CS467: Introduction to Quantum Information Processing

Felix Zhou [1]

June 1, 2023

[1]From Professor Ashwin Nayak's Lectures at the University of Waterloo in Winter 2021

Contents

Chapter 1

Introduction

1.1 Qubits, Quantum States, and Measurement

The quantum bit, or *qubit*, is the quantum equivalent of the classic unit of information: the bit. Qubits are the elementary building blocks of quantum computers. We can think of a qubit as a register in the classical computer.

The state of a qubit is described by a unit vector in $\mathbb{C}^2$. This space is imbued with the natural Euclidean inner product.

Definition 1.1.1 (Ket)
A vector $|\psi\rangle \in \mathbb{C}^2$.

Example 1.1.1
The following are examples of states of a qubit.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Definition 1.1.2 (Probability Amplitude)
Given a quantum state $\alpha |0\rangle + \beta |1\rangle$. The coefficients α, β are the probability amplitudes of $|0\rangle, |1\rangle$ respectively.

Recall that an orthonormal basis of a finite dimensional inner product space is a basis of unit vectors such that is pairwise orthogonal (inner product 0).

Example 1.1.2
The canonical basis $|0\rangle, |1\rangle$ is an orthonormal basis of $\mathbb{C}^2$. The following are all orthonormal bases of $\mathbb{C}^2$

$$\{|+\rangle, |-\rangle\}$$
$$\{|v_0\rangle, |v_1\rangle\} := \{|0\rangle + i|1\rangle, |0\rangle - i|1\rangle\}$$

Definition 1.1.3 (Bra)
The linear algebraic dual (conjugate transpose) of $|v\rangle$ is denoted $\langle v|$.

Example 1.1.3
The following are duals

$$\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}$$
$$\langle v_0| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1, -i \end{pmatrix}$$

We can succintly represent the inner product between vectors $|u\rangle, |v\rangle \in \mathbb{C}^d$ as $\langle u|v\rangle$.

Example 1.1.4
The following are examples of inner products expressed with bra-ket notation.

$$\langle 0|+\rangle = \frac{1}{\sqrt{2}}$$
$$|\langle +|v_0\rangle| = \left|\frac{1}{2}(1+i)\right|$$
$$= \frac{1}{\sqrt{2}}$$

Definition 1.1.4 (Complete Projective Measurement)
Let $B := \{|u_0\rangle, |u_1\rangle\}$ be an orthonormal basis of $\mathbb{C}^2$.
The effect of a measurement in B on a qubit in state $|v\rangle$ is an outcome of either $|u_0\rangle, |u_1\rangle$, with probability $|\langle u_b|v\rangle|^2$ for $b = 0, 1$.
The qubit is then left in the state $|u_b\rangle$. In other words, the state collapses to $|u_b\rangle$.

1.2 Bit Commitment

Consider a simple, single message protocol consisting of two stages.

Commit Stage:

1) Alice has a bit $a \in \{0, 1\}$. She sends a message m (depending on a) to Bob.
2) Bob receives m and stores it.

Reveal Phase:

1) Alice sends bit a, and message r to Bob.
2) Bob uses r to check that m is "consistent" with a. If so, he "accepts" and otherwise "rejects".

Requirements

1) Bob cannot learn bit a from the message m alone (Hiding Property).
2) Alice cannot send bit $\bar{a}$ and some message r such that Bob accepts (Binding Property).

Classically, with probability 1, either Bob learns the bit a from m alone, or Alice can claim she committed to $\bar{a}$ and Bob is unable to distinguish.

1.2.1 A Quantum Protocol

Let $|\psi_0\rangle$ be the state which is rotated $\frac{\pi}{8}$ from $|0\rangle$ to $|1\rangle$. Similarly, let $|\psi_1\rangle$ be the state which is rotated $\frac{\pi}{8}$ from $|1\rangle$ to $|0\rangle$. Explicitly,

$$|\psi_0\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle, |\psi_1\rangle = \sin\frac{\pi}{8}|0\rangle + \cos\frac{\pi}{8}|1\rangle.$$

Commitment Stage

1) Let $a \in \{0,1\}$ be Alice's bit. She prepares a qubit M in the state $|\psi_a\rangle$.
2) She sends M to Bob, who stores this qubit.

Let $\left|\tilde{\psi}_0\right\rangle$ be some state orthogonal to $|\psi_0\rangle$ and $\left|\tilde{\psi}_1\right\rangle$ be some state orthogonal to $|\psi_1\rangle$.

Reveal Stage

1) Alice sends bit a, and no other message.
2) Bob measures qubit M in basis $\left\{|\psi_a\rangle, \left|\tilde{\psi}_a\right\rangle\right\}$. He accepts a if the qubit is left in state $|\psi_a\rangle$.

Proposition 1.2.1
The quantum protocol satisfies the hiding and binding properties probabilistically:
(a) Given the qubit M, $\min_{a\in\{0,1\}} P(\text{outcome} = a) \leq \delta$ for some $\delta < 1$. This happens regardless of the measurement Bob makes.
(b) For any state in which Alice prepares M, if she wishes to claim bit b in the reveal stage, $\min_{b\in\{0,1\}} P(\text{Bob accepts } b) \leq \epsilon$ for some $\epsilon < 1$.

Proof (a)
Here, we are considering the case when Bob can "cheat" and perform some other measurement not as indicated in the protocol.

We can show that the optimal measurement for Bob is along the standard basis $\{|0\rangle, |1\rangle\}$ as this maximizes $P(\text{outcome} = a|\,|\psi_a\rangle)$.

By computation

$$P(\text{outcome} = 0|\,|\psi_0\rangle) = \cos^2\frac{\pi}{8}$$
$$P(\text{outcome} = 1|\,|\psi_1\rangle) = \sin^2\frac{\pi}{8}.$$

Thus $\delta \approx 0.85$ and the hiding property is satisfied with this probability.

Proof (b)
Here, we are considering the case when Alice can "cheat" by preparing some state that is not either $|\psi_0\rangle, |\psi_1\rangle$, and claim she had the right answer later.

Let $|\psi\rangle := \frac{1}{\sqrt{2}}(|\psi_0\rangle + |\psi_1\rangle)$. $|\psi\rangle$ is a state maximizing

$$\begin{aligned}\min_{a\in\{0,1\}} |\langle\psi_a|\psi\rangle|^2 &= P(\text{outcome} = a|\,|\psi\rangle\,, \text{Alice claims } a)\\ &= \cos^2\frac{\pi}{8}\\ &\approx 0.85.\end{aligned}$$

Hence the concealing property is satisfied with $\epsilon \approx 0.85$.

1.3 Multiple Qubits

As the quantum analog to the classical memory, we have sequences of, say n qubits. There are 2^n perfectly distinguishable states labelled as n-bit binary strings.

Definition 1.3.1 (Pure State)
A general pure quantum state is a unit vector in $\mathbb{C}^d$ where $d = 2^n$.

Using the Dirac notation, we write

$$|\psi\rangle := \sum_{x\in\{0,1\}^n} \alpha_x\,|x\rangle\,.$$

Since $|\psi\rangle$ is a unit vector,

$$\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1.$$

Example 1.3.1
The following are examples of multiple qubit states

$$\begin{aligned}|\psi_1\rangle &= \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\\ |\psi_2\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)\\ |\psi_3\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)\\ |\psi_4\rangle &= \frac{2}{5}i\,|001\rangle + \frac{2\sqrt{3}}{5}\,|011\rangle - \frac{3}{5}\,|101\rangle\,.\end{aligned}$$

1.3.1 Tensor Products

Definition 1.3.2 (Tensor Product)
For vectors $|u\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}$, the tensor product, denoted $|u\rangle \otimes |v\rangle$, is a vector in $\mathbb{C}^{d_1 \times d_2}$.

Suppose $\{e_i\}, \{f_j\}$ are the standard basis vectors for $\mathbb{C}^{d_1}, \mathbb{C}^{d_2}$, respectively. Suppose we index the standard basis vectors $\{g_{i,j}\}$ of $\mathbb{C}^{d_1 d_2}$ by pairs (i,j) for $i \in [d_1], j \in [d_2]$.

Let $|u\rangle = \sum_i u_i e_i \in \mathbb{C}^{d_1}$ and $|v\rangle = \sum_j v_j f_j \in \mathbb{C}^{d_2}$. Then

$$\begin{aligned} |u\rangle \otimes |v\rangle &= \left(\sum_i u_i e_i\right) \otimes \left(\sum_j v_j f_j\right) \\ &:= \sum_{ij} u_i v_j g_{i,j} \\ &\in \mathbb{C}^{d_1 d_2}. \end{aligned}$$

In Dirac notation,

$$|u\rangle \otimes |v\rangle = \sum_{ij} u_i v_j |i,j\rangle .$$

For simplicity, $|u\rangle \otimes |v\rangle$ is written as

$$|u\rangle |v\rangle , |u, v\rangle , |uv\rangle$$

when there is no confusion.

Proposition 1.3.2
The tensor product operation is bilinear:

(a) For all $\alpha \in \mathbb{C}, |u\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}$,

$$(\alpha |u\rangle) \otimes |v\rangle = \alpha(|u\rangle \otimes |v\rangle) = |u\rangle \otimes (\alpha |v\rangle).$$

(b) For all $|u_1\rangle , |u_2\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}$,

$$(|u_1\rangle = |u_2\rangle) \otimes |v\rangle = |u_1\rangle \otimes |v\rangle + |u_2\rangle \otimes |v\rangle .$$

(c) For all $u \in \mathbb{C}^{d_1}, |v_1\rangle , |v_2\rangle \in \mathbb{C}^{d_2}$,

$$|u\rangle \otimes (|v_1\rangle + |v_2\rangle) = |u\rangle \otimes |v_1\rangle + |u\rangle \otimes |v_2\rangle .$$

Thus the tensor product is a function $\otimes : \mathbb{C}^{d_1} \times \mathbb{C}^{d_2} \to \mathbb{C}^{d_1 d_2}$. The question is whether the function is surjective.

Proposition 1.3.3
$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ is not the tensor product of two vectors.

It follows that

$$\mathbb{C}^{d_1 d_2} \neq \{|u\rangle \otimes |v\rangle : |u\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}\}.$$

However,

$$\mathbb{C}^{d_1 d_2} = \text{span}\{|u\rangle \otimes |v\rangle : |u\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}\} =: \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}.$$

Definition 1.3.3 (Product State)
Vectors which may be written as the tensor product of two vectors are product states.

The rest of the states unable to be written this way are *entangled states*.

1.3.2 Tensor Products of Operators

Definition 1.3.4 (Tensor Product)
Suppose $U : \mathbb{C}^{d_1} \to \mathbb{C}^{d_1}, V : \mathbb{C}^{d_2} \to \mathbb{C}^{d_2}$ are linear. The tensor product $U \otimes V$ is a linear operator on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ given on product states by

$$|u\rangle \otimes |v\rangle \mapsto U|u\rangle \otimes V|v\rangle$$

and extended by linearity throughout the entire space.

In matrix form, $U \otimes V$ is given by the block matrix

$$\begin{pmatrix} U_{1,1}V & U_{1,2}V & \dots \\ U_{2,1}V & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix}.$$

Properties

Let $U, U_1, u_2 \in \mathbb{C}^{d_1^2}, V, V_1, V_2 \in \mathbb{C}^{d_2^2}$ and $\alpha \in \mathbb{C}$.

The tensor product is bilinear. That is

(a) $(\alpha U) \otimes V = \alpha(U \otimes V) = U \otimes (\alpha V)$
(b) $(U_1 + U_2) \otimes V = U_1 \otimes V = u_2 \otimes V$

(c) $U \otimes (V_1 + V_2) = U \otimes V_1 + U \otimes V_2$

Moreover,

$$(U_1 \otimes V_2) \cdot (U_2 \otimes V_2) = (U_1 U_2) \otimes (V_2 V_2).$$

The adjoint of a tensor product is

$$(U \otimes V)^* = U^* \otimes V^*.$$

The inverse of a tensor product is simply

$$(U \otimes V)^{-1} = U^{-1} \otimes V^{-1}.$$

The operator norm of a tensor product is

$$\|U \otimes V\| = \|U\| \cdot \|V\|.$$

1.3.3 Inner Product

The inner product in $\mathbb{C}^{d_1} \otimes C^{d_2}$ is inherited from the tensored spaces.

Suppose $|u_1\rangle |v_1\rangle \in |u_2\rangle |v_2\rangle \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$. Their inner product on prouduct vectors is

$$\langle u_1| \langle v_1| \cdot |u_2\rangle |v_2\rangle = \langle u_1|u_2\rangle \cdot \langle v_1|v_2\rangle .$$

For general vectors, we simply extend by conjugate linearity.

1.3.4 Outer Product

Definition 1.3.5 (Outer Product)
Let $|u\rangle \in \mathbb{C}^{d_1}, |v\rangle \in \mathbb{C}^{d_2}$. The outer product of $|u\rangle, |v\rangle$ is

$$|u\rangle \cdot \langle v| .$$

$$|u\rangle \langle v| = \begin{pmatrix} u_1 v^* \\ u_2 v^* \\ \cdots \\ u_{d_1} v^* \end{pmatrix} .$$

The outer product is helpful in writing and manipulating matrices. Take $U := (\alpha_{ij})$. Then

$$U = \sum_{ij} \alpha_{ij} e_i e_j^*.$$

1.4 Measurement

A sequence of qubits (registers) M in state $|\psi\rangle \in \mathbb{C}^d$.

1.4.1 Complete Projective Measurement

Definition 1.4.1 (Complete Projective Measurement)
A complete projective measurement of M is specified by an orthonormal basis of $\mathbb{C}^d$, say

$$B := \{|u_i\rangle : i \in [d]\}.$$

When M is measured, we get a probabilistic outcome $i \in [d]$. Outcome i occurs with probability

$$|\langle u_i|\psi\rangle|^2.$$

When the outcome is observed to be i, the state of M collapsed to $|u_i\rangle$.

Coarser measurements of states are given by projective measurements.

Definition 1.4.2 (Projective Measurement)
A measurement specified by a sequence of orthogonal projection operators

$$\left\{P_i : i \in [k], \sum_{i=1}^{k} P_i = I\right\}.$$

On measurement of M in the state $|\psi\rangle \in \mathbb{C}^d$, we observe a probabilistic outcome $i \in [k]$. We observe outcome i with probability

$$\|P_i |\psi\rangle\|^2.$$

On outcome i, the state of M becomes

$$\frac{P_i |\psi\rangle}{\|P_i |\psi\rangle\|}.$$

1.4.2 Measuring a Subsystem

Consider registers AB, with state space $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$. Say the state is $|\psi\rangle$.

We wish to measure register A according to the projective measurement $\{P_i : i \in [k]\}$. This is equivalent to measuring the entire system according to

$$\{P_i \otimes I : i \in [k]\}.$$

Example 1.4.1
Consider when the P_i's correspond to a complete measurement in the orthonornal basis $\{|u_i\rangle\}$.

By the Schmidt decomposition theorem, we can write

$$\psi = \sum_{i=1}^{d_1} \alpha \, |u_i\rangle \, |\psi_i\rangle$$

where $\{\psi_j : j \in [d_2]\}$ is an orthonormal basis.

Then on measurement, we observe outcome i with probability $|\alpha_i|^2$, and the state becomes

$$\frac{(P_i \otimes I) \, |\psi\rangle}{|\alpha_i|} = |u_i\rangle \, |\psi_i\rangle \, .$$

Remark on measurements: Since $\sum_i P_i = I$ for any projective measurement with the $\{P_i\}'s$ being orthogonal projectors, we have for each $i \neq j$,

$$\operatorname{Im} P_i \perp \operatorname{Im} P_j$$

(or else the sum cannot be the identity) as well as

$$\begin{aligned} |\psi\rangle &= \sum_{i=1}^{d} P_i \, |\psi\rangle \\ 1 &= \| |\psi\rangle \|^2 \\ &= \sum_{i=1}^{d} \| P_i \, |\psi\rangle \|^2. \end{aligned}$$

So some outcome $i \in [k]$ is observed and $P(\text{outcome} = i)$ is well-defined.

1.4.3 General Measurements

Definition 1.4.3 (General Measurement)
A general measurement of a register M with state space $\mathbb{C}^{d_1}$ consists of preparing another register M' with state space $\mathbb{C}^{d_2}$ in the fixed pure state $|\bar{0}\rangle$, and measuring the composite system MM' together according to a projective measurement on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$.

The register M' is an *ancillary register* or *ancilla*.

Any other kind of measurement can be formulated as a general measurement.

To see how general measurements are more powerful than projective measurements, consider the following example. We have $|\psi_i\rangle, i = 0, 1, 2$, each equi-angle in the real plane.

Proposition 1.4.2
If each of the 3 states occur with probability $\frac{1}{3}$, then with a projective measurement in $\mathbb{C}^2$,

$$P(\text{outcome is } i, \text{ the correct index}) \leq \frac{2}{3}\cos^2\frac{\pi}{12}.$$

The proof is similar to the upper bound for bit commitment cheating probability. However if we use a general measurement, the probability is $\frac{2}{3}$.

