}\right) = \mathbf{QMA}(1 - 2^{-r(n)}, 2^{-r(n)})$ for any polynomial $r(n)$.

QMA

This means that **QMA** is the set of languages for which:

- when the answer is YES, there is a polynomial-size quantum proof (a quantum state) that convinces a polynomial time quantum verifier with high probability
- when the answer is NO, every polynomial-size quantum state is rejected by the verifier with high probability.

Analogous relationships: **QMA** with **BQP**, **NP** with **P**, with **MA** and **BPP**.

QAM: Arthur generates a random string, Merlin answers with a quantum certificate and Arthur verifies it as a **BQP** machine.

P NP MA QMA PP PSPACE

k-Local Hamiltonian

The Hamiltonian of a system is an operator corresponding to the total energy of that system.

A **k-local Hamiltonian** H is a Hermitian matrix acting on n qubits in which every term involves at most k qubits each.

$$H = \sum_{i=1}^{m} H_i$$

The general problem is to find the smallest eigenvalue of H, which expresses the ground state energy.

k-Local Hamiltonian

k-Local Hamiltonian

Given a k-local Hamiltonian on n qubits, $\sum_{i=1}^{r} H_i$, where r = poly(n) and each Hi acts non-trivially on at most k qubits and has bounded operator norm k *jjHjj* poly(n), determine whether:

- (yes case) H has an eigenvalue less than a
- (no case) all of the eigenvalues of H are larger than b, promised one of these to be the case, where $b \quad a = 1/poly(n)$

The general problem is to find the smallest eigenvalue of H, which expresses the ground state energy.

k-Local Hamilton problem is **QMA**-complete for k 2

The class PP

PP (Probabilistic Polynomial Time) [Gill '77]

A language L is in PP iff there exists a probabilistic Turing machine M, such that

- M runs for polynomial time on all inputs
- For all x in L, M outputs 1 with probability strictly greater than 1/2
- For all x not in L, M outputs 1 with probability less than or equal to 1/2

PP: the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than 1/2 for all instances.

Question: is PP closed under intersection?

The class PP

Theorem (Beigel, Reingold and Spielman)

PP is closed under intersection

However, the proof uses rational functions and is complicated.

The class PostBQP

Postselection refers to the process of conditioning the experiment on getting the outcome that you are looking for, and discarding the outcome otherwise.
We define the following class:

PostBQP (Postselected Bounded-Error Quantum Polynomial-Time)

PostBQP is the class of languages $L \quad f0;1g$ for which there exists a uniform family of polynomial-size quantum circuits $fC_n g_{n \ 1}$ such that for all inputs x,

1. After C_n is applied to the state $j0:::0i \quad jxi$, the first qubit has a nonzero probability of being measured to be $j1i$.
2. If $x \ 2 \ L$, then conditioned on the first qubit being $j1i$, the second qubit is $j1i$ with probability at least $2/3$.
3. If $x \not\in L$, then conditioned on the first qubit being $j1i$, the second qubit is $j1i$ with probability at most $1/3$.

Some interesting facts about PostBQP

- NP PostBQP
- **[Adleman, DeMorrai, Huang, '91]** PostBQP PP
- PostBQP is trivially closed under union, intersection, and complement

Theorem [Aaronson '04]

PostBQP = PP

The proof is short and does not use heavy-duty mathematics

PostBQP = PP implies Beigel et al. results!

Consequences of PostBQP=PP

We define BQP_p similarly to BQP , except that when we measure, the probability of obtaining a basis state $|x\rangle$ equals $|\alpha_x|^p / \sum_y |\alpha_y|^p$ instead of $|\alpha_x|^2$. Thus, $\mathrm{BQP}_2 = \mathrm{BQP}$

Theorem [Aaronson '04]

PP $\subseteq$ BQP_p $\subseteq$ PSPACE, $\forall p \neq 2$, with $\mathrm{BQP}_p = \mathrm{PP}$ when $p \in \{4, 6, 8, \ldots\}$

If we changed the measurement probability rule from $|\alpha|^2$ to $|\alpha|^p$; $p > 2$ or allowed linear but nonunitary evolution, then we could simulate postselection.

