# Introduction to Quantum Computing Lecture Notes

Ahmet Çevik

## Contents

# 1 Introduction

Quantum mechanics is a deeply strange theory that challenges the traditional notions about the physical reality that we know of the macroscale world. It is inherently probabilistic and it forbids us from measuring certain kinds of physical quantities. For example, due to *Heisenberg's uncertainity principle* we cannot know, for certain, both the momentum and the position of a subatomic particle. It forbids us from asking questions about the physical system in consideration. If a subatomic particle has travelled from point $A$ to point $B$ in a trajectory motion, we may not know which path it followed. Yet quantum mechanics has been very successful in explaning physical reality, such as the conductivity of metals, transparency of glass, colors, chemical reactions and etc.

Classical computing has served us well for many purposes in the history of science. Computations relied on the rules of Newtonian physics. The seminal work of Einstein, Schrödinger, Dirac and other influential scientists about a new kind of physics tells us that it may actually be time for a change towards a different computational paradigm.

Quantum computing is a type of natural computing which uses subatomic matter and quantum mechanical principles that we will explain shortly such as *superposition*, *interference* and *entanglement* to solve various problems more efficiently than classical algorithms. It is worth noting beforehand that quantum computers do not solve problems that classical computers fail to solve. In other words, quantum computers do not have any computational power over classical computers as they compute the same class of functions. It is just that quantum computers solve certain hard problems more efficiently than classical computers. This does not to say that classical computers cannot solve these hard problems given a sufficient amount of time.

Quantum computational models give us a considerable amount of asymptotic speed-up in solving different kinds of problems including unordered searching, integer factorization, period finding and many others. For our purpose, we will look at popular quantum algorithms starting from the simplest one, moving to more complicated quantum algorithms.

The content of this lecture notes is taken from multiple sources, but the topics are carefully chosen with the aim of teaching the most "standard" subjects in a typical quantum computing course.

We assume some basic familiarity with linear algebraic notions. We shall also assume some high school trigonometry and complex numbers. Necessary definitions will be given in this chapter. No background on quantum mechanics is assumed, although it would be certainly helpful to understand the intuition behind the course material. It is also encouraged that whenever the reader is not clear about some point, he could quickly skim through the rest of the chapter as the answer to those points might be provided at a later paragraph.

## 1.1 Classical Deterministic Systems

In the next two subsections we will show how matrices can be used in computing systems. The reader may skip to Section 1.3 without loss of generality. Now consider children's marbles placed on the vertices of a graph.

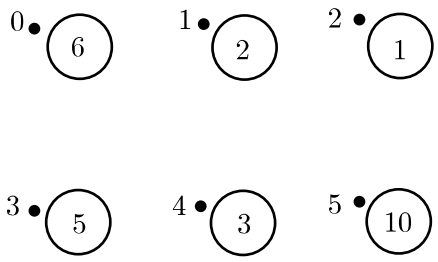



Figure 1: A deterministic state.

We shall denote the state of this system via the column vector

$$M = \begin{bmatrix} 6 \\ 2 \\ 1 \\ 5 \\ 3 \\ 10 \end{bmatrix}.$$

We may also denote it by the row matrix to save some space $M^T = [6, 2, 1, 5, 3, 10]$, where $M^T$ is the *transpose* of $M$ defined as $M[i, j]^T = M[j, i]$, but we will mostly use the column matrix form.

We are also interested in the dynamics of the system, i.e. how states change. Let us represent this by the following directed graph.

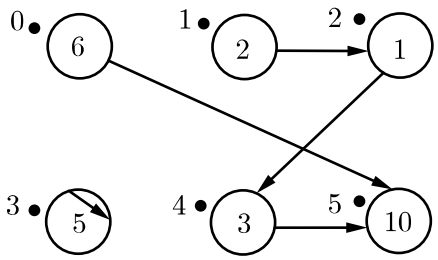



Figure 2: Dynamics of the system.

If there is an arrow from vertex $i$ to vertex $j$, then let us say in one time click all the marbles on vertex $i$ will move to vertex $j$. Since we are considered with deterministic systems in this example, the types of graphs we shall be concerned with are those with exactly one outgoing edge from each vertex.

The adjacency matrix of this graph is equivalent to the matrix $M$

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

where $M[i, j] = 1$ if and only if there is an arrow from vertex $j$ to vertex $i$.

Let us multiply $M$ by the state $X = [6, 2, 1, 5, 3, 10]^T$. Then we have,

$$MX = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 6 \\ 2 \\ 1 \\ 5 \\ 3 \\ 10 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 12 \\ 5 \\ 1 \\ 9 \end{bmatrix} = Y$$

This corresponds to the fact that if $X$ describes the state of the system at stage $s$, then $Y$ is the state at stage $s+1$. Therefore, $M$ is a way of describing how the state of the system changes. Finite dimensional quantum systems works in the same way. We may also apply the matrix $M$ $n$ many times consecutively on the state to see how the system changes after $n$ stages.

