Quantum Computing

BQP is in PSPACE

Term Project:

E0 224: Computational Complexity Theory

IISc Bengaluru

Manish Kumar

M. Tech, IISc Bengaluru

BS (PHY), IIT Kanpur

Course Instructor: Prof. Chandan Saha (IISc)

TA: Bhargav Thankey (IISc)

Prelude

Why Quantum Computing?

Prelude

Why Quantum Computing?

Mentioned at least twice in the course(!)

Prelude

Why Quantum Computing?

Mentioned at least twice in the course(!)

- **Strong Church-Turing thesis**: *"Every physically realizable computation device – whether it's based on silicon, DNA, neurons or some other alien technology – can be simulated efficiently by a Turing machine".*

Possible exception: Quantum machines!

Image source: Slide #14 Lecture-2 @ CCT by prof. Chandan [Course Link]

Prelude

Why Quantum Computing?

Mentioned at least twice in the course(!)

- Possible violation of strong Church-Turing thesis
- FACTORING is in BQP

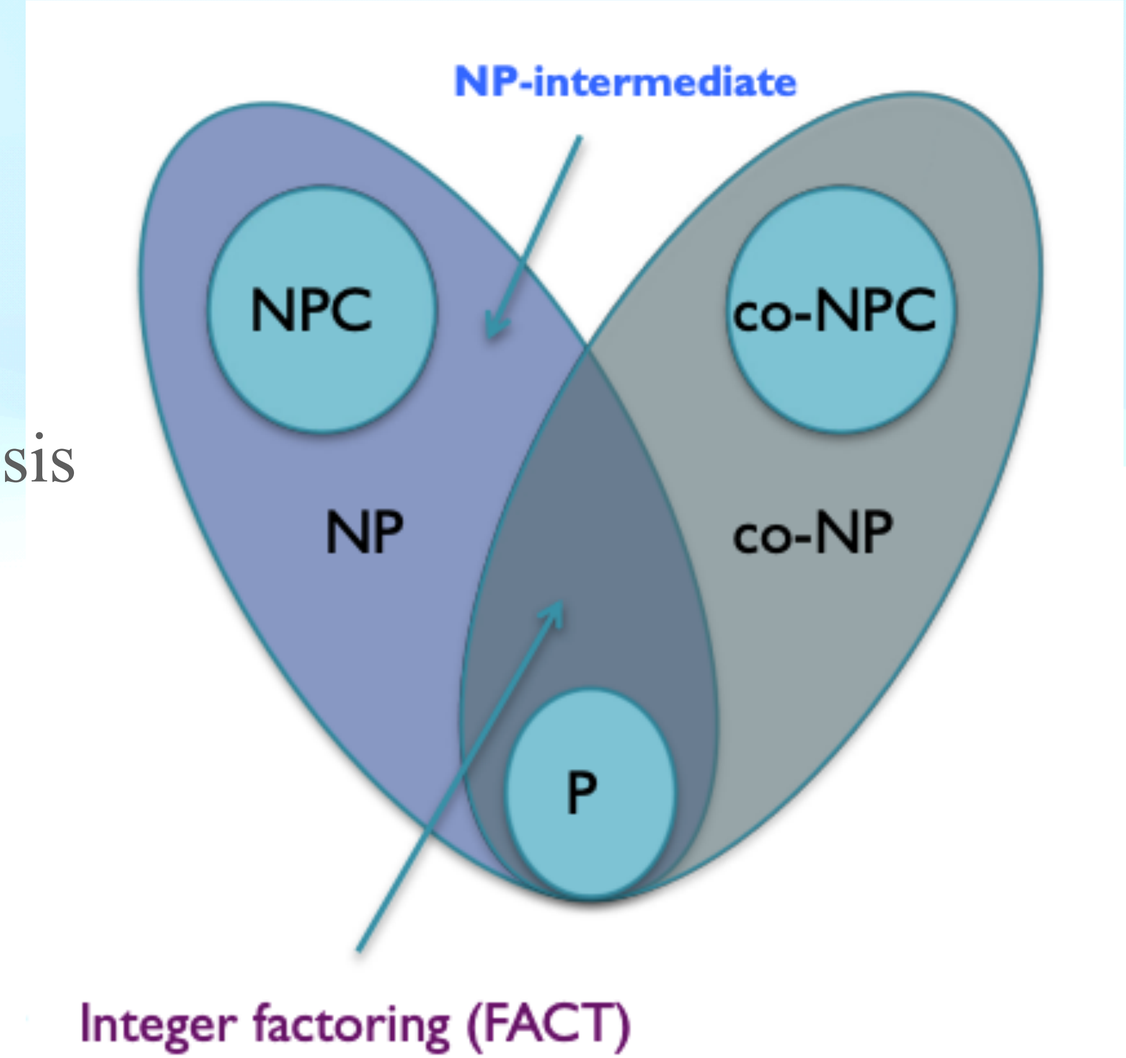


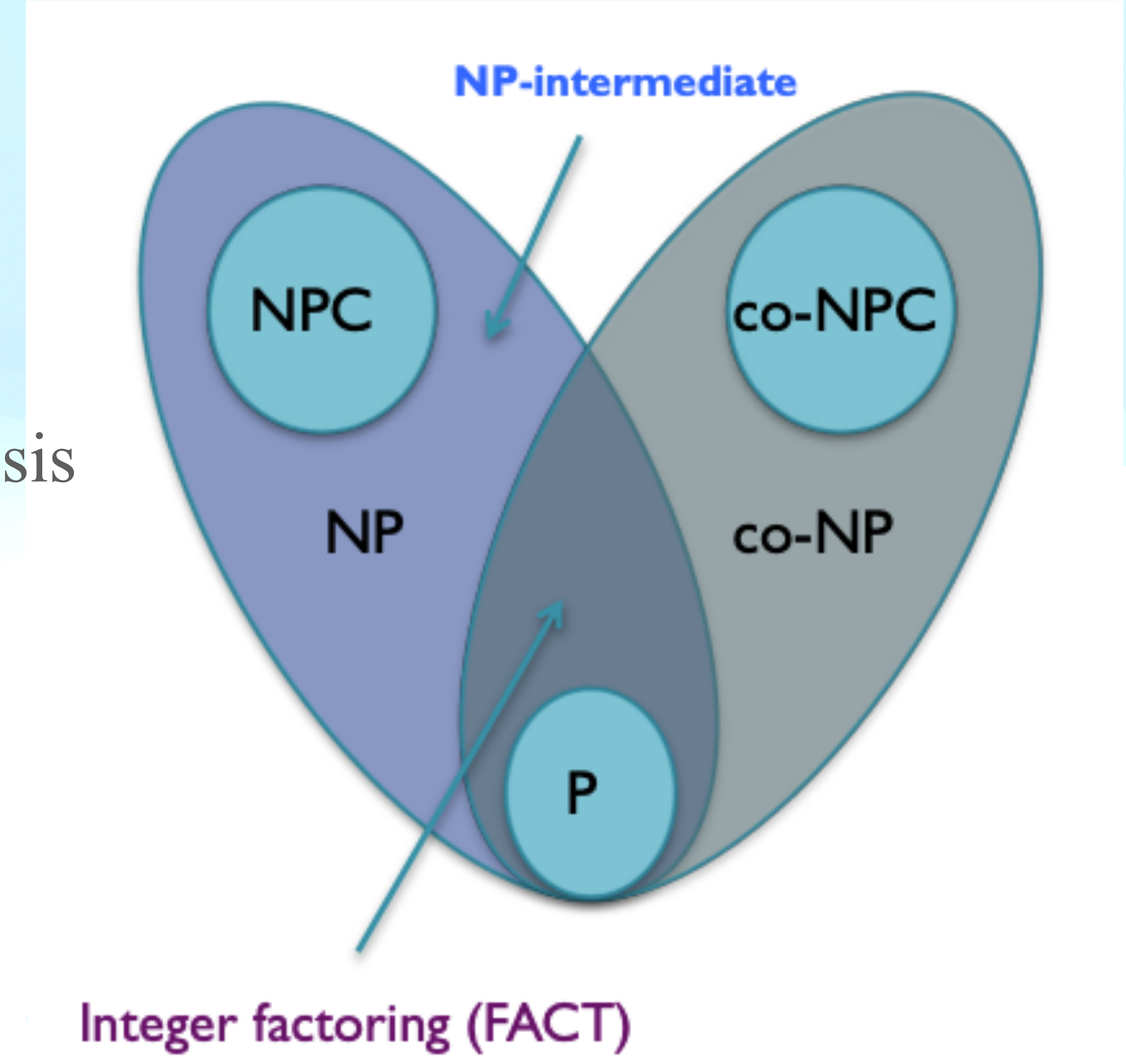
Image sources: lectures (Prof. Chandan)

Prelude

Why Quantum Computing?

Mentioned at least twice in the course(!)

- Possible violation of strong Church-Turing thesis
- FACTORING is in BQP



Other reasons:

$$P \neq BQP \implies P \neq PSPACE$$

(Due to $BQP \subseteq PSPACE$)

Image sources: the lectures

MAIN GOAL FOR TODAY

MAIN GOAL FOR TODAY

Make sense of the below diagram

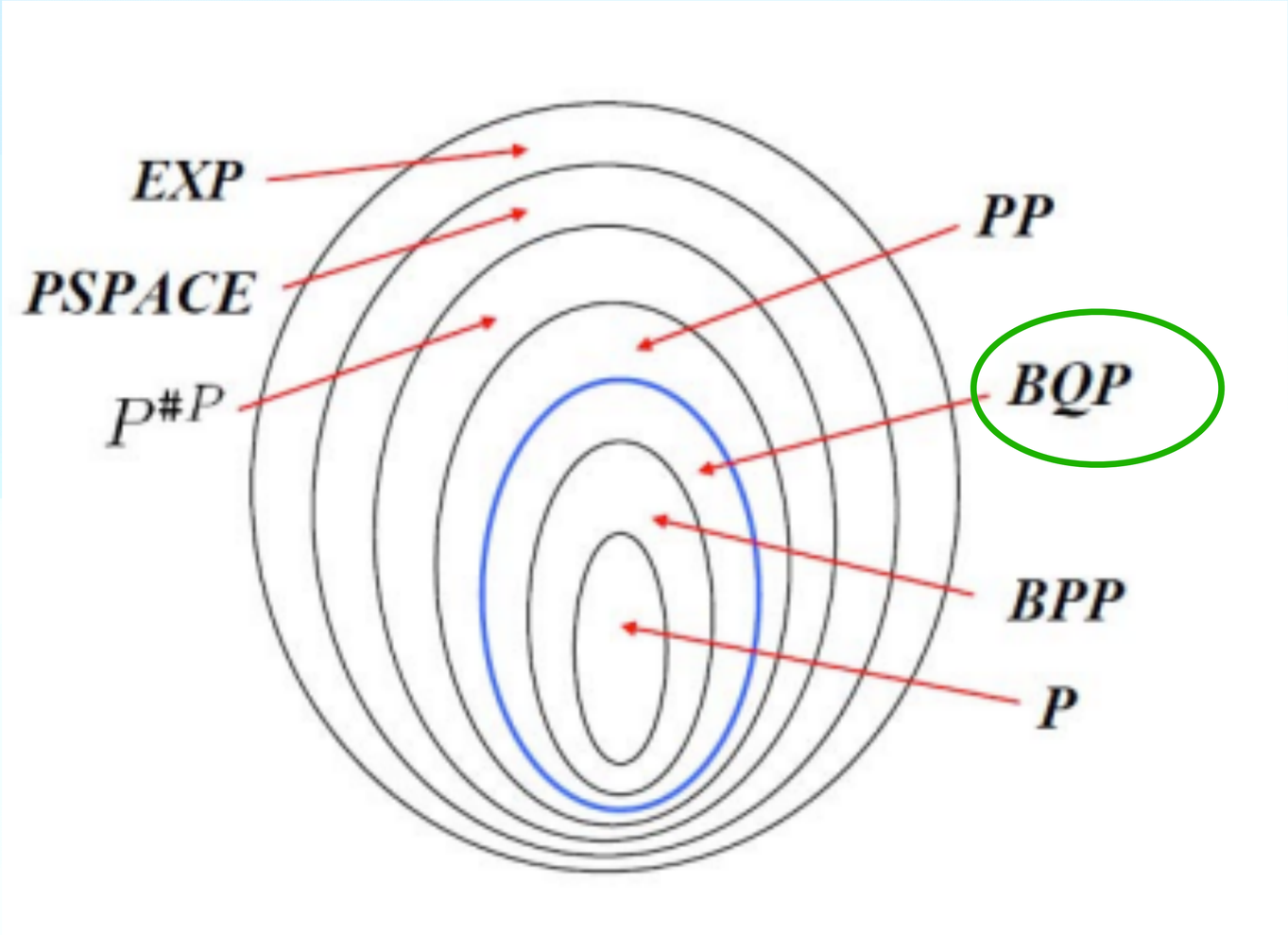
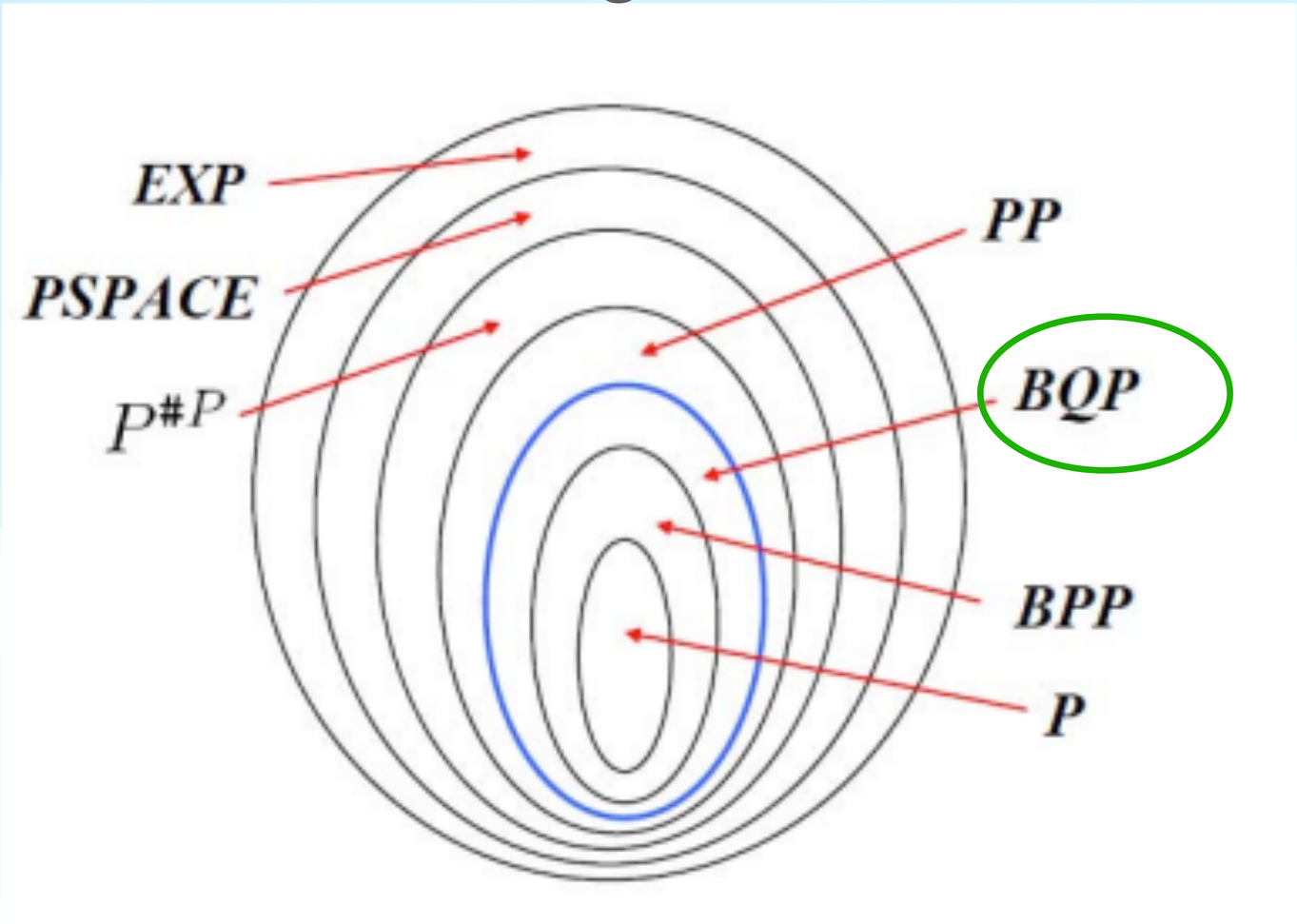


Image Source: Scott Aaronson (MIT6_845F10_lec04)

MAIN GOAL FOR TODAY

Make sense of the below diagram



$$P \subseteq BPP \subseteq BQP \subseteq PSAPCE \subseteq EXP$$

Image Source: Scott Aaronson (MIT6_845F10_lec04)

MAIN GOAL FOR TODAY

If time permits…

Non-Determinism VS Quantum-ness

(Comparison of two computational resources)

MAIN GOAL FOR TODAY

If time permits…

Non-Determinism VS Quantum-ness

Could there be a problem in **NP** but not in **BQP**?

Theorem: $\exists O \ s.t \ NP^O \neq BQP^O$ [BBBV-97]

MAIN GOAL FOR TODAY

If time permits…

Non-Determinism VS Quantum-ness

Could there be a problem in **NP** but not in **BQP**?

Theorem: $\exists O \ s.t \ NP^O \neq BQP^O$ [BBBV-97]

Could there be a problem in **BQP** but not in **NP**?

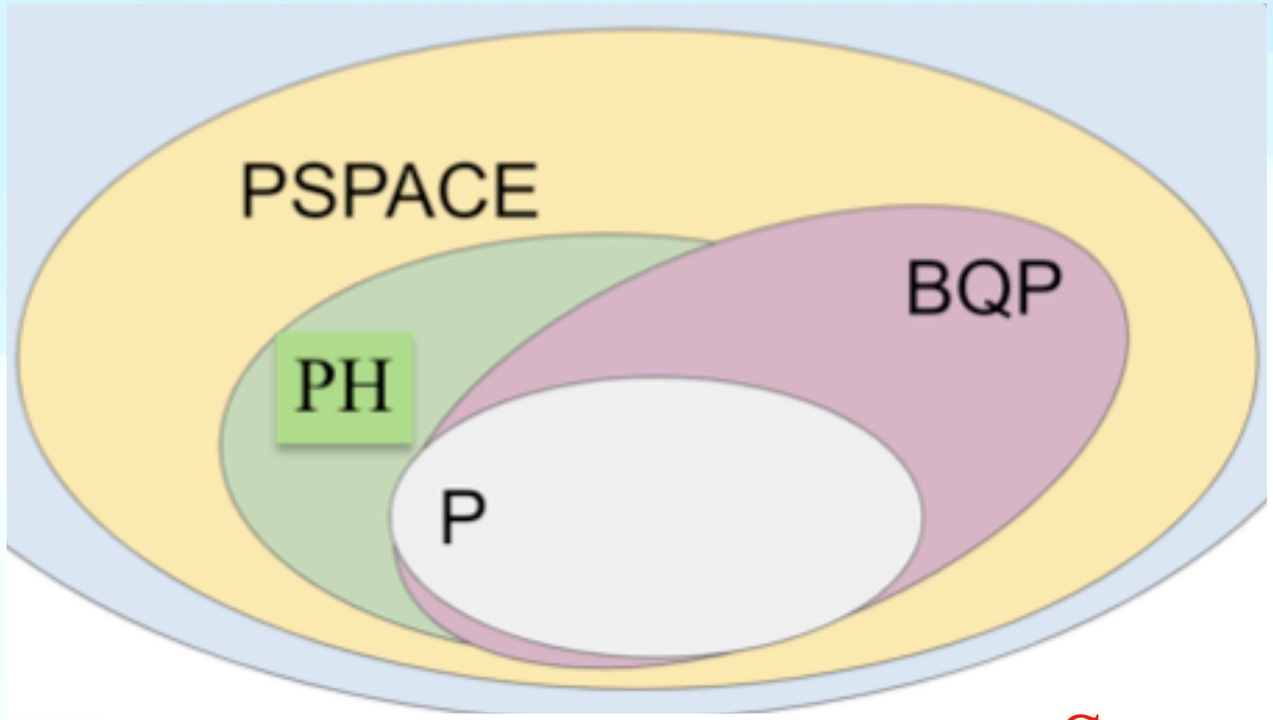
Theorem: $\exists O \ s.t \ BQP^O \neq PH^O$ [Raz-Tal-18]

(Note: $NP \subseteq PH$)

MAIN GOAL FOR TODAY

If time permits…

Non-Determinism VS Quantum-ness



Suspected Relationship

Google images

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

Quantum Circuit Model: Register and Gate Operation

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

Quantum Circuit Model: Register and Gate Operation

Defining BQP : Efficient Quantum Computation

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

Quantum Circuit Model: Register and Gate Operation

Defining BQP : Efficient Quantum Computation

Upper bounds on BQP: APPROX. QUANTUM CIRCUIT problem

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

Quantum Circuit Model: Register and Gate Operation

Defining BQP : Efficient Quantum Computation

Upper bounds on BQP: APPROX. QUANTUM CIRCUIT problem

Promise problem

$BQP \subseteq PSAPCE \subseteq EXP$

CONTENT of the TALK

Quantum Bits: Quantum state and Superposition

Basic Maths: Linear algebra over complex field

Quantum Circuit Model: Register and Gate Operation

Defining BQP : Efficient Quantum Computation

Upper bounds on BQP: APPROX. QUANTUM CIRCUIT problem

Promise problem

$BQP \subseteq PSAPCE \subseteq EXP$

Reversible Computation: $P \subseteq BPP \subseteq BQP$

CONTENT of the TALK

…. if time permits, then:

Unstructured Search Problem: $NP \not\subset BQP$ **conjecture**

CONTENT of the TALK

…. if time permits, then:

Unstructured Search Problem: $NP \not\subset BQP$ **conjecture**

Forrelation Problem: $BQP \not\subset PH$ **conjecture**

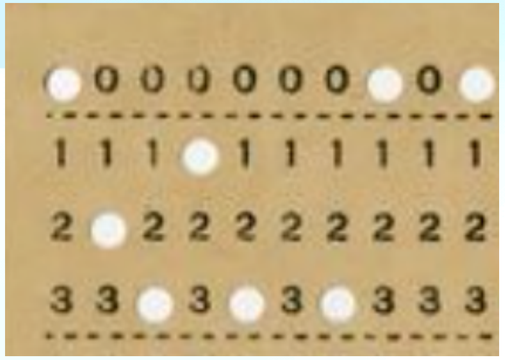
Quantum Bits:
(Quantum state and Superposition)

Classical information: Bits (**B**inary Dig**its**) → State-I: **0**, State-II: **1**

Quantum Bits:
Quantum state and Superposition

Classical information: Bits (Binary Digits)

Punched Card:

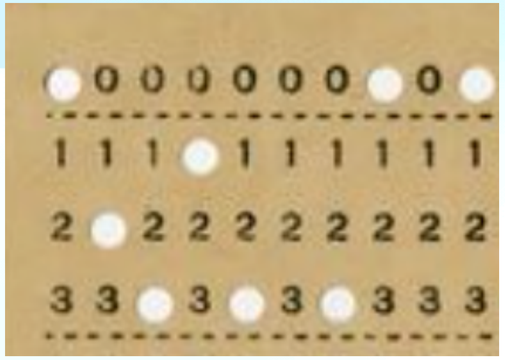


Google images

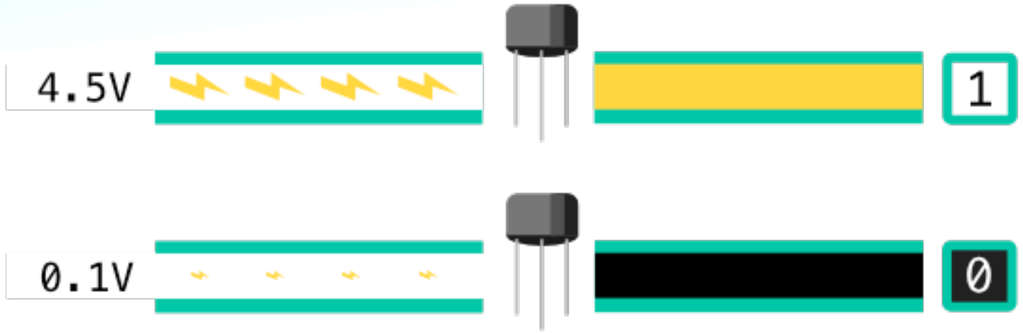
Quantum Bits:
Quantum state and Superposition

Classical information: Bits (Binary Digits)

Punched Card:



State of transistor:

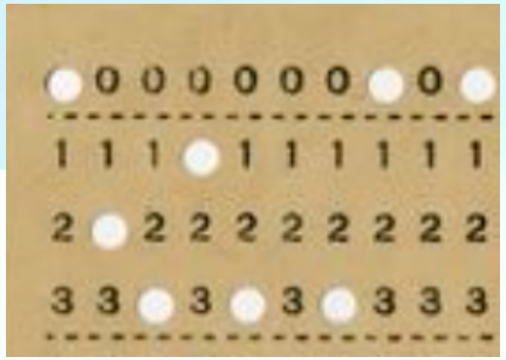


Google images

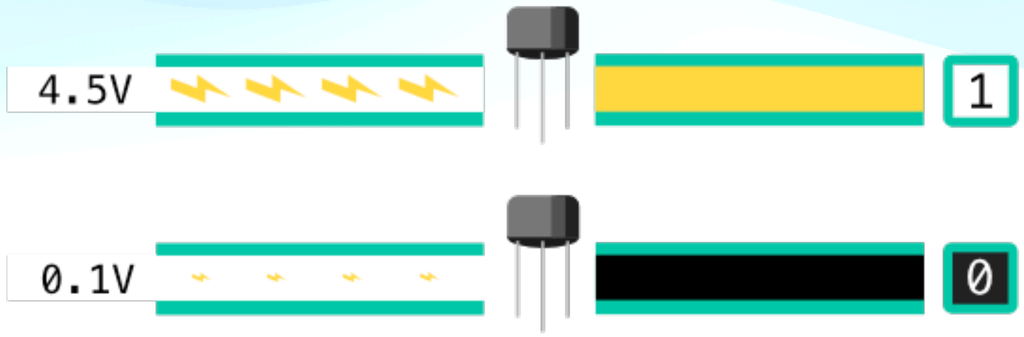
Quantum Bits:
Quantum state and Superposition

Classical information: Bits (Binary Digits)

Punched Card:



State of transistor:



What is classical about Classical Computation?