So we solve NP-complete and even PP-complete problems in polynomial time!

Relativization

Relativization (i.e. black-box complexity):

- has been a central tool for complexity theorists.
- lets us make well-defined progress even when the original questions we wanted to answer are out of reach.
- is an imperfect tool, though.

In quantum complexity theory, relativization has been an inextricable part of progress from the very beginning.
e.g we have quantum algorithms that query all oracle bits in superposition.

Some relativized results

- Watrous, 2000: There is an oracle A, such that $\mathbf{BQP}^A \not= \mathbf{BPP}^A$
- Watrous, 2000: There is an oracle A, such that $\mathbf{NP}^A \not= \mathbf{BQP}^A$
- Raz and Tal, 2019: There is an oracle A, such that $\mathbf{BQP}^A \not= \mathbf{PH}^A$

IP

IP (Interactive Proof)

$L \in \mathbf{IP}$:

- $x \in L$) there exists prover P that the verifier accepts w.p. $2/3$
- $x \notin L$) for all provers P the verifier rejects w.p. $2/3$

Theorem (Shamir)

IP = PSPACE

MIP

MIP (Multi-Prover Interactive Proof)

Same as IP, except that now the verifier can exchange messages with many provers, not just one. The provers cannot communicate with each other during the execution of the protocol, so the verifier can "cross-check" their assertions (as with suspects in separate interrogation rooms).

Theorem (Babai, Fortnow, and Lund)

MIP = NEXP

QIP

QIP (Quantum Interactive Proof)

The class of decision problems such that a "yes" answer can be verified by a **quantum interactive proof**. Here the verifier is a **BQP** algorithm, the prover has unbounded computational resources.
The prover and verifier exchange a polynomial number of messages, which can be quantum states. Thus, the verifier's and prover's states may become entangled during the course of the protocol. Given the verifier's algorithm, we require that:

- $x \in L$) there exists prover P that the verifier accepts w.p. $2/3$
- $x \notin L$) for all provers P the verifier rejects w.p. $2/3$

Properties

QIP(1) is called **QMA**
QIP[k] = **QIP[3]** = **QIP** (Kitaev and Watrous)
QIP = **IP** = **PSPACE** (Jain, Ji, Upadhyay, and Watrous)

QMIP, MIP*

QMIP (Quantum Multi-Prover Interactive Proofs)

The quantum generalization of MIP, and the multi-prover generalization of QIP.

MIP* (MIP With Quantum Provers)

Same as MIP, except that the provers can share arbitrarily many entangled qubits. The verifier is classical, as are all messages between the provers and verifier.

Properties

NEXP $\subseteq$ MIP*
QMIP $=$ MIP*
MIP* $=$ NEXP

MIP*=RE

$\mathsf{MIP}^* = \mathsf{RE}$

Zhengfeng Ji*[1], Anand Natarajan†[2,3], Thomas Vidick‡[3], John Wright§[2,3,4], and Henry Yuen¶[5]

We show that the class MIP^* of languages that can be decided by a classical verifier interacting with multiple all-powerful quantum provers sharing entanglement is equal to the class RE of recursively enumerable languages. Our proof builds upon the quantum low-degree test of (Natarajan and Vidick, FOCS 2018) and the classical low-individual degree test of (Ji, et al., 2020) by integrating recent developments from (Natarajan and Wright, FOCS 2019) and combining them with the recursive compression framework of (Fitzsimons et al., STOC 2019).