Proposition 1.4.3
Even with general measurements, the upper bound on cheating for bit commitment holds:

$$\min_{a\in\{0,1\}} P(\text{outcome is } i : M \text{ in } |\psi_i\rangle) \leq \cos^2\frac{\pi}{8}.$$

1.5 Information Content

An n-qubit state

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

requires $\Omega(2^n)$ parameters to describe.

However, we may only reliably encode $\Theta(n)$ bits into n qubits (Holevo '73).

Theorem 1.5.1 (Nayak '99)
Let $X \in \{0,1\}^m$ be uniformly random. Suppose we encode $x \in \{0,1\}^m$ by an n-qubit state $|\psi_x\rangle$. Let Y be the outcome of any measurement of the state $|\psi_X\rangle \in \mathbb{C}^d, d = 2^n$. Then

$$P(Y = X) \leq \frac{2^n}{2^m}.$$

Proof
Suppose we measure the state according to projection operators

$$\{P_y : y \in \{0,1\}^m\},$$

where outcome y indicates our guess for the encoded string. The operators P_y act on $\mathbb{C}^d \otimes \mathbb{C}^{d'}$, where $d := 2^n$ and d' is arbitrary but finite.

Given state $|\psi_x\rangle$, we append $|\bar{0}\rangle \in \mathbb{C}^{d'}$, and measure according to the projective measurement $\{P_y\}$.

Note that $\sum_y P_y = I$ and each P_y being an orthogonal projector implies that there is a orthonormal basis for $\mathbb{C}^d \otimes \mathbb{C}^{d'}$,

$$\{|f_{yi}\rangle : u \in \{0,1\}^m, i \in [d']\}$$

such that

$$P_y = \sum_i |f_{yi}\rangle\langle f_{yi}| .$$

Consider the subspace $H := \text{span}\{|\psi_x, \bar{0}\rangle : x \in \{0,1\}^m\}$ with dimension at most 2^n. Let $\{|g_j\rangle\}$ be a basis of H. We can write

$$|\psi_x, \bar{0}\rangle = \sum_j \langle g_j|\psi_x, \bar{0}\rangle \, |g_j\rangle .$$

We have

$$\begin{aligned}
P(Y = X) &= \frac{1}{2^m} \sum_x P(Y = x \mid |\psi_x\rangle) \\
&= \frac{1}{2^m} \sum_x \|P_x |\psi_x, \bar{0}\rangle\|^2 \\
&= \frac{1}{2^m} \sum_x \|\sum_{xi} \sum_j |f_{xi}\rangle\langle f_{xi}| \cdot \langle g_j|\psi_x, \bar{0}\rangle |g_j\rangle\|^2 \\
&= \frac{1}{2^m} \sum_x \|\sum_{xi} \sum_j \langle f_{xi}|g_j\rangle \cdot \langle g_j|\psi_x, \bar{0}\rangle |f_{x_i}\rangle\|^2 \\
&= \frac{1}{2^m} \sum_x \sum_{xi} \sum_j \langle f_{xi}|g_j\rangle^2 \cdot \langle g_j|\psi_x, \bar{0}\rangle^2 \\
&\leq \frac{1}{2^m} \sum_x \sum_{xi} \sum_j \langle f_{xi}|g_j\rangle^2 \\
&= \frac{1}{2^m} \sum_j \sum_x \sum_{xi} \langle f_{xi}|g_j\rangle^2 \\
&= \frac{1}{2^m} \sum_j \|g_j\|^2 \\
&= \frac{1}{2^m} \dim H \\
&\leq \frac{2^n}{2^m}.
\end{aligned}$$

If we wish to encode m random bits into n, and ask that $P(Y = X) = p > 0$, then

$$n \geq m - \log_2 \frac{1}{p}.$$

This is the same as the classical bound, thus there is no quantum advantage in terms of encoding with distribution.

1.6 Operations on Quantum Bits

The evolution of a closed quantum system is

1) linear
2) reversible (invertible)
3) norm-preserving

Thus they are given by unitary operators on $\mathbb{C}^d$, the state space of the system.

Example 1.6.1
The hadamard matrix is unitary

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Example 1.6.2 (Pauli Operators)
The following are all unitary

$$\begin{aligned} \text{NOT} = X &:= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ Z &:= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ Y = iXZ &:= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \end{aligned}$$

Computation with qubits thus may be implemented by allowing the entire system to evolve in isolation. If we allow subregister A of AB to evolve but not B. The evolution is described by

$$U \otimes I.$$

A sophisticated computation may involve a sequence of such unitary operations applied to different subregisters.

All operations allowed by quantum physics can be reduced to a composition of

1) addition of ancillas
2) unitary evolutions of the entire system
3) projective measurments

Chapter 2

Superdense Coding & Teleportation

2.1 Superdense Coding

Theorem 2.1.1 (Bennett, Wiesner '92)
Suppose Alice and Bob each hold single qubit registers E_1, E_2, where the joint state of E_1E_2 is

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

(EPR pair).
Suppose Alice has two bits $a, b \in \{0, 1\}$. She can convey with no error, ab to Bob, by sending only 1 qubit to Bob.

1. Alice applies the unitary operator $U_{ab} := X^a Z^b$ on E_1.
2. Alice sends E_1 to Bob.
3. Bob measures E_1E_2 in an orthonormal basis B, the Bell basis $\{|\phi_{xy}\rangle : x, y \in \{0, 1\}\}$.
4. Bob gets outcome ab upon measurement.

The Bell basis is

$$|\phi_{00}\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$
$$|\phi_{01}\rangle := \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\phi_{10}\rangle := \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$$
$$|\phi_{11}\rangle := \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle).$$

We can verify that

$$|\phi_{ab}\rangle = (X^a Z^b \otimes I)\,|\phi_{00}\rangle\,.$$

Thus Bob indeed gets the desired outcome.

2.1.1 Remarks

We cannot send any bits using the EPR pair and only local operations. Moreover, no classical analogue of this protocol exists.

2.1.2 Bit Commitment Revisited

In analyzing the binding property when Alice cheats, we assumed she prepares qubit M in a pure state, and later reveals a bit of her choice. Consider if Alice uses and entanged state in $M'M$, with some local register M' and later measure M'. Can she improve her strategy?

In such a strategy, the bit she sends in the reveal stage may be random, and we need to reformulate the cheating criterion.

Proposition 2.1.2
Even with the most general strategy allowed by quantum physics,

$$\min_{a\in\{0,1\}} P(\text{Alice reveals } a, \text{ Bob accepts}) \leq \cos^2\frac{\pi}{8}.$$

In other words, Alice still cannot reveal both values 0 and 1 and force Bob to accept with probability 1 in each case.

2.2 Teleportation

Suppose Alice and Bob are physically separated but are connected by a classical channel. Suppose Alice has a qubit M in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and would like to help Bob construct $|\psi\rangle$. With only classical messages, she would need to send infinite length messages to send α, β exactly, even if she knew α, β.

Theorem 2.2.1 (Teleportation)
Suppose Alice and Bob share E_1, E_2, respectively, two qubits in the state

$$|\phi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

She can help Bob construct $|\psi\rangle$ by sending 2 classical bits.

Alice has qubits M, E_1 with M is state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Bob has E_2 and E_1E_2 is in state $|\phi_{00}\rangle$.

1) Alice measures qubits ME_1 in the Bell basis $\{|\phi_{ab}\rangle : a, b \in \{0, 1\}\}$.
2) She sends the 2-bit outcome to Bob.
3) Bob receives $a, b \in \{0, 1\}$, and applies the correctional operator X^aZ^b.
4) E_2 is in state $|\psi\rangle$.

Proposition 2.2.2
The final state of E_2 is $|\psi\rangle$.

Recall that the Bell basis vectors were

$$\begin{aligned} za\,|\phi_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\phi_{01}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\phi_{10}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \\ |\phi_{11}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle). \end{aligned}$$

Proof
The state of ME_1E_2 is

$$
\begin{aligned}
&|\psi\rangle |\phi_{00}\rangle \\
&= \frac{1}{\sqrt{2}}(\alpha |000\rangle + \alpha |011\rangle + \beta |100\rangle + \beta |111\rangle) \\
&= \frac{1}{\sqrt{2}}\left[\frac{\alpha}{\sqrt{2}}(|\phi_{00}\rangle + |\phi_{01}\rangle)|0\rangle + \frac{\alpha}{\sqrt{2}}(|\phi_{10}\rangle - |\phi_{11}\rangle)|1\rangle + \frac{\beta}{\sqrt{2}}(|\phi_{10}\rangle + |\phi_{11}\rangle)|0\rangle + \frac{\beta}{\sqrt{2}}(|\phi_{00}\rangle - |\phi_{01}\rangle)|1\rangle\right] \\
&= \frac{1}{2}\left[|\phi_{00}\rangle |\psi\rangle + |\phi_{01}\rangle (Z|\psi\rangle) + |\phi_{10}\rangle (X|\psi\rangle) + |\phi_{11}\rangle (ZX|\psi\rangle)\right]
\end{aligned}
$$

Alice's measurement of ME_1 in the Bell basis yields the result ab and leaves the system E_1E_2 in state $Z^bX^a|\psi\rangle$ with probability $\frac{1}{4}$. By sending a, b, Bob can then apply the correctional operator X^aZ^b to recover $|\psi\rangle$.

Example 2.2.3
Suppose Alice and Bob have an imperfect communication channel, which applies the Z operator with probability 1% to any qubit sent over it.

Suppose Alice teleports $|\psi\rangle$, and instead sends 2 classical bits (through superdense coding perhaps) using the communical channel. Then she can repeatedly send these 2-bits and Bob can correctly apply the correctional operator with arbitrarily high probability.

Chapter 3

Quantum Circuits

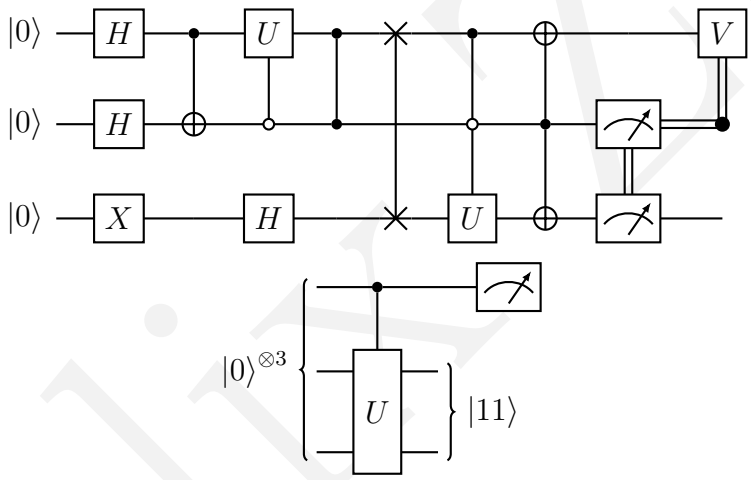

Figure 3.1: Some quantum circuits.

3.1 Classical Circuits

Computers are physical devices which perform a prescribed set of instructions. We can model them using the circuit model.

A classical computer is made of

1) memory, consisting of bits (registers)
2) logical gates, which perform operations on the memory

A bit is a unit of memory taking values 0/1.

Logical gates are functions on a small constant number of bits, mapping them to other bits.

An algorithm (program/circuit) for computing a function $f : \{0,1\}^n \to \{0,1\}^m$ is specified by

1) The number of bits in the memory (*size*). This holds the *input* as well as the *workspace*.
2) A string of s bits, to which the memory is *initialized*: the first n bits to $x \in \{0,1\}^n$ and the rest to 0.
3) A sequence of logic gates, from a fixed set $\mathcal{G}$, along with the indices of memory bits on which they act, as well as the indices of memory bits which contain the output: say the last m bits.

The output of the circuit is the string in the output register after applying the gates in sequence.

A circuit C that takes n-bit inputs and produces m-bit outputs computes a function $\{0,1\}^n \to \{0,1\}^m$. We refer to this function as C.

We say C *computes* a function $f : \{0,1\}^n \to \{0,1\}^m$ if $C(x) = f(x)$ for all inputs x.

Theorem 3.1.1
Any function $f : \{0,1\}^n \to \{0,1\}^m$ can be computed by a boolean circuit over the gate set

$$\{\text{NOT}, \text{AND}\}.$$

Definition 3.1.1 (Universal)
A set of gates $\mathcal{G}$ is universal if any function $\{0,1\}^n \to \{0,1\}^m$ can be computed by a boolean circuit over $\mathcal{G}$.

3.1.1 Efficiency of Circuits

We would like to compute boolean functions with as few resources (time, space) as we can.

The *space complexity* of a circuit is the amount of memory in bits it requires.

The *time complexity* of a circuit is the number of gates. We assume here that each gate takes unit time.

Some functions are easier to compute than others.

Definition 3.1.2 (Efficient)
A circuit / algorithm is efficient if its time complexity is $O(n^k)$ for some constant k, where n denotes the number of input bits.

A point of subtlety is that we wish for the circuits for a problem/function to be *uniform*. Informally, they should use the same method for all input lengths.

3.1.2 Randomized Circuits

A randomized algorithm is a circuit where some its memory bits are initialized to uniformly random bits. The output of such a circuit is also random.

Definition 3.1.3 (High Probability)
Let $f : \{0,1\}^n \to \{0,1\}^m$. We say a random circuit C computes f if

$$P(C(x) = f(x)) \geq \frac{2}{3}$$

for all inputs x.

3.2 Quantum Circuits

Definition 3.2.1 (Quantum Circuit)
A quantum circuit consists of

1) The memory, consisting of qubits (memory)
2) Quantum gates, which perform unitary operations on the memory
3) Measurements

We draw qubits as wires Figure 3.2.

$$|\psi\rangle \text{ —}$$

Figure 3.2: A qubit.

Quantum gates are unitary operators on a small constant number of bits. We depict them

as labelled boxes Figure 3.3.

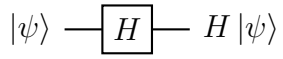

Figure 3.3: A quantum gate.

An important gate is the Toffoli gate Figure 3.4

$$\sum_{a,b\in\{0,1\}} |ab\rangle\langle ab| \otimes X^{a\cdot b}.$$

The Toffoli gate, along with the ability to prepare ancilla in states $|0\rangle, |1\rangle$, gives us universal classical computation. Indeed, if $c = 0$, this it the AND gate. If $b = c = 1$, this is the NOT gate.

The measurements are all single-qubit measurements in the standard (Z) basis Figure 3.5.

Measurements may be complicated in general, and may need a lot of resources. Our choice allows us to quantify the time and space needed. We will see how to implement other measurements using unitary gates and standard basis measurements.

A quantum algorithm/circuit with boolean inputs and outputs is specified by

1) the number of qubits (*size* s) in the memory. This holds the *input* and the *workspace*.
2) The s-qubit state, to which the memory is initialized: The first n qubits to $|x\rangle, x \in \{0,1\}^n$ and the rest to $|\bar{0}\rangle$.
3) A sequence of quantum gates from a fixed set $\mathcal{G}$, along with the induces of memory qubits on which they act. The result of the gate is contained in the same qubits.
4) The indices of memory qubits which contain the *output*: say the last m bits.

To obtain the output of a circuit, we apply the gates in the specified sequence to the initial state $|x, \bar{0}\rangle$.

$$U_t U_{t-1} \dots U_2 U_1 |\psi, \bar{0}\rangle .$$

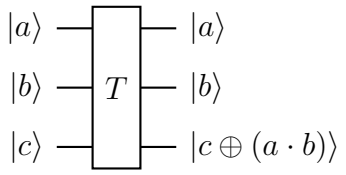

Figure 3.4: The Toffoli gate.

Figure 3.5: A measurement.

We measure the output qubits in the standard basis to obtain a random outcome $C(x)$. The probability of correctness if $P(C(x) = f(x))$.

We say the circuit computes $f : \{0,1\}^n \to \{0,1\}^m$ if the probability of sucess is at least $\frac{2}{3}$ for all inputs x.

Note that we may have circuits that implement unitary operations, or measurements, or any combination of those, which need not compute boolean functions.

Note that we use registers as superscripts to denote

(1) the registers on which an operator acts and which order ($\text{CNOT}^{E_2E_1}$)

(2) the registeres which are in that state ($|00\rangle^{E_1E_2}$)

Example 3.2.1
Consider the registers $ABCD$ in state

$$\frac{1}{2}(|0000\rangle + |1010\rangle + |0101\rangle + |1111\rangle).$$

A compact way of expresing the state is

$$|\phi_{00}\rangle^{AC} |\phi_{00}\rangle^{BD}.$$

where the tensor factors are permuted, but the superscripts help represent the state correctly.

Recall in the decoding phase for superdense coding, we are given bits a, b and apply X^bZ^a on E_2. We can implement this as in Figure 3.6.