## 1.2 Classical Probabilistic Systems

First thing to note is that quantum mechanics is not deterministic. That is, neither the state of the system nor the dynamics of the system are deterministic. There is always an indeterminacy in our knowledge of a state. Moreover, the states change with probabilistic laws as opposed to deterministic laws.

Continuing our example, now let us deal with a single marble. The state of the system will tell us the probabilities of the single marble being on each vertex. For a 3-vertex graph, a typical state would look like $X = [\frac{1}{5}, \frac{3}{10}, \frac{1}{2}]^T$. This will correspond to the fact that there is 1/5 chance that the marble is on vertex 0, and 3/10 chance on vertex 1, and 1/2 chance on vertex 2. The marble must be somewhere in the graph. So the sum of the probabilities is 1.

We should also generalize the dynamics of the system. Instead of having exactly one edge leaving each vertex, we allow multiple edges with the condition that the sum of the probabilities of outgoing and incoming edges add up to 1.

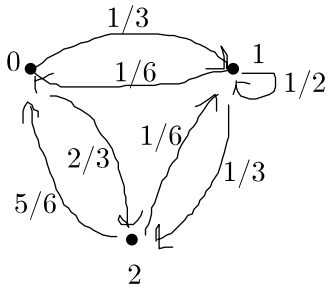



Figure 3: An example to a probabilistic system dynamics.

The matrix representation is

$$M = \begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}.$$

Suppose that we have a state $X = [\frac{1}{6}, \frac{1}{6}, \frac{2}{3}]^T$. If we apply $M$ on $X$ we get $MX = Y$ which is

$$\begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} \frac{21}{36} \\ \frac{9}{36} \\ \frac{6}{36} \end{bmatrix} = Y$$

Quantum systems are generally in a probabilistic state. Manipulating the system will correspond to multiplying the state by a matrix. Each computation step will correspond to one matrix transformation. The resulting vector will describe the state of the system.

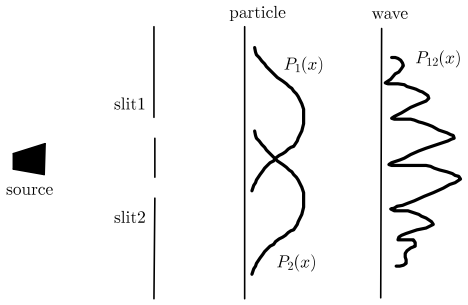
### 1.3 Quantum systems

In 1801, Thomas Young conducted the *double-slit experiment* that explained the concept of wave-particle duality, which later gave us hints regarding the behavior of matter at the subatomic level. The experiment also helped us to understand the physics of the microscopic world. Subatomic matters are neither particles nor waves, but they behave in their own quantum mechanical way. The double-slit experiment demonstrates that a subatomic particle of matter behaves sometimes like a macroscale particle and sometimes like

a wave. It also shows that the very act of observing a subatomic particle has dramatic effects on its behavior.

In this experiment, we have a source emitting light or electrons. Let us suppose that intensity of the source is so low that it emits the photons or electrons as discrete particles once in every second, let us say. We also have a panel with two slits, and along way from the panel there is a screen on which we measure the energy of the source admitted at some point $x$ on the screen. What is the probability that an electron is detected at point $x$?

First, if we were to shoot machine gun bullets, what would happen in the macroscale? If only one slit is opened, we observe that bullet hole will more likely to be detected on the screen closer to position of the slit.




What happens if we open both slits? Naturally a similar behavior will be observed for macroscale objects. Now what about the behavior of the light? If one slit is opened, we have no problem. It behaves exactly like a bullet. When two slits are opened, it forms an interference pattern on the screen.

Now if we were to shoot photons, discrete particles of light, we could expect that it would behave like a bullet. The strange part is that it forms the exact interference pattern.

The double-slit experiment demonstrates that physics at the subatomic level is a lot different than that of the macroscale level. In the next subsection we shall we looking at how to model this physics mathematically for the purpose of our study.

## 1.4 Complex Numbers and Linear Algebra

Complex numbers play a major role in quantum computing. Probabilities of states and transitions are given in the form of a complex number $c = a + bi$

such that $|c|^2 = a^2 + b^2$ is a real number between 0 and 1. We call $|c|$ the *modulus* of $c$, i.e., $|c| = \sqrt{a^2 + b^2}$.

Why complex numbers? The necessary reason has to do with time evolution of quantum systems in *Schrödinger's equation*. The auxiliary reason is that real number probabilities necessarily add together to obtain larger real numbers. On the other hand, complex numbers can *cancel* each other and lower their probability. In detail, if $p_1$ and $p_2$ are two real numbers between 0 and 1, then $(p_1 + p_2) \geq p_1$ and $(p_1 + p_2) \geq p_2$. However, if $c_1$ and $c_2$ are two complex numbers, $|c_1 + c_2|^2$ need not be bigger than $|c_1|^2$ or than $|c_2|^2$.