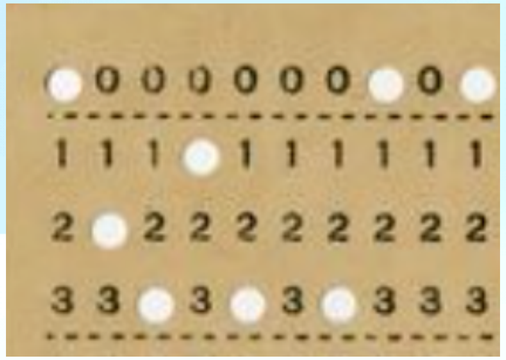
Google images

Quantum Bits:

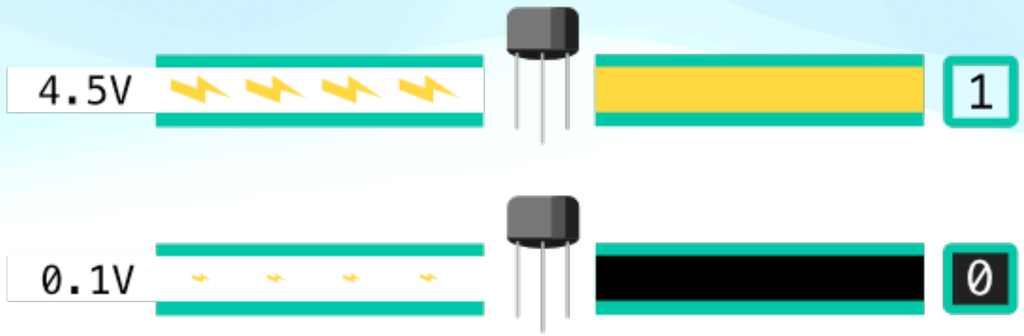
Quantum state and Superposition

Classical information: Bits (Binary Digits)

Punched Card:



State of transistor:



What is classical about Classical Computation?

Information: Bits
Operation: Manipulation of bits

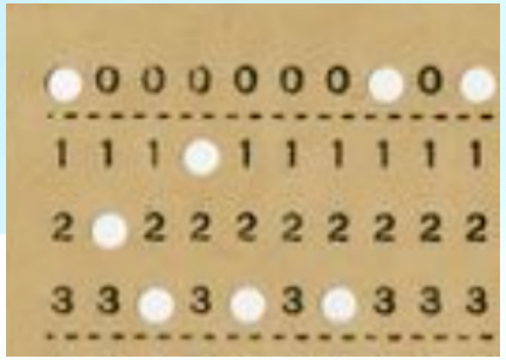
Google images

Quantum Bits:

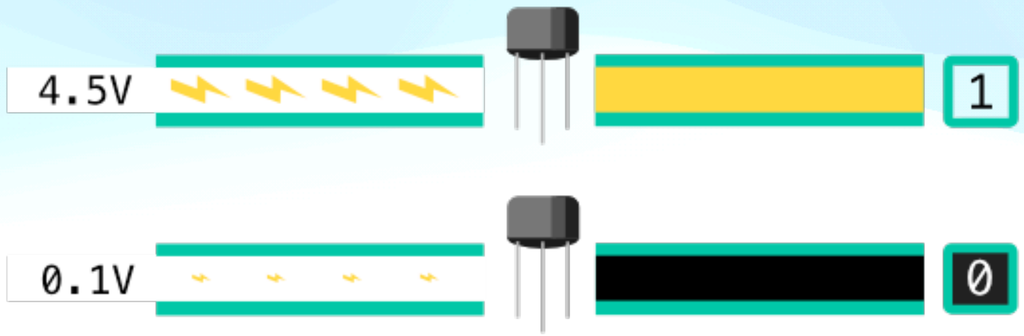
Quantum state and Superposition

Classical information: Bits (Binary Digits)

Punched Card:



State of transistor:



What is classical about Classical Computation?

Information: Bits

Operation: Manipulation of bits

Classical Physics can explain it.

Google images

Quantum Superposition: Informal notion

Quantum information: Qubits (**Qu**antum **Bits**)

How are they differ from classical?

Quantum Superposition: Informal notion

Quantum information: Qubits (**Qu**antum **Bits**)

How are they differ from classical?

Quantum System*

State-I: **0**

State-II: **1**

---

* photons, electrons and any two level Quantum system

Quantum Superposition: Informal notion

Quantum information: Qubits (**Qu**antum **Bits**)

How are they differ from classical?

Quantum System*

State-I: **0**

State-II: **1**

State-III: Both **0** and **1** at the same time

---

* photons, electrons and any two level Quantum system

Quantum Bits:
Quantum state and Superposition

Making sense of state-III: Both **0** and **1** at the same time

System
(in state-III)

Measure

$$\begin{cases} \mathbf{0} & ;\ p_0 \text{ probability} \\ \mathbf{1} & ;\ p_1 \text{ probability} \end{cases}$$

$$p_0 + p_1 = 1$$

Quantum Bits:
Quantum state and Superposition

Sense making of state-III: Both **0** and **1** at the same time

System
(in state-III)

Measure →

$$\begin{cases} \mathbf{0} & ;\ p_0 \text{ probability} \\ \mathbf{1} & ;\ p_1 \text{ probability} \end{cases}$$

$$p_0 + p_1 = 1$$

- (Inherently*) probabilistic event
- p_0 *and* p_1 quantified by Quantum state (wave-function)

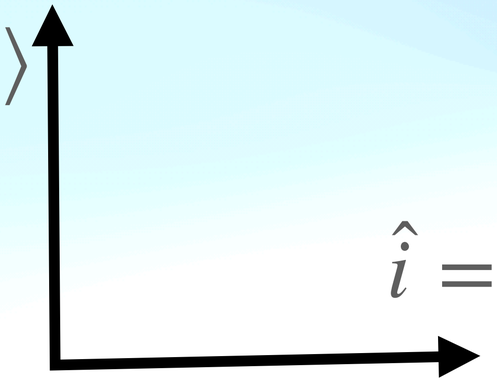
* see Hidden variable theory: Link

Quantum Superposition: Formal notion

Linear algebra (over Real field):

Various notations for a unit vector:

$$\hat{j} = [0,\ 1]^T = |1\rangle$$

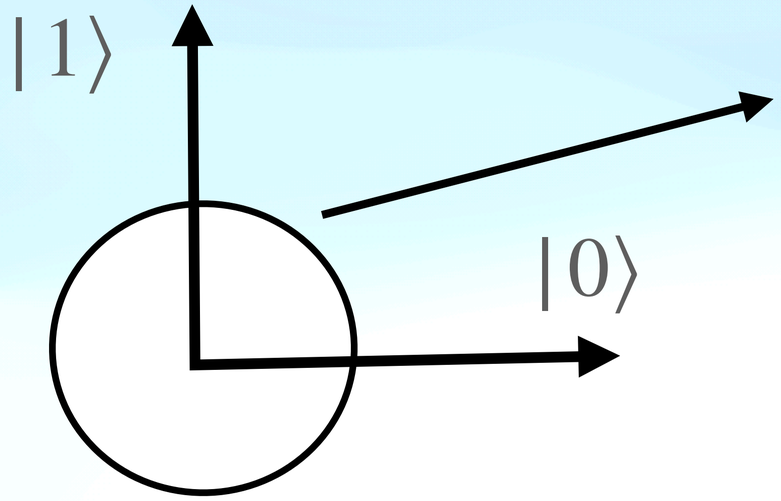


$$\hat{i} = [1,\ 0]^T = |0\rangle$$

Quantum Superposition: Formal notion

Linear algebra (over Real field):

A general unit vector:



$$a|0\rangle + b|1\rangle$$

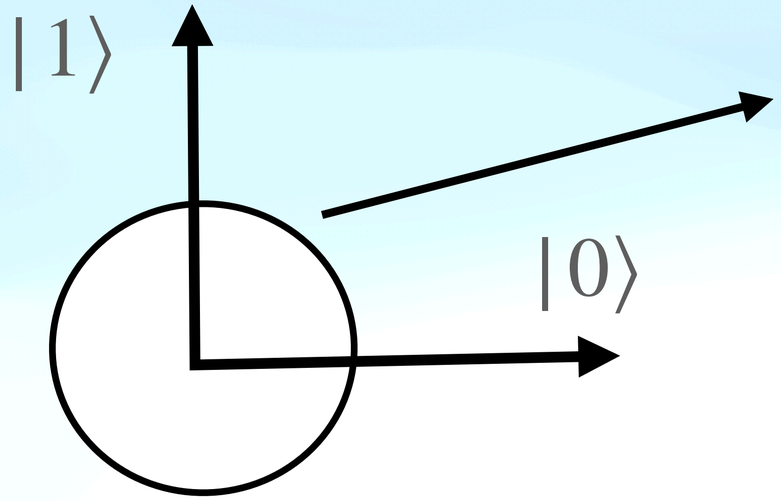
$$a^2 + b^2 = 1$$

$$a, b \in \mathbb{R}$$

Quantum Superposition: Formal notion

Linear algebra (over Real field):

A general unit vector:



$$a|0\rangle + b|1\rangle$$

$$a^2 + b^2 = 1$$

$$a, b \in \mathbb{R}$$

Note: One parameter is sufficient to represent a point on the unit circle.

Quantum Superposition: Formal notion

Linear algebra (over Complex field):

What if we allow complex numbers?

$$\alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Quantum Superposition: Formal notion

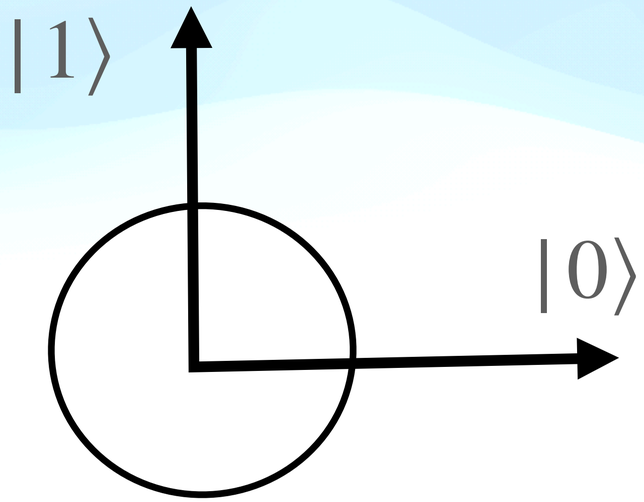
Linear algebra (over Complex field):

What if we allow complex numbers?

$$\alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$



Euclidean Space

Can we represent the vector on the circle?

Quantum Superposition: Formal notion

Linear algebra (over Complex field):

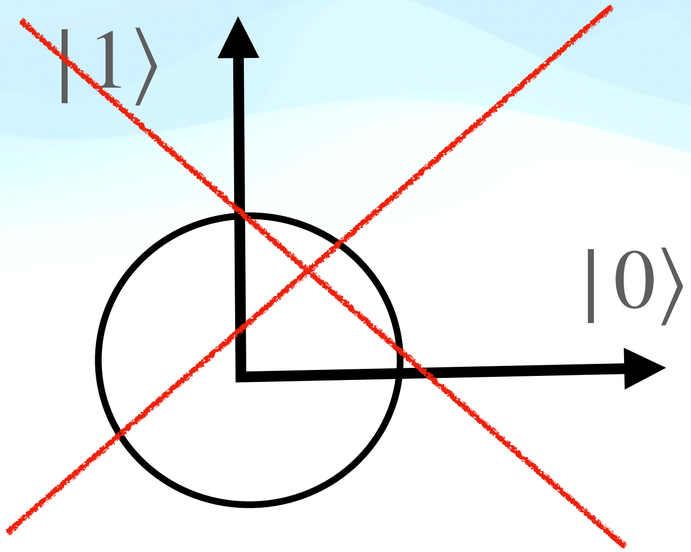
What if we allow complex numbers?

$$\alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Can we represent the vector on the circle?



Euclidean Space

Quantum Superposition: Formal notion

Linear algebra (over Complex field):

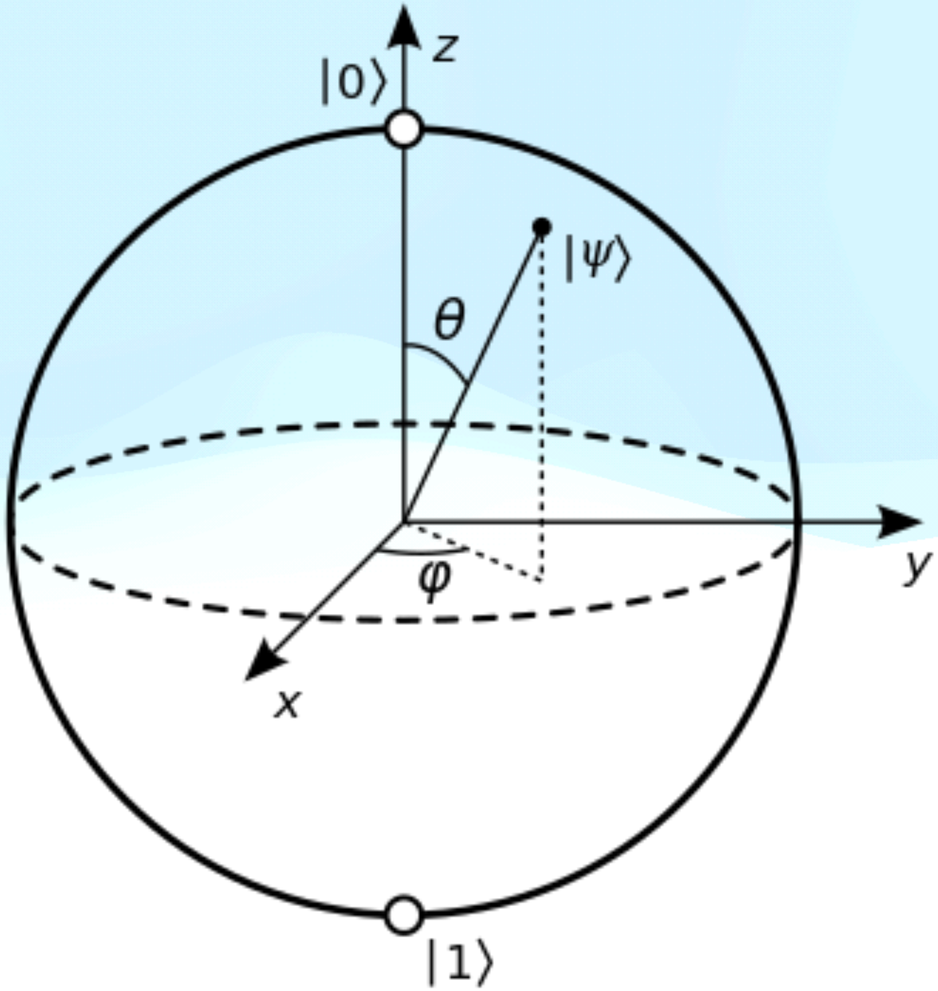
What if we allow complex numbers?

$$\alpha|0\rangle + \beta|1\rangle = |\psi\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Bloch sphere: Where state vectors for a qubit resides



Bloch Sphere

Equivalent to 2D Hilbert Space: more at stack exchange Link

Image: Wikipedia

Quantum Superposition: Formal notion

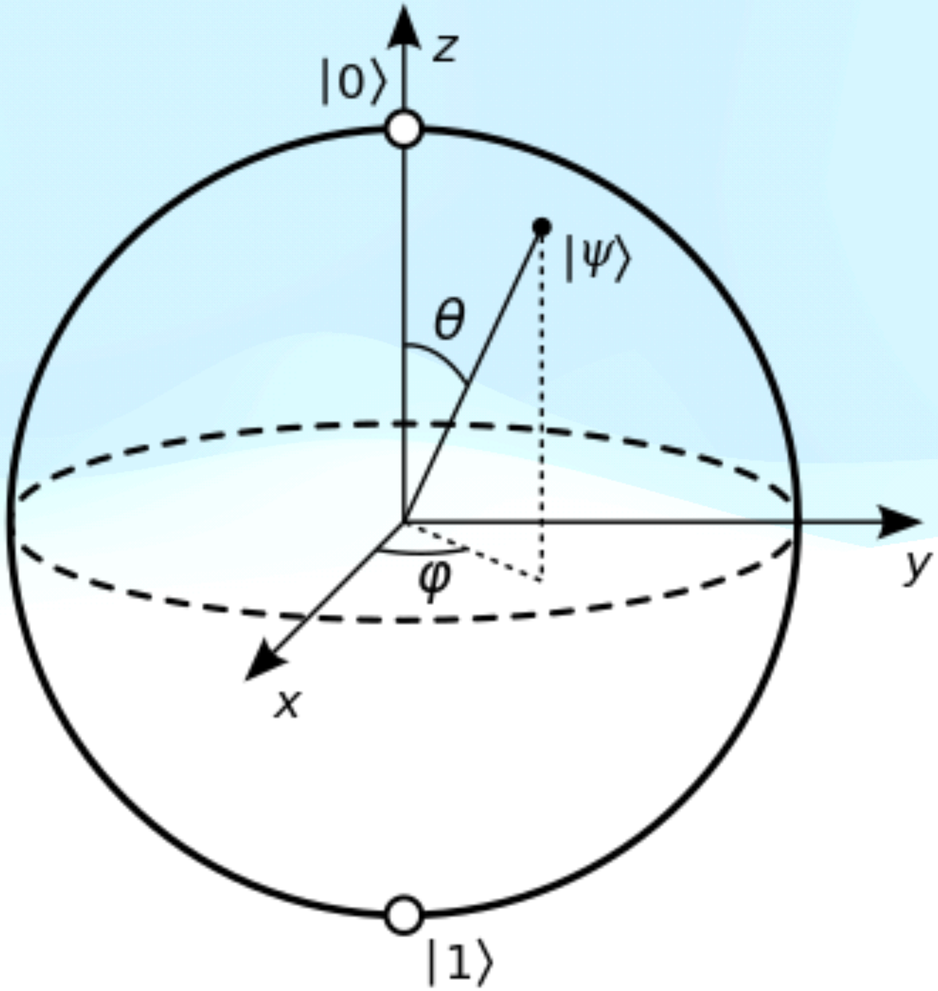
Linear algebra (over Complex field):

What if we allow complex numbers?

$$\alpha|0\rangle + \beta|1\rangle = |\psi\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

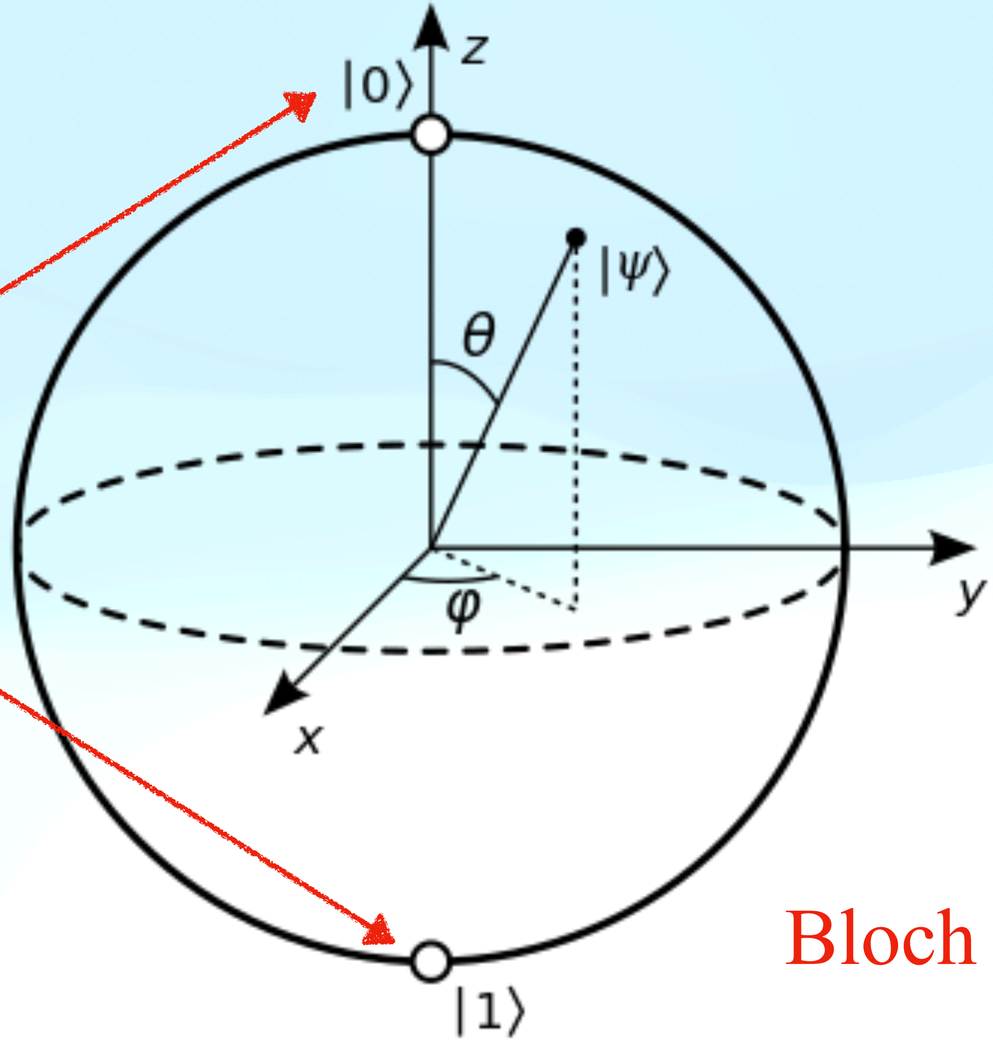


Note: Two parameter is sufficient to represent a point on the Bloch sphere.

Equivalent to 2D Hilbert Space: more at stack exchange Link

Quantum Superposition: Formal notion

Classical states

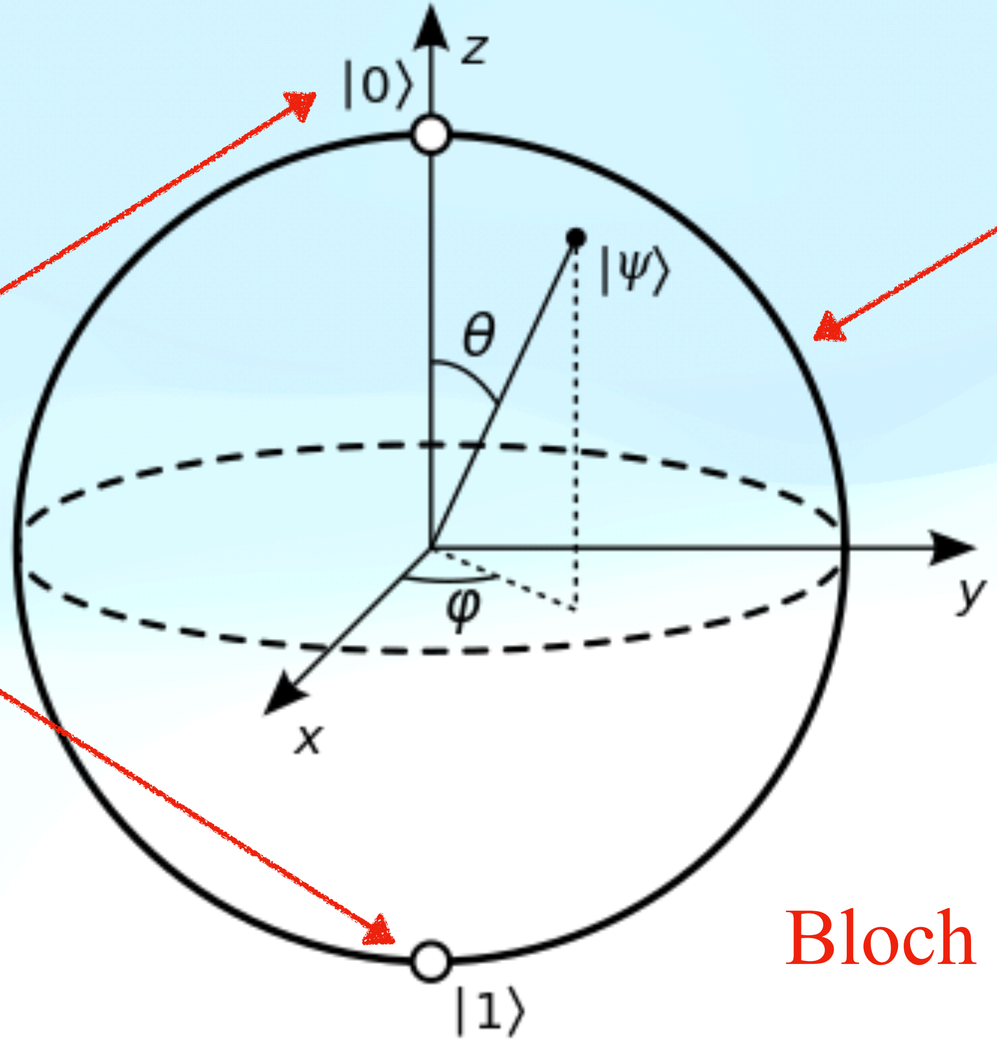


Bloch Sphere

\

Quantum Superposition: Formal notion

Classically available



(Purely) Quantum

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Bloch Sphere

Quantum Superposition

Informal

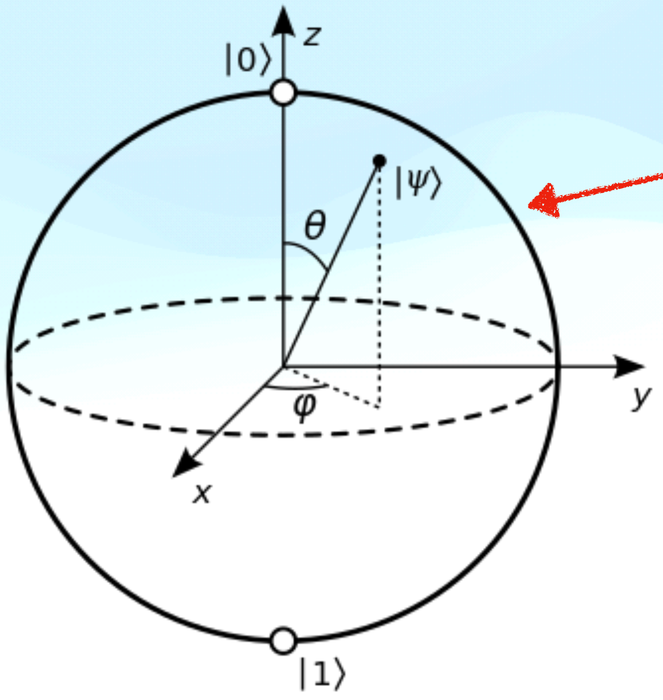
Quantum System*

State-I: **0**

State-II: **1**

State-III: Both **0** and **1** at the same time

Formal



Various realisations of state-III

Quantum Measurement

Two key questions to ask:

What is measured?:

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

How it is measured?:

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

How it is measured?: Measurement basis: $\{|0\rangle, |1\rangle\}$

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

How it is measured?: Measurement basis: $\{|0\rangle, |1\rangle\}$

Linear Algebra Recap:

Orthonormal Basis of vector space V

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

How it is measured?: Measurement basis: $\{|0\rangle, |1\rangle\}$

Linear Algebra Recap:

(Orthonormal) Basis of vector space V

Def: a subset of orthonormal vectors in V that spans V.