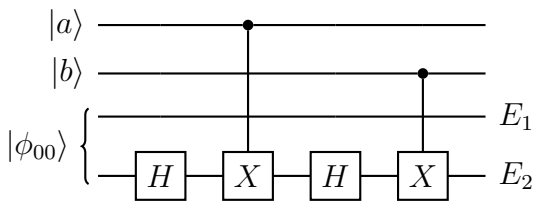

Figure 3.6: Superdense Coding decoding phase.

Example 3.2.2
A common trick is that

$$\begin{aligned} HXH &= Z \\ HYH &= i(HXZH) \\ &= iHXHHZH \\ &= iZX \\ &= -Y. \end{aligned}$$

We may think of $-Y \equiv Y$ from the perspective of measurment. However, $c - Y \not\equiv c - (-Y)$ since they no longer differ only by a global phase.

3.2.1 Measurement in Non-Standard Basis

Say we wish to measurement two qubits along the orthonormal basis $\{|\phi_{ab}\rangle\}$. Take the unitary change of basis matrix $U := \sum_{a,b} |ab\rangle \phi_{ab}$ which sends

$$|\phi_{ab}\rangle \mapsto |ab\rangle .$$

1) Apply U to the registers being measured
2) Measure the registers in the standard basis
3) Apply $U^{-1} = U^*$

Example 3.2.3
Consider this in the example of the Bell basis. Recall that

$$|\phi_{00}\rangle = \mathrm{CNOT}^{E_2E_1}(I \otimes H) |00\rangle^{E_1E_2} .$$

Observe that

$$\begin{aligned} \mathrm{CNOT}^{E_2E_1}(I \otimes H) |ab\rangle^{E_1E_2} &= \mathrm{CNOT}^{E_2E_1} |a\rangle \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle) \\ &= \frac{1}{\sqrt{2}}(|a0\rangle + (-1)^b |\bar{a}1\rangle) \\ &= (X^a Z^b \otimes I) |\phi_{00}\rangle \\ &= |\phi_{00}\rangle . \end{aligned}$$

Also notice that both gates are self-inverses. Hence

$$(I \otimes H) \mathrm{CNOT}^{E_2E_1}$$

yields the inverse change of basis matrix.

3.3 Universality of Gate Sets

We say a quantum circuit C *computes* a unitary operator U on n qubits, if for every n-qubit input state $|\psi\rangle \in \mathbb{C}^d, d = 2^n$, the final state of the circuit is

$$U|\psi\rangle|\bar{0}\rangle$$

with probability 1.

Theorem 3.3.1
For any unitary operation U on $\mathbb{C}^d, d = 2^n$, there is a quantum circuit that uses only the CNOT gate as well as single qubit gates, and computes U.

Definition 3.3.1 (Universal Gate Set)
A gate set $\mathcal{G}$ is universal if for any unitary operation U on $\mathbb{C}^d, d = 2^n$, there is a quantum circuit that uses only gates from $\mathcal{G}$ and computes U.

Hence

$$\{\text{CNOT}\} \cup \mathcal{U}(2)$$

is universal.

Definition 3.3.2 (Operator Approximation)
We say that a unitary operator V on $\mathbb{C}^d, d = 2^n$ approximates another, U, with error $\epsilon \in (0, 1)$, if

$$\|U - V\| \leq \epsilon.$$

Definition 3.3.3 (Circuit Approximation)
We say a quantum circuit approximates a unitary operator U with error $\epsilon \in (0, 1)$ if it computes a unitary operator V such that

$$\|U - V\| \leq \epsilon.$$

Proposition 3.3.2
Suppose $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^d$ are pure quantum states such that

$$\| |\psi\rangle - |\varphi\rangle \| \leq \epsilon$$

for some $\epsilon \in (0, 1)$.
Let $p, q \in \mathbb{R}^k$ be the distributions over outcomes obtained by measuring $|\psi\rangle, |\varphi\rangle$ according to some measurement, respectively. Then

$$\|p - q\|_1 \leq 2\epsilon.$$

Proof
Suppose we add $|\bar{0}\rangle$ and measure according to

$$\{P_i : i \in [k]\}.$$

Then $p_i = \|P_i |\psi, \bar{0}\rangle\|^2$ and $q_i = \|P_i |\varphi, \bar{0}\rangle\|^2$.

Then

$$\begin{aligned}
\|p - q\|_1 &= \sum_i |p_i - q_i| \\
&= \sum_i |\sqrt{p_i} - \sqrt{q_i}| \cdot |\sqrt{p_i} + \sqrt{q_i}| \\
&\leq \|\sqrt{p} - \sqrt{q}\| \cdot \|\sqrt{p} + \sqrt{q}\|.
\end{aligned}$$

Now,

$$\begin{aligned}
\|\sqrt{p} - \sqrt{q}\| &= \sqrt{\sum_i |\|P_i |\psi, \bar{0}\rangle\| - \|P_i |\varphi, \bar{0}\rangle\||^2} \\
&\leq \sqrt{\sum_i \|P_i |\psi, \bar{0}\rangle - P_i |\varphi, \bar{0}\rangle\|^2} && \text{reverse } \Delta \\
&= \sqrt{\sum_i \|P_i(|\psi, \bar{0}\rangle - |\varphi, \bar{0}\rangle)\|^2} \\
&= \sqrt{\||\psi, \bar{0}\rangle - |\varphi, \bar{0}\rangle\|^2} \\
&\leq \epsilon.
\end{aligned}$$

On the other hand,

$$\begin{aligned}\|\sqrt{p}+\sqrt{q}\| &\leq \|\sqrt{p}\|+\|\sqrt{q}\| \\ &= 2\end{aligned}$$

as p, q being unit vectors under the 1-norm means that $\sqrt{p}, \sqrt{q}$ are unit vectors under the 2-norm.

This concludes the proof.

Say a circuit C computes a boolean function f with probability $\frac{2}{3}$ and we approximate the unitary operator U implemented by all its gates together by V with

$$\|U - V\| \leq \epsilon.$$

Then the new circuit C' computes f with probability at least

$$\frac{2}{3} - 2\epsilon.$$

Proposition 3.3.3
Let p, q be probability distributions on $[k]$, and E be any event. Then

$$|p(E) - q(E)| \leq \frac{1}{2}\|p - q\|_1.$$

Definition 3.3.4 (Universal Gate Set)
Let $\mathcal{G}$ be a gate set. We say $\mathcal{G}$ is universal if for any unitary operation U on $\mathbb{C}^d, d = 2^n$, and any $\epsilon \in (0, 1)$, there is a quantum circuit that uses only gates from $\mathcal{G}$ and computes a unitary operation V on $\mathbb{C}^d$ such that

$$\|U - V\| \leq \epsilon.$$

Theorem 3.3.4
The gate set

$$\{\text{CNOT}, H, T\}$$

is universal

The intuition is to approximate arbitrary rotations by rotations of degree θ such that $\frac{\theta}{2\pi}$ is irrational.

Example 3.3.5
We can implement c-Y, c-CNOT exactly with CNOT, H, T.

Theorem 3.3.6 (Solovay-Kitaev)
For any $\epsilon \in (0,1)$ and any operator $U \in \mathcal{U}(2)$, there is a sequence of

$$O\left(\log^c \frac{1}{\epsilon}\right)$$

gates from $\{H, T\}$ which computes $V \in \mathcal{U}(2)$ such that

$$\|V - U\| \leq \epsilon,$$

where c is a universal constant.

Proposition 3.3.7
If $\|U_i - V_i\| \leq \epsilon_i$ for $i \in [t]$, then

$$\|V_t V_{t-1} \dots V_1 - U_t U_{t-1} \dots U_1\| \leq t\epsilon.$$

Proof
Observe that

$$\begin{aligned}\|V_2 V_1 - U_2 U_1\| &\leq \|(V_2 - U_2)V_1\| + \|U_2(V_2 - U_2)\| \\ &\leq 2\epsilon.\end{aligned}$$

The rest follows from induction.

To get an overall error ϵ when approximating m gates in the circuit C, we need only approximate each with error $\frac{\epsilon}{m}$. Thus to approximate a circuit of m gates, we need at most a sequence of

$$O\left(m \log^c \frac{m}{\epsilon}\right)$$

gates.

Due to the above result, we may as well assume any gate is available to us!

3.4 Implementing Measurements

As seen before, we can implement a complete measurement in terms of a change of basis basis to the standard basis, a measurement in the standard basis, and the inverse change of basis matrix.

Consider a projective measurement according to $\{P_i : i \in [k]\}$. Recall we can decompose

$$P_i = \sum_j |v_{ij}\rangle\langle v_{ij}| .$$

We implement the basis change operator

$$U := \sum_{i,j} |i,j\rangle\langle v_{ij}|$$

and

(a) Apply U to get the indices in registers A_1A_2.

(b) Copy A_1 containing the first index into an ancilla B.

(c) Measure B in the standard basis.

(d) Apply U^* to A_1A_2.

In other words, we apply the unitary operator U_P

$$|v_{ij}\rangle |0\rangle \mapsto |v_{ij}\rangle |i\rangle ,$$

then measure the ancilla in the standard basis.

A general measurement involves addition of an ancilla before a projective measurement. This is straightforward in a circuit. In summary, the complexity of implementing a measurement is captured by that of the basis change operation.

3.5 Efficiency of a Quantum Circuit

The *space complexity*, or memory size is the number of qubits in the memory. The *time complexity* or size of the circuit is the number of quantum gates in the circuit.

We say that a family of quantum circuits is *efficient* if its time complexity is $O(n^c)$ for a constant c, AND the family is uniform. Here n is the size of the input.

Definition 3.5.1 (Polynomial Time)
The complexity class $\mathcal{P}$ consists of all families of boolean functions

$$\{f_n : \{0,1\}^n \to \{0,1\}\}$$

(decision problems) which have efficient deterministic classical circuits.

Definition 3.5.2 (Bounded-Error Probabilistic Polynomial Time)
The complexity class $\mathcal{BPP}$ consists of all families of boolean functions

$$\{f_n : \{0,1\}^n \to \{0,1\}\}$$

(decision problems) which have efficient randomized classical circuits.

Definition 3.5.3 (Bounded-Error Quantum Polynomial Time)
The complexity class $\mathcal{BQP}$ consists of all families of boolean functions

$$\{f_n : \{0,1\}^n \to \{0,1\}\}$$

(decision problems) which have efficient quantum circuits.

3.5.1 Simulating Classical Computation

Since classical physics is a special case of quantum physics, we expect to be able to efficiently simulate classical computation with quantum computation.

We have seen that the Toffoli gate can simulate $\{\text{AND}, \text{NOT}\}$, given ancillas in state $|0\rangle, |1\rangle$.

However, classical computation leaves "side-effects", which is not in general unitary. To be specific, we can compute $f : \{0,1\}^n \to \{0,1\}^m$ by a unitary operation

$$U_f : |x, \bar{0}, \bar{0}\rangle \mapsto |x, f(x), \bar{0}\rangle .$$

However, classical computation may leave intermediate results and instead have

$$\tilde{U}_f : |x, \bar{0}, \bar{0}\rangle \mapsto |x, f(x), g(x)\rangle ,$$

where $g(x)$ are the intermediate results.

Instead, we can implement the circuit in Figure 3.7.

1) Compute $\tilde{U}_f$.
2) Copy $f(x)$ into fresh ancilla
3) Reverse the computation $\tilde{U}_f$

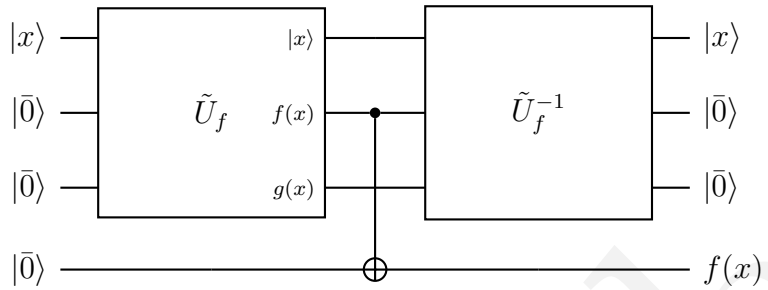

Figure 3.7: reversible classical computation.

If the classical computation U_f takes space s, time t, and m-bit output. Then the quantum space is at most

$$s + 2t + m$$

accounting for bits, ancilla for gates, and ancilla for output. The quantum time is at most

$$2t + 2t + m$$

accounting for the NOT, c-CNOT, CNOT gates.

Thus if the deterministic circuit is efficient, so is the quantum simulation.

Theorem 3.5.1
The two complexity classes $\mathcal{P}, \mathcal{BQP}$ satisfy

$$\mathcal{P} \subseteq \mathcal{BQP}.$$

3.5.2 Simulating Randomized Circuits

A randomized classical circuits are deterministic circuits with some memory bits inititalized to uniformly random bits. See Figure 3.8 where r indicates some random bits. Then

$$P(C(x) = y) = P(h(x, R) = y)$$

where R is uniformly distributed over $\{0, 1\}^k$.

In order to simulate this randomized circuit, we first simulated the deterministic circuit, then apply $H^{\otimes k}$ to the registers corresponding to the randomized bits r. See Figure 3.9.

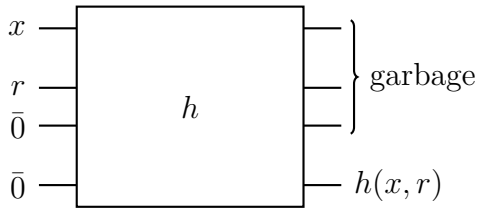

Figure 3.8: randomized classical computation.

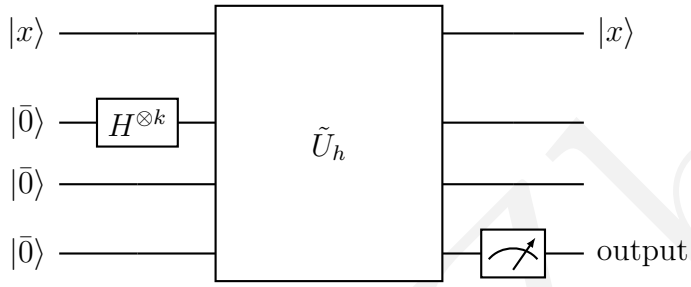

Figure 3.9: quantum simulation of randomized classical computation.

Observe that

$$|\psi\rangle := H^{\otimes k} \left|0^k\right\rangle = \frac{1}{\sqrt{2^k}} \sum_{r\in\{0,1\}^k} |r\rangle .$$

Then before the measurement, the output state is

$$\begin{aligned} |\varphi\rangle &:= \tilde{U}_h |x\rangle |\psi\rangle |\bar{0}\rangle \\ &= \frac{1}{\sqrt{2^k}} \sum_r |x\rangle |r\rangle |g(x,r)\rangle |h(x,r)\rangle . \end{aligned}$$

Upon measuring the output registers,

$$P(\text{output} = y) = P(h(x,R) = y).$$

Thus if the randomized circuit computes $f : \{0,1\}^n \to \{0,1\}^m$, so does the quantum simulation. Moreover, if the randomized circuit is efficient, so is the quantum simulation.

Theorem 3.5.2
The complexity classes $\mathcal{BPP}, \mathcal{BQP}$ satisfy

$$\mathcal{BPP} \subseteq \mathcal{BQP}.$$

Chapter 4

Quantum Algorithms

4.1 Early Quantum Algorithms

Let us warm-up with the black-box or query model of computation.

Definition 4.1.1 (Black-Box/Oracle)
Given a function $f : D \to \{0,1\}$ for some domain D and a circuit that computes it. We call the circuit a black-box/oracle for f.

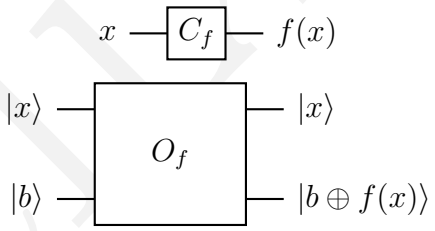

Figure 4.1: classical and quantum oracles.

Example 4.1.1
Given a boolean formula φ over n variables x in CNF, an assignment $a \in \{0,1\}^n$ of truth values to x satisfies φ if $\varphi(a) = 1$.

Then a classical circuit C_φ or a quantum circuit O_φ for checking if a given assignment a satisfies φ is an oracle with domain $D := \{0,1\}^n$ and function is φ.

Given the circuit C_φ, how many assignments do we need to evaluate to determine if it is satisfiable?

The number of queries is the *query complexity*.

4.2 The Deutsch-Jozsa Problem

We say a function $f : \{0,1\}^n \to \{0,1\}$ is *constant* if $f(x) = 0$ for all x or $f(x) = 1$ for all x.

On the other hand, f is *balanced* if $f(x) = 0$ for exactly half of the domain and $f(x) = 1$ for the other half of the domain.