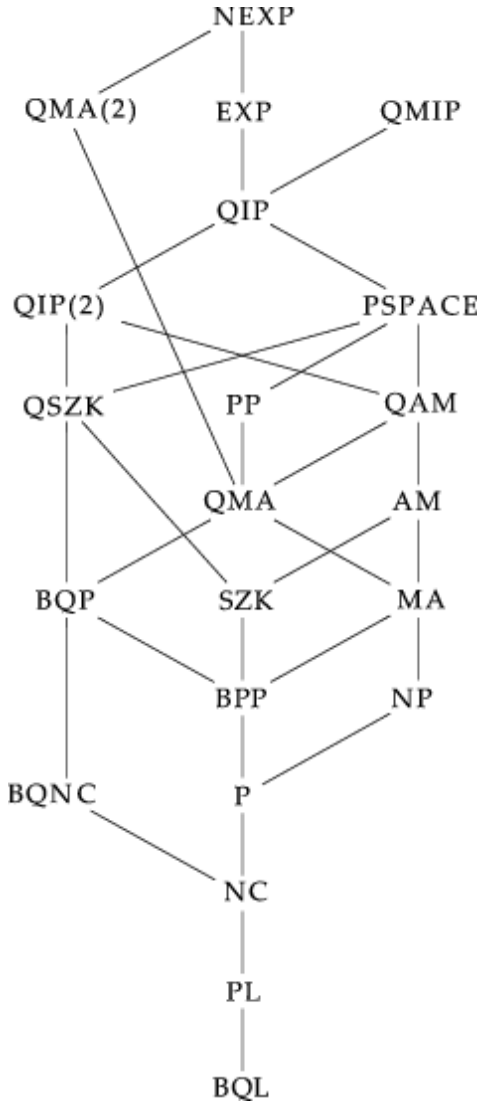
An immediate byproduct of our result is that there is an efficient reduction from the Halting Problem to the problem of deciding whether a two-player nonlocal game has entangled value 1 or at most $\frac{1}{2}$. Using a known connection, undecidability of the entangled value implies a negative answer to Tsirelson's problem: we show, by providing an explicit example, that the closure C_{qa} of the set of quantum tensor product correlations is strictly included in the set C_{qc} of quantum commuting correlations. Following work of (Fritz, Rev. Math. Phys. 2012) and (Junge et al., J. Math. Phys. 2011) our results provide a refutation of Connes' embedding conjecture from the theory of von Neumann algebras.

Implications

1. There is a protocol by which two entangled provers can convince a polynomial-time verifier of the answer to any computable problem whatsoever, or indeed that a given Turing machine halts.
2. There is a two-prover game for which Alice and Bob can do markedly better with an infinite amount of entanglement than they can with any finite amount of entanglement.
3. There is no algorithm even to approximate the entangled value of a two-prover game (i.e., the probability that Alice and Bob win the game, if they use the best possible strategy and as much entanglement as they like). Instead, this problem is equivalent to the halting problem.

Implications of MIP* = RE

4. There are types of correlations between Alice and Bob that can be produced using infinite entanglement, but that can't even be approximated using any finite amount of entanglement.
5. The undecidability result disproves the Connes embedding conjecture, a central conjecture from the theory of operator algebras and the Tsirelson conjecture in quantum information theory.
6. It is one of the first non-relativizing results in quantum computability theory.


NEXP
QMA(2)
EXP
QMIP
QIP
QIP(2)
PSPACE
QSZK
PP
QAM
QMA
AM
BQP
SZK
MA
BPP
NP
BQNC
P
NC
PL
BQL

Further Reading

- An Introduction to Quantum Computing
  P. Kaye, R. Laflamme, M. Mosca
- Quantum Computation and Quantum Information
  M.A. Nielsen, I.C. Chuang
- Lecture notes of John Preskill on Quantum Computation
  `http://theory.caltech.edu/~preskill/ph229/`
- Classical and Quantum Computation
  A. Yu. Kitaev, A.H. Shen, M.N. Vyalyi
- Quantum Computing Since Democritus
  Scott Aaronson
- Shtetl-Optimized
  Scott Aaronson's blog: `https://scottaaronson.blog/`