Quantum Measurement

Two key questions to ask:

What is measured?: System/ Sub-system: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

How it is measured?: Measurement basis: $\{|0\rangle, |1\rangle\}$

Linear Algebra Recap:

(Orthonormal) Basis of vector space V

Def: a subset of orthonormal vectors in V that spans V.

Ex: set $\{|0\rangle, |1\rangle\}$ on Bloch Sphere

Orthonormality: $\langle 0|1\rangle = 0$ *and* $\langle 0|0\rangle = \langle 1|1\rangle = 1$

Quantum Measurement

Linear Algebraic view:

State of the system: a vector in the vector space

Note: a vector quantity.

Projection of the sate: Projecting the state vector on the basis vector

Note: a (complex) scalar quantity

Quantum Measurement

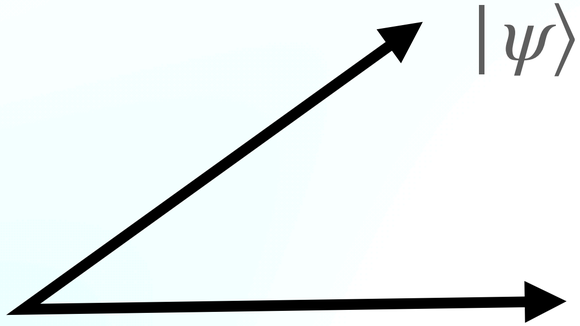
Linear Algebraic view:

State of the system: a vector in the vector space

Note: a vector quantity.

Projection of the sate: Projecting the state vector on the basis vector

Note: a (complex) scalar quantity



$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|0\rangle$$

$$Projection : \langle 0|\psi\rangle = \alpha$$

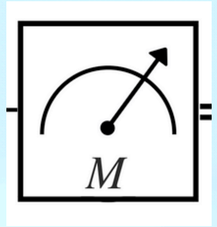
$$|\langle 0|\psi\rangle|^2 = |\alpha|^2$$

It has a physical meaning

Meaning of measurement

Step wise process of the measurement in standard basis:

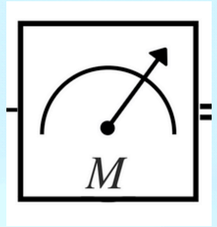
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$



Meaning of measurement

Step wise process of the measurement in standard basis:

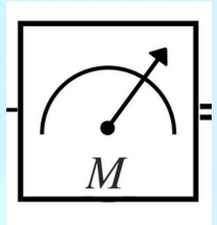
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$



$$Event - I \ or \ Event - II$$

Meaning of measurement

Step wise process of the measurement in standard basis:

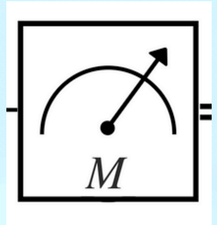
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$ 

$$Event - I \ or \ Event - II$$

Event-I: $|\psi\rangle \xrightarrow{\textbf{Measure}} |0\rangle$ **Pr**(Event-I) $= |\langle 0|\psi\rangle|^2 = |\alpha|^2$

Meaning of measurement

Step wise process of the measurement in standard basis:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$ 

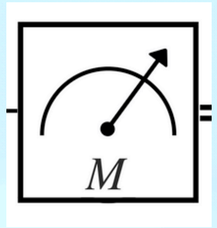
$$Event - I \ or \ Event - II$$

Event-I: $|\psi\rangle \xrightarrow{\textbf{Measure}} |0\rangle$ **Pr**(Event-I) $= |\langle 0|\psi\rangle|^2 = |\alpha|^2$

Event-II: $|\psi\rangle \xrightarrow{\textbf{Measure}} |1\rangle$ **Pr**(Event-II) $= |\langle 1|\psi\rangle|^2 = |\beta|^2$

Meaning of measurement

Step wise process of the measurement in standard basis:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$ 

$$Event - I \ or \ Event - II$$

Event-I: $|\psi\rangle \xrightarrow{\textbf{Measure}} |0\rangle$ **Pr**(Event-I) $= |\langle 0|\psi\rangle|^2 = |\alpha|^2$

Event-II: $|\psi\rangle \xrightarrow{\textbf{Measure}} |1\rangle$ **Pr**(Event-II) $= |\langle 1|\psi\rangle|^2 = |\beta|^2$

$$|\langle i|\psi\rangle|^2$$

It has a physical meaning

What's the probability that system collapse to state $|i\rangle$ on measurement.

Summary of Quantum Jargon

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$: State Vector (Wave function)

$\{|0\rangle, |1\rangle\}$: Standard Basis

$Projection : \langle 0|\psi\rangle = \alpha$: Probability Amplitude

$\langle 0|\psi\rangle^2 = |\alpha|^2$: Measurement Probability

Quantum Operations (Q. Ops.)

Computation requires manipulation of information

Quantum Operation: Manipulating the state in a specific way.

$$|\psi\rangle \xrightarrow{\textbf{Q. Ops.}} |\phi\rangle$$

Quantum Operations (Q. Ops.)

Computation requires manipulation of information

Quantum Operation: Manipulating the state in a specific way.

$$|\psi\rangle \xrightarrow{\textbf{Q. Ops.}} |\phi\rangle$$

Def: a function F that maps its previous state to the new state and satisfies the following conditions:

Linearity: $F(\alpha|0\rangle + \beta|1\rangle) = \alpha \cdot F|0\rangle + \beta \cdot F|1\rangle)$

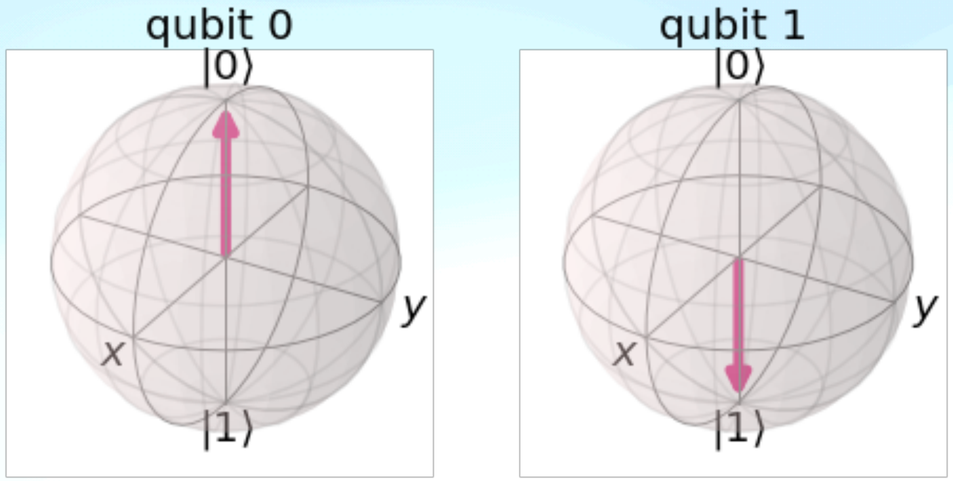
Norm Preserving: $||(|\psi\rangle)||_2 = ||(F|\psi\rangle)||_2$

Quantum Operations

Ex: Bit flip operation

$$|0\rangle \xrightarrow{\textbf{Bit Flip}} |1\rangle$$

Function view



Evolution on Bloch sphere

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

Matrix view

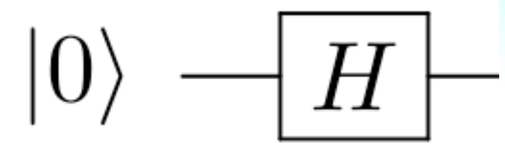
Quantum Operations

More on Matrix view:

$$Quantum\ Operation \leftrightarrow F \leftrightarrow Unitary\ matrix$$

$$Matrix\ Dim : U_{2\times 2}\ (for\ a\ qubit)$$

Hadamard Gate:



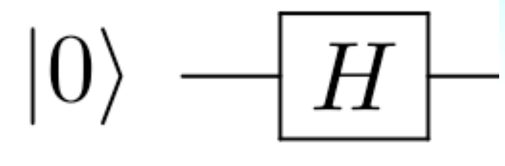
Quantum Operations

More on Matrix view:

$$Quantum\ Operation \leftrightarrow F \leftrightarrow Unitary\ matrix$$

$$Matrix\ Dim : U_{2\times 2}\ (for\ a\ qubit)$$

Hadamard Gate:



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

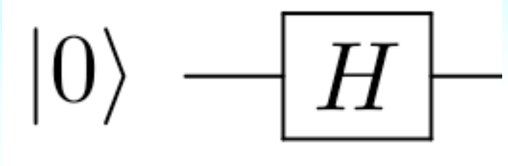
Quantum Operations

More on Matrix view:

$$Quantum\ Operation \leftrightarrow F \leftrightarrow Unitary\ matrix$$

$$Matrix\ Dim : U_{2\times 2}\ (for\ a\ qubit)$$

Hadamard Gate:



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

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

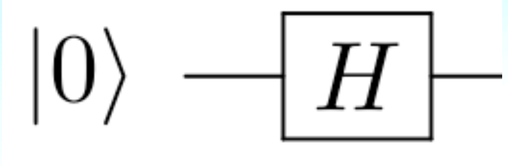
Quantum Operations

More on Matrix view:

$$Quantum\ Operation \leftrightarrow F \leftrightarrow Unitary\ matrix$$

$$Matrix\ Dim : U_{2\times 2}\ (for\ a\ qubit)$$

Hadamard Gate:

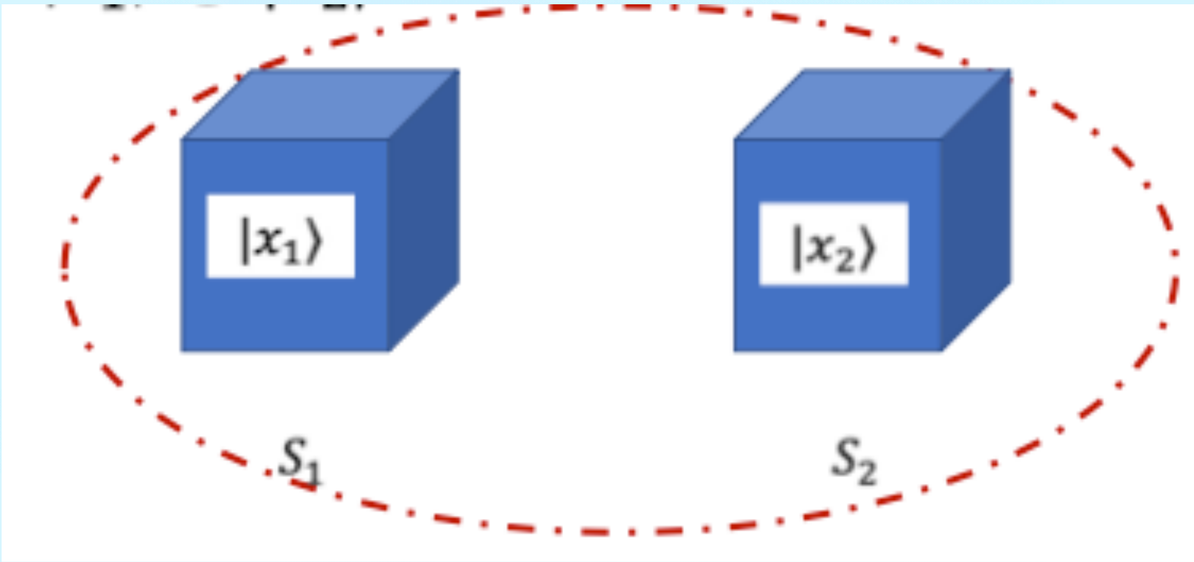


$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

- **Equal superposition state**
- **Measurement outcome is "like" Coin toss**

Multi-Qubit system



Composite system:

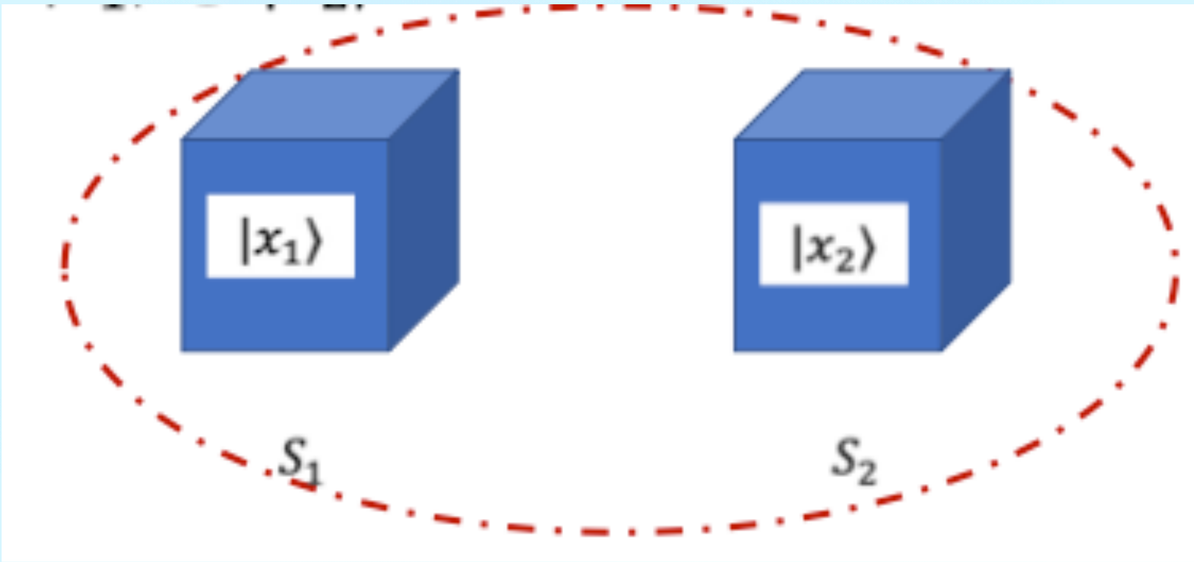
What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

How to measure?

Basis: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

Multi-Qubit system



What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

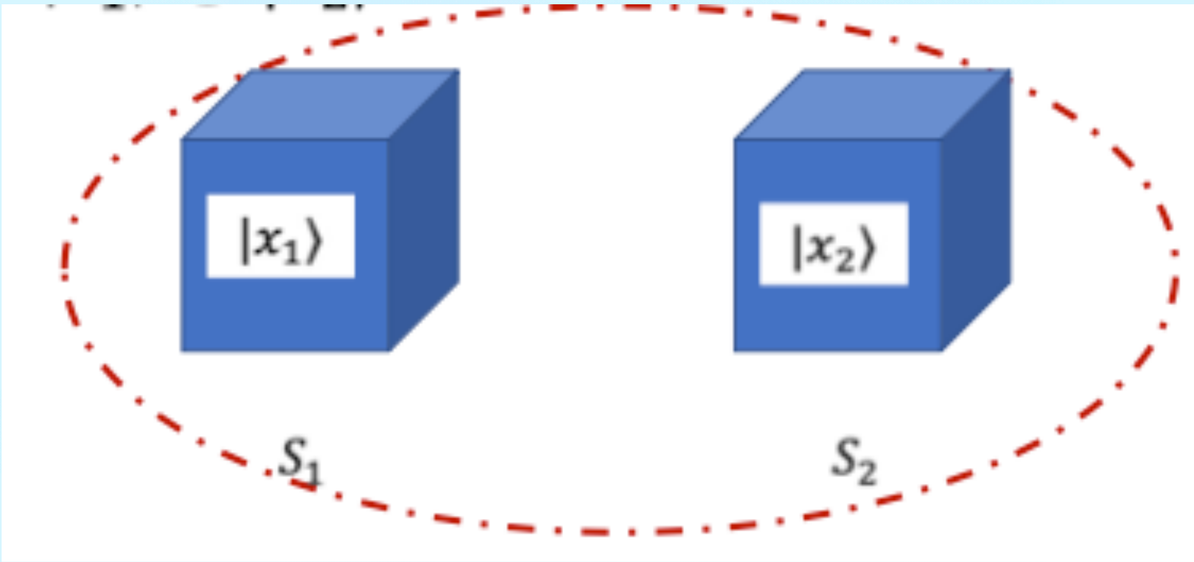
How to measure?

Basis: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

Composite system:

$$\begin{bmatrix} x \\ y \end{bmatrix} \otimes \begin{bmatrix} w \\ z \end{bmatrix} = \begin{bmatrix} xw \\ xz \\ yw \\ yz \end{bmatrix}$$

Multi-Qubit system



What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

How to measure?

Basis: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

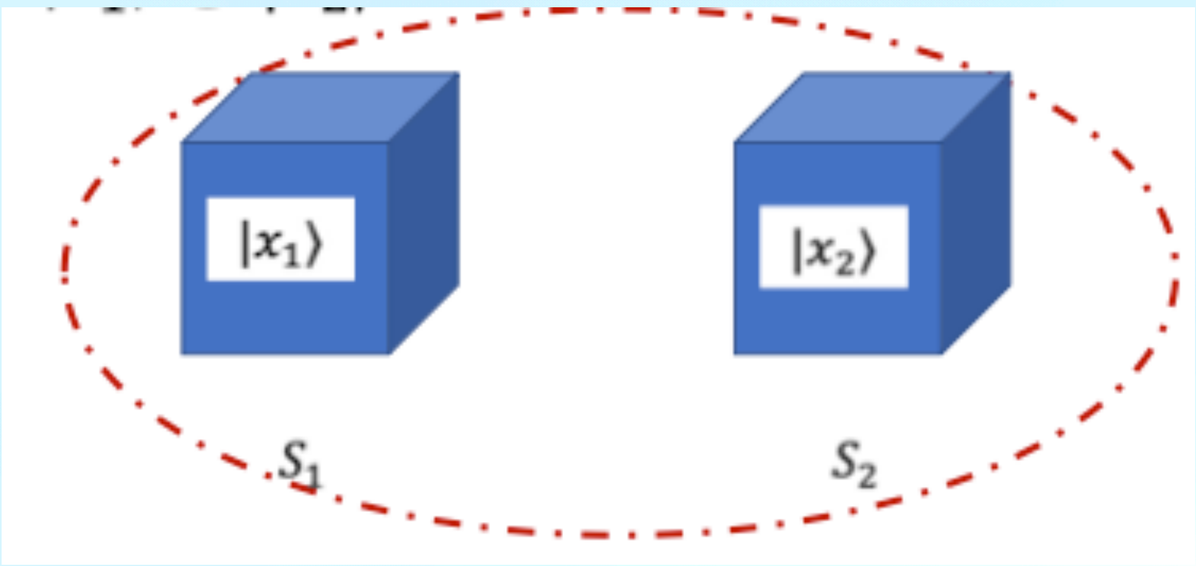
$$|x_1\rangle, |x_2\rangle \in \mathbb{C}^2$$

$$|\psi\rangle \in \mathbb{C}^{2^2} \in \mathbb{C}^4$$

Composite system:

$$\begin{bmatrix} x \\ y \end{bmatrix} \otimes \begin{bmatrix} w \\ z \end{bmatrix} = \begin{bmatrix} xw \\ xz \\ yw \\ yz \end{bmatrix}$$

Multi-Qubit system



Composite system:

What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

How to measure?

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

$$|x_1\rangle, |x_2\rangle \in \mathbb{C}^2$$

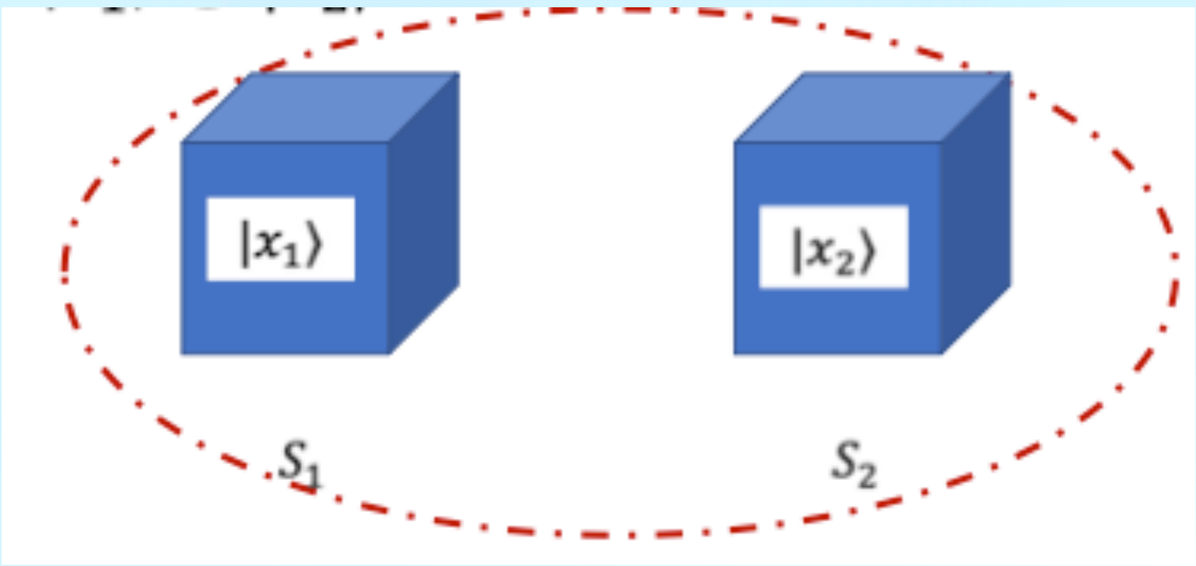
$$|\psi\rangle \in \mathbb{C}^{2^2} \in \mathbb{C}^4$$

N-qubit generalisation:

$$|x_1\rangle \otimes \ldots \otimes |x_n\rangle = |\phi\rangle$$

$$|\phi\rangle \in \mathbb{C}^{2^n}$$

Multi-Qubit system



Composite system:

What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

How to measure?

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

$$|x_1\rangle, |x_2\rangle \in \mathbb{C}^2$$

$$|\psi\rangle \in \mathbb{C}^{2^2} \in \mathbb{C}^4$$

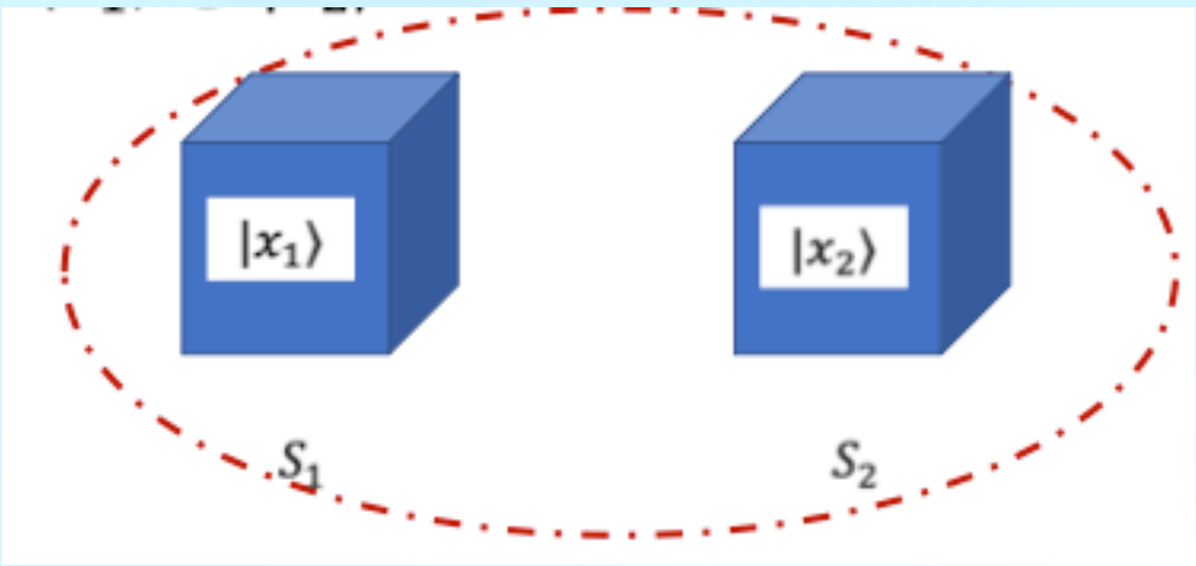
N-qubit generalisation:

$$|x_1\rangle \otimes \ldots \otimes |x_n\rangle = |\phi\rangle$$

$$|\phi\rangle \in \mathbb{C}^{2^n}$$

Note: 2^n **parameters required**

Multi-Qubit system



Composite system:

N-qubit generalisation:

$$|x_1\rangle \otimes \ldots \otimes |x_n\rangle = |\phi\rangle$$

$$|\phi\rangle \in \mathbb{C}^{2^n}$$

What is the state vector?