Problem 1 (Deutsch-Jozsa)
Given an oracle $f : \{0,1\}^n \to \{0,1\}$ where f is either constant or balanced, decide if f is constant or balanced with as few queries as possible.

For deterministic algorithms, we need at least $2^{n-1} + 1$ queries in the worst case.

Theorem 4.2.1 (Deutsch-Jozsa)
There is a quantum algorithm DJ making 2 queries which gives the correct output with probability 1.

Note that in fact only one query suffices. See Figure 4.2 for a circuit.

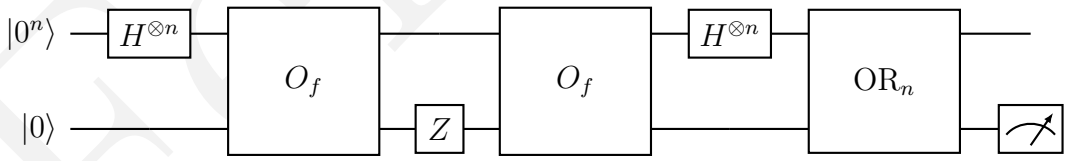

Figure 4.2: The Deutsch-Jozsa algorithm.

For n-bit strings x, y

$$x \cdot y := \oplus_{i=1}^{n} x_i y_i.$$

Given an arbitrary $x \in \mathbb{Z}_2^n$,

$$\begin{aligned} H^{\otimes n} |x\rangle &= \otimes_{i=1}^{n} \frac{1}{\sqrt{2}} (|0\rangle + (-1)^{x_i} |1\rangle) \\ &= \otimes_{i=1}^{n} \frac{1}{\sqrt{2}} ((-1)^{0 \cdot x_i} |0\rangle + (-1)^{1 \cdot x_i} |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} \otimes_{i=1}^{n} \sum_{y_i \in \{0,1\}} (-1)^{y_i \cdot x_i} |y_i\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} |y\rangle . \end{aligned}$$

Proposition 4.2.2
For n-bit strings x, y,

$$\sum_x (-1)^{x \cdot y} = \begin{cases} 2^n, & y = 0^n \\ 0, & y \neq 0^n \end{cases}$$

Proof
Observe that

$$\begin{aligned} \frac{1}{2^n} \sum_x (-1)^{x \cdot y} &= \langle 0^n | H^{\otimes n} \cdot H^{\otimes n} |y\rangle \\ &= \langle 0^n | y \rangle \\ &= \begin{cases} 1, & y = 0^n \\ 0, & y \neq 0^n \end{cases} \end{aligned}$$

Let $|\psi_i\rangle$ be the state of the system after the i-th gate.

$$
\begin{aligned}
|\psi_1\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} |x\rangle\,|0\rangle \\
|\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} |x\rangle\,|f(x)\rangle \\
|\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle \\
|\psi_5\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)}\,|x\rangle\,|0\rangle \\
|\psi_5\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)} H^{\otimes n}\,|x\rangle\,|0\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{y\in\mathbb{Z}_2^n} (-1)^{x\cdot y}\,|y\rangle\,|0\rangle \\
|\psi_6\rangle &= \frac{1}{2^n} \sum_y \left(\sum_x (-1)^{f(x)\oplus x\cdot y} \right) |y\rangle\,|\mathrm{OR}_n(y)\rangle
\end{aligned}
$$

Case I(a): $f(x) = 0$ for all x

$$
\begin{aligned}
|\psi_6\rangle &= \frac{1}{2^n} \sum_y \left(\sum_x (-1)^{x\cdot y} \right) |y\rangle\,|\mathrm{OR}_n(y)\rangle \\
&= |0^n\rangle\,|\mathrm{OR}_n(0^n)\rangle \\
&= |0^n\rangle\,|0\rangle\,.
\end{aligned}
$$

Case I(b): $f(x) = 1$ for all x

$$
\begin{aligned}
|\psi_6\rangle &= -\frac{1}{2^n} \sum_y \left(\sum_x (-1)^{x\cdot y} \right) |y\rangle\,|\mathrm{OR}_n(y)\rangle \\
&= -\,|0^n\rangle\,|\mathrm{OR}_n(0^n)\rangle \\
&= -\,|0^n\rangle\,|0\rangle\,.
\end{aligned}
$$

Case II: $f(x)$ is balanced

$$\begin{aligned}|\psi_6\rangle &= \frac{1}{2^n}\sum_y \left(\sum_x (-1)^{f(x)\oplus x\cdot y}\right)|y\rangle\,|\mathrm{OR}_n(y)\rangle \\ &= 0\cdot|0^n\rangle\,|0\rangle + \frac{1}{2^n}\sum_{y\neq 0^n}\left(\sum_x (-1)^{f(x)\oplus x\cdot y}\right)|y\rangle\,|\mathrm{OR}_n(y)\rangle \\ &= \frac{1}{2^n}\sum_{y\neq 0^n}\left(\sum_x (-1)^{f(x)\oplus x\cdot y}\right)|y\rangle\,|1\rangle\end{aligned}$$

Thus on the input $|0^n\rangle$, the last register will be in state $|0\rangle$ if and only if f is constant. It is $|1\rangle$ otherwise.

Hence with 2 queries to the oracle, we decide with a circuit with $O(n)$ gates and $O(n)$ qubits whether f is constant or balanced. This is an exponential speed-up.

4.2.1 Randomized Classical Algorithm

1) Pick x, y uniformly at random from $\{0,1\}^n$.
2) Output 0 if $f(x) = f(y)$ and 1 otherwise.

If the function is constant, this succeeds with probability 1. If it is balanced, the probability of success if $\frac{1}{2}$. By repeating this algorithm t times, we can boost the worst case probability to $1 - \frac{1}{2^t}$.

Hence by making $2t$ queries, the randomized algorithm solves the Deutsch-Jozsa problem with one-sided error at most $1 - \frac{1}{2^t}$.

Hence quantum algorithms do not have any asymptotic advantage in term of query complexity when compared to randomized algorithms allowing for small constant error.

4.3 The Simon Problem

Recall that $\mathbb{Z}_2^n$ is an n-dimensional vector space over the finite field $\mathbb{Z}_2$.

Suppose $f : \{0,1\}^n \to \{0,1\}^n$ is such that there exists $s \in \mathbb{Z}_2^n$ for which

$$\forall x, y \in \mathbb{Z}_2^n, f(x) = f(y) \iff y = x \oplus s.$$

We say that f *hides* the string s.

Problem 2 (Simon)
Given an oracle for the function $f : \{0,1\}^n \to \{0,1\}^n$ that hides a non-zero string s, output s.

Consider a classical and quantum oracle as in Figure 4.3.

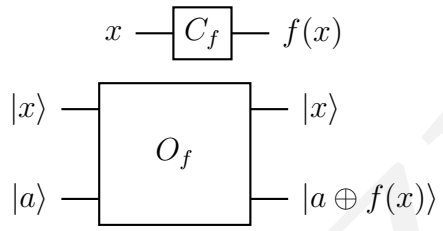

Figure 4.3: classical and quantum oracles.

How many queries do we need to find s? What about the size of the circuit needed?

4.3.1 A Classical Algorithm

The idea is based on the birthday paradox. We wish to find two distinct elements x, y such that $f(x) = f(y)$. Then $s = x \oplus y$.

There are 2^{n-1} pairs $\{x, x \oplus s\}$. If we pick t elements $x_1, \ldots, x_t$ uniformly random from $\mathbb{Z}_2^n$ for $t := 2\sqrt{2^n}$, then the probability that the $f(x_i)$'s are all distinct are at most $\frac{1}{4}$. However, it is possible that $x_i = x_j$ for some $i \neq j$.

Given $f(x_i) = f(x_j)$ we have with probability $\frac{1}{2}$ that $x_i \neq x_j$. Hence we find s with probability at least $\frac{1}{2} \cdot \frac{3}{4} = \frac{3}{8}$.

Theorem 4.3.1
There is a randomized algorithm that finds s with probability at least $\frac{2}{3}$, the string s that the function f hides with $O\left(2^{\frac{n}{2}}\right)$ queries to the oracle for f.

Theorem 4.3.2
Any randomized algorithm that solvces the Simon problem with success probability at least $\frac{2}{3}$ makes $\Omega\left(2^{\frac{n}{2}}\right)$ queries.

4.3.2 The Simon Algorithm

The heart of the algorithm is the hehaviour of the Hadamard operator on the uniform superpositions over a subspace of $\mathbb{Z}_2^n$. Fix $0 \neq s \in \mathbb{Z}_2^n$, then $S := \{0^n, s\}$ is a subspace of $\mathbb{Z}_2^n$.

Indeed,

$$\begin{aligned} H^{\otimes n}\frac{1}{\sqrt{2}}(|0^n\rangle + |s\rangle) &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^n}}\sum_y |y\rangle + \frac{1}{\sqrt{2^n}}\sum_y (-1)^{s\cdot y}|y\rangle\right) \\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_y (1+(-1)^{s\cdot y})|y\rangle \end{aligned}$$

If $s \cdot y = 0$, then the amplitdue is $\frac{1}{\sqrt{2^{n-1}}}$. Otherwise, it is 0.

Recall that

$$S^\perp := \{x \in \mathbb{Z}_2^n : x \cdot s = 0\}$$

is a subspace of $\mathbb{Z}_2^n$ with dimension $n-1$. Hence $H^{\otimes n}\frac{1}{\sqrt{2}}(|0^n\rangle + |s\rangle)$ is an equal superposition of the orthogonal complement of S.

More generally, consider any coset of S, namely $x + S := \{x, x \oplus s\}$. Then

$$\begin{aligned} H^{\otimes n}\frac{1}{\sqrt{2}}(|x\rangle + |x\oplus s\rangle) &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^n}}\sum_y (-1)^{x\cdot y}|y\rangle + \frac{1}{\sqrt{2^n}}(-1)^{(x\oplus s)\cdot y}\right) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^n}}\sum_y (-1)^{x\cdot y}|y\rangle + \frac{1}{\sqrt{2^n}}(-1)^{(x\cdot y)\oplus(s\cdot y)}\right) \\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_y (-1)^{x\cdot y}(1+(-1)^{s\cdot y})|y\rangle \\ &= \frac{1}{\sqrt{2^{n-1}}}\sum_{y\in S\perp} (-1)^{x\cdot y}|y\rangle\,. \end{aligned}$$

The idea now is to repeatedly sample vectors from $S^\perp$ until we have a basis. We are then able to recover s.

Proposition 4.3.3
Let $Y_1, Y_2, \ldots, Y_t$ be iid random variables uniform over the subspace $S^\perp$. Then the probability that the Y_i's span $S^\perp$ when $t = 8n$ is at least $\frac{3}{4}$.

Proof
Suppose we have sampled $Y_1, \ldots, Y_m$ and $V_m := \text{span}\{Y_1, \ldots, Y_m\}$. If $V_m = S^\perp$, we are done. Suppose otherwise.

We argue that

$$P(Y_{m+1} \notin V_m) \geq \frac{1}{2}.$$

Then with probability at least $\frac{1}{2}$,

$$\dim V_{m+1} = \dim V_m + 1.$$

In particular, the expected number of samples needed for the dimension of the samples to increase by 1 is at most 2. Thus the expected number of samples needed for the dimension to attain $n-1$ is at most $2(n-1)$. Let T be the random variable equal to the number of samples needed to obtain dimension $n-1$. By Markov's inequality,

$$\begin{aligned} P(T \geq 8n) &\leq \frac{E[T]}{8n} \\ &\leq \frac{2(n-1)}{8n} \\ &\leq \frac{1}{4}. \end{aligned}$$

In other words,

$$P(T \leq 8n) \geq \frac{3}{4}.$$

To see the claim, we know that $Y_{m+1} \in S^\perp$. If $Y_{m+1} \notin V_m$, then $Y_{m+1} \in S^\perp \setminus V_m$. We know that there is some $y \in S^\perp \setminus V_m$. For each $z \in V^m$, $y \oplus z \in S^\perp$.

However $y \oplus z \notin V_m$, or else

$$y = (y \oplus z) \oplus z \in V_m$$

which is absurd. Moreover, if $z_1 \neq z_2 \in V_m$, then $z_1 \oplus y \neq z_2 \oplus y$. Hence translation by y is an injective function $V_m \to S^\perp \setminus V_m$.

It follows that $|V_m| \leq |S^\perp \setminus V_m|$ and that we pick an element outside of V_m with probability at least $\frac{1}{2}$.

Thus using $8n$ samples from $S^\perp$, we can find a basis for $S^\perp$ with probability at least $\frac{3}{4}$. Say $y_1, \ldots, y_{n-1} \in \mathbb{Z}_2^n$ spans $S^\perp$. Then we can find the hidden substring s by solving the linear equations

$$y_i \cdot s = 0, \forall i \in [n-1].$$

A key subroutine of Simon's algorithm is SAMPLE(f) illustrated in Figure 4.4. The random

variable Y is uniform over $S^\perp$.

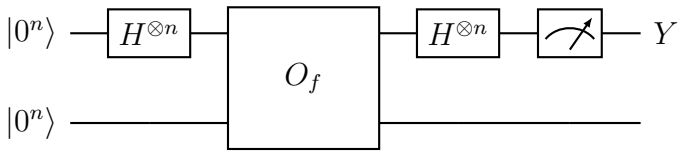

Figure 4.4: The SAMPLE(f) subroutine.

Let $|\psi_i\rangle$ be the state of the system after the i-th gate. Put $F := f(\mathbb{Z}_2^n)$ as the image of f.

$$\begin{aligned}
|\psi_1\rangle &= \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0^n\rangle \\
|\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle \\
&= \frac{1}{\sqrt{2^{n-1}}} \sum_{f(x)\in F} \frac{1}{\sqrt{2}} (|x\rangle + |x \oplus s\rangle) |f(x)\rangle \\
|\psi_3\rangle &= \frac{1}{\sqrt{2^{n-1}}} \sum_{f(x)\in F} \frac{1}{\sqrt{2^{n-1}}} \sum_{y\in S^\perp} (-1)^{x\cdot y} |y\rangle |f(x)\rangle .
\end{aligned}$$

After measurement, we see some $y \in S^\perp$ with probability

$$\begin{aligned}
\frac{1}{2^{2(n-1)}} \sum_{f(x)\in F} |(-1)^{x\cdot y}|^2 &= \frac{2^{n-1}}{2^{2(n-1)}} \\
&= \frac{1}{2^{n-1}}.
\end{aligned}$$

‘

The Algorithm

1) Run SAMPLE(f) for $t := 8n$ times using fresh qubits initialized to $|0^{n+1}\rangle$ for each run
2) Let $Y_i, i \in [t]$ be the t measurement outcomes obtained. Solve the system of linear equations $\{Y_i \cdot w = 0 : i \in [t]\}$ over the variable $w \in \mathbb{Z}_2^n$ (using Gaussian elimination). This is a basis for the solution space.
3) If the solution space has dimension more than 1, output “fail”. Otherwise, output $\tilde{s}$, the unique non-zero element.

Note that Gaussian elimination over $\mathbb{Z}_2$ takes $O(n^3)$ time and $O(n^2)$ space.

Theorem 4.3.4
Simon's algorithm has query complexity $O(n)$, uses $O(n^2)$ qubits, and $O(n^2)$ quantum gates. It has classical post-processing using $O(n^2)$ bits of memory and time complexity $O(n^3)$. Moreover, it produces an output $\tilde{s}$ with probability at least $\frac{3}{4}$, and

$$P(\tilde{s} = s : \text{not fail}) = 1.$$

Proof
The complexity of the algorithm is clear.

Remarks

We can combine all the steps of Simon's algorithm into one quantum circuit. The simulation of classical computation is already clear. Thus the difficulty lies in the composition of quantum circuits. Observe the circuit identity Figure 4.5.

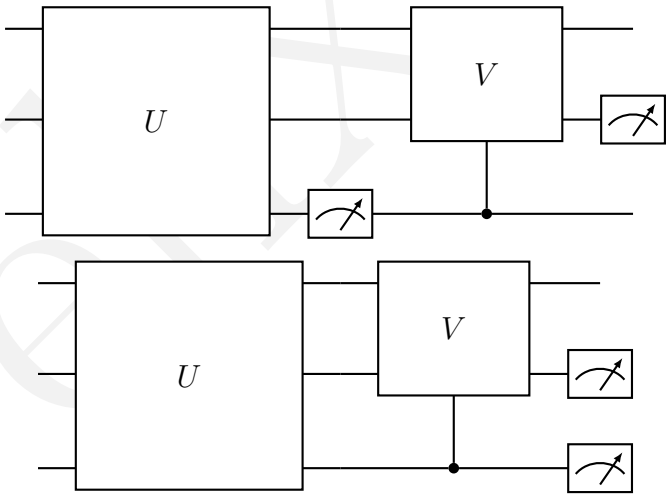

Figure 4.5: The composition of two SAMPLE(f) subroutines.