**Example.** Let $c_1 = 5 + 3i$ and let $c_2 = -3 - 2i$. Then $|c_1|^2 = 34$ and $|c_2|^2 = 13$ but $|c_1 + c_2|^2 = |2 + i|^2 = 5$. Notice that $|c_1 + c_2|^2 < |c_1|^2$.

But why do we need this kind of probability behavior? Subatomic particles both behave like a particle and a wave. When two waves interfere, they either cancel or rainforce each other (see Double Slit Experiment). The former is a *destructive interference* whereas the latter is called *constructive interference*. Complex numbers are used, for convenience, to model the *interference* phenomenon in quantum mechanics. In fact, complex numbers are necessary in quantum computing. As we will discuss later, this has to do with time evolution and reversibility.

Given a complex number $c = a + bi$, we denote the *complex conjugate* of $c$ by $c^* = a - bi$. The set of complex numbers is denoted by $\mathbb{C}$ and the set of real numbers is denoted by $\mathbb{R}$.

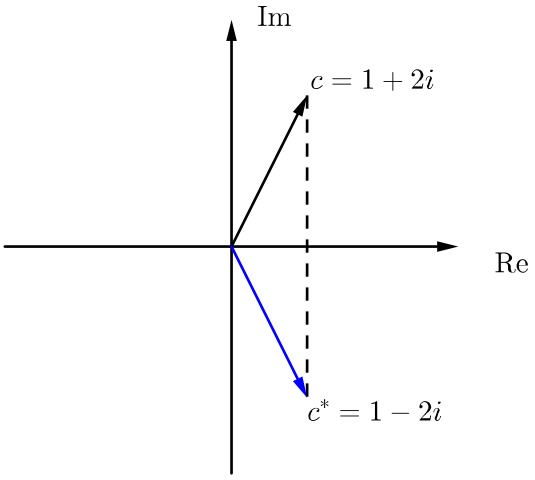



Figure 4: Complex conjugate of a complex number is just a reflecting its imaginary part around the real axis.

The reader should note that any complex number $c = a + bi$ can also be represented by polar coordinates $(\rho, \theta)$, where $\rho$ is called the *magnitude* and $\theta$ is called the *phase*. Given $c = a + bi$ such that $a, b \in \mathbb{R}$, from elementary trigonometry we compute $\rho$ and $\theta$ as

$$\rho = \sqrt{(a^2 + b^2)}, \qquad \theta = \arctan\left(\frac{b}{a}\right)$$

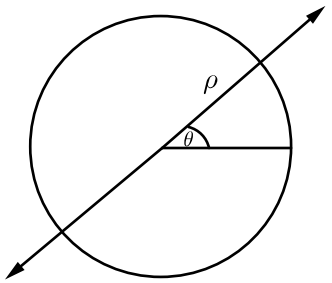



Figure 5: Polar representation of a complex number. The phase gives the degree and the magnitude gives the length of the vector.

We can go back from polar to Cartesian representation, again using trigonometry,

$$a = \rho\cos(\theta), \qquad b = \rho\sin(\theta).$$

We now give a few definitions of some linear algebraic concepts that we will use.

**Definition 1.** A *group* $(G, \cdot)$ is a set $G$ together with a binary operator $\cdot$ that satisfies the following axioms:

1. For all elements $a, b, c \in G$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
2. There exists an *identity element* $e \in G$, such that for any element $a \in G$, $a \cdot e = e \cdot a = a$.
3. For every element $a \in G$, there exists an inverse element $a^{-1} \in G$ such that $a \cdot a^{-1} = a^{-1} \cdot a = e$.

If a group $G$ also satisfies that for all elements $a \cdot b = b \cdot a$, then $G$ is called an *abelian group*.

**Definition 2.** Let $G$ be a group. If $H$ is a group and $H \subset G$, then $H$ is called a *subgroup* of $G$. If $H$ is a subgroup of $G$, then for each $g \in G$, the set

$$gH = \{gh : h \in H\}$$

is called a *coset* of $H$ (determined by $g$).

**Definition 3.** A *vector space* over complex numbers is an abelian group $V$ that is equipped with *scalar multiplication*, which is a mapping $\mathbb{C} \times V \to V$, which satisfies the following axioms: For every $c, c_1, c_2 \in \mathbb{C}$ and for every $\boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2 \in V$,

1. $c_1(c_2\boldsymbol{x}) = (c_1c_2)\boldsymbol{x}$.
2. $(c_1 + c_2)\boldsymbol{x} = c_1\boldsymbol{x} + c_2\boldsymbol{x}$.
3. $c(\boldsymbol{x}_1 + \boldsymbol{x}_2) = c\boldsymbol{x}_1 + c\boldsymbol{x}_2$.
4. $1\boldsymbol{x} = \boldsymbol{x}$.