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle$$

How to measure?

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

$$|x_1\rangle, |x_2\rangle \in \mathbb{C}^2$$

$$|\psi\rangle \in \mathbb{C}^{2^2} \in \mathbb{C}^4$$

Note: 2^n parameters required

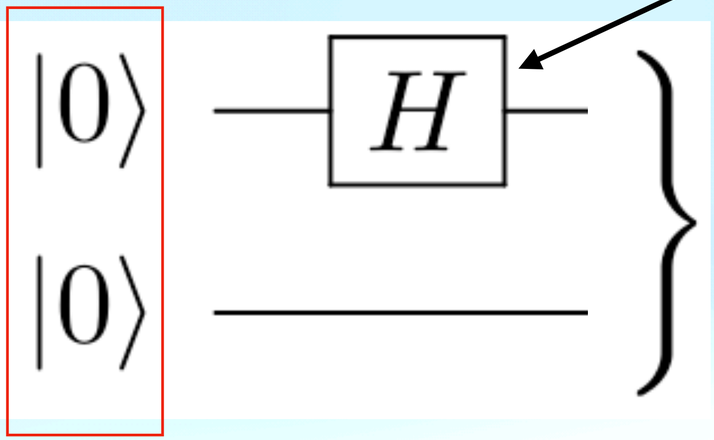
(Exponentially of Quantum mechanics*)

*See: Arora & Barak section 10.3.5

Operations on Multi-Qubit

Circuit (Gate) model:

Qubit Operation (Qubit Gate)



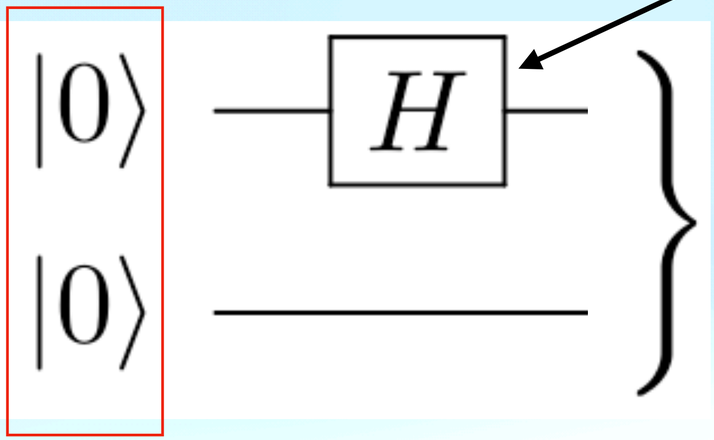
$$(H \otimes I)|00\rangle = |\psi\rangle$$

Registers: $|00\rangle$

Operations on Multi-Qubit

Gate model:

Qubit Operation (Qubit Gate)



$$(H \otimes I)|00\rangle = |\psi\rangle$$

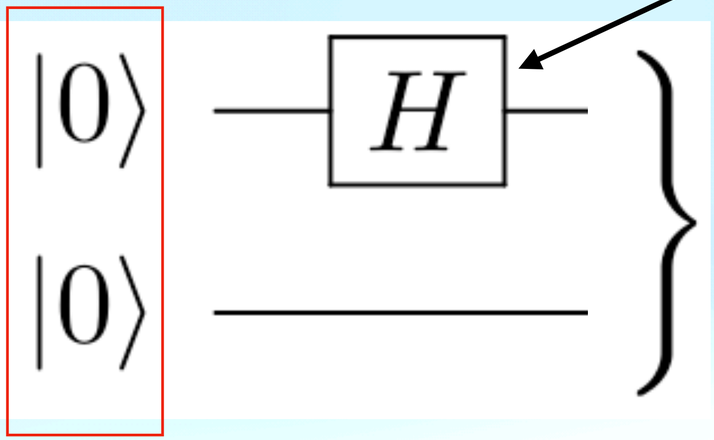
Registers: $|00\rangle$

$$H \otimes I = \begin{bmatrix} h_{11}\mathbf{I} & h_{12}\mathbf{I} \\ h_{21}\mathbf{I} & h_{22}\mathbf{I} \end{bmatrix} = U_{2^2 \times 2^2} = U_{4\times 4}$$

Operations on Multi-Qubit

Gate model:

Qubit Operation (Qubit Gate)



$$(H \otimes I)|00\rangle = |\psi\rangle$$

Registers: $|00\rangle$

$$H \otimes I = \begin{bmatrix} h_{11}\mathbf{I} & h_{12}\mathbf{I} \\ h_{21}\mathbf{I} & h_{22}\mathbf{I} \end{bmatrix} = U_{2^2 \times 2^2} = U_{4\times 4}$$

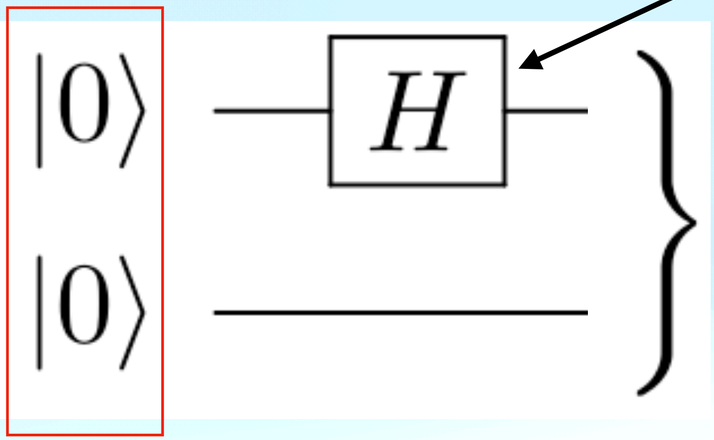
N-qubit generalisation:

$$O_1 \otimes \ldots \otimes O_n = U_{2^n \times 2^n}$$

Operations on Multi-Qubit

Gate model:

Qubit Operation (Qubit Gate)



$$(H \otimes I)|00\rangle = |\psi\rangle$$

Registers: $|00\rangle$

$$H \otimes I = \begin{bmatrix} h_{11}\mathbf{I} & h_{12}\mathbf{I} \\ h_{21}\mathbf{I} & h_{22}\mathbf{I} \end{bmatrix} = U_{2^2 \times 2^2} = U_{4\times 4}$$

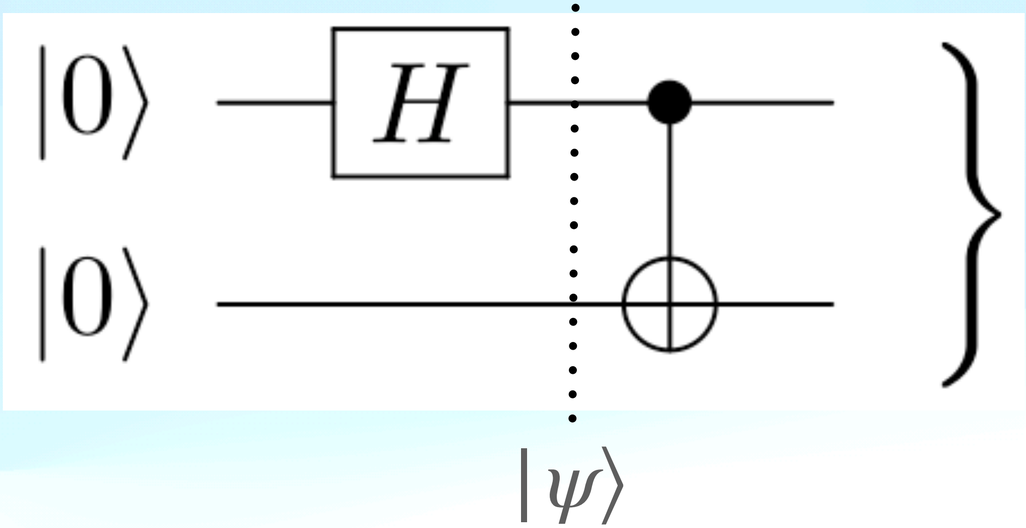
N-qubit generalisation:

$$O_1 \otimes \ldots \otimes O_n = U_{2^n \times 2^n}$$

Note: 2^{2n} **parameters required**

Operations on Multi-Qubit

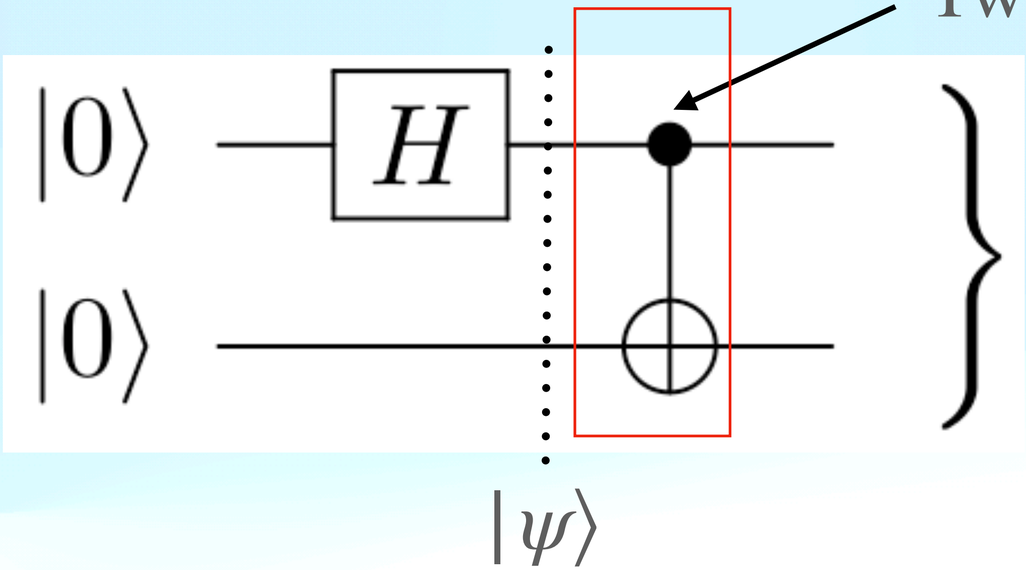
Gate model:



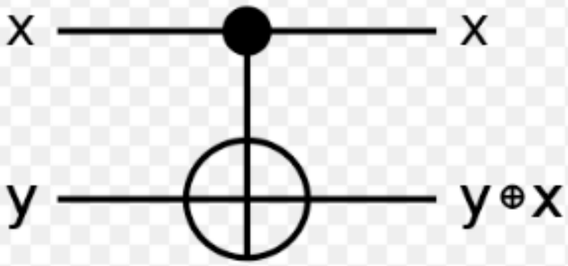
Operations on Multi-Qubit

Gate model:

Two Qubit Gate



Working of CNOT

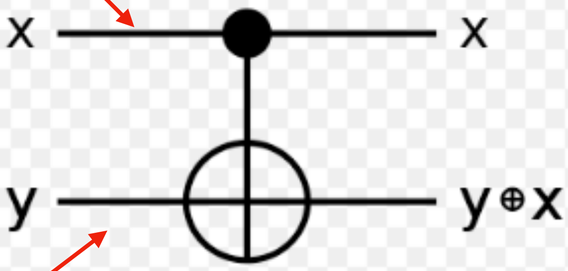


input		output	
x	y	x	y+x
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Controlled-NOT operation

Working of CNOT

Control line



Target Line

input		output	
x	y	x	y+x
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

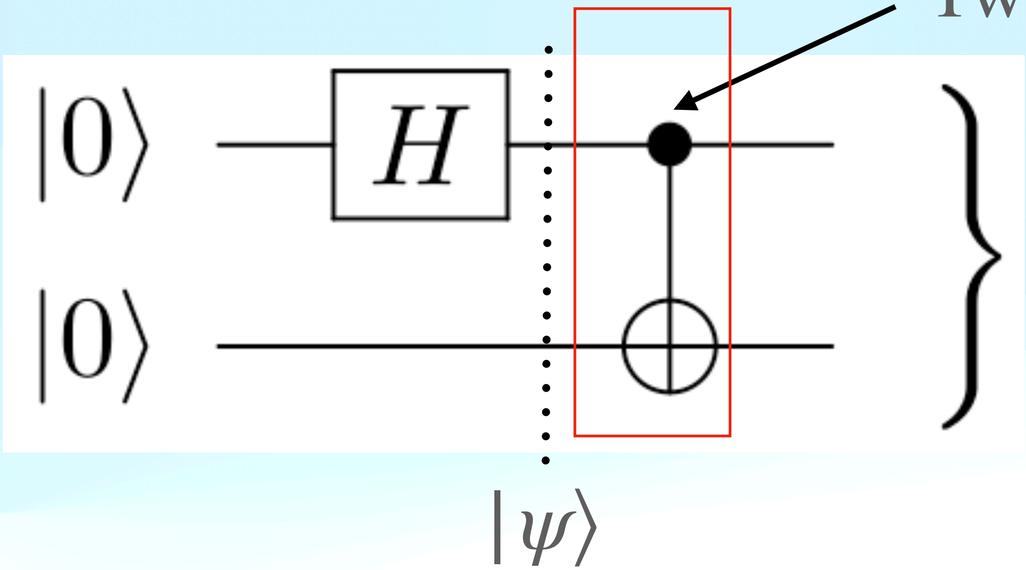
Controlled-NOT operation

Flip the target if the control has value 1.

Operations on Multi-Qubit

Gate model:

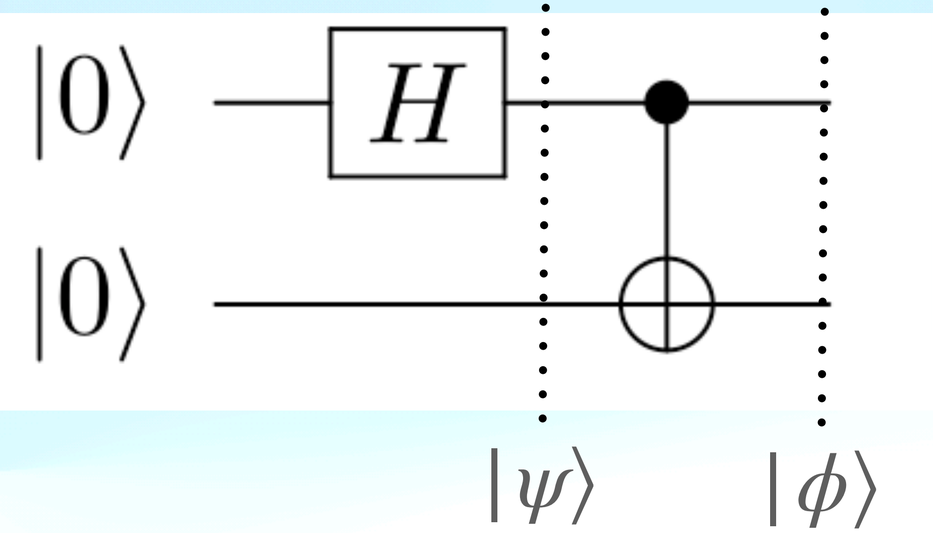
Two Qubit Gate



$$CNOT|\psi\rangle = |\phi\rangle$$

Operations on Multi-Qubit

Gate model:



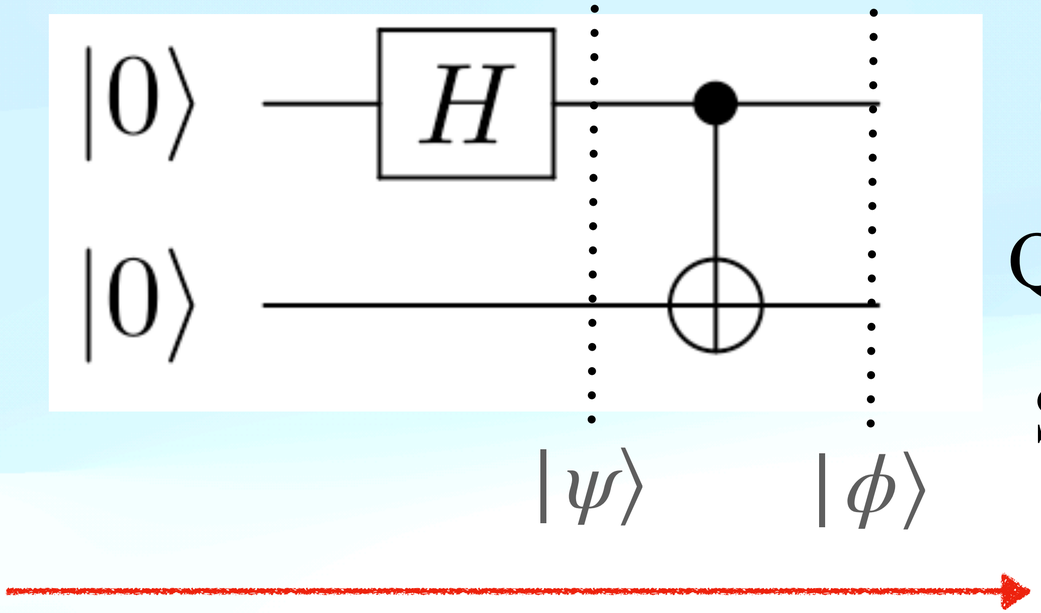
Computational Resources:

Qubits (Registers) ↔ Space

Sequence of Gates ↔ Time

Operations on Multi-Qubit

Gate model:



Flow of time

Computational Resources:

Qubits (Registers) ↔ Space

Sequence of Gates ↔ Time

Quantum Algorithms ↔ Quantum Circuit ↔ sequence of Quantum Gates

Universal Gate set

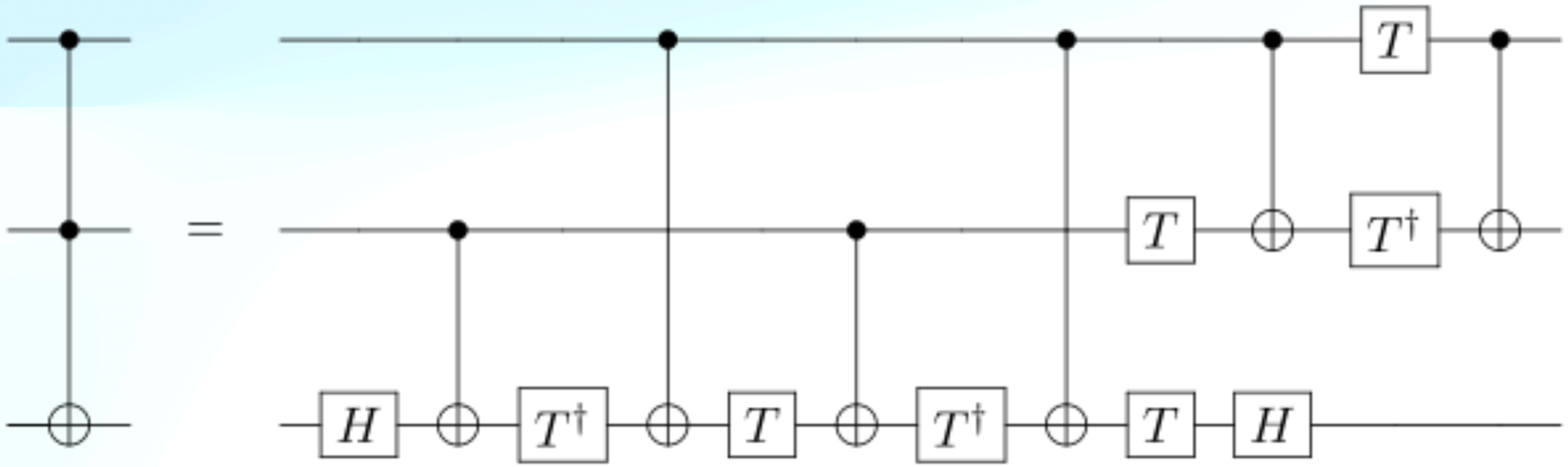
Classical Computation: {NOT, AND} or {NAND} or {…}

Quantum Computation: {H, S, T and CNOT } or, {H, Toffoli} or {…}

Universal Gate set

Classical Computation: {NOT, AND} or {NAND} or {…}

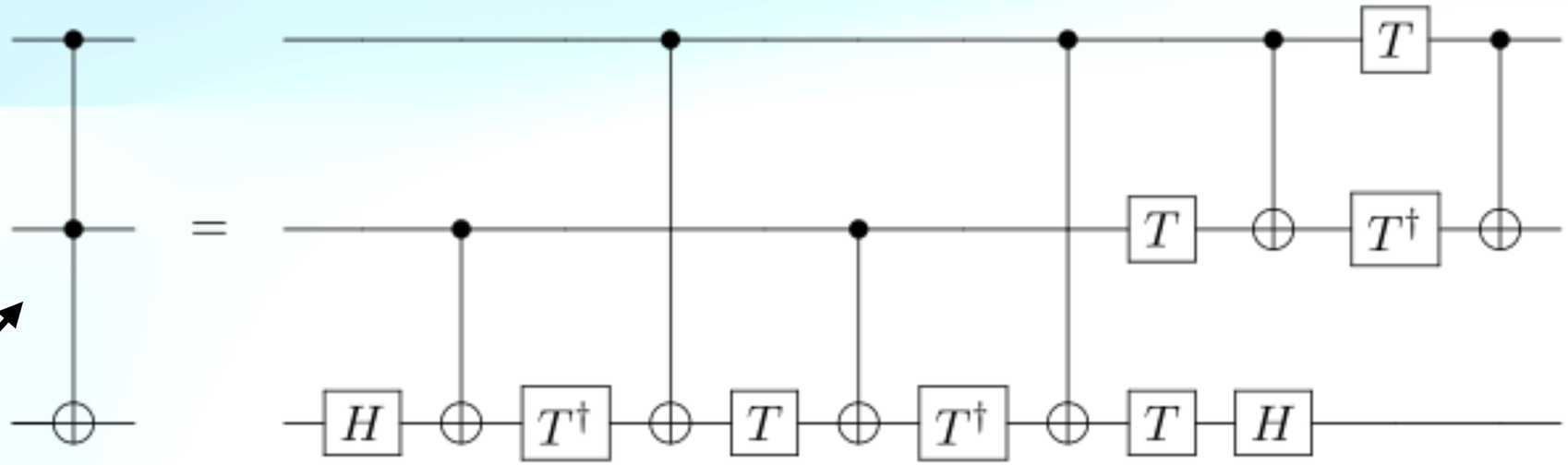
Quantum Computation: {H, S, T and CNOT } or, {H, Toffoli} or {…}



Universal Gate set

Classical Computation: {NOT, AND} or {NAND} or {…}

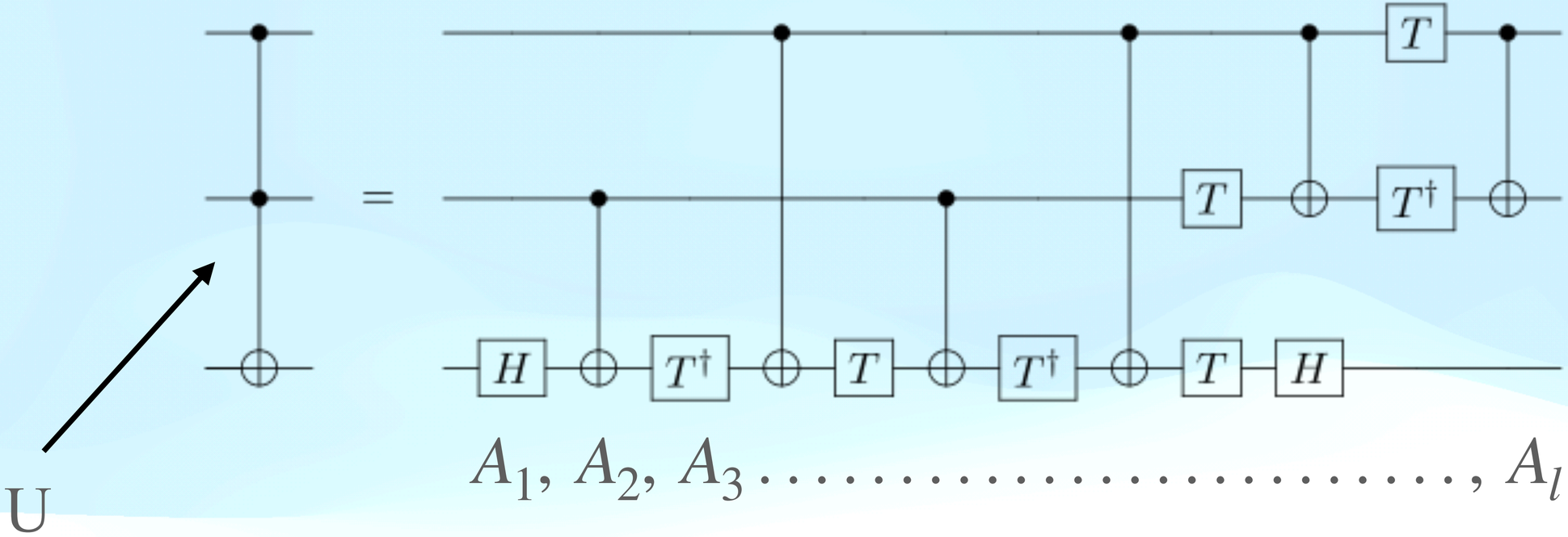
Quantum Computation: {H, S, T and CNOT } or, {H, Toffoli} or {…}



U

$A_1, A_2, A_3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, A_l$

Gate Approximation: Solvay-Kitaev theorem



Theorem* [Sol95, Kit97]: A m-qubit gate can be approximated to ϵ by a quantum circuit of $\mathcal{O}(m \cdot log^c[\frac{m}{\epsilon}])$ gates from a desired finite universal gate set.