The identity holds as the two sequences of operations have the same distribution over final states. But it is easier to conceptually understand algorithms as subroutines, hence we may similarly design other algorihtms by composing simpler quantum circuits.

4.3.3 Extensions

The Simon problem is a special instance of the Hidden Subgroup Problem (HSP). Simon's algorithm extends to efficient quantum algorithms for HSP over abelian groups.

In fact, integer factorization and discrete logarithm, two problems underlying public key cryptography, can both be reduced to abelian HSP. The famous algoirhtms for these probelms due to Shor were inspried by the Simon algorithm.

4.4 Phase Estimation

Let $\varphi := 2\pi\theta$

Recall the beam-splitter experiment which applies the gates

$$HR(\varphi)H$$

where $R(\varphi)$ is the phase gate of corresponding angle. Then we observe the light at detector D_1 with probability

$$\sin^2 \frac{\varphi}{2}.$$

If $\varphi << \frac{\pi}{2}$, then $\sin^2 \frac{\varphi}{2} \approx \frac{\varphi^2}{4}$. Hence by repeating the experiment, we can approximate φ.

Consider the binary representation of θ

$$\theta = 0.\theta_1\theta_3\ldots.$$

Then

$$\begin{aligned} 2^m\varphi &= 2\pi 2^m\theta \\ &= 2\pi\left(2^{m-1}\theta_1 + 2^{m-2}\theta_2 + \cdots + \theta_m + \frac{1}{2}\theta_{m+1} + \frac{1}{2^2}\theta_{m+2} + \ldots\right) \\ &\equiv 2\pi\left(\frac{1}{2}\theta_{m+1} + \frac{1}{2^2}\theta_{m+2} + \ldots\right). \end{aligned}$$

The idea is to then learn each bit in succession.

Theorem 4.4.1 (Kitaev)
Given a circuit for applying U^{2^k} where

$$U := \begin{pmatrix} 1 & 0 \\ 0 & 2^{2\pi i\theta} \end{pmatrix}, k \in [m],$$

we can learn θ up to m bits of precision using $O(m)$ qubits and $O(m)$ other gates and measurements.

4.4.1 Efficient Phase Estimation

Let U denote hte phase gate

$$U := \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i\theta} \end{pmatrix}.$$

Recall that Kitaev's algorithm applies the powers U^{2^k} for $0 \leq k < m$ and

$$U^{2^k} H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(2^k\theta)}).$$

Now, let $|\psi_\theta\rangle$ be the tensor product of the above for $0 \leq k < m$.

$$\begin{aligned}
|\psi_\theta\rangle &:= \frac{1}{\sqrt{2^m}} \otimes_{k=m-1}^{0} (|0\rangle + e^{2\pi i(2^k\theta)} |1\rangle) \\
&= \frac{1}{\sqrt{2^m}} \otimes_{k=m-1}^{0} (e^{2\pi i(2^k\theta)\cdot 0} |0\rangle + e^{2\pi i(2^k\theta)\cdot 1} |1\rangle) \\
&= \frac{1}{\sqrt{2^m}} \otimes_{k=m-1}^{0} \sum_{y_k \in \{0,1\}} e^{2\pi i(2^k\theta)\cdot y_k} |y_k\rangle \\
&= \frac{1}{\sqrt{2^m}} \sum_{y \in \{0,1\}^m} e^{2\pi i\theta\left(\sum_{k=m-1}^{0} 2^k y_k\right)} |y\rangle \\
&= \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} e^{2\pi i\theta y} |y\rangle .
\end{aligned}$$

Suppose $\theta = \frac{a}{2^m}$ for some $a \in \mathbb{Z}_{2^m}$. Then

$$e^{2\pi i y \frac{a}{2^m}} = \omega^{ay}$$

for the 2^m-th root of unity $\omega := \exp\left(\frac{2\pi i}{2^m}\right)$. Define

$$|\chi_x\rangle := \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} \omega^{xy} |y\rangle .$$

Then from complex analysis,

$$\begin{aligned}\langle \chi_a | \chi_b \rangle &= \frac{1}{2^m} \sum_{y=0}^{2^m-1} \omega^{(b-a)y} \\ &= \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}\end{aligned}$$

Definition 4.4.1 (Fourier Basis)
The Fourier basis is the set

$$\{|\chi_x\rangle : 0 \leq x < 2^m\}.$$

We have just shown that the Fourier basis is an orthonormal basis of $\mathbb{C}^{2^m}$. Hence if $\theta = \frac{a}{2^m}$ for some a, we need only measure in the Fourier basis, and we determine θ exactly.

Definition 4.4.2 (Quantum Fourier Transform)
The quantum fourier transform (over $\mathbb{Z}_{2^m}$) is the change of basis operator

$$F_{2^m} := \sum_{x=0}^{2^m-1} |\chi_x\rangle\langle x| \,.$$

In order to implement this measurement, we need an efficient implementation of the quantum fourier transform.

What if θ is NOT an integer multiple of $\frac{1}{2^m}$? Let a be such that $\frac{a}{2^m}$ is the closest integer multiple of $\frac{1}{2^m}$ to θ. If we measure $|\psi_\theta\rangle$ in the Fourier basis,

$$\begin{aligned} P(\text{outcome} = a) &= |\langle \chi_a | \psi_\theta \rangle|^2 \\ &= \frac{1}{2^{2m}} \left| \sum_{y=0}^{2^m-1} \exp\left(-\frac{a}{2^m} + \theta\right) y \right|^2 \\ &= \frac{1}{2^{2m}} \left| \frac{\exp\left(\theta - \frac{a}{2^m}\right) \cdot 2^m - 1}{\exp\left(2\pi i(\theta - \frac{a}{2^m})\right) - 1} \right|^2 && \text{geometric sum} \\ &= \frac{1}{2^{2m}} \frac{\sin^2\left(2^m \pi \left(\theta - \frac{a}{2^m}\right)\right)}{\sin^2\left(\pi(\theta - \frac{a}{2^m})\right)} \\ &=: \frac{1}{2^{2m}} \frac{\sin^2 \alpha}{\sin^2\left(\frac{\alpha}{2^m}\right)}. \end{aligned}$$

Lemma 4.4.2
For $\varphi \in \left[0, \frac{\pi}{2}\right]$,

$$\frac{\varphi}{2} \leq \sin \varphi \leq \varphi.$$

Since

$$\left|\frac{a}{2^m} - \theta\right| \leq \frac{1}{2} \cdot \frac{1}{2^m},$$

the absolute value of the angle α in the numerator is at most

$$2^m \pi \frac{1}{2^{m+1}} \leq \frac{\pi}{2}.$$

The absolute value of the angle $\frac{\alpha}{2^m}$ is positive and at most

$$\frac{\pi}{2^{m+1}}.$$

By our work above,

$$\begin{aligned} P(\text{outcome} = a) =: \frac{1}{2^{2m}} \frac{\sin^2 \alpha}{\sin^2\left(\frac{\alpha}{2^m}\right)} \\ \geq \frac{1}{2^{2m}} \frac{\left(\frac{\alpha}{2}\right)^2}{\left(\frac{\alpha}{2^m}\right)^2} \\ = \frac{1}{4}. \end{aligned}$$

Thus with probability at least $\frac{1}{4}$, the measurement outcome a is such thaat $\frac{a}{2^m}$ gives us an approximation of θ with error at most $\frac{1}{2^{m+1}}$.

Theorem 4.4.3 (Cleve, Ekert, Macchiavello, Mosca)
Let

$$U := \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i \theta) \end{pmatrix}.$$

With probability at least $\frac{1}{4}$, the algorithm described above produces outcome $a \in \mathbb{Z}_{2^m}$ such that

$$\left|\theta - \frac{a}{2^m}\right| \leq \frac{1}{2^{m+1}}.$$

We can boost the probability by using more qubits and taking the most popular output among repetitions.

The memory size is m qubits. The time complexity is $O(m)$ gates, the size of the circuit for F_{2^m}, the quantum fourier transform.

4.5 Quantum Fourier Transform

Recall the state used for phase estimation. Let

$$U := \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i\theta) \end{pmatrix}.$$

Then $(\otimes_{k=m-1}^{0} U^{2^k}) H^{\otimes n}$ maps

$$|0^m\rangle \mapsto |\psi_\theta\rangle = \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} e^{2\pi i\theta y} |y\rangle .$$

If we take $\theta := \frac{x}{2^m}$,

$$|0^m\rangle \mapsto |\chi_x\rangle .$$

To see how we implement U^{2^k},

$$\begin{aligned} U^{2^k} &:= \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i\theta) \end{pmatrix}^{2^k} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & \exp\big(2\pi i(2^k\theta)\big) \end{pmatrix}. \end{aligned}$$

Now,

$$2\pi(2^k\theta) = 2\pi(0.\theta_{k+1}\theta_{k+2}\dots) \mod 2\pi.$$

Hence

$$2\pi(\theta) = 2\pi\left(\frac{x}{2^m}\right) = 0.x_{m-1-k}x_{m-2-k}\dots x_1x_0 \mod 2\pi.$$

We can write

$$e^{2\pi i(2^k\theta)} = \prod_{j=0}^{m-1} e^{2\pi i \frac{x_{m-1-k-j}}{2^{j+1}}}.$$

Let $\alpha_j := 0.\underbrace{00\dots0}_{j-1}1$ and

$$R_j := \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i\alpha_j) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^j}\right) \end{pmatrix}.$$

Then

$$U^{2^k} = \prod_{j=1}^{m-k} R_j^{x_{m-j-k}}.$$

For the special case of R_1,

$$R_1^{x_j} = Z^{x_j}$$

and

$$R_1^{x_j} H |0\rangle = Z^{x_j} |+\rangle = H |x_j\rangle .$$

So we simply apply the H gate.

We can apply the c-R_j gate controlled by the $(m-j-k)$-th bit. The order is such that we apply $U^{2^0} = \prod_{j=2}^{m-j} R_j^{x_{m-j}} H$ to $|x_m\rangle$ first, as it is conditioned on all other qubits. Then in decreasing order of qubits, we can apply U^{2^k} with increasing powers. This ensures we are "done" with all qubits as controls before applying a non-identity operator.

All in all, we have a circuit sending

$$|x\rangle \mapsto |\chi_x\rangle .$$

Theorem 4.5.1
There is a quantum circuit for F_{2^m} over $\mathbb{Z}_{2^m}$ that uses m qubits, has no measurements, and has $O(m^2)$ single or two-qubit gates.

4.6 Eigenvalue Estimation

Problem 3 (Eigenvalue Estimation)
Given a quantum state $|\psi\rangle \in \mathbb{C}^d$ which is an eigenvector of V with eigenvalue $e^{2\pi i\theta}$ for $\theta \in [0,1)$, and accuracy parameter $\epsilon \in \left[\frac{1}{2^m}, 1\right)$, output an estimate $\tilde{\theta} \in [0,1)$ such that $|\tilde{\theta} - \theta| \leq \epsilon$.

The idea is to use phase estimation. It suffices to implement

$$U^{2^k} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i (2^k \theta)\right) \end{pmatrix} .$$

Observe that

$$\begin{aligned} (\text{c-V}^{2^k} H)(|0\rangle |\psi\rangle) &= \frac{1}{\sqrt{2}} \left(|0\rangle |\psi\rangle + |1\rangle V^{2^k} |\psi\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 2^k \theta} \right) |\psi\rangle . \end{aligned}$$

We can simply this further Figure 4.6.

$$\begin{aligned}
|\xi_1\rangle &:= H^{\otimes m} |0^m\rangle \otimes |\psi\rangle \\
&= \frac{1}{\sqrt{2^m}} \sum_{x \in \{0,1\}^m} |x_{m-1} \dots x_0\rangle |\psi\rangle \\
|\xi_2\rangle &:= \prod_{j=1}^{m} \text{c-V}^{2^{\text{m-j}} E_j E_\psi} |\xi_1\rangle \\
&= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle \otimes V^x |\psi\rangle .
\end{aligned}$$

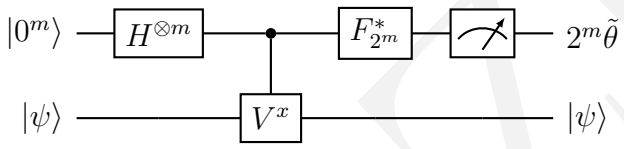

Figure 4.6: The simplified eigenvalue estimation circuit.

Theorem 4.6.1
Let $m := \lfloor \log \frac{1}{\epsilon} \rfloor$. With probability at least $\frac{1}{4}$, the circuit depicted in Figure 4.6 produces an estimate $\tilde{\theta} \in [0, 1)$ such that $|\theta - \tilde{\theta}| \leq \epsilon$, when $|\psi\rangle$ is an eigenvector of V with eigenvalue $e^{2\pi i \theta}$.

4.7 Period Finding

For $n > 1$, let $\mathbb{Z}_n^*$ denote its multiplicative group

$$\mathbb{Z}_n^* := \{a \in \mathbb{Z}_n : \gcd(a, n) = 1\}.$$

Problem 4 (Period Finding)
Given an integer $n > 1$ and $a \in \mathbb{Z}_n^*$, output the order of $a \mod n$.

The brute force algorithm is exponential with respect to the input n.

Using an algorithm for this problem as a subroutine, we can factor the integer n efficiently using a classical algorithm. No efficient classical algorithm for this problem is known.

4.7.1 Main Idea

Multiplication modulo n is in general not invertible, but since $\gcd(a, n) = 1$, multiplication by a modulo 10 is invertible!

Lemma 4.7.1
Le $1 < n \in \mathbb{Z}$ and $\gcd(a, n) = 1$. The function $f_a : \mathbb{Z}_n \to \mathbb{Z}_n$ given by

$$f_a(x) := ax \mod n$$

is invertible.

Consider the unitary operator V_a on $\mathbb{C}^n$ given by

$$|x\rangle \mapsto |ax \mod n\rangle .$$

Thus it permutes the basis elements via multiplication modulo n.

Let r denote the order $|a|$ of a modulo n, and $\omega := e^{-\frac{2\pi i}{r}}$. Define the quantum state

$$|\psi_1\rangle := \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \omega^j \left|a^j \mod n\right\rangle .$$

We have

$$\begin{aligned} V_a |\psi_1\rangle &= \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \omega^j \left|a^{j+1} \mod n\right\rangle \\ &= \frac{\omega^{-1}}{\sqrt{r}} \sum_{j=0}^{r-1} \omega^{j+1} \left|a^{j+1} \mod n\right\rangle \\ &= \omega^{-1} |\psi_1\rangle \\ &= e^{\frac{2\pi i}{r}} |\psi_1\rangle . \end{aligned}$$

Thus we can use the eigenvalue estimation algorithm to determine r! The precision needed is an error bound of $< \frac{1}{2n}$ since $1 \leq r \leq n$. Thus given a register initialized to $|\psi_1\rangle$, we can efficiently compute r, provided we can implement

$$\text{c-V}_\text{a}^{2^\text{k}}, 0 \leq k \leq m$$

efficiently, where $m := O(\log n)$.

4.7.2 Controlled Phase Gates

The brute force implementation of V^{2^k} takes 2^k operations and is exponential in the input length of n. The more efficient method is the repeated squaring method.

Let T be the classical operator which sends

$$a \mapsto a^2.$$

Then we can compute a^{2^k} as

$$T^k a.$$

Note that we need at most $O(\log n)$ applications for this method. Since this can be done classically, we can also simulate this with a quantum operator.

4.7.3 Preparing the State

Consider the related states

$$|\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \omega^{kj} \left| a^j \mod n \right\rangle, 0 \leq k \leq r-1.$$

The state $|\psi_k\rangle$ is also an eigenvector of V_a, with eigenvalue ω^{-k}.

Recall that

$$\sum_{k=0}^{r-1} \omega^{kj} = \begin{cases} r, & j = 0 \\ 0, & j \neq 0 \end{cases}$$

We have

$$\begin{aligned} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \omega^{kj} \left| a^j \mod n \right\rangle \\ &= \frac{1}{r} \sum_{j=0}^{r-1} \left(\sum_{j=0}^{r-1} \omega^{kj} \right) \left| a^j \mod n \right\rangle \\ &= \frac{1}{r} \cdot r \left| a^0 (\mod n) \right\rangle \\ &= |1\rangle . \end{aligned}$$

Thus $|1\rangle$ is a uniform superposition of the r eigenvectors $|\psi_k\rangle$. If we run the eigenvalue estimation algorithm on $|1\rangle$, we will get an approximation of $\frac{k}{r}$ in the component $|\psi_k\rangle$.

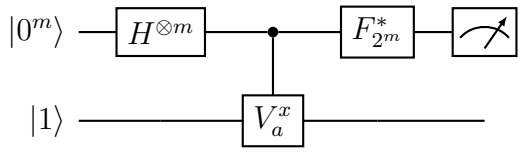

Figure 4.7: The order sampling circuit.