---

* This version is mentioned in pp.617-624 of Neilson & Chuang (reprint 2021)

BQP: Complexity Class

BQP: Bounded Error Quantum Polynomial Time

Set of decision problems that can be solved by a poly(n) sized quantum circuit with high probability.

BQP: Complexity Class

BQP: Bounded Error Quantum Polynomial Time

Set of decision problems that can be solved by a poly(n) sized quantum circuit with high probability.

Captures the notion of efficient Quantum computation

Examples: P, BPP, FACTORING,
Quantum Simulation (Decision version)

BQP: Formal Definition

Requires formalism of two concepts:

Acceptance Probability of Quantum Circuit

P-unifrom circuit families

BQP: Formal Definition

Requires formalism of two concepts:

Acceptance Probability of Quantum Circuit

P-unifrom circuit families

P-unifrom circuit families:

Def: A circuit family $\{C_n\}$ is P-uniform if there is a polynomial-time TM that on input 1^n, output the description of the circuit C_n.

BQP: Formal Definition

Requires formalism of two concepts:

Acceptance Probability of Quantum Circuit

P-unifrom circuit families

Acceptance Probability of Quantum Circuit

Probability of output being $|1\rangle$

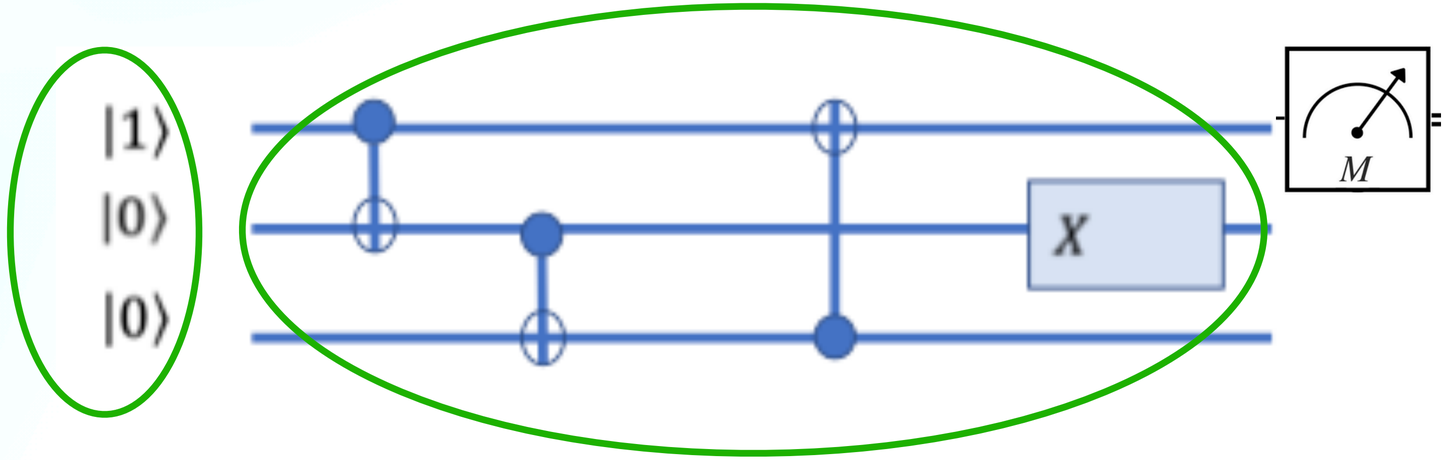
BQP: Formal Definition

Requires formalism of two concepts:

Acceptance Probability of Quantum Circuit

P-unifrom circuit families

Acceptance Probability of Quantum Circuit



Probability of output being $|1\rangle$

$|\psi\rangle$

C_n

BQP: Formal Definition

Requires formalism of two concepts:

Acceptance Probability of Quantum Circuit

P-unifrom circuit families

Acceptance Probability of Quantum Circuit

Def: Acceptance Probability of a quantum circuit C on input $|x\rangle$ is the *probability to measure the first qubit* of $C|x\rangle$ yields $|1\rangle$

Handy notation for acceptance probability: $Pr_{acc}(C_n|x\rangle)$

BQP: Formal Definition

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

BQP: Formal Definition

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

- Similar to BPP, error bound can be $\left[\frac{1}{2} + n^{-c}\right]$ without changing the complexity class.
- Similar probability boosting method applicable.

Proof strategy
(for BQP is in PSPACE and EXP)

Discuss hardest problem in BQP.

Then show this problem lie in PSPACE (and EXP)

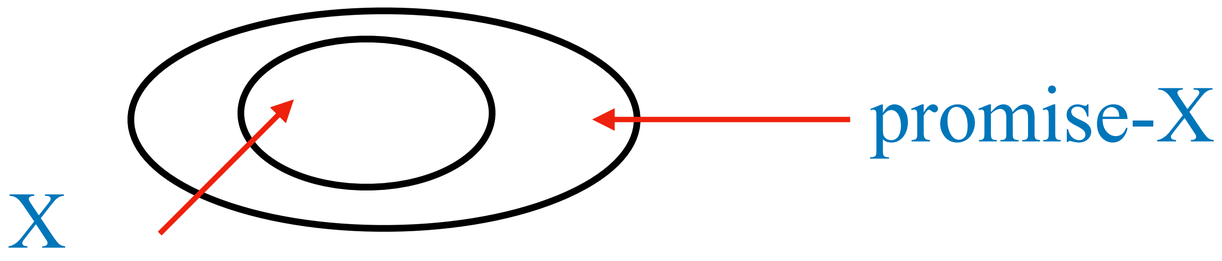
Promise Problem

Similar to BPP, existence of complete problems in BQP is unknown

Promise Problem: A generalisation of decision problem

Def: Take set L_{YES} *and* $L_{NO} \in \{0,1\}^*$, *such that* $L_{YES} \cap L_{NO} = \emptyset$.
If $x \in L_{YES}$ must be accepted, while $x \in L_{NO}$ must be rejected.
The union $L_{YES} \cup L_{NO}$ is called promise.

Note: If $L_{YES} \cup L_{NO} = \{0,1\}^*$, then It is decision Problem.



Source: Henry Yeun lecture link:

Promise BQP problem

APPROX. Q- CIRCUIT problem (AQC):

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq \alpha$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq \beta$

Where, α, $\beta \in [0,1]$

Output: Determine which is the case

AQC is a promise problem because input (C) is promised to satisfy some condition

Complete problem for Promise BQP

Def: A problem is promiseBQP-complete if and only if it lie in promiseBQP and all the problem in promiseBQP has a randomised reduction to the problem.

We will use a known result: AQC is promise-BQP complete

Lemma: For every $L \in BQP$, *relation* $L \leq_r AQC$ *holds*

Reduction type: randomized reduction

Note: An efficient classical algorithm for AQC implies an efficient algorithm for all the languages in BQP.

Proof of the Lemma

Lemma: If $L \in BQP, then\ L \leq_r AQC$

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Proof of the Lemma

Lemma: If $L \in BQP, then\ L \leq_r AQC$

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Without loss of generality, fix input size to be n.

$x \in \{0,1\}^n$

Proof of the Lemma

Lemma: If $L \in BQP, then\ L \leq_r AQC$

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Without loss of generality, fix input size to be n.

$$x \in \{0,1\}^n$$

Claim-1: $\forall x, |x\rangle$ can be generate via a poly(n)-size quantum circuit (say, C'_x) acting on state $|0_1 \ldots 0_n\rangle$

Poof: Via an explicit circuit construction algorithm (on next slide)

Proof of the Lemma

Lemma: If $L \in BQP, then\ L \leq_r AQC$

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Without loss of generality, fix input size to be n.

$$x \in \{0,1\}^n$$

Claim-1: $\forall x, |x\rangle$ can be generate via a poly(n)-size quantum circuit (say, C'_x) acting on state $|0_1 \ldots 0_n\rangle$

Poof: Via an explicit circuit construction algorithm (on next slide)

Conclusion: $C_{|x|}|x\rangle = C_{|x|}C'_x|0_1 \ldots 0_n\rangle, where, C_{|x|}C'_x$ is poly(n) size

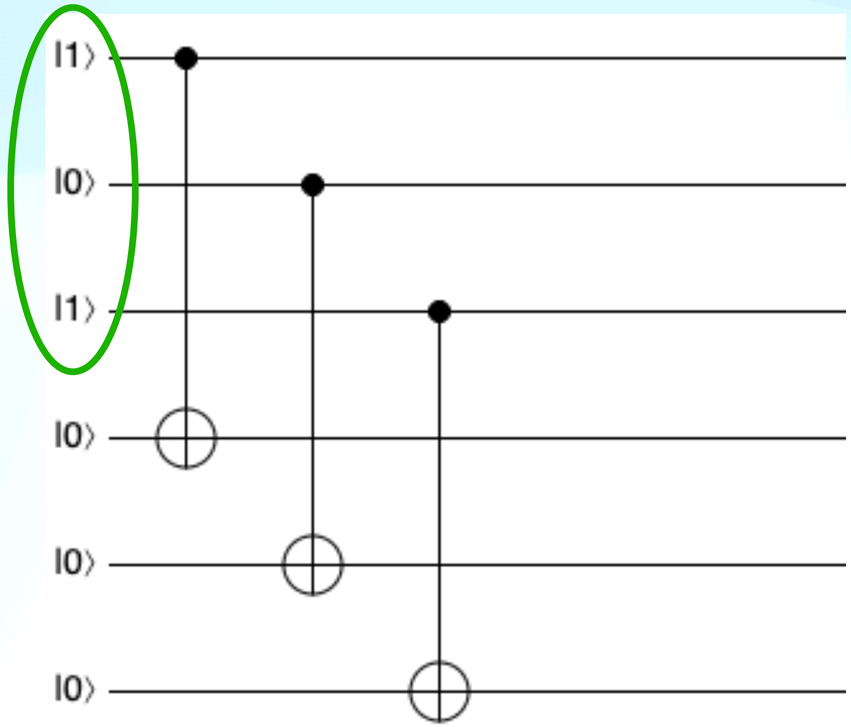
Arbitrary basis state construction

To construct $|x_1 \ldots x_n\rangle$ from $|0...0\rangle$:

Algorithm: Hardwire the desired input state to the circuit (as shown below) and use CNOT gate flip the sate $|0...0\rangle$ in qubit by qubit manner.

Example: $|101\rangle$ *from* $|000\rangle$

hardwiring $|101\rangle$



Link to circuit designer

Arbitrary basis state construction

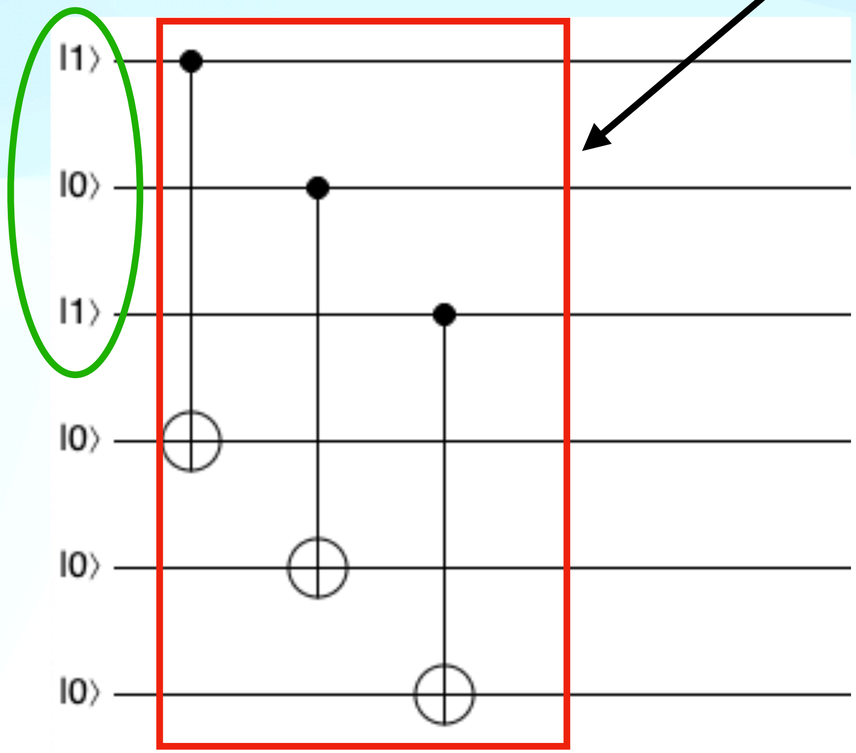
To construct $|x_1 \ldots x_n\rangle$ from $|0...0\rangle$:

Algorithm: Hardwire the desired input state to the circuit (as shown below) and use CNOT gate flip the sate $|0...0\rangle$ in qubit by qubit manner.

Example: $|101\rangle$ *from* $|000\rangle$

C'_{101}

hardwiring $|101\rangle$

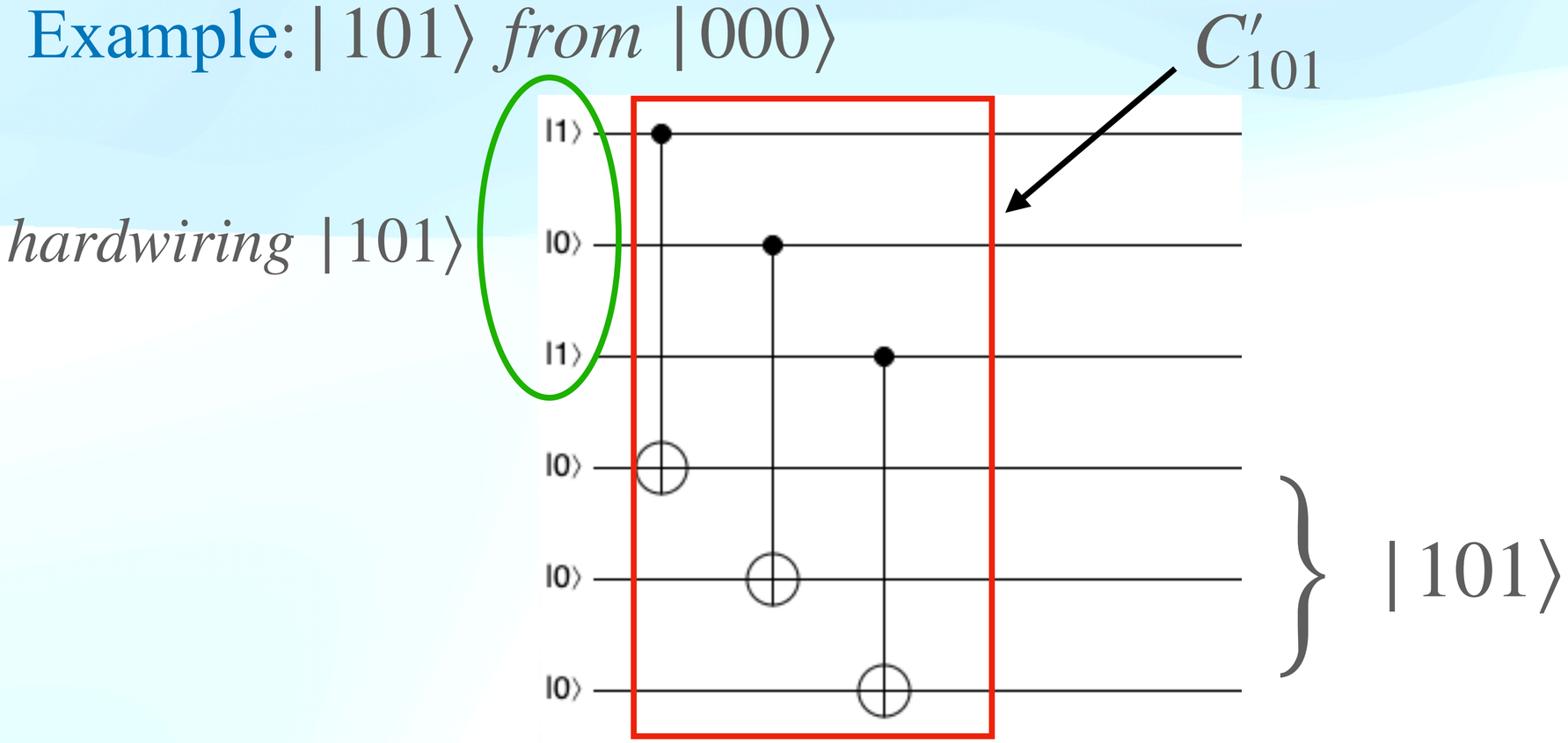


Arbitrary basis state construction

To construct $|x_1 \ldots x_n\rangle$ from $|0...0\rangle$:

Algorithm: Hardwire the desired input state to the circuit (as shown below) and use CNOT gate flip the sate $|0...0\rangle$ in qubit by qubit manner.

Example: $|101\rangle$ *from* $|000\rangle$



Arbitrary basis state construction

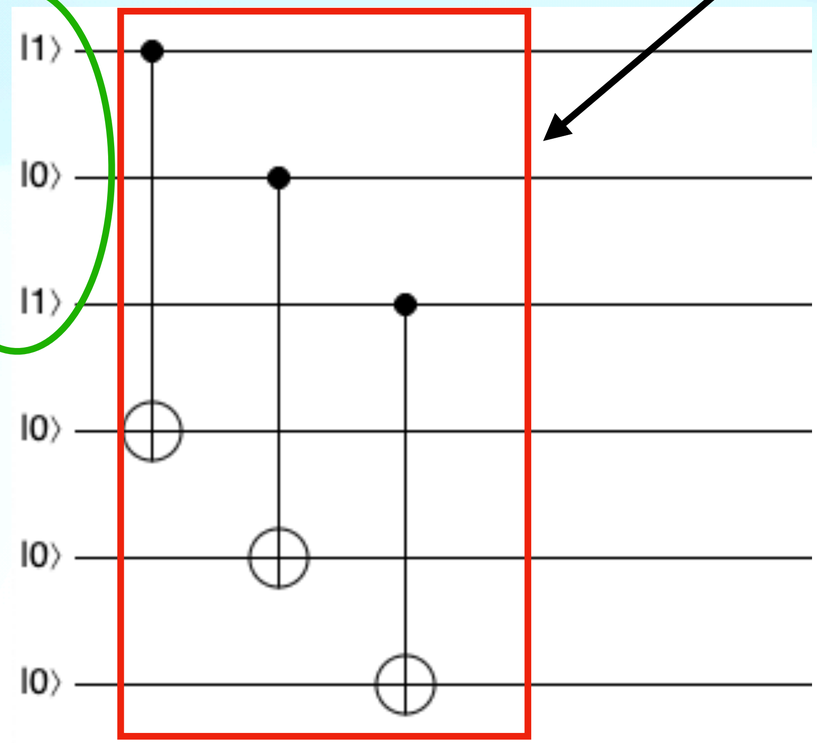
To construct $|x_1 \ldots x_n\rangle$ from $|0...0\rangle$:

Algorithm: Hardwire the desired input state to the circuit (as shown below) and use CNOT gate flip the sate $|0...0\rangle$ in qubit by qubit manner.

Example: $|101\rangle$ *from* $|000\rangle$

C'_{101}

hardwiring $|101\rangle$



$|101\rangle$ $\xrightarrow{\textit{apply } C}$

Proof continue…

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

BQP definition

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

AQC definition

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \geq \alpha$
2. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \leq \beta$

Where, $\alpha, \beta \in [0,1]$

Output: Determine which is the case

Reduction: Step-I: Fix the value of $\alpha = 2/3$ *and* $\beta = 1/3$

Proof continue…

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \leq 1/3$

Output: Determine which is the case

Reduction: Step-I: Fix the value of $\alpha = 2/3$ *and* $\beta = 1/3$

Step-II: for any x, produce circuit $C_{|x|}C'_x$

Proof continue…

A language L is in BQP if and only if there exists a P-unifrom Quantum circuits $\{C_n : n \in \mathbb{N}\}$, such that:

For all $x \in L$, $Pr_{acc}(C_{|x|}|x\rangle) \geq 2/3$

For all $x \notin L$, $Pr_{acc}(C_{|x|}|x\rangle) \leq 1/3$

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1 \ldots 0_n\rangle) \leq 1/3$

Output: Determine which is the case

Reduction: Step-I: Fix the value of $\alpha = 2/3 \ and \ \beta = 1/3$

Step-II: for any x, produce circuit $C_{|x|}C'_x$

Step-III: Use AQC-solver to determine the membership of $x \ in \ L$

AQC is in EXP

Classical simulation of ACQ in exponential time

AQC is in EXP

Classical simulation of ACQ in exponential time

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

Let the description of circuit $C_n = g_1 \cdot g_2 \cdots g_T$

Time estimation of matrix operation: $C_n|0_1\ldots0_n\rangle$

AQC is in EXP

Classical simulation of ACQ in exponential time

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

Let the description of circuit $C_n = g_1 \cdot g_2 \ldots g_T$

Time estimation of matrix operation: $C_n|0_1\ldots0_n\rangle$

Recall: 1. $|0_1\ldots0_n\rangle$ is vector of dimension $2^n \times 1$

2. Each g_i is matrix of dimension $2^n \times 2^n$

AQC is in EXP

Classical simulation of ACQ in exponential time

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

Let the description of circuit $C_n = g_1 \cdot g_2 \ldots g_T$

Time estimation of matrix operation: $C_n|0_1\ldots0_n\rangle$

Recall: 1. $|0_1\ldots0_n\rangle$ is vector of dimension $2^n \times 1$

2. Each g_i is matrix of dimension $2^n \times 2^n$

3. Operation $A_{m\times m} \cdot a_{m\times 1}$ requires $\mathcal{O}(m^2)$ time.

AQC is in EXP

Classical simulation of ACQ in exponential time

Let the description of circuit $C_n = g_1 \cdot g_2 \ldots g_T$

Time estimation of matrix operation: $C_n|0_1 \ldots 0_n\rangle$

Recall: 1. $|0_1 \ldots 0_n\rangle$ is vector of dimension $2^n \times 1$

2. Each g_i is matrix of dimension $2^n \times 2^n$

3. Operation $A_{m\times m} \cdot a_{m\times 1}$ requires $\mathcal{O}(m^2)$ time.

4. Operation $g_i|0_1 \ldots 0_n\rangle$ requires $\mathcal{O}((2^n)^2)$ time.

AQC is in EXP

Classical simulation of ACQ in exponential time

Let the description of circuit $C_n = g_1 \cdot g_2 \ldots g_T$

Time estimation of matrix operation: $C_n|0_1 \ldots 0_n\rangle$

Recall: 1. $|0_1 \ldots 0_n\rangle$ is vector of dimension $2^n \times 1$

2. Each g_i is matrix of dimension $2^n \times 2^n$

3. Operation $A_{m\times m} \cdot a_{m\times 1}$ requires $\mathcal{O}(m^2)$ time.

4. Operation $g_i|0_1 \ldots 0_n\rangle$ requires $\mathcal{O}((2^n)^2)$ time.

Conclusion:

$C_n|0_1 \ldots 0_n\rangle = g_1 \cdot g_2 \ldots g_T|0_1 \ldots 0_n\rangle$ requires $\mathcal{O}((2^n)^2 \cdot T)$ time

AQC is in EXP

Classical simulation of ACQ in exponential time

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

AQC is in EXP

Classical simulation of ACQ in exponential time

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

AQC is in EXP

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

Expansion of $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \alpha_{0\ldots0}|0\ldots0\rangle + \alpha_{0\ldots1}|0\ldots1\rangle + \ldots + \alpha_{1\ldots1}|1\ldots1\rangle$

AQC is in EXP

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

Expansion of $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \alpha_{0\ldots0}|0\ldots0\rangle + \alpha_{0\ldots1}|0\ldots1\rangle + \ldots + \alpha_{1\ldots1}|1\ldots1\rangle$

Probability that first qubit collapse to state $|1\rangle$

AQC is in EXP

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

Expansion of $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \alpha_{0\ldots0}|0\ldots0\rangle + \alpha_{0\ldots1}|0\ldots1\rangle + \ldots + \alpha_{1\ldots1}|1\ldots1\rangle$

Probability that first qubit collapse to state $|1\rangle = |\langle 1|C_n|0_1\ldots0_n\rangle|^2$

AQC is in EXP

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

Expansion of $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \alpha_{0\ldots0}|0\ldots0\rangle + \alpha_{0\ldots1}|0\ldots1\rangle + \ldots + \alpha_{1\ldots1}|1\ldots1\rangle$

Probability that first qubit collapse to state $|1\rangle = \sum_{x\in\{0,1\}^n\,:\,x_1=1} |\alpha_x|^2$

AQC is in EXP

Computational Cost for $Pr_{acc}(C_n|0_1\ldots0_n\rangle)$

Let $C_n|0\ldots0\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$

Expansion of $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \alpha_{0\ldots0}|0\ldots0\rangle + \alpha_{0\ldots1}|0\ldots1\rangle + \ldots + \alpha_{1\ldots1}|1\ldots1\rangle$

Probability that first qubit collapse to state $|1\rangle = \sum_{x\in\{0,1\}^n\,:\,x_1=1} |\alpha_x|^2$

Thus cost of $Pr_{acc}(C_n|0_1\ldots0_n\rangle) = \mathcal{O}((2^n/2)$

Cost Summary: BQP is in EXP

Time: $\mathcal{O}((2^n)^2 \cdot T)$

Space: Dominated by the need to store T number of matrices of size $2^n \times 2^n$ in the work space to evaluate the matrix product. Thus, order $\mathcal{O}((2^n)^2 \cdot T)$ space required.

Cost Summary: BQP is in EXP

Time: $\mathcal{O}((2^n)^2 \cdot T)$

Space: Dominated by the need to store T number of matrices of size $2^n \times 2^n$ in the work space to evaluate the matrix product Thus, order $\mathcal{O}((2^n)^2 \cdot T)$ space required.

Can we do it in polynomial space (while trading off the time)?

Is BQP is in PSAPCE?

ACQ is in PSPACE

Input: A description of Quantum circuit C where either

1. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \geq 2/3$
2. $Pr_{acc}(C_n|0_1\ldots0_n\rangle) \leq 1/3$

Output: Determine which is the case

General Strategy: $|\psi\rangle \xrightarrow[\text{operation}]{\text{Sequence of Gate}} |\phi\rangle \xrightarrow[\text{Prob. estimation}]{\text{Measure Qubit-1}} |1\rangle$

Called: Operator Formalism

Alternative Strategy:

Source: Henry Yeun lecture link:

ACQ is in PSPACE

General Strategy: $|\psi\rangle \xrightarrow[\text{operation}]{\text{Sequence of Gate}} |\phi\rangle \xrightarrow[\text{Prob. estimation}]{\text{Measure Qubit-1}} |1\rangle$

Named: Operator Formalism

Alternative Strategy:

ACQ is in PSPACE

General Strategy: $|\psi\rangle \xrightarrow[\text{operation}]{\text{Sequence of Gate}} |\phi\rangle \xrightarrow[\text{Prob. estimation}]{\text{Measure Qubit-1}} |1\rangle$

Named: Operator Formalism

Named: Feynman Path Formulation

Alternative Strategy:

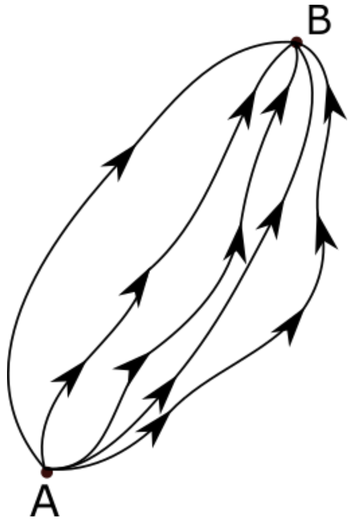
ACQ is in PSPACE

General Strategy: $|\psi\rangle \xrightarrow[\text{operation}]{\text{Sequence of Gate}} |\phi\rangle \xrightarrow[\text{Prob. estimation}]{\text{Measure Qubit-1}} |1\rangle$

Named: Operator Formalism

Named: Feynman Path Formulation

Alternative Strategy: Feynman Path Integral approach



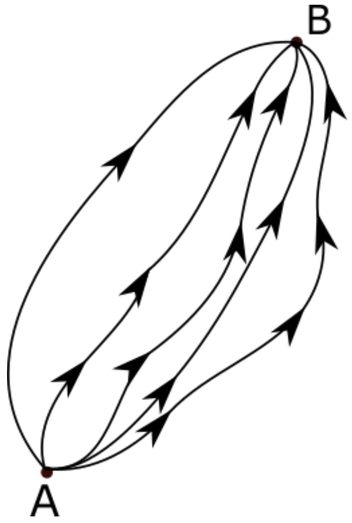
ACQ is in PSPACE

General Strategy: $|\psi\rangle \xrightarrow[\text{operation}]{\text{Sequence of Gate}} |\phi\rangle \xrightarrow[\text{Prob. estimation}]{\text{Measure Qubit-1}} |1\rangle$

Named: Operator Formalism

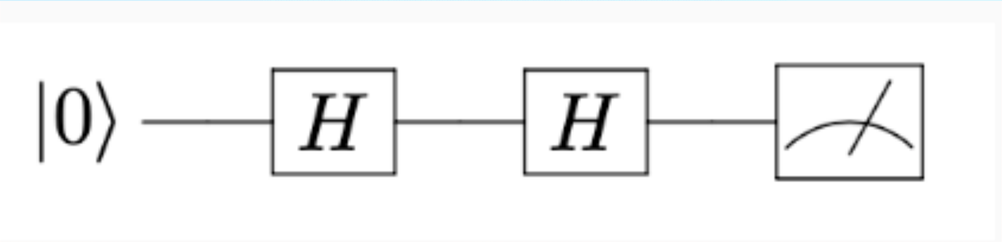
Named: Feynman Path Formulation

Alternative Strategy: Feynman Path Integral approach

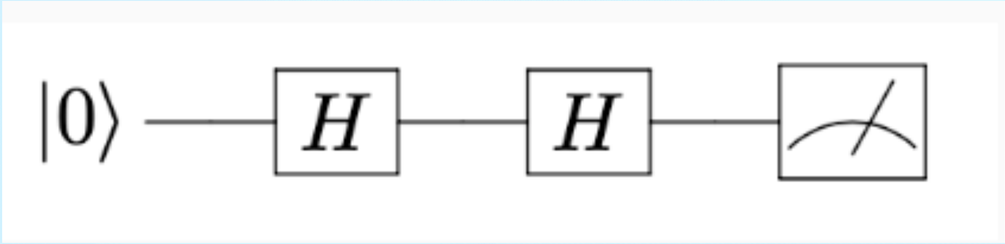


Key idea: In polynomial space, iterate over exponentially many path of quantum computation, and keep adding the result of each path.

Path Integral: An Example



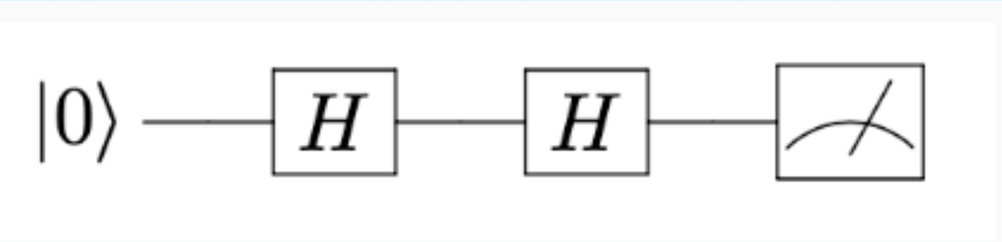
Path Integral: An Example



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

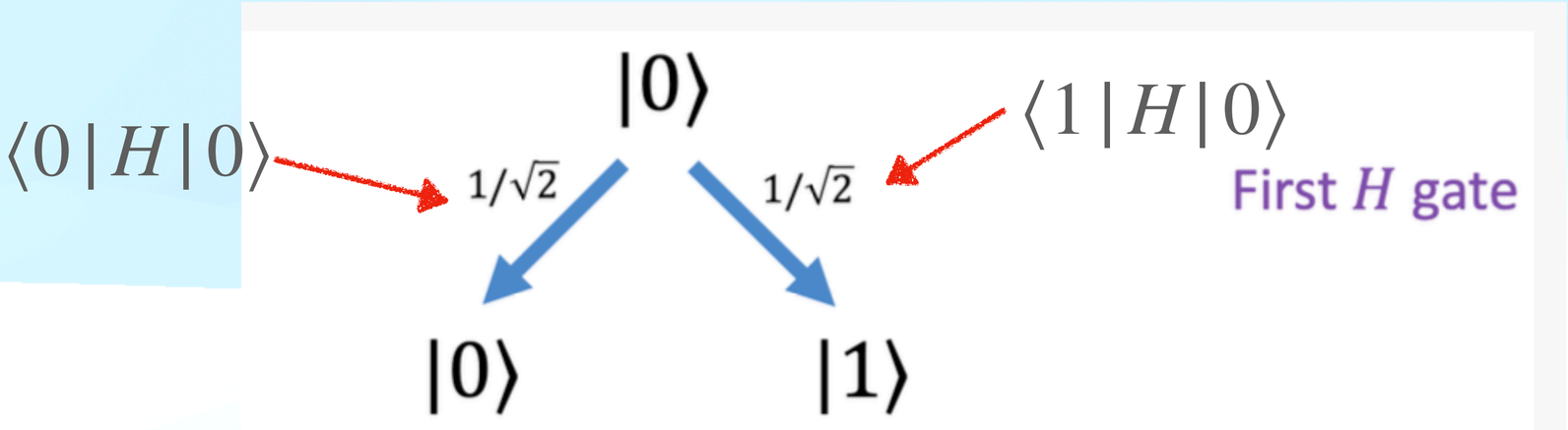
$$|1\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Path Integral: An Example

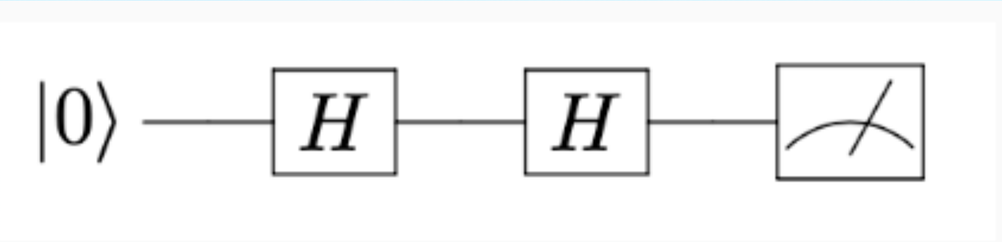


$$|0\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

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

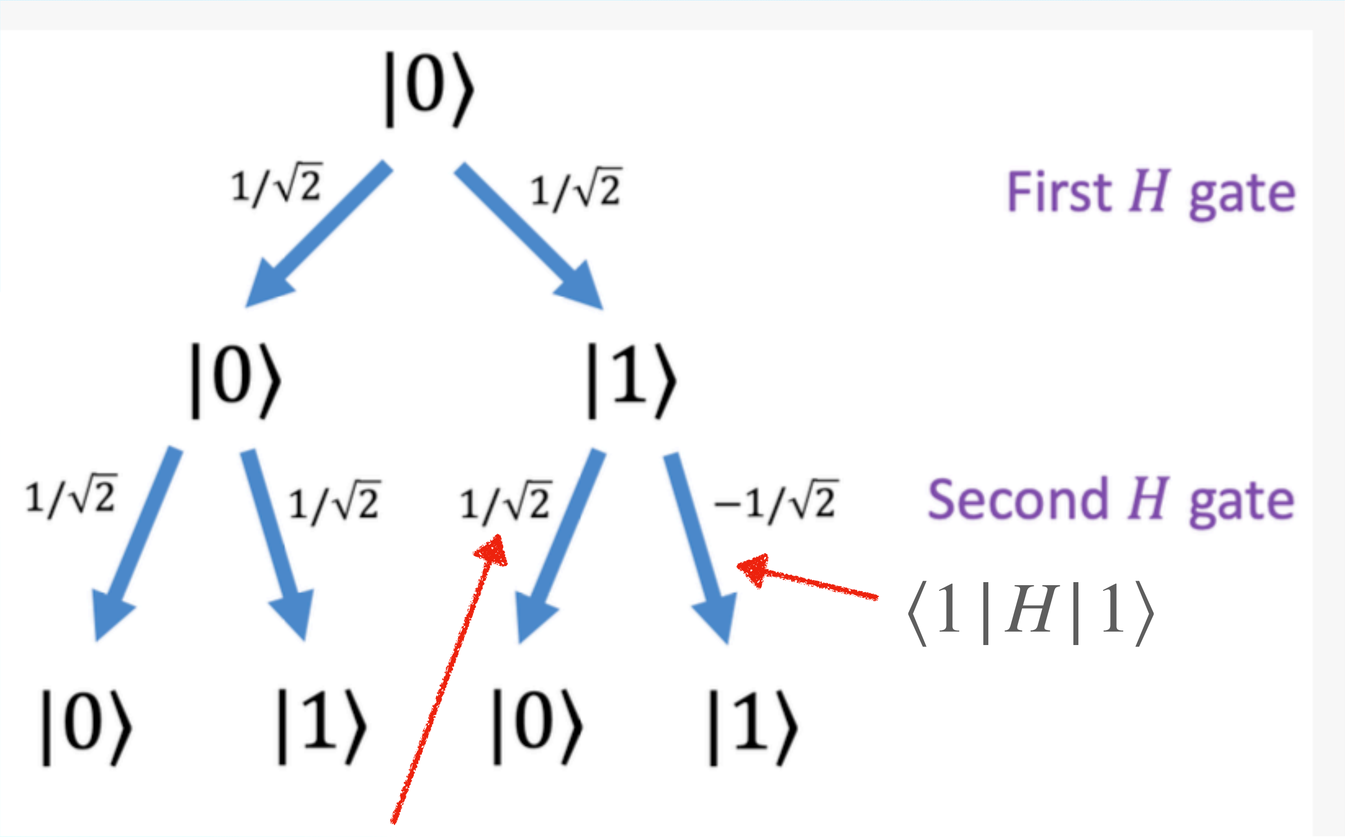


Path Integral Example

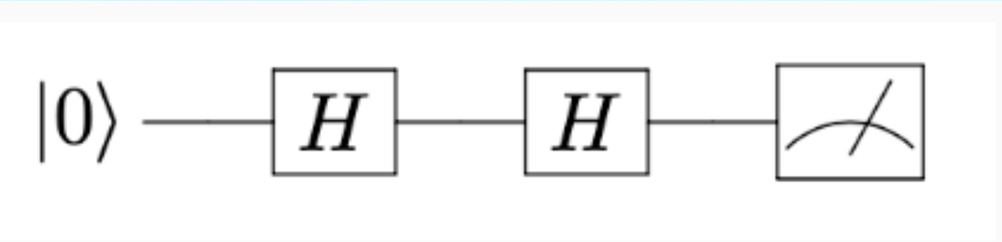


$$|0\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

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

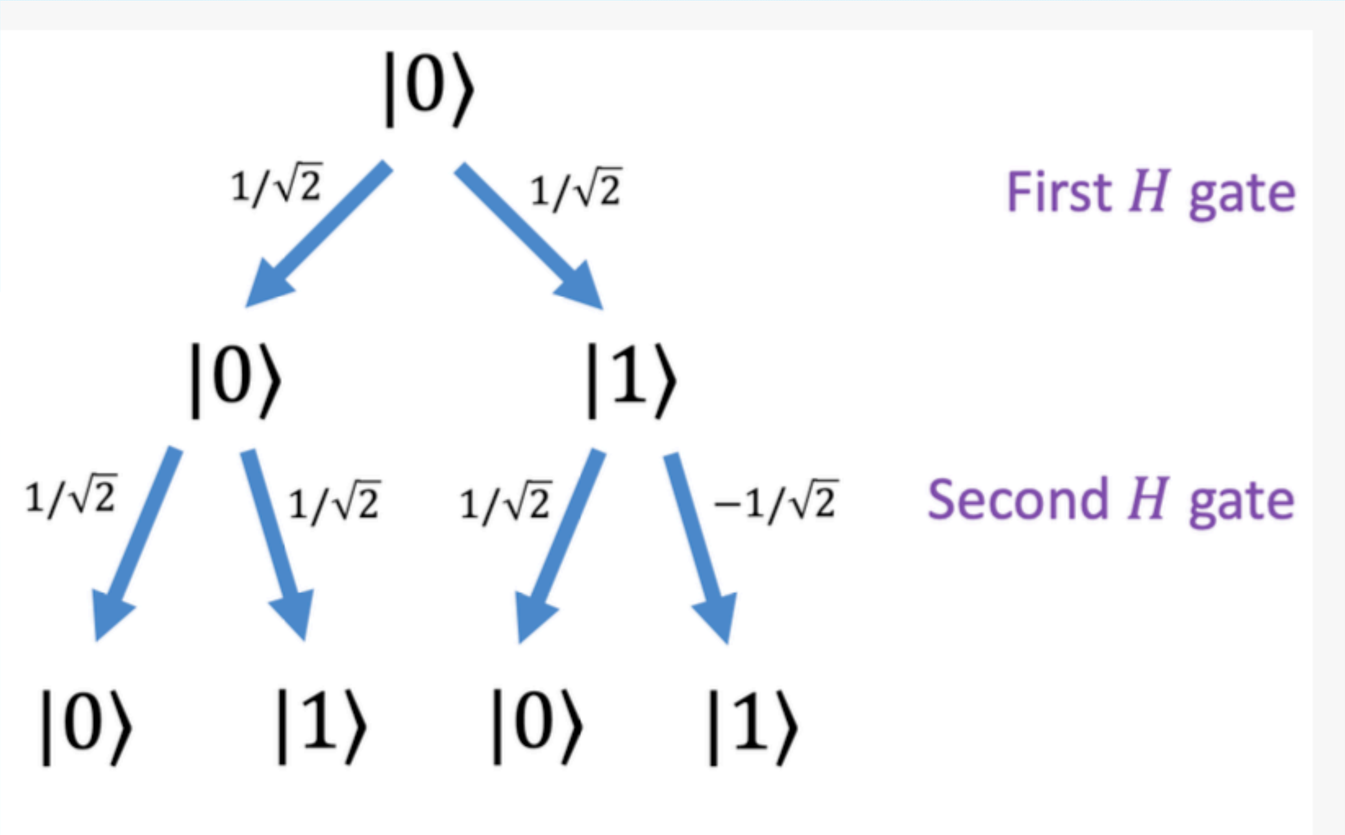


Path Integral Example



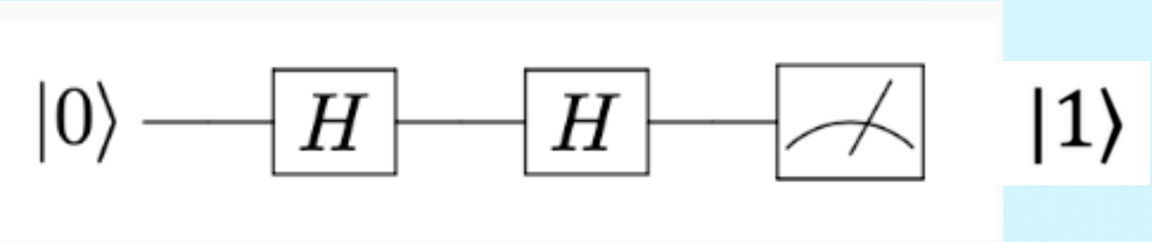
$$|0\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

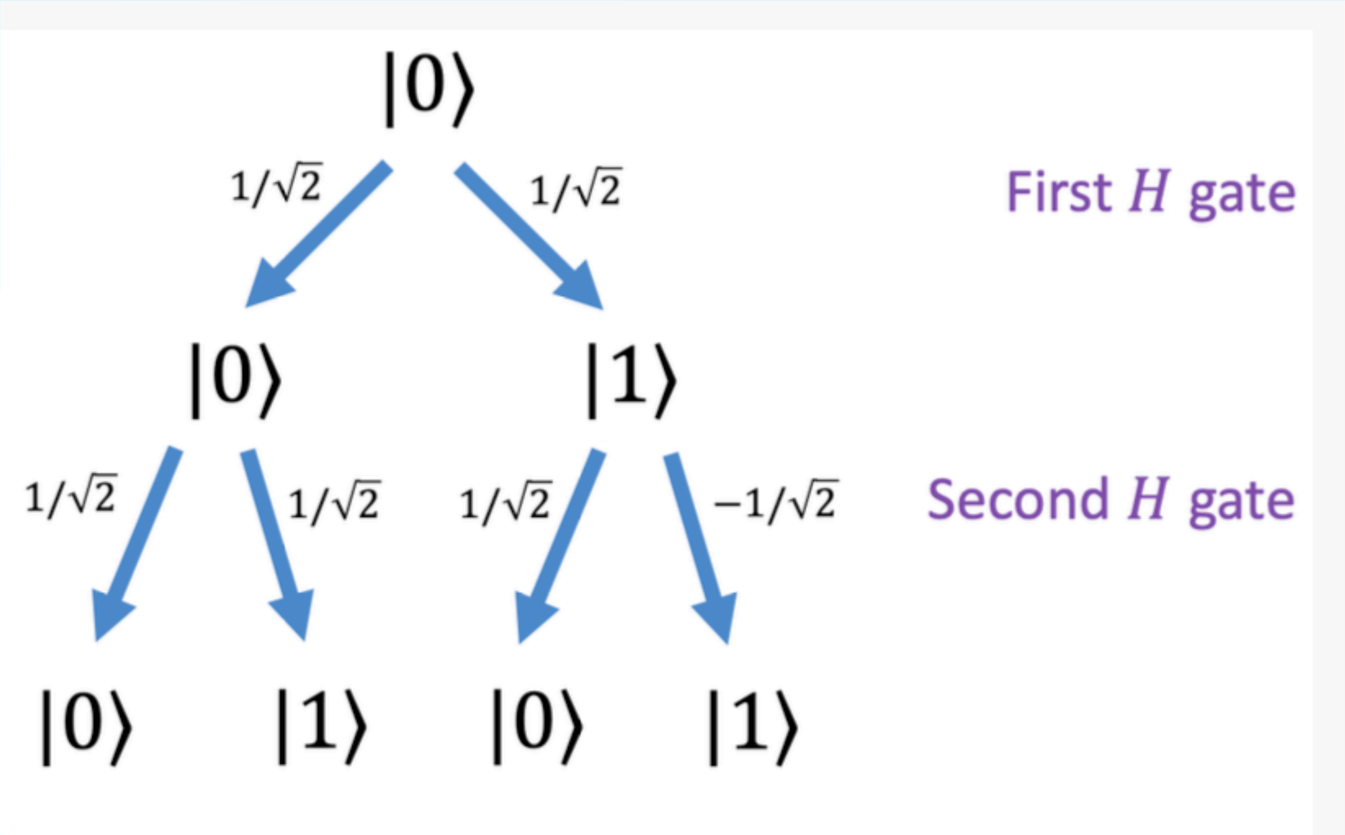
$$|1\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Path Integral Example



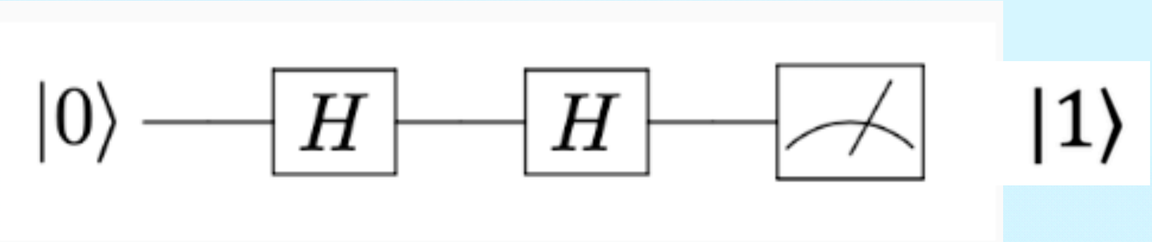


$$|0\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

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

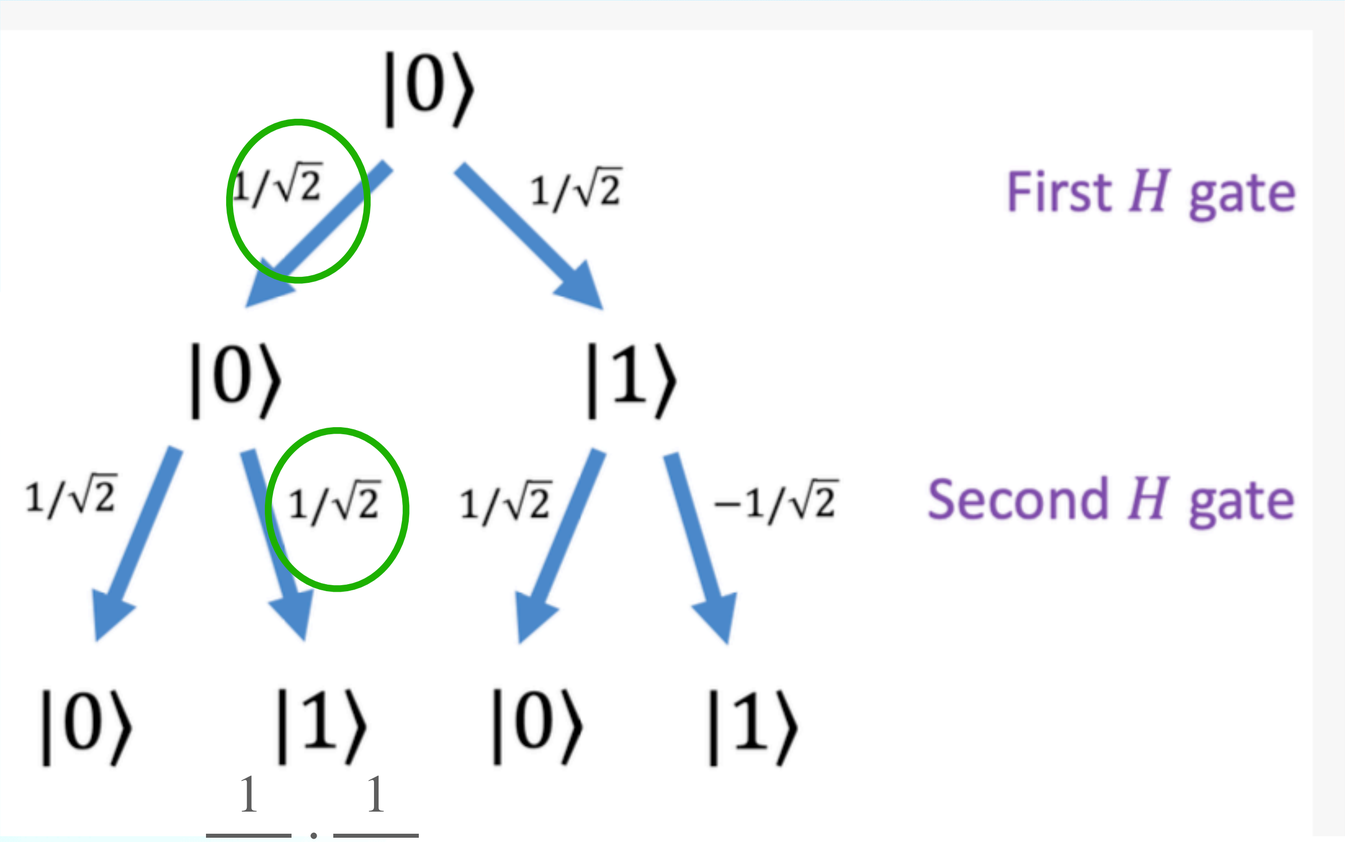
ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Path Integral Example