Consider the circuit depicted in Figure 4.7. Put $|\xi\rangle$ as the state of the system before the measurement. We have

$$|\xi\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\xi_k\rangle |\psi_k\rangle .$$

Measuring $|\xi_k\rangle$ gives $2^m \tilde{\theta}_k$, where $\tilde{\theta}_k$ is within $\frac{1}{2^{m+1}}$ of $\frac{k}{r}$ with probability at least $\frac{1}{4}$.

Since the $|\psi_k\rangle$ Are orthonormal, we get outcome $2^m \tilde{\theta}_k$ with probability at least $\frac{1}{4r}$ for each $k = 0, 1, \ldots, r-1$.

Now, we can extract $\frac{k}{r}$ from $\tilde{\theta}_k$ using the method of continued fractions. This involves an efficient classical algorithm, which requires an acccuracy parameter m such that

$$\frac{1}{2^m} \leq \frac{1}{n^2}.$$

Hence $m \in O(\log n)$ suffices.

We can recover r from $\frac{k}{r}$ provided that k, r are coprime, given $\frac{k}{r}$ is expressed as a fraction in the lowest terms.

Proposition 4.7.2
The number of $k \in [r-1]$ that are coprime with r is

$$\Omega\left(\frac{r}{\log\log r}\right).$$

Hence if we repeat the sampling procedure $O(\log\log r) \subseteq O(\log\log n)$ times, we sample $2^m \tilde{\theta}_k$ for k coprime with r at least once. Thus with high probability, we can extract r.

We do not know when we get a useful sample, but we can check that the denominator $\tilde{r}$ that we compute is correct by checking if $a^{\tilde{r}} \equiv 1 \mod n$. Indeed, if k, r share a common factor, then we output a strictly smaller $\tilde{r} < r$ and $a^{\tilde{r}} \not\equiv 1$. Thus we can make the algorithm zero-error.

4.7.4 Period Finding Algorithm

1) Let $m := 2 \log n$ and $t := 25 \log \log n$.
2) For $i \in [t]$:
   (a) Let $2^m \tilde{\theta}$ be the output of our sampling algorithm.
   (b) Compute a fraction $\frac{\tilde{k}}{\tilde{r}}$ in lowest terms such that $\left|\tilde{\theta} - \frac{\tilde{k}}{\tilde{r}}\right| \leq \frac{1}{2n^2}$.
   (c) If $a^{\tilde{r}} \equiv 1 \mod n$, stop and output $\tilde{r}$.
3) If none of the iterations succeed, output "FAIL".

Theorem 4.7.3
There is a polynomial time algorithm which outputs the order of $a \mod n$ with probability at least $\frac{2}{3}$. Moreover, the algorithm does not output incorrect answers and outputs "FAIL" if it fails to find the order.

4.8 Integer Factorization

Problem 5 (Integer Factorization)
Given an integer $n > 1$, output "prime", or $1 < n_1 < n$ such that $n_1 \mid n$.

No polynomial time classical algorithm is known for this problem. The best known classical algorithm runs in time

$$e^{O\left(\log^{\frac{1}{3}} n (\log \log n)^{\frac{2}{3}}\right)}.$$

The computational hardness of this problem is the basis of the RSA public-key encryption scheme that has been used for decades.

4.8.1 Reduction to Period Finding

If $n = p^k$ for some prime p and $k \geq 1$, we can classically verify this efficiently, and factor n.

Suppose now that n has at least two distinct prime factors.

Theorem 4.8.1
If the integer $n > 1$ has at least two distinct factors, at least a constant fraction of numbers $a \in \mathbb{Z}_n^*$ satisfy the following property:
Let $r := |a|$ be the order of $a \mod n$. Then r is even and $a^{\frac{r}{2}} - 1$ has a non-trivial factor in common with n.

To apply this theorem, we repeat the following a constant number of times:

1) Pick $a \in \mathbb{Z}_n$ uniformly at random.
2) If $\gcd(a, n) \neq 1$, we have a non-trivial factor on n.
3) Otherwise, we compute r, the order of $a \mod n$ with the period finding algorithm.
4) If r is even, we compute $b := a^{\frac{r}{2}} - 1$ and $\gcd(b, n)$.
5) If the GCD is not 1, we succeed.

Thus we get a non-trivial factor of n with constant probability.

Theorem 4.8.2 (Shor)
There is an efficient quantum algorithm for integer factorization that computes all the prime factors of a given integer n, in time $\text{poly}(\log n)$, with probability at least $\frac{2}{3}$.

4.9 NP-Hard Problems

Definition 4.9.1 (Non-Deterministic Polynomial Time)
$\mathcal{NP}$ consists of computational problems solvable by non-deterministic Turing machines.
In other words, for which given a potential solution, we can verify if it is a valid solution efficiently.

It seems intuitive that the classical $\mathcal{NP}$-complete problem 3-SAT can be tackled using quantum parallelism. We shall see that quantum effects alone do NOT suffice for this purpose.

4.9.1 Oracle Lower Bound

Let O_φ be an oracle for the boolean formula φ. How many queries does a quantum algorithm need to determine if φ is satisfiable?

Problem 6 (Unordered Search)
Given an oracle for $f : \{0,1\}^n \to \{0,1\}$, output 1 if there is some a such that $f(a) = 1$ and 0 otherwise.

Theorem 4.9.1 (Bennet, Bernstein, Brassard, Vazirani)
Any quantum algorithm for the unordered search problem with oracle access requires $\Omega\left(2^{\frac{n}{2}}\right)$ queries.

Proof
Consider any quantum circuit that uses $m \geq n+1$ qubits of memory, makes t queries to f, and outputs the correct answer with probability at least $\frac{2}{3}$. We show that $t \in \Omega(2^{\frac{n}{2}})$.

We may as well assume that O_f is only applied on the first $n+1$ qubits, by applying a linear number of SWAP gates of the correct dimensions. Moreover, we measure on the last qubit in the standard basis and output the result. Hence the circuit consists of

$$U_t(O_f \otimes I)U_{t-1}(O_f \times I)\dots U_0$$

and a measurement, where each U_i is unitary and acts on the entire system.

We claim that there is an element $y \in \{0,1\}^n$ that is queried with squared amplitude at most $\frac{t}{2^n}$ over the entire course of the algorithm.

Suppose we run the algorithm on $f_0 \equiv 0$. Let $|\psi_i\rangle$ be the state of the memory just before the $(i+1)$-th query for $i = 0, 1, \dots, t-1$, and let $|\psi_t\rangle$ the state before measurement.

Express

$$|\psi_i\rangle = \sum_{x \in \mathbb{Z}_2^n, b \in \mathbb{Z}_2} \alpha_{ixb} |x\rangle |b\rangle |\psi_{ixb}\rangle ,$$

where $|\psi_{ixb}\rangle$ are some quantum states of the remaining qubits and $\sum_{x,b}|\alpha_{ixb}|^2 = 1$ for all $i = 0, \dots, t$.

Summing all squared amplitudes over i,

$$\sum_{i=0}^{t-1}\sum_{x,b}|\alpha_{ixb}|^2 = t$$

$$\sum_{x}\sum_{i=0}^{t-1}\sum_{b}|\alpha_{ixb}|^2 = t.$$

Hence there must be some $y \in \{0,1\}^n$ such that

$$\sum_{i=0}^{t-1} \sum_{b} |\alpha_{iyb}|^2 \leq \frac{t}{2^n}.$$

In other words, the element y is quried with probability at most $\frac{t}{2^n}$, over the course of the algorithm.

Now suppose we run the algorithm with an oracle f_y such that $f(y) = 1$ and $f(x) = 0$ for all $x \neq y$. Let the final state be $|\phi_t\rangle$. We claim that

$$\| |\psi_t\rangle - |\phi_t\rangle \| \leq 2\sqrt{2} \frac{t}{2^{\frac{n}{2}}}.$$

We apply the "Hybrid Argument". Consider $t+1$ runs of the algorithm. In the j-th run, we give the algorithm the oracle for f_0 for the first j queries, and then the oracle for f_y for the remaining $(t-j)$ queries.

Let $|\xi_j\rangle$ be the final state just before measurement on the j-th run. Then

$$\begin{aligned} |\xi_0\rangle &= |\phi_t\rangle \\ |\xi_t\rangle &= |\psi_t\rangle . \end{aligned}$$

Now,

$$\begin{aligned} \| |\phi_t\rangle - |\psi_t\rangle \| &= \| |\xi_0\rangle - |\xi_t\rangle \| \\ &\leq \sum_{j=0}^{t-1} \| |\xi_j\rangle - \|\xi_{j+1}\| \|. \end{aligned}$$

Let us bound $\| |\xi_j\rangle - |\xi_{j+1}\rangle \|$. Now, unitary operators preserve norm difference. Moreover, the circuits between the $j, j+1$-th runs differ only by a single oracle call. Thus

$$\| |\xi_j\rangle - |\xi_{j+1}\rangle \| = \left\| O_{f_y} |\psi_j\rangle - O_{f_0} |\psi_{j+1}\rangle \right\|.$$

Recall that

$$|\psi_j\rangle = \sum_{x,b} \alpha_{jxb} |x\rangle |b\rangle |\psi_{jxb}\rangle .$$

By definition, $O_{f_y} |\psi_j\rangle = |\psi_j\rangle$ and

$$O_{f_y} |\psi_j\rangle = \sum_{b} \alpha_{jyb} |y\rangle |b \oplus 1\rangle |\psi_{jyb}\rangle + \sum_{x \neq y, b} \alpha_{jxb} |x\rangle |b\rangle |\psi_{jxb}\rangle .$$

Thus

$$\begin{aligned}\left\|O_{f_y}|\psi_j\rangle - O_{f_0}|\psi_j\rangle\right\| &\leq \left\|\sum_b \alpha_{jyb}|y\rangle|b\rangle|\psi_{jyb}\rangle + \sum_b \alpha_{jyb}|y\rangle|b\oplus 1\rangle|\psi_{jyb}\rangle\right\| \\ &\leq 2\sum_b |\alpha_{jyb}|.\end{aligned}$$

Substituting this back,

$$\begin{aligned}\||\phi_t\rangle - |\psi_t\rangle\| &\leq \sum_{j=0}^{t-1} \||\xi_j\rangle - |\xi_{j+1}\rangle\| \\ &\leq 2\sum_{j=0}^{t-1}\sum_b |\alpha_{jyb}| \\ &\leq 2\sqrt{2t}\left(\sum_{jb}|\alpha_{jyb}|^2\right)^{\frac{1}{2}} && \text{Cauchy-Schwartz} \\ &\leq 2\sqrt{2t}\sqrt{\frac{t}{2^n}} \\ &= 2\sqrt{2}\frac{t}{2^{\frac{n}{2}}}.\end{aligned}$$

Finally, we claim that $\||\phi_t\rangle - |\psi_t\rangle\| \geq \frac{1}{3}$ (exercise). Intuitively, the algorithm has error at most $\frac{1}{3}$, hence the probability that the measurement outcome is 1 MUST be very different for the two states.

Combining this inequality with the one obtained above,

$$\begin{aligned}\frac{1}{3} &\leq \frac{2\sqrt{2}t}{2^{\frac{n}{2}}} \\ t &\geq \frac{2^{\frac{n}{2}}}{6\sqrt{2}}.\end{aligned}$$

Note that the ame proof gives an $\Omega(\sqrt{m})$ bound if f had a domain of size m. We can also adapt this argument for classical algorithms, and obtain a $\Omega(2^n)$ query lower bound.

This lower bound is actually tight!

4.9.2 Grover's Search Algorithm

Theorem 4.9.2 (Grover)
There is a quantum algorithm for the unordered search problem, that has query complexity $O\left(2^{\frac{n}{2}}\right)$ and $O(n2^{\frac{n}{2}})$ other single/two-qubit gates.
Moreover, the algorithm makes no error when $f \equiv 0$, and makes error at most $\frac{1}{3}$ otherwise.

For functions $f : [m] \to \{0,1\}$ the theorem holds with query complexity $O(\sqrt{m})$. By repeating the argument t ties, we can reduce the error arbitrarily small.

4.9.3 Implications

Unordered search is a component in many classical algorithms.

Problem 7 (Element Distinctness)
Given a list of numbers $x_1, \ldots, x_n \in [n]$, output 1 if they are all distinct and 0 otherwise.

This can be reduced to sorting and comparing adjacent elements.

Theorem 4.9.3
Any classical algorithm for element distinctness requires $\Omega(n \log n)$ comparisons.

There is a straightforward quantum algorithm: Define $f : [n] \times [n] \to \{0,1\}$ as

$$f(i,j) := \begin{cases} 1, & i \neq j, x_i = x_j \\ 0, & \text{else} \end{cases}$$

The function can be computed with one comparison (per evaluation). We run the Grover search algorithm with f as the oracle. We output 1 if there is no collision and 0 otherwise. The number of comparisons is $O(\sqrt{n^2}) = O(n)$.

In fact, we can obtain a more efficient algorithm that uses Grover search in a nested fashion and solves the problem with

$$O\left(n^{\frac{3}{4}} \log n\right)$$

comparisons.

There is an even more sophisticated algorithm for element distinctness using quantum walk that achieves the optimum $O\left(n^{\frac{2}{3}}\right)$ comparisons.

4.10 Grover's Algorithm

We first restrict to the scenario when f takes on value 1 at most at a single element $x \in \mathbb{Z}_2^n$. The intuition is to start with the uniform super position

$$|u\rangle := \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} |x\rangle$$

and rotate it towards the vector $|a\rangle$ at which $f(a) = 1$.

Consider the rotation

$$R_f := I - 2\,|a\rangle\langle a|\,.$$

For $|\psi\rangle = \alpha_a\,|a\rangle + \sum_{x \neq a} \alpha_x\,|x\rangle$, we get

$$R_f\,|\psi\rangle = -\alpha_a\,|a\rangle + \sum_{x \neq a} \alpha_x\,|x\rangle\,.$$

To implement R_f. consider the circuit depicted in Figure 4.8. Note we employ the phase kickback method.

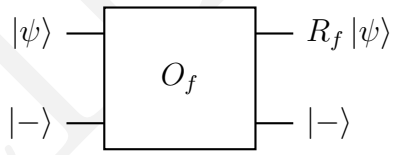

Figure 4.8: The circuit implementing R_f.

Now consider the rotation $R := 2\,|u\rangle\langle u| - I$. R flips the phase of any state perpendicular to $|u\rangle$, but not that of $|u\rangle$.

Recall that $|u\rangle = H^{\otimes n}\,|0^n\rangle$. Hence we may as well implement the rotation reflection about $|0^n\rangle$.

$$\begin{aligned} |0^n\rangle &\mapsto |0^n\rangle \\ |x\rangle &\mapsto |x\rangle \qquad\qquad\qquad\qquad x \neq 0^n \end{aligned}$$

Lemma 4.10.1
There is an efficient quantum circuit that implements the reflection about $|0^n\rangle$ without any queries to f with $O(n)$ single and two-qubit gates.

So we can rotate the whole space by $H^{\otimes n}$, reflect about $|0^n\rangle$, and the rotate the space back. Indeed,

$$\begin{aligned} R &= 2\,|u\rangle\langle u| - I \\ &= 2H^{\otimes n}\,|0^n\rangle\langle 0^n|\,H^{\otimes n} - I \\ &= H^{\otimes n}(2\,|0^n\rangle\langle 0^n| - I)H^{\otimes n}. \end{aligned}$$

Let C denote some circuit implementing $2\,|u\rangle\langle u| - I$. We can implement R as in the circuit depicted in Figure 4.9.

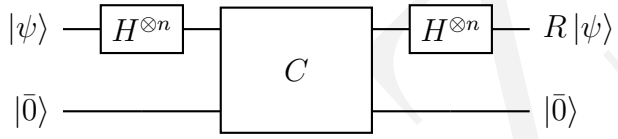

Figure 4.9: The circuit implementing R.

When $f \equiv 0$, both $R_f, R = I$. When f has a 1, the input state $|u\rangle$ is slowly rotated towards $|a\rangle$. We can distinguish the two cases by evaluating f on the query register and measuring the answer register.

4.10.1 The Algorithm

1) Initialize the query register Q, the answer register A, the work register W to all $|\bar{0}\rangle$.
2) Appy $H^{\otimes n}$ to Q to prepare $|u\rangle$.
3) Apply HX to A to prepare $|-\rangle$.
4) Apply $(RR_f)^t$ to registers QA, using W, where $t := \lfloor \frac{\pi}{2\theta} \rfloor$ where $\theta := \arcsin\left(2^{-\frac{n}{2}}\right)$.
5) Apply XH to A to reset it to $|0\rangle$. query f, and measure A to obtain the output.

4.10.2 Complexity

The algorithm uses $O(n)$ qubits, $O(\frac{n}{\theta})$ gates, and makes $O(\frac{1}{\theta})$ queries to the oracle for f. Since

$$\frac{1}{\sqrt{2^n}} = \sin\theta \leq \theta, \frac{1}{\theta} \leq \sqrt{2^n},$$

the complexity is as stated in the theorem.

4.10.3 Correctness

When $f \equiv 0$, the output is always 0. Suppose $f(a) = 1$ for some $a \in \mathbb{Z}_2^n$, and $f(x) = 0$ for $x \neq a$. We analyze the state of the circuit before measurement.

Note that

$$\begin{aligned} |u\rangle &:= \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \\ |v\rangle &:= \frac{1}{\sqrt{2^n - 1}} \sum_{x \neq a} |x\rangle \\ \sin\theta &:= \langle a|u\rangle \\ &= \frac{1}{\sqrt{2^n}}. \end{aligned}$$

Lemma 4.10.2
Let $R_f := I - 2|a\rangle\langle a|$ and $F := 2|0^n\rangle\langle 0^n| - I$. Suppose $V := \text{span}\{|a\rangle, |v\rangle\}$ is the 2-dimensional subspace spanned by $|a\rangle, |v\rangle$. Then RR_f preserves the subspaces $V, V^\perp$, and acts as the rotation

$$\begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

through angle 2θ in the basis $|a\rangle, |v\rangle$ on V, and as $-I$ on $V^\perp$.

The initial state of the query register Q is

$$|u\rangle = \sin\theta a + \cos\theta |v\rangle .$$

After t iterations of RR_f, the state of Q is

$$|u_t\rangle := \sin((2t+1)\theta)|a\rangle + \cos((2t+1)\theta)|v\rangle .$$

So the probability of outputting 1 is $\sin^2((2t+1)\theta)$. Since $t := \lfloor \frac{\pi}{4\theta} \rfloor$,

$$\begin{aligned} \frac{\pi}{2\theta} - 1 \leq 2t + 1 \leq \frac{\pi}{2\theta} + 1 \\ \frac{\pi}{2} - \theta \leq (2t+1)\theta \leq \frac{\pi}{2} + \theta. \end{aligned}$$

So the probability of correctness is

$$\begin{aligned}\sin^2((2t+1)\theta) &\geq \cos^{\theta} \\ &= 1 - \sin^2\theta \\ &= 1 - \frac{1}{2^n}.\end{aligned}$$

Note that this algorithm can be adapted for functions $f : [m] \to \{0,1\}$ and has query complexity $O(\sqrt{m})$.

There is a generalization of this algorithm for when there are multiple 1's. When then are k 1's, the expected query complexity is $O\left(\sqrt{\frac{m}{k}}\right)$. When $f \equiv 0$, it is $O(\sqrt{m})$.

In all versions of the algorithm, we get a uniformly random element of the pre-image $f^{-1}(1)$. The algorithm may be used to speed up a host of classical algorithms by a polynomial in the run-time.

Chapter 5

Error Correction

Quantum states are fragile. Unwanted evolution may occur since perfect isolation is not possible. Moreover, implementations of quantum gates themselves may not be exact, and may introduce the system to external noise.

How can we model noise mathematically and make quantum memory and computation robust in the presence of noise?

5.1 Classical Codes

We can represent 0 as 000, 1 as 111. If there is a bit flip in one of the bits, we can detect this by comparing the value of the 3 bits.

However, if there are two or more bit flips, this redundancy is insufficient. In general, we would like to use as little redundancy as possible to correct as many errors as possible.

For two strings $x, y \in \mathbb{Z}_2^n$,

$$\Delta(x, y) := |\{i \in [n] : x_i \neq y_i\}|.$$

We would like to encode k-bit strings into n-bit strings that are far apart from each other in Hamming distance.

Definition 5.1.1 (Error-Correcting Code)
A $(n, k, d)_2$ error-correcting code is a subset $C \subseteq \mathbb{Z}_2^n$ of size 2^k such that

$$\min\{\Delta(x, y) : x \neq y \in C\} = d.$$

n is the *block-length*, k is the *message length*, and d is the *minimum distance* of C. The subscript 2 indicates that the alphabet is $\{0, 1\}$, which is the default when the subscript is omitted.

Elements of the code are *codewords* and the ratio $\frac{k}{n}$ is the *information-rate* of the code.

Thus the Hamming code $\{000, 111\}$ is a $(3, 1, 3)_2$ code.

Lemma 5.1.1
Suppose we encode a k-bit string x as $C(x)$ using an $(n, k, d)_2$ code C. Suppose the codeword $C(x)$ is subject to t errors, giving us the "received word" $y \in \mathbb{Z}_2^n$. We can recover x provided that

$$t \leq \frac{d-1}{2}.$$

Example 5.1.2 (Hadamard Code)
The code maps $x \in \mathbb{Z}_2^k$ to a 2^k-bit string $C(x)$. The y-th bit in $C(x)$ is $x \cdot y, y \in \mathbb{Z}_2^k$. This yields a $(2^k, k, 2^{k-1})_2$ code.

Suppose we wish to encode k bits into n wiht sufficient redundancy so that we can correct ϵ fraction of errors in the codewords. How large can k be?

Let

$$H(p) := -p \log_2 p - (1-p) \log_2 (1-p)$$

be the binary entropy function.

Theorem 5.1.3
For any error rate $\epsilon \in \left[0, \frac{1}{4}\right)$ and information rate $r < 1 - H(2\epsilon)$, for all n sufficiently large, there are (n, k, d) error correction codes with

$$k := \lfloor rn \rfloor, d \geq 2\epsilon + 1.$$

Thus there are codes that have constant information rate and can withstand a sufficiently small error-rate. Moreover, there are good codes with efficient encodings, error-correction, and decoding procedures.

5.1.1 Binary Symmetric Channel

A simple error model is one in which each bit in a codeword is flipped with probability $p \in \left[0, \frac{1}{2}\right)$, independently.

If a codeword of length n is subjected to such noise, the fraction of errors that occur are at most $p+\nu$ for a small constant ν, with high probability. Hence it suffices to use a code that can handle $p+\nu$ fraction errors.

5.1.2 Linear Error-Correcting Codes

A code is linear if the sum of codewords in $\mathbb{Z}_2^n$ is also a codeword. Both the Hamming and Hadamard codes are linear.

Definition 5.1.2 (Linear Code)
An (n, k, d) code is linear if the encoding function

$$C : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$$

is linear.

We denote such a linear ocde as an $[n, k, d]$ code.

It suffices to give the matrix representing the linear function C to specify the corresponding linear code. This matrix is the generator matrix for C.

Recall the $(3, 1, 3)$ Hamming code. Let $y \in \mathbb{Z}_2^3$ denote the received word and we compute the *error syndrome*

$$y_1 \oplus y_2, y_2 \oplus y_3.$$

We can determine the error (if any) simply by looking at these two bits and correct it.

There is a similar error detection procedure for every linear error-correcting code. It computes a sequence of parities of the bits of the received word, the error syndrome. The location of the errors can be unquely determined from the syndrome, assuming the number of errors is at most $\frac{d-1}{2}$, where d is the minimum distance. The errors can thus be corrected.

5.2 Quantum Error Correction

Extending classical error correction techniques is very non-trivial. This is due to the existence of infinite possibilities for noise and we are unable to clone arbitrary unknown quantum states, unlike classical states.

Another issue is that measuring a quantum state may potentially disturb it, due to the collapse phenomenon. How can we recover from errors when the action is not unitary?

Finally, we also wish for the error correction to be done efficiently.

5.2.1 Bit Flips

Let us start with simple errors, say a unitary analogue of the bit flip: X. We can try to encode a qubit state by repetition in the standard basis. Thus we encode

$$\alpha |0\rangle + \beta |1\rangle \mapsto \alpha |000\rangle + \beta 111.$$

Suppose there is a flip in the first bit. We cannot directly measure the outcome, but we can compute its error syndrome through ancillary qubits:

$$(\alpha |100\rangle + \beta |011\rangle) |00\rangle \mapsto (\alpha |100\rangle + \beta |011\rangle) |10\rangle .$$

Thus we can conditionally apply $X \otimes I$ to correct the error.

5.2.2 Phase Flips

Another type of error is the Z gate:

$$\alpha |0\rangle + \beta |1\rangle \mapsto \alpha |0\rangle - \beta |1\rangle .$$

Recall that

$$HZH = X$$

hence phase flips ARE bit flips in the hadamard basis. We can use repetition in the Hadarmard basis to protect against phase flips.

$$\alpha |0\rangle + \beta |1\rangle \mapsto \alpha |+\rangle |+\rangle |+\rangle + \beta |-\rangle |-\rangle |-\rangle .$$

Computation of the error syndrome and correction is identical.

5.2.3 Shor Code

We can handle XZ errors on one qubit by composing the encoding procedures for Z, X errors:

1) First encode 1 qubit into 3 for an Z error.

2) Then encode each of the 3 qubits into 3 qubits for an X error.

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha\left[\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3}\right] + \beta\left[\frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3}\right]$$

We first correct for X errors and then for Z errors.

Suppose an XZ error occurs in the 1st qubit.

$$\frac{1}{2\sqrt{2}}(|100\rangle - |011\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2}.$$

The error syndrome for qubits 1, 2, 3 is $|10\rangle$ and $|00\rangle$ for the other two triplets. Thus we can correct the X error.

Once the X error is corrected, this leads to the state

$$\frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2}.$$

Reversing the repetition code for bit flips yield

$$\frac{1}{2\sqrt{2}}(|-\rangle|00\rangle) \otimes (|+\rangle|00\rangle)^{\otimes 2}.$$

Follwoing the error correction procedure for a phase flip in qubits 1, 4, 7 allows us to recover

$$\frac{1}{2\sqrt{2}}(|+\rangle|00\rangle) \otimes (|+\rangle|00\rangle)^{\otimes 2}.$$

The 9-qubit Shor code can be summarized as

1) Compute the syndrome for a bit flip for each consecutive triplet of qubits.
2) Apply the error reversal for a bit flip to each triplet, using the corresponding syndrome.
3) Decode the repetition code for bits withint each triplet. Qubits 1, 4, 7 now contain the state encoded for a phase flip error, potentially with an error.
4) Use the error-correction procedure for a phase flip to correct any Z errors.
5) Encode qubits 1, 4, 7 using the repetition code for a bit flip to reverse step 3). This recovers the original 9-qubit codeword.

In fact, this procedure correct an X-error and a Z-error, even if they occur in different qubits.

5.2.4 Unitary Errors

Lemma 5.2.1
The operators I, X, Z, XZ form a basis for the vector space of 2×2 complex matrices.

Proof
Any element of the standard basis $|i\rangle\langle j|$ can be expressed using these operators.

When a 1-qubit unitary error occurs on any of the 9 qubits of the Shor codeword $\left|\hat{\psi}\right\rangle$, the error-correction procedure results in $\left|\hat{\psi}\right\rangle$ in tensor product with the superposition over error syndromes for I, X, Z, XZ at the location of the error.

Thus the Shor code code can be used to correct any 1-qubit unitary error.

5.2.5 General Single-Qubit Errors

Recall a general error on the qubit in state $|\psi\rangle$ and environment in state $|\varphi\rangle$ is a unitary operator U on the qubit and environment. We can write

$$U = \sum_{i,j \in \{0,1\}} |i\rangle\langle j| \otimes U_{i,j}$$

where we split U into 4 submatrices $U_{i,j}$.

Since each of the standard basis can be expressed as a linear combination of the Pauli basis, we can also write

$$U = I \otimes V_0 + X \otimes V_1 + Z \otimes V_2 + XZ \otimes V_3$$

where each V_i is a linear combination of the matrices $U_{i,j}$.

Thus a general single-qubit error also manifests itself as one of X, Z, XZ.

Takeaways

1) Even though there maybe an infinite number of possibilities for noise, we need only correct bit-flip or phase-flip errors.
2) An arbitrary unknown quantum state cannot be cloned, but need only introduce redundancy for bit-flip and phase-flip errors

3) Measuring a quantum state may disturb it, but we can compute and measure the error syndrome, which is independent of the state itself.

4) The action of an error on the state is not necessarily unitary, but we need only reverse X and Z errors, and these are unitary

5) Quantum codes with constant information rate and ability to correct a contant fraction of error exist. However, such codes with efficient encoding, error correction, and decoding operations are not yet known.

5.2.6 Calderbank-Shor-Steane Codes

Suppose we wish to encode a k-qubit state $|\psi\rangle$ as an n-qubit state $\left|\hat{\psi}\right\rangle$. Let us derive sufficient conditions for it to correct t-qubit errors. We need only consider a combination of X, Z errors.

In order to correct X errors, is suffices for $\left|\hat{\psi}\right\rangle$ to be a super position over codewords of a classical code C_1 with length n and minimum distance $d \geq 2t+1$. Say

$$\left|\hat{\psi}\right\rangle := \sum_{x \in C} \alpha_x |x\rangle .$$

Similarly, to be able to handle phase-flip errors, it suffices for $\left|\hat{\psi}\right\rangle$ to be a superposition over a similar code in the Hadamard basis. That is, we would like $H^{\otimes n}\left|\hat{\psi}\right\rangle$ to be a superposition over codewords of an (n,k,d)-classical code C_2. Thus

$$\begin{aligned} H^{\otimes n}\left|\hat{\psi}\right\rangle &= \sum_{x \in C} \alpha_x \frac{1}{\sqrt{2^n}} \sum_y (-1)^{x \cdot y} |y\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_y \left(\sum_{x \in C} (-1)^{x \cdot y} \alpha_x \right) |y\rangle . \end{aligned}$$

If α_x is arbitrary, then $H^{\otimes n}\left|\hat{\psi}\right\rangle$ has no hope of being another code. However, recall that

$$H^{\otimes n}|S\rangle = \left|S^{\perp}\right\rangle$$

and if C is a linear code, then its image is a linear subspace. Hence we would like codes where $C^{\perp}$ is also a linear code!

Lemma 5.2.2
If C is an $[n, k_1, d_1]$ code and $C^\perp$ is an $[n, k_2, d_2]$ code with $d_1, d_2 \geq 2t + 1$, the state

$$\left|\hat{\psi}\right\rangle := \frac{1}{\sqrt{2^{k_1}}} \sum_{x \in C} |x\rangle$$

is a quantum codeword that can correct t errors.

This gives a codeword to encode a single state.

Proposition 5.2.3
Let C_1 be an $[n, k, d_1]$-code and C_2 be an $[n, k_2, d_2]$-code such that $C_1 \subseteq C_2$. Then the cosets of C_1 in C_2 are also (n, k_1, d_2) codes.

Lemma 5.2.4
For two elements $a, b \in C_2$, either
(1) $a \oplus C_1 = b \oplus C_1$, or
(2) $(a \oplus C_1) \cap (b \oplus C_1)$ are disjoint

Proof
The number of cosets of C_1 in C_2 is

$$\frac{|C_2|}{|C_1|} = 2^{k_2 - k_1}.$$

Fix a set of C_1 coset representatives

$$\{u_1, \ldots, u_{2^\ell}\} \subseteq C_2$$

and define

$$\left|\hat{\psi}_i\right\rangle := |u_i \oplus C_1\rangle = \frac{1}{\sqrt{2^{k_1}}} \sum_{c \in C_1} |u_i + c\rangle .$$

These are all orthogonal, as they correspond to distinct cosets. They all correct t bit-flip errors by construction.

Recall that

$$H^{\otimes n} |a \oplus C_1\rangle = \frac{1}{\sqrt{2^{n-k_1}}} \sum_{c \in C_1^\perp} (-1)^{a \cdot c} |c\rangle .$$

Hence if $C_1^\perp$ is also an error-correction code with distance at least $2t + 1$, we can correct t phase flips.

This gives us all the conditions we need to construct codes for multi-qubit states that can handle multi-qubit errors.

Summary

Let C_1 be an $[n, k_1, d_1]$ code, and C_2 be an $[n, k_2, d_2]$ code such that

1) $C_1 \subseteq C_2$
2) $C_1^\perp$ is an $[n, n-k_1, d_3]$ code
3) $k_1 < k_2$ and $d_2, d_3 \geq 2t-1$

Let $\{u_1, \ldots, u_{2^\ell}\}$ be a set of representatives for the cosets of C_1 in C_2, where $\ell := k_2 - k_1$.

define for each $i \in [2^\ell]$, and n-qubit state

$$\left|\hat{\psi}\right\rangle := |u_i \oplus C_1\rangle = \frac{1}{\sqrt{2^{k_1}}} \sum_{c \in C_1} |u_i + c\rangle .$$

Since the $\left|\hat{\psi}\right\rangle$'s are orthogonal, the mapping

$$|i\rangle \mapsto \left|\hat{\psi}_i\right\rangle$$

defines a quantum error-correction code with which we can encode ℓ-qubit states.

The code can correct t-qubit errors, using the error-correction procedures for C_2 in the standard basis and for $C_1^\perp$ in the Hadamard basis.