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

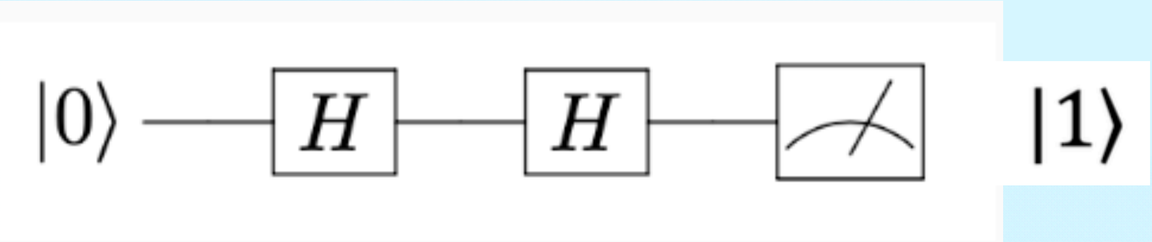
$$|1\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



$$\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}$$

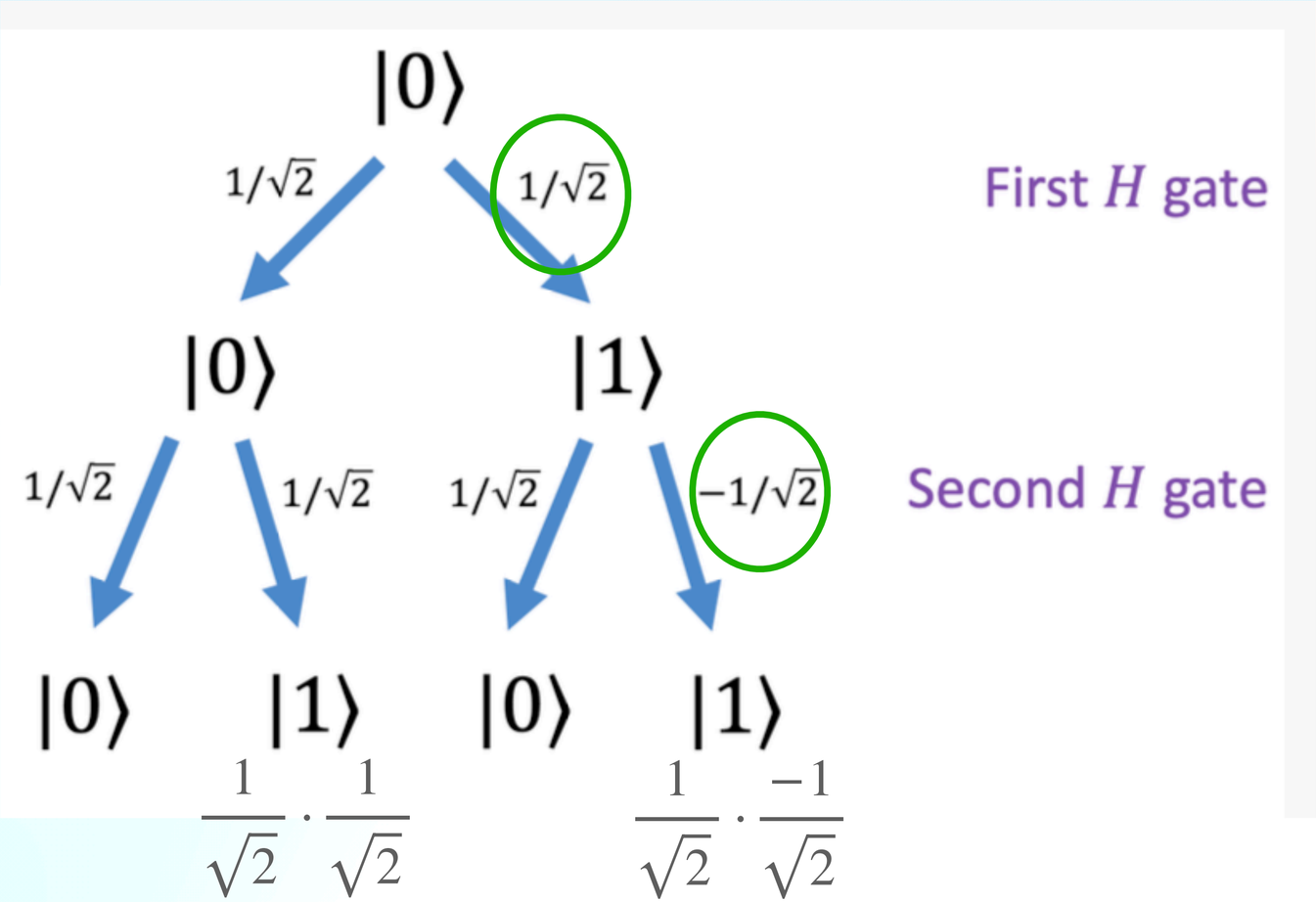
ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Path Integral Example



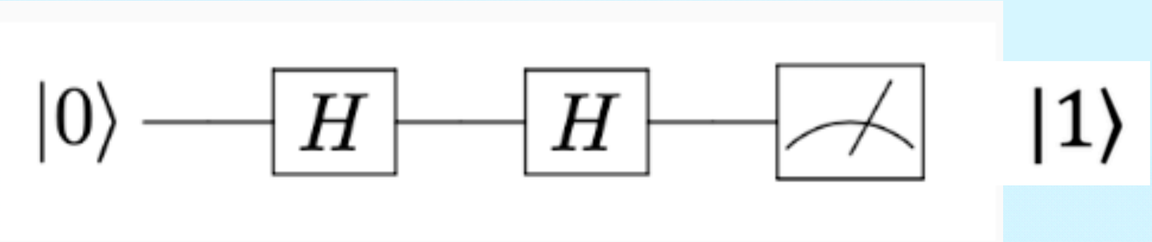
$$|0\rangle \xrightarrow{\mathbf{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

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



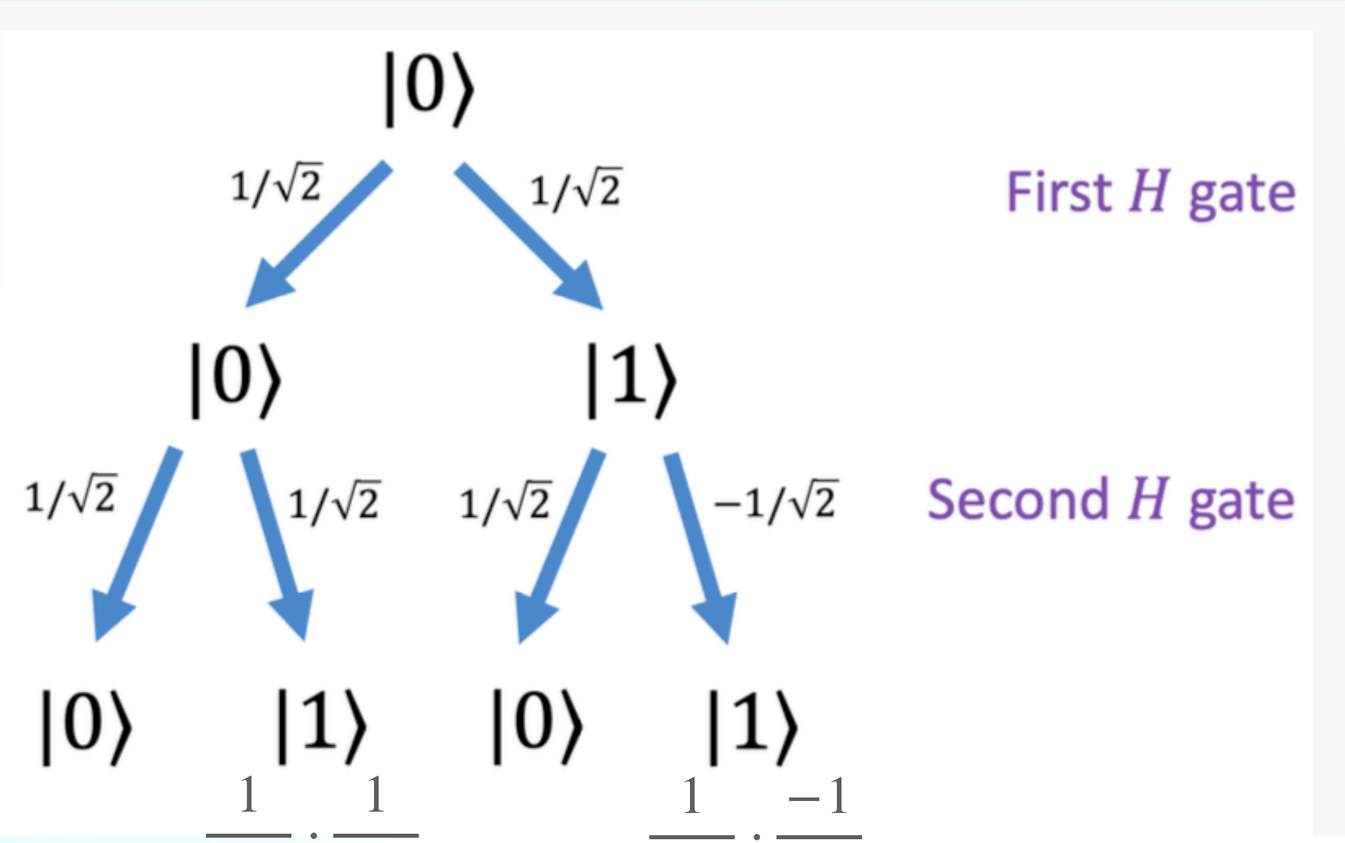
ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Path Integral Example



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

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



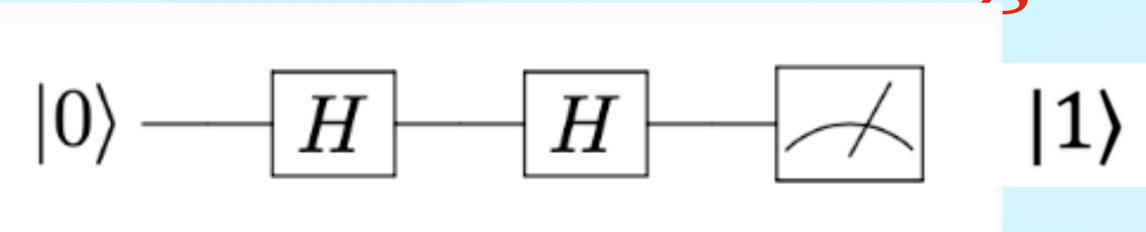
$$\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \qquad \frac{1}{\sqrt{2}} \cdot \frac{-1}{\sqrt{2}}$$

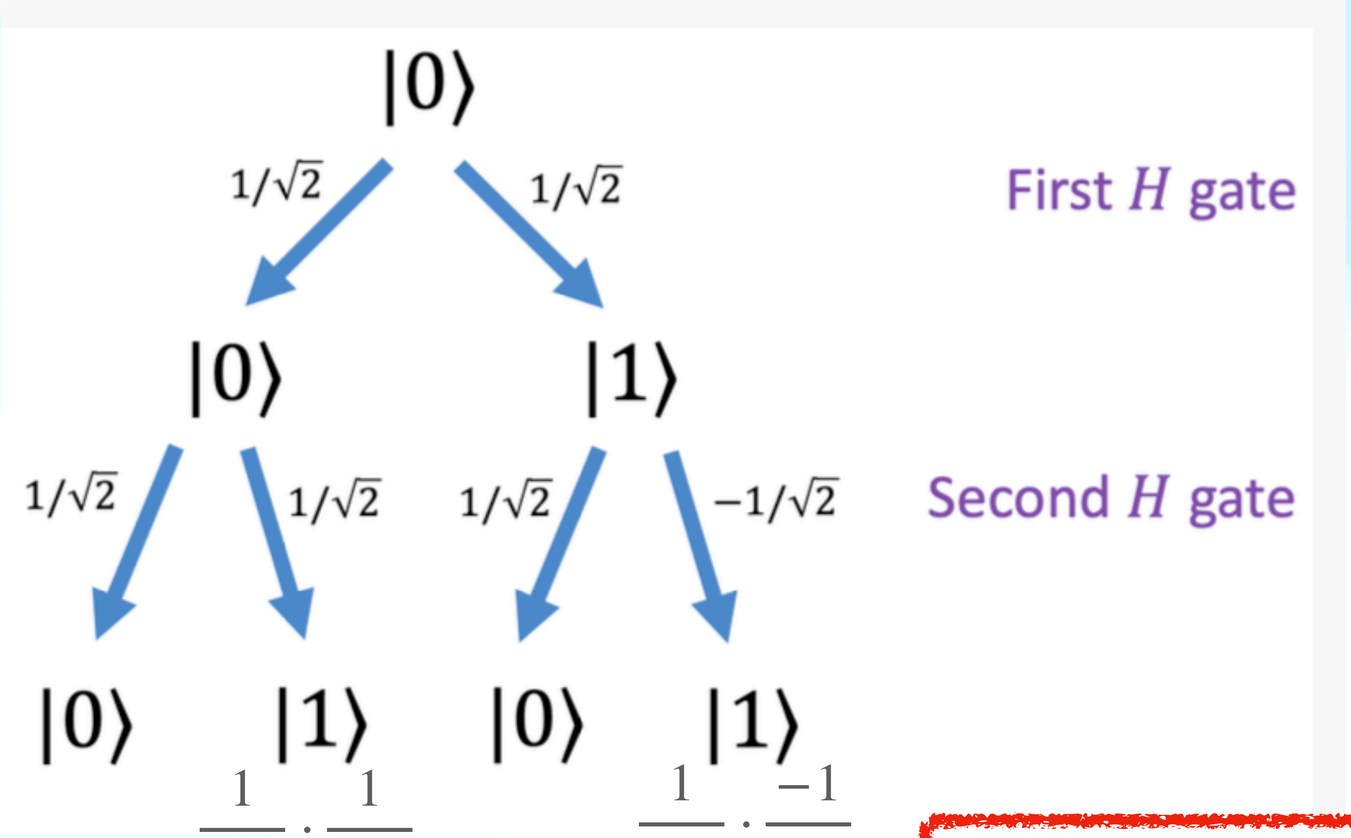
ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Transition amplitude of event $|0\rangle \rightarrow |1\rangle$ = sum of amplitude of all paths from $|0\rangle \rightarrow |1\rangle$

$$= \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{-1}{\sqrt{2}} = 0$$

Path Integral Example





ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Transition amplitude of event $|0\rangle \rightarrow |1\rangle$ = sum of amplitude of all paths from $|0\rangle \rightarrow |1\rangle$

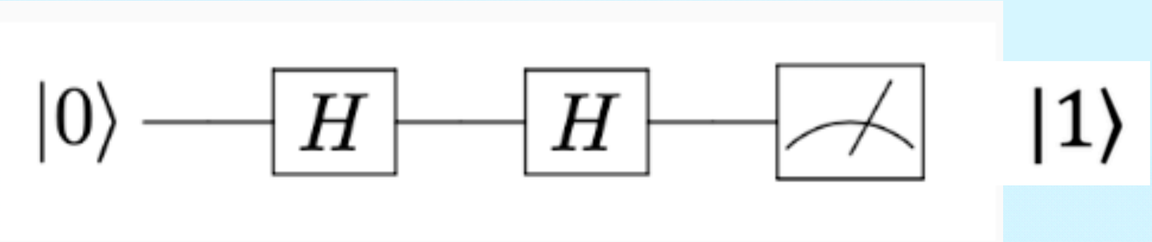
$$= \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{-1}{\sqrt{2}} = 0$$

This phenomena is called inference. Two paths with opposite amplitude value cancel each other.

Suspected to be one of the main reason that power Quantum machines over classical machines.(*)

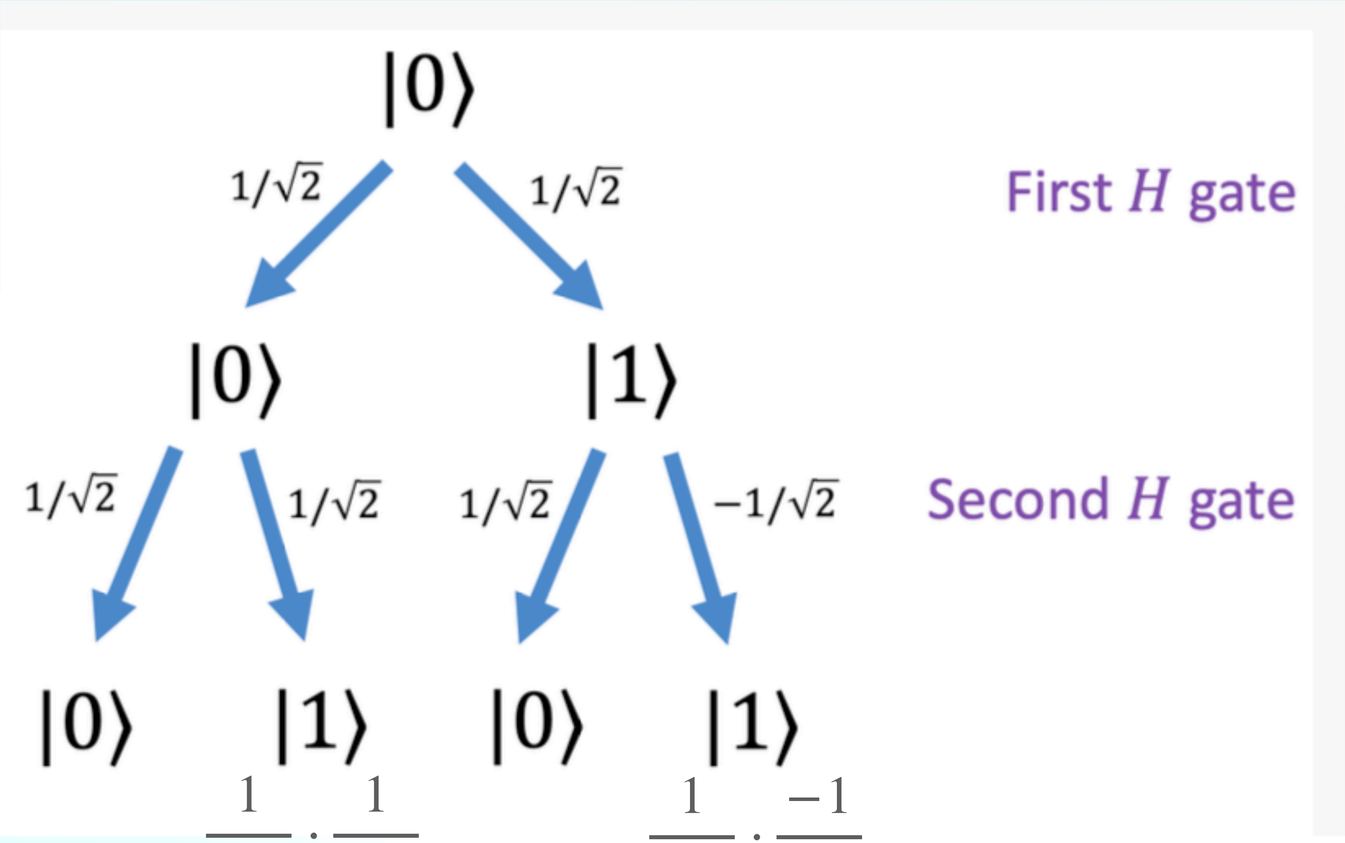
(*) source: Arora & Barak (13.3.5)

Path Integral Example



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

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



ACQ requirement:
Probability of getting $|1\rangle$ on measuring first qubit

Transition amplitude of event $|0\rangle \rightarrow |1\rangle$ = sum of amplitude of all paths from $|0\rangle \rightarrow |1\rangle$

Get the probability of the event by squaring transition amplitude

General case
(Transition Tree)

Gate sequence:

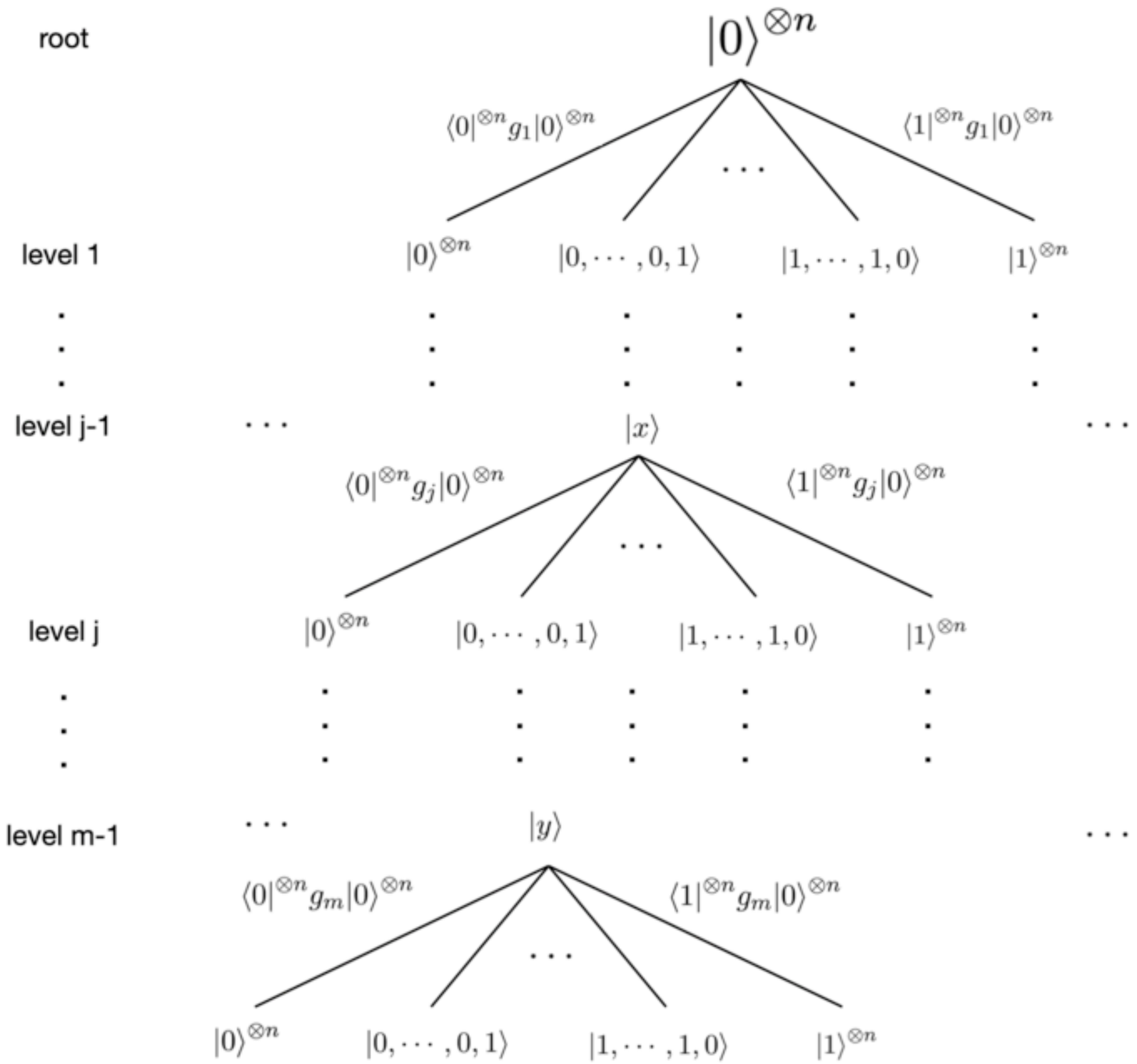
$g_1 \cdot g_2 \cdot g_3 \cdots g_m$

Basis states:

$|i\rangle \in \{0,1\}^n$

Notation:

$|0\rangle^{\otimes n} = |0...0\rangle$



General case

High level idea:

- we calculate weight of each edges (in the path).
- Store the value in workspace.
- To get probability amplitude(PA) of the path, multiply all the edge values.
- Several paths might end up at a certain basis state. Add probability amplitude of such **paths**. [Call it an event. Ex: $|0...0\rangle \rightarrow |10...0\rangle$]

General case

High level idea:

- we calculate weight of each edges (in the path).
- Store the value in workspace.
- To get probability amplitude(PA) of the path, multiply all the edge values.
- Several paths might end up at a certain basis state. Add probability amplitude of such **paths**. [Call it an event. Ex: $|0...0\rangle \rightarrow |10...0\rangle$]
- Get the probability of an event by squaring PA, and delete all the spaces used for calculating PA for the event.
- Repeat it for all cases(*) contributing to AQC
- Sum the probability for all the cases to get value associated with AQC

(*) For AQC, cases means all those paths ending at basis states having 1 as the first bit. (Ex- 10…0, 110…0 etc)

Computing weight of the edge in a path

Claim: For a certain path, computing weight of edges and the path amplitude probability require polynomial amount of space.

Computing weight of the edge in a path

Claim: For a certain path, computing weight of edges and the path amplitude probability require polynomial amount of space.

Theorem: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g , the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial Space. [Due to the next lemma]

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g, the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial Space.