There are such codes C_1, C_2 for sufficiently large n such that $\frac{k_2-k_1}{n}$ and $\frac{t}{n}$ are both constant. In fact, for any $\epsilon \in \left[0, \frac{1}{2}\right)$ and constant $\epsilon \in k \in [k, 1-2h(2\epsilon))$, where h is the binary entropy function, we can achieve

$$\frac{k_2 - k_1}{n} \geq k, \frac{t}{n} \geq \epsilon.$$

Theorem 5.2.5 (CSS)
Good quantum error-correction codes exist.

We can construct CSS codes for encoding more qubits, and correcting more errors from classical Reed-Muller codes. CSS codes are special cases of stablizer codes. They can be used to design algorithms that are resistant to errors in memory and gate-implementations.

Chapter 6

Encryption

6.1 Encryption

Encryption are methods to discretely send messages between two parties.

6.1.1 Vernam Cypher

Consider the following scenario.

1) Alice and Bob share a uniformly random string $K \in \mathbb{Z}_2^n$ (*private key*), that is known only to them.
2) When Alice wishes to send a message $X \in \mathbb{Z}_2^n$, she computes $C(X) := X \oplus K$ (*cypher-text*) and sends this to Bob.
3) Bob decrypts the message as $D := C \oplus K = X$.

Any eavesropper who has no information about K, gets no information about K as C is uniformly random and independent of X.

This scheme has perfect security, but has disadvantages:

(i) The private key must be kept secret from everyone else. Hence a different private key must be used to communicate between Alice and Charlie.

(ii) A private key can be used only once. If two messages X_1, X_2 are encrypted with the same key K, anyone who intercepts the cyphertexts can learn $X_1 \oplus X_2$.

(iii) In order to keep the private key K secret, there must be a safe way for them to

communicate the key.

Public key encryption was proosed to overcome these limitations.

6.1.2 Public Key Encryption

1) Bob generates a pair of keys (K_1, K_2), keeps K_2 a secret (*private key*), and publishes K_1 (*public key*).
2) To encrypt X, Alice computes some function $f(X, K_1)$, and sends this to Bob.
3) Bob uses the private key K_2 to compute $g(C, K_2)$, where C is the ciphertext, and g is some function, to recover X.

This scheme is based on the assumption that K_2 is computationally hard to compute, given the public key K_1 (ie f is hard to invert given only K_1). Such schemes have several advantages:

(i) Only one private key per recipient is required.
(ii) The public key cna be reused.
(iii) The public key can be published or transmitted in the open.

Unfortunately, current such schemes assume either the hardness of integer factorization (RSA) or the hardness of discrete logarithm over an elliptic curve. Both these problems can be solved efficiently with a quantum computer.

6.1.3 Quantum Key Distribution

This is an unconditionally secure protocol whose validity is bcased on quantum theory alone. In addition, it only requires the ability to prepare simple single-qubit states $(|0\rangle, |1\rangle, |+\rangle, |-\rangle)$.

We need only a way to communicate a private key securely between Alice and Bob. the Vernam Cypher does the rest.

Consider the following setting:

1) Alice and Bob have secure computers, to which the eavesdropper Eve does not have access.
2) Alice and Bob can commmunicate over two channels, a classical one, and a quantum one. Both are bi-directional.

3) Alice and Bob know they are not talking to impersonations (the classical channel is *authenticated*). Moreover, Alice and Bob can detect if any message over that channel has been tampered.

4) Eve is computationally unbounded, and can intercept messages over both channels.

5) Alice, Bob, and Eve start in a tensor product state $|\bar{0}\rangle^A |\bar{0}\rangle^E |\bar{0}\rangle^B$. Alice and Bob communicate with each other over the two channels and compute either "FAIL" or "PASS" and strings K_A, K_B, respectively.

6) When there is no eavesdropper, the output is PASS with probability 1, and $K_A = K_B$, with K_A uniformly distributed.

7) When an eavesdropper is present, with high probability, one of the following two events occur:

   (a) Both Alice and Bob output FAIL

   (b) Both Alice and Bob output PASS, and compute uniformly random strings $K_A = K_B$, and the eavesdropper has "very little information" about K_A.

The idea is to share some number of Bell state pairs $|\phi\rangle^{\otimes k}$ where one qubit is with Alice and the other with Bob. A measurement in the standard basis yields the uniformly random key.

Suppose at the end of a protocol, the final state before measurement of any registers is

$$|PASS\rangle^{A_0B_0} \otimes (|\phi\rangle^{\otimes k})^{A_1B_1} \otimes |\psi\rangle^{A_2B_2E},$$

then whatever measurement Eve makes will be independent from that of the generated key.

First Attempt

Alice prepares $2k$ qubits in state $|\phi\rangle^{\otimes k}$, and sends one qubit from each Bell pair to Bob.

The issue is that Eve may make a copy of the state in the standard basis

$$|\phi\rangle^{\otimes k} \mapsto \frac{1}{\sqrt{2^k}}(|000\rangle + |111\rangle)^{\otimes k}$$

so she learns the key exactly.

We must be able to detect eavesdropping. Eavesdropping manifests itself as errors in the system state. Hence we check the Bell states for errors.

Alice and Bob can measure their qubits in the standard basis and compare the results to detect bit flip errors. For a phase flip error, Alice and Bob can measure in the Hadamard basis and compare the results to detect phase errors.

Unfortunately, they can measure either in the standard basis OR in the Hadarmard basis, but not both. The solution is to uniformly randomly measure half the Bell states in the standard basis, and the other half in the Hadamard basis. This helps us estimate with a small constant, the fraction of each kind of error.

The issue is that no more Bell states are left for generating the key. Thus we can measure a uniformly random subset of half the Bell states as before, but leave the rest for the key.

Now, the unmeasured Bell states may have errors. The solution is to output FAIL if the estimated fraction of errors is larger than a threshold $\epsilon \in \left(0, \frac{1}{2}\right)$. Fewer than ϵ fractions can be corrected with quantum error-correction.

Decide in advance which Bell states will be used to error-estimation, and which for key-generation. Use a suitable code to encode the key qubits, and send both kinds of qubits to Bob. Since Eve does not know which qubits are checked and which are used, the error-rate in both will be roughly the same.

Protocol II

All steps starting from the 5th step are done through the (authenticated) classical channel.

1) Alice prepares k Bell states $|\phi\rangle^{\otimes k}$ in registers A_1B_1.
2) Alice encodes the state in register B, using a CSS code C with message length k that can handle error-rate $\epsilon \in \left[0, \frac{1}{2}\right)$. Let the block-length of the code C be n, and the encoded state be in register B_1'.
3) Alice prepares n Bell states $|\phi\rangle^{\otimes n}$ in registers A_2B_2, with one qubit of each Bell state in A_2 and the other in B_2.
4) Alice permutes the qubits in registers B_1' and B_2 uniformly at random, and sends all these qubits to Bob over the quantum channel.
5) Bob acknowledges recept of $2n$ qubits.
6) Alice sends the permutation she used for the qubits in B_1' and B_2.
7) Bob inverts the permutation of the qubits and obtains the registers B_1' and B_2.
8) Alice selects a uniformloy random string $S \in \{0,1\}^n$ and sends it to Bob.
9) Alice measures the i-th qubit in A_2 in the standard basis if $S_i = 0$, and in the Hadamard basis if $S_i = 1$.
10) Alice sends the outcome of the measurement to Bob.
11) Bob measures the i-th qubit in B_2 in the standard basis if $S_i = 0$, and in the Hadamard basis if $S_i = 1$.

12) Bob compares his outcomes with those sent by Alice.

13) Bob compares his outcomes with those by Alice. Let δ_0 be the fraction of indices such that $S_i = 0$ and the outcomes of Alice's and Bob's measurements of the i-th qubits of A_2 and B_2. respectively, are different. Similarly, let δ_1 be the fraction of indices such that $S_i = 1$ but the outcomes of Alice and Bob's measurements of the i-th qubits of A_2, B_2, respectively, are different.

14) If either $\delta_0 n$ or $\delta_1 n$ is at least $(\epsilon - \nu)\frac{n}{2}$, Bob informs Alice and they output FAIL. Here $\nu \in (0, \epsilon)$ is a constant parameter.

15) Otherwise, Bob uses the error-correction procedure for CSS to decode the state in B_1' into register B_1.

16) Alice and Bob measure the k-qubits in A_1, B_1 in the standard basis, and output PASS as well as the outcomes K_A, K_B, respectively.

The protocol Π for QKD can be shown to be unconditionally secure. We consider some simple cases within the proof to illustrate the key ideas.

When there is no eavesdropper, with probability 1, Alice and Bob output PASS, $K_A = K_B$, and K_A is uniformly distributed over $\{0,1\}^k$.

Proposition 6.1.1
For any $m \geq 1$ and integer $0 \leq \ell \leq \frac{m}{2}$,

$$\sum_{i=0}^{\ell} \binom{m}{i} \leq 2^{mh(\ell/m)},$$

where h is the binary entropy function.

Proposition 6.1.2
Suppose Eve applies a unitary operator $U := P \otimes V$ to $2n$ qubits sent by Alice and a private register E. Suppose $P := \otimes_{i=1}^{2n} P_i$, where each P_i is one of the four operators $\{I, X, Z, XZ\}$.
If at least $\epsilon(2n)$ of the $P_i \in \{X, XZ\}$, then Alice and Bob output FAIL with probability at least

$$1 - 2e^{-cn}$$

for a positive constant c that depends only on ϵ, ν.
Let T be the set of indices i for which $P_i \in \{X, XZ\}$. We have $|T| \geq \epsilon(2n)$ by assumption. We argue that the probability that Alice's choice of permutations that fewer than $(\epsilon - \nu/2)n$ locations in T contain check qubits is at most e^{-cn}.
If the check qubits located in T are measured in the standard basis, a bit-flip is detected. We expect close to a half fraction of these check qubits to be measured in the standard basis.
By the Hoeffding bound, the probability that fewer than $(\epsilon - \nu)\frac{n}{2}$ out of the $(\epsilon - \nu/2)n$ check qubits in T are measured in the standard basis is at most e^{-cn}.
By the union bound, the probability that either one of the above overlaps is at most $2e^{-cn}$. When the overlap of $T \cap B_2$ and $\{i : S_i = 0\}$ is at least $(\epsilon - \nu)\frac{n}{2}$. Alice and Bob detect this number of bit-flips, and output FAIL.
Now we show the probabilistic upper bound. The probability is

$$\sum_{i=0}^{(\epsilon-\nu/2)n-1} \frac{\binom{2\epsilon n}{i}\binom{2n(1-\epsilon)}{n-i}}{\binom{2n}{n}}$$

From the binomial expansion of $(1+1)^{2n}$ and $(1+1)^{2n(1-\epsilon)}$, we have

$$\binom{2n}{n} \geq \frac{2^{2n}}{2n+1}$$
$$\binom{2n(1-\epsilon)}{n-i} \leq 2^{2n(1-\epsilon)}.$$

It follows from these bounds as well as the previous proposition that the probability is bounded above by

$$\frac{2^{2n(1-\epsilon)}}{2^{2n}/(2n+1)} \sum_{i=0}^{(\epsilon-\nu/2)n-1} \binom{2\epsilon n}{i}. \leq (2n+1)2^{-2\epsilon n}2^{2\epsilon n \cdot h(\alpha)}$$
$$\leq e^{-cn}$$
$$\alpha := \frac{\epsilon - \nu/2}{2\epsilon}$$

Here c is a suitable constant.

Since B_1' is the encoding of a state in a CSS code which can correct ϵ fraction errors, Bob can decode the state correctly. The joint state of Alice, Bob, and Eve is then

$$|\text{PASS}\rangle \otimes (|\phi\rangle^{\otimes k})^{A_1 B_1} \otimes |\psi\rangle^{ABE},$$

where A, B are all the remaining registers of Alice, Bob, respectively.

In general, Eve's action is a linear combination of different cases. Hence the analysis extends to the general case by the linearity of quantum operations. The security guarantee must be formulated carefully to account for the range of possible eavesdropping.

Chapter 7

Implementation

DiVicenzo Criteria for implementing a quantum computer.

1. Scalable physical system with well-characterized qubits.
2. Ability to initialize the state of the qubits to a simple fiducial state.
3. A universal set of quantum gates.
4. A qubit-specific measurement capability.
5. Long relevant decoherence times, much longer than the gate operation time.

7.1 Nuclear Magnetic Resonance

Qubits are encoded in the magnetic moment (spin) of certain nuclei in molecules. Many molecules are used to simulate a few qubits since measurement of a single molecule is difficult.

One qubit gates are implemented as Rabi-oscillations with RF pulses. to force rotations about the Bloch sphere.

As for two-qubit gates, nuclei already interact in a phenomenon known as "J-coupling". Thus our challenge is to regulate this coupling process. We are able to implement a CNOT gate this way.

Rotating fields will induce a current in the detection coils. The detected current is used to determine the final state of the qubits as an ensemble measurement. This is the reason to use many molecules: It is simply impossible to detect the current from a single molecule.

Note that we can initialize the nuclei in a pseudo-pure state. In the sense that "most" nuclei as in the $|0\rangle$ state and the rest contribute a net effect of zero.

The decoherence time is on the order of seconds while the 1-qubit gates takes on the order of microseconds and the 2-qubit gates the order of milliseconds.

7.1.1 Strengths & Weaknesses

This is a well-studied technology and there is excellent control with a few qubits.

However, the scalability is limited by molecules and signal to noise ratio.

7.2 Linear Optical Quantum Computing

The accessibility and cost of required devices are relatively better. The equipment does not need cooling. Moreover, photons interact weakly with the environment and thus we have longer decoherence times.

7.2.1 Qubits

One implementation is with spontaneous emission, where we rely on the energy level of an atom to drop and emit a photon in some random direction. We know the exact frequency of the photon but need an isolated quantum system. Moreover, the photon is emitted in a random direction (but can be fixed).

Another implementation is weak coherent light. The idea is that for a sufficiently weak light, it is approximately 1 photon. This is easy but not consistent or reliable. But it could be sufficient for QKD.

A third implementation is with Heralded single photons. We pump a non-linear crystal which emits 2 photons. We can detect one of them to know the existence of the other. However, most photons pass through and we cannot create a qubit on-demand.

States are typically implemented as dual-rail encoding, where the phonton being in the top rail encodes the $|0\rangle$ state and vice versa.

7.2.2 Measurements

We wish to convert optimal signals to electric signals.

The first way is through the photoelectric effect, where we can free electrons from a charged plate by hitting it with a sufficiently powerful photon.

Another implementation is through photomultiplier tubes. Once a photon hits the photocathode, it releases an electron which in turns releases more electrons, etc, until there is a sufficiently strong electrical signal we can detect.

7.2.3 Single Qubit Gates

These are accomplished through linear optical elements such as polarizers, wave plates, and polarizing beam splitters.

The Z-gate is a phase shifter which causes the phase to shift on the second rail. The Y-gate is a beam splitter on both rails.

7.2.4 Two Qubit Gates

We are able to implement a probabilistic control-Z gate. The success rate is enhanced through quantum teleportation while error-correction can reduce the resources needed.

7.2.5 Docoherence

The decoherence times can be very large while both 1 (time to go through optimal element) and 2-qubit (depends on measurement/feedback times) gates are very fast.

7.2.6 Strengths & Weaknesses

Multiple photons can be generated in terms of scalability.

The Decoherence times can be very large and the gate times can be very fast.

7.3 Trapped Ion

The qubits are encoded as electonic states of a trapped ion. We can scale through modular traps connected together.

Initialization is done through laser cooling and optimal pumping.

1-qubit gates are implemented as Rabi oscillations/Raman transitions with RF fields. On the other hand, 2-qubit gates are done via the motional state of the ion.

Measurements are implemented through state-dependent fluroescence.

The decoherence time is on the scale of $10^1 - 10^5$ milliseconds while the 1-qubit gates take 0.001 ms with the 2-qubit gates taking 0.1 ms.

7.3.1 Strengths & Weaknesses

The best part of this implementation are the long coherence/gate times with high gate fidelities. Moreover, all qubits are inherently identical which is excellent for reproducibility and reduction of calibration time.

However, gate operation times are very slow and the number of lasers scale linearly with the number of qubits. We also need to sort out the modular structures.

7.4 Superconducting Quantum Computing

The qubits are encoded as the electronic states of a superconducting LC circuit.

Initialization is done through cooling in a dilution fridge.

1-qubit gates are implemented as microwave field connected to a resonator. On the other hand, 2-qubit gates for tunable qubits consist of adjusting the frequency and for fixed qubits consists of driving the resonator.

Measurements consist of measuring the transmission spectra.

Decoherence times are on the order of 1 ms. 1-qubit gates take around 10-30 ns and 2-qubit gates around 10-100 ns.

7.4.1 Strength & Weaknesses

The main strengths of this implementation are the ability to tailor the qubit properties, stronger coupling to field than natural atoms (larger dipole moments) with faster rates, as well as the familiarity of silicon based architecture.

Unfortunately, we must cool our material much further than ions and we achieve worse gate/coherence times.