Proof: Assume (w.l.g) that gate g acts on first two qubits. Or,

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g, the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial Space.

Proof: Assume (w.l.g) that gate g acts on first two qubits. Or,

$$\langle y_1 y_2 \ldots y_n | g_{12} \otimes I_3 \otimes \ldots \otimes I_n | x_1 x_2 \ldots x_n \rangle$$

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g, the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial Space.

Proof: Assume (w.l.g) that gate g acts on first two qubits. Or,

$$\langle y_1 y_2 \ldots y_n | g_{12} \otimes I_3 \otimes \ldots \otimes I_n | x_1 x_2 \ldots x_n \rangle$$

$$\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle \cdot \langle y_3 y_4 \ldots y_n | I_3 \otimes \ldots \otimes I_n | x_3 x_4 \ldots x_n \rangle$$

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g, the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial Space.

Proof: Assume (w.l.g) that gate g acts on first two qubits. Or,

$$\langle y_1 y_2 \ldots y_n | g_{12} \otimes I_3 \otimes \ldots \otimes I_n | x_1 x_2 \ldots x_n \rangle$$

$$\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle \cdot \langle y_3 y_4 \ldots y_n | I_3 \otimes \ldots \otimes I_n | x_3 x_4 \ldots x_n \rangle$$

Notation: **I_i is identity matrix acting on qubit number i**

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g , the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial time.

Proof: Assume w.l.g that gate g acts on first two qubits. Or,

$$\langle y_1 y_2 \ldots y_n | g_{12} \otimes I_3 \otimes \ldots \otimes I_n | x_1 x_2 \ldots x_n \rangle =$$
$$\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle \cdot \langle y_3 y_4 \ldots y_n | I_3 \otimes \ldots \otimes I_n | x_3 x_4 \ldots x_n \rangle$$

Note: g_{12} is 4×4 matrix

Computing weight of the edge in a path

Lemma: Given n-qubit basis state $|x\rangle$ *and* $|y\rangle$, and two-qubit gate g , the transition amplitude $\langle y|g|x\rangle$ can be computed in (classical) polynomial time.

Proof: Assume w.l.g that gate g acts on first two qubits. Or,

$$\langle y_1 y_2 \ldots y_n | g_{12} \otimes I_3 \otimes \ldots \otimes I_n | x_1 x_2 \ldots x_n \rangle =$$
$$\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle \cdot \langle y_3 y_4 \ldots y_n | I_3 \otimes \ldots \otimes I_n | x_3 x_4 \ldots x_n \rangle$$

Note: g_{12} is 4×4 matrix

Computation Strategy: $\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle$ is the entry of g_{12} in the row indeed by (y_1, y_2) and column indexed by (x_1, x_2)

Computing weight of the edge in a path

Computation Strategy: $\langle y_1 y_2 | g_{12} | x_1 x_2 \rangle$ is the entry of g_{12} in the row indeed by (y_1, y_2) and column indexed by (x_1, x_2)

$$\begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \end{array}$$

Quary the matrix and write down the value in the workspace

General case
(Transition Tree)

Properties of the Tree:

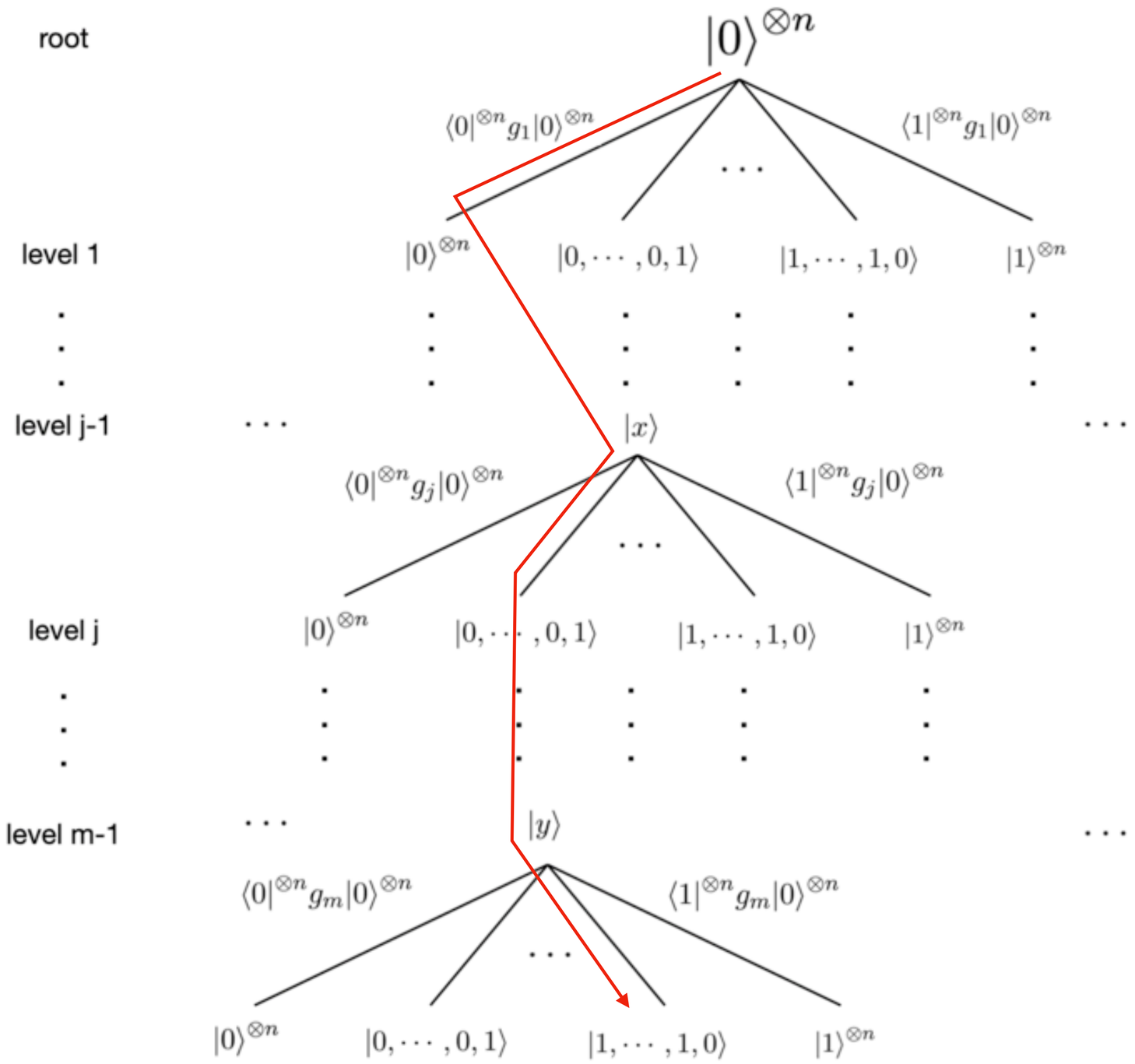
Depth : m; [Poly(n)]

Node at last level:

$$(2^n)^{m-1}$$

Node at last level having first bit as 1.

$$\frac{[(2^n)^{m-1}]}{2}$$



Resource Analysis

(For a specific path) probability amplitude calculation

To store a path information: A path contains ‘m’ edges. We shall store the weights of each edges. Hence, at most $\mathcal{O}(m \cdot n)$ bit space.

To store amplitude value: $poly(n)$ bit space (because, it is product of all the weights in the path)

Resource Analysis of a general path

To store a path information: A path contains 'm' edges. We shall store the weights of each edges. Hence, at most $\mathcal{O}(m \cdot n)$ bit space.

To store amplitude value: $poly(n)$ bit space

AQC Requirement

Path start at: $|0...0\rangle$

Path end at: $|y\rangle$, $s.t,\ y \in \{0,1\}^n\ and\ y_1 = 1$

Circuit description: $C_n = g_1 \ldots g_n$

As per Lemma, $\langle y|C|0...0\rangle$ can be estimated in polynomial space

AQC Requirement

Path start at: $|0...0\rangle$ Path end at: $|y\rangle$, $s.t,\ y \in \{0,1\}^n$ *and* $y_1 = 1$

Circuit description: $C_n = g_1 \ldots g_n$

As per Lemma, $\langle y|C|0...0\rangle$ can be estimated in polynomial space

AQC Calculation: High level idea

Recall: Multiple path ends at certain y (say, $y_{10..0}$). We first add the amplitude associated with each of paths ending at $y_{10..0}$. Then square the resulted sum to get the probability of event $|0...0\rangle \rightarrow |10...0\rangle$.

For AQC, start estimating the probability from basis $y_{10..0}$, and then keep adding probability of each y's having first bit as 1.

AQC Requirement

Path start at: $|0...0\rangle$ Path end at: $|y\rangle$, $s.t,\ y \in \{0,1\}^n$ *and* $y_1 = 1$

Circuit description: $C_n = g_1 \ldots g_n$

As per Lemma, $\langle y|C|0...0\rangle$ can be estimated in polynomial space

AQC Calculation

Algorithm: 1. $prob \leftarrow 0$

2. For all $y \in \{0,1\}^n$ $s.t$ $y_1 = 1$

2.1 prob += $|\langle y|C|0...0\rangle|^2$

3. If promise is satisfies, output YES, o.w NO

AQC Requirement

Path start at: $|0...0\rangle$ Path end at: $|y\rangle$, $s.t,\ y \in \{0,1\}^n$ *and* $y_1 = 1$

Circuit description: $C_n = g_1 \ldots g_n$

As per Lemma, $\langle y|C|0...0\rangle$ can be estimated in polynomial space

AQC Calculation

Estimate probability for certain y (say $y_{10..0}$), then keep adding probability of each y's of interest for AQC problem

Algorithm: 1. $prob \leftarrow 0$

2. For all $y \in \{0,1\}^n$ $s.t$ $y_1 = 1$

2.1 prob += $|\langle y|C|0...0\rangle|^2$

3. If promise is satisfies, output YES, o.w NO

AQC Requirement

Path start at: $|0...0\rangle$ Path end at: $|y\rangle$, $s.t,\ y \in \{0,1\}^n$ *and* $y_1 = 1$

Circuit description: $C_n = g_1 \ldots g_n$

As per Lemma, $\langle y|C|0...0\rangle$ can be estimated in polynomial space

AQC Calculation

Estimate probability for certain y (say $y_{10..0}$), then keep adding probability of each y's of interest for AQC problem

Algorithm: 1. $prob \leftarrow 0$

2. For all $y \in \{0,1\}^n$ $s.t$ $y_1 = 1$

2.1 prob += $|\langle y|C|0...0\rangle|^2$

3. If promise is satisfies, output YES, o.w NO

Overall process requires polynomial space using Feynman path formalism (But it is quite worse in time complexity if compared to operator formalism)

Conclusion

$$BQP \subseteq PSAPCE$$

Corollary: $P \subset BQP \implies P \subset PSPACE$

Reversible Computation

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NAND is an irreversible gate

Quantum gates are reversible

Claim:

Irreversible gate $\longleftrightarrow$ Reversible gate

(With polynomial overhead of resources)

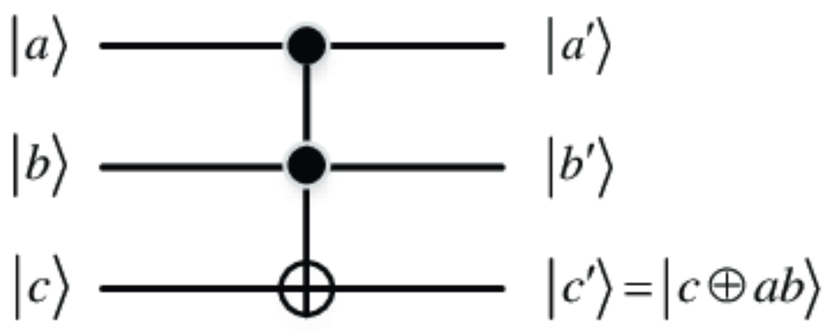
Source: John Watrous lecture link:

Reversible Computation

Toffoli Gate:

An universal gate for reversible computation

Inputs			Ouputs		
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



Source: John Watrous lecture link:

NAND gate from Toffoli Gate

Fix the value of the third bit to be 1.

Inputs			Ouputs		
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

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Reversible Computation

Recall:

{H, Toffoli} forms universal set of Quantum computation

How to generate random bit in Quantum Computer?

Use Hadamard gate trick:

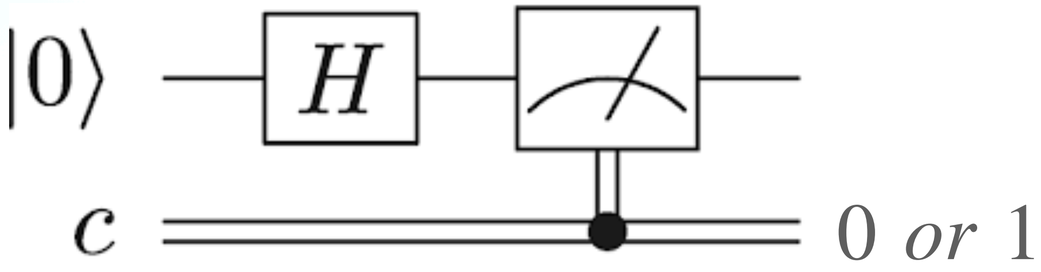
Reversible Computation

Recall:

{H, Toffoli} form universal set of Quantum computation

How to generate random bit in Quantum Computer?

Use Hadamard gate trick:



(Detailed in slide 66)

Circuit Definition of class P

Definition:

A given promise problem $\mathbb{A} = A_{Yes}, A_{No}$ is in P if and only if there exists a polynomial-time generated family $C = \{C_n : n \in \mathbb{N}\}$ of Boolean circuits, where each circuit C_n takes n and output 1 bit, such that:

1. $C(x) = 1 \; for \; all \; x \in A_{Yes}$
2. $C(x) = 0 \; for \; all \; x \in A_{No}$

Quantum Simulation of P

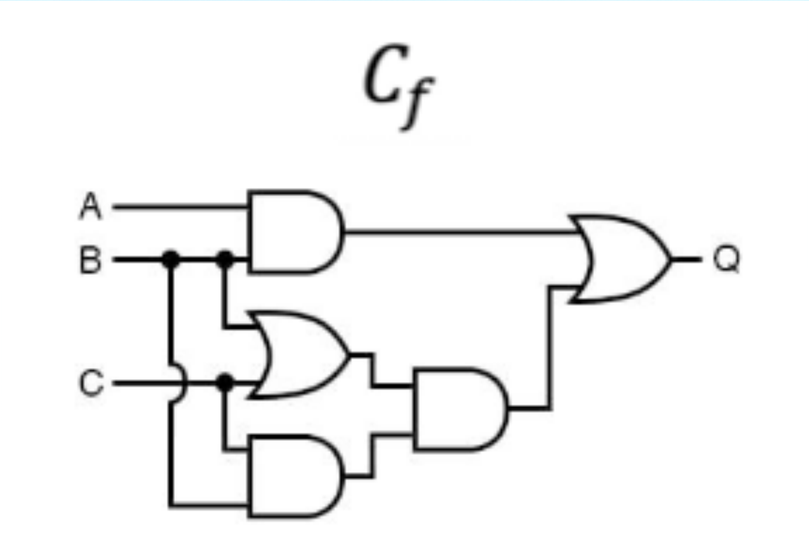
Simulation Requirement:

- The definition talks about Boolean circuit.
- w.l.g the circuit contains NAND gates.

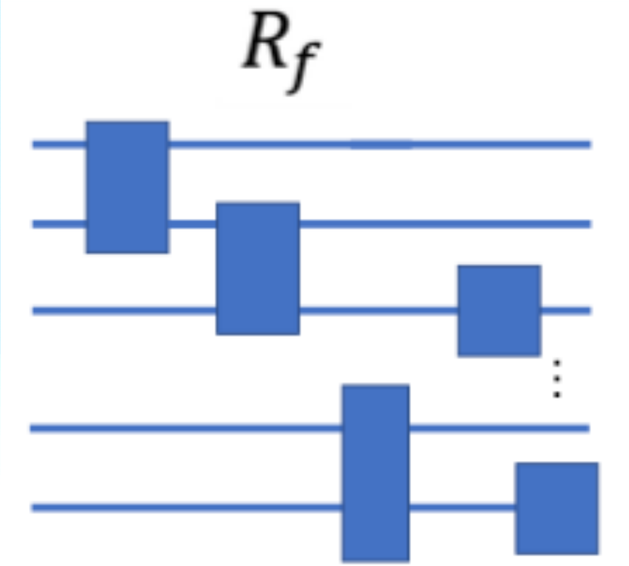
Simulation Strategy:

- Recall, NAND can be implemented by Toffoli gate
- Simulate circuit family $C = \{C_n : n \in \mathbb{N}\}$ gate-by-gate to obtain quantum circuit family $Q = \{Q_n : n \in \mathbb{N}\}$

High level understanding of the conversion



Boolean Circuit



Quantum Circuit

- Some extra ancilla (helper) bits are required
- Conventionally, the first qubit contain the answer

Circuit Definition of class BPP

Definition:

A given promise problem $\mathbb{A} = A_{Yes}, A_{No}$ is in BPP if and only if there exists a polynomial bounded function 'r' and polynomial-time generated family $C = \{C_n : n \in \mathbb{N}\}$ of Boolean circuits, where each circuit C_n takes $n + r(n)$ input bits and output 1 bit, such that:

1. $Pr[C(x, y) = 1] \geq 2/3 \; for \; all \; x \in A_{Yes}$
2. $Pr[C(x, y) = 1] \leq 1/3 \; for \; all \; x \in A_{No}$

Where $y = \Sigma^{r(|x|)}$ is chosen uniformly at random

Circuit Definition of class BPP

Definition:

A given promise problem $\mathbb{A} = A_{Yes}, A_{No}$ is in BPP is and only if there exists a polynomial bounded function 'r' and polynomial-time generated family $C = \{C_n : n \in \mathbb{N}\}$ of Boolean circuits, where each circuit C_n takes $n + r(n)$ input bits and output 1 bit, such that:

1. $Pr[C(x, y) = 1] \geq 2/3 \; for \; all \; x \in A_{Yes}$
2. $Pr[C(x, y) = 1] \leq 1/3 \; for \; all \; x \in A_{No}$

Where $y = \Sigma^{r(|x|)}$ is chosen uniformly at random

Note: $y = \Sigma^{r(|x|)}$ depends on input size, not on actual value of the input

Quantum Simulation of BPP

Simulation Requirement:

- The definition talks about Boolean circuit.
- With loss of generality, assume the circuit contains NAND gates only.
- Random bits are essential

Simulation Strategy:

- Recall, NAND can be implemented by Toffoli gate
- Use Hadamard-trick to generate random number
- Simulate circuit family $C = \{C_n : n \in \mathbb{N}\}$ gate-by-gate to obtain quantum circuit family $Q = \{Q_n : n \in \mathbb{N}\}$

Oracle separation of NP and BQP

[Bennett-Bernstein-Brassard-Vazirani (1997)]

Result: $\exists O,\ s.t\ NP^O \neq BQP^O$

Paper: Strength and weakness of Quantum computing

Oracle separation of NP and BQP

[Bennett-Bernstein-Brassard-Vazirani (1997)]

Result: $\exists O,\ s.t\ NP^O \neq BQP^O$

Unstructured Search Problem(USP)

Input: **A black function** $f: \{0,1\}^n \rightarrow \{0,1\}$

Output: **Does there exist** $x, such\ that\ f(x) = 1$

Paper: Strength and weakness of Quantum computing

Oracle separation of NP and BQP

[Bennett-Bernstein-Brassard-Vazirani (1997)]

Result: $\exists O,\ s.t\ NP^O \neq BQP^O$

Unstructured Search Problem(USP)

Input: **A black function** $f: \{0,1\}^n \rightarrow \{0,1\}$

Output: **Does there exist** x, *such that* $f(x) = 1$

Claim-1: There exist an NP (oracle) algorithm that solves the USP with single query.

Reason: If there is a solution x, the NP (oracle) algorithm guesses x and query the oracle to see if $f(x) = 1$

Paper: Strength and weakness of Quantum computing

Oracle separation of NP and BQP

[Bennett-Bernstein-Brassard-Vazirani (1997)]

Unstructured Search Problem(USP)

Claim-2: Any Quantum (oracle) algorithm must make $\mathcal{O}(\sqrt{2^n})$ query to the oracle.

Reason:

- There exists a quantum (oracle) algorithm (named Grover's algorithms) that solves USP in $\mathcal{O}(\sqrt{2^n})$ query to the oracle.
- The paper proved Grover's algorithm is optimal. Or, the lower bound on the number of the oracle query is tight.

Paper: Strength and weakness of Quantum computing

Oracle separation of BQP and PH
[Raz-Tal (2018)]

Result: $\exists O,\ s.t\ BQP^O \nsubseteq PH^O$

Paper: Oracle separation of BQP and PH

Oracle separation of BQP and PH
[Raz-Tal (2018)]

Result: $\exists O,\ s.t\ BQP^O \nsubseteq PH^O$

Some terminology

- Let $\mathscr{D}$ *and* $\mathscr{D}'$ be two probability distribution over a finite set X.
- Definition: An algorithm $\mathscr{A}$ distinguish between $\mathscr{D}$ *and* $\mathscr{D}'$ with advantage ϵ if:

$$\epsilon = |Pr_{(x\sim\mathscr{D})}[\mathscr{A}\ accept\ x] - Pr_{(x'\sim\mathscr{D}')}[\mathscr{A}\ accept\ x']|$$

Paper: Oracle separation of BQP and PH

Oracle separation of BQP and PH
[Raz-Tal (2018)]

Result: $\exists O,\ s.t\ BQP^O \nsubseteq PH^O$

Some terminology

- Let $\mathscr{D}$ *and* $\mathscr{D}'$ be two probability distribution over a finite set X.
- Definition: An algorithm $\mathscr{A}$ distinguish between $\mathscr{D}$ *and* $\mathscr{D}'$ with advantage ϵ if:

$$\epsilon = |Pr_{(x\sim\mathscr{D})}[\mathscr{A}\ accept\ x] - Pr_{(x'\sim\mathscr{D}')}[\mathscr{A}\ accept\ x']|$$

Key idea

Authors designed an explicit distribution $\mathscr{D}$ over inputs in $\{\pm 1\}^{2N}$ that is Quantumly easy but classically hard to distinguish $\mathscr{D}$ from the uniform distribution. [Formal statement on next page]

Paper: Oracle separation of BQP and PH

Theorem:

There exists an explicit distribution $\mathscr{D}$ over inputs in $\{\pm 1\}^{2N}$ such that following two statements holds:

(1) There exits a Quantum algorithms that makes one query to the input, and runs in $\mathcal{O}(log(N))$ times, that distinguishes between $\mathscr{D}$ and the uniform distribution (U) with advantage $\Omega[\frac{1}{log(N)}]$

Theorem:

There exists an explicit distribution $\mathscr{D}$ over inputs in $\{\pm 1\}^{2N}$ such that following two statements holds:

(1) There exits a Quantum algorithms that makes one query to the input, and runs in $\mathscr{O}(log(N))$ times, that distinguishes between $\mathscr{D}$ and the uniform distribution (U) with advantage $\Omega[\frac{1}{log(N)}]$

(2) No Boolean circuit of size $quasipoly(N)$ and constant depth distinguishes between $\mathscr{D}$ and U with advantage than $\frac{polylog(N)}{\sqrt{N}}$.

Theorem:

There exists an explicit distribution $\mathscr{D}$ over inputs in $\{\pm 1\}^{2N}$ such that following two statements holds:

(1) There exits a Quantum algorithms that makes one query to the input, and runs in $\mathcal{O}(log(N))$ times, that distinguishes between $\mathscr{D}$ and the uniform distribution (U) with advantage $\Omega[\frac{1}{log(N)}]$

(2) No Boolean circuit of size $quasipoly(N)$ and constant depth distinguishes between $\mathscr{D}$ and U with advantage than $\frac{polylog(N)}{\sqrt{N}}$.

Talks about a limitation of AC^0

Relationship between PH and AC^0

Replace every ∀ by conjunction
Replace every ∃ by disjunction

Paper: Oracle separation of BQP and PH

Relationship between PH and AC^0

Replace every $\forall$ by conjunction
Replace every $\exists$ by disjunction

Two corollaries

Corollary-I: In the black-box model, promise-BQP$\not\subseteq$promise-PH

Corollary-II: There exists an oracle O relative to which, $BQP^O \not\subseteq PH^O$

Paper: Oracle separation of BQP and PH

Take aways

- Basic Rules of Quantum Computation
- BQP complexity class
- Promise-BQP complete problems
- BQP is in EXP
- BQP is PSPACE
- Reversible Computation
- P and BPP are in BQP
- NP and BQP separation in relativized world
- BQP and PH separation in relativized world

Source/Citations

- Arora & Barak: Complexity Theory (A modern Approach)
- Nielsen & Chuang: Quantum Computation & Information
- Henry Yean (Introduction to Quantum Computing (Fall 2023))
- Quantun Complexity theory: John Watrous
- [BBBV97] Power and limitation of Quantum computing
- [Raz-Tal18] Oracle separation of BQP and PH

Q & A from the talk

Q: How promiseBQP-completeness is defined?

A: The promiseBQP-complete problem shall lie in promiseBQP and all the problem in promiseBQP has a randomised reduction to that problem.

Note: Efficient algorithm for promiseBQP-complete problem implies an efficient algorithm for BQP, but not vice-a-versa. [Source(*)]

Q: AQC is in promiseBQP or BQP?

A: AQC is in promiseBQP. It is also promiseBQP-complete. It is still unknown if BQP has any complete problem.

Q: What is the meaning of APPROX. in 'APPROX Q-CIRCUIT' problem?

A: The sources I consulted for AQC don't mention any reason for this choice. I will do further inspection on it.

(*) source link

Thanks!