The elements of $V$ are called *vectors*.

**Definition 4.** Let $V$ and $W$ be two vector spaces. A *linear map* from $V$ to $W$ is a function $f : V \to W$ such that for all $\boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2 \in V$ and $c \in \mathbb{C}$,

1. $f(\boldsymbol{x}_1 + \boldsymbol{x}_2) = f(\boldsymbol{x}_1) + f(\boldsymbol{x}_2)$,
2. $f(c \cdot \boldsymbol{x}) = c \cdot f(\boldsymbol{x})$.

We call any linear map from a vector space to itself a *linear operator*.

**Definition 5.** Let $V$ be a vector space. A *linear combination* of vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in V$ is a finite sum $c_1\boldsymbol{x}_1 + \ldots + c_n\boldsymbol{x}_n$, where $c_1, \ldots, c_n \in \mathbb{C}$. A set $S \subset V$ is called *linearly independent* if

$$c_1\boldsymbol{x}_1 + \ldots + c_n\boldsymbol{x}_n = \boldsymbol{0}$$

implies $c_1 = \cdots = c_n = 0$ whenever $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in S$. This means that the only way that a linear combination of vectors can be zero is if all the $c_i$'s are zero. A set that is not linearly independent is *linearly dependent*.

**Definition 6.** Let $V$ be a vector space and let $W = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}\} \subset V$ be a set. We say that $W$ is a *basis* for $V$ if the following two conditions are satisfied:

1. Every $\boldsymbol{x} \in V$ can be written as a linear combination of vectors from $W$, i.e. in this case we say that $W$ *spans* $V$.
2. $W$ is linearly independent.

The number of elements in $W$ gives the *dimension* of $W$. Throughout this course we will only consider finite dimensional complex vector spaces.

A natural way to introduce geometry into a complex vector space $V$ is to define the inner product.

**Definition 7.** Let $V$ be a complex vector space. An *inner product* on $V$ is a mapping $V \times V \to \mathbb{C}$, denoted as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, that satisfies the following conditions for any $c_1, c_2 \in \mathbb{C}$ and any $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ in $V$.

1. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle^*$.
2. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$ and $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$.
3. $\langle \boldsymbol{x}, c_1\boldsymbol{y} + c_2\boldsymbol{z} \rangle = c_1\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c_2\langle \boldsymbol{x}, \boldsymbol{z} \rangle$.

A vector space equipped with an inner product is also called an *inner product space*.

For vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1 \ldots, y_n)$ in $\mathbb{C}^n$, the formula

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = x_1^* y_1 + \ldots + x_n^* y_n$$

defines the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$.

**Definition 8.** Two vectors are called *orthogonal* if their inner product is 0.

From trigonometry, we understand that, if $V$ and $V'$ are unit vectors, the inner product $\langle V, V' \rangle$ is the directional length of the projection of $V$ onto the direction of $V'$. The inner product can be geometrically interpreted in that way as we shall be dealing with unit vectors.

**Definition 9.** Let $C$ be an $n \times n$ matrix. The *trace* of $C$, denoted by Trace($C$), is the sum of its diagonal elements. That is,

$$\text{Trace}(C) = \sum_{i=0}^{n-1} C[i, i].$$

**Definition 10.** We define the *norm* or the *length* of a vector $V$ as

$$||V|| = \sqrt{\langle V, V \rangle}.$$

If $||V|| = 1$, then $V$ is called a *unit vector*.

**Definition 11.** A set $V$ of vectors is called *orthonormal* if every vector in $V$ is pairwise orthogonal and is of unit length.

**Definition 12.** A complex inner product space $V$ is *complete* if for any Cauchy sequence of vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$, there exists a vector $\boldsymbol{y} \in V$ such that

$$\lim_{n \to \infty} |\boldsymbol{x}_n - \boldsymbol{y}| = 0.$$

The intuition behind this is that a vector space with an inner product is complete if any sequence accumulating somewhere converges to a point.

**Definition 13.** A *Hilbert space* is a complex inner product space that is complete.

It is worth noting that every finite dimensional complex inner product space is automatically complete. Hence any finite dimensional complex inner product space is a Hilbert space. Since we only work with finite dimensional vector spaces in quantum computing, we do not have to worry about the completeness criteria.

**Definition 14.** For a matrix $A$ in $\mathbb{C}^{n\times n}$, if there is a number $c \in \mathbb{C}$ and a vector $V \in \mathbb{C}^n$ such that

$$AV = cV$$

then $c$ is called an *eigenvalue* of $A$ and $V$ is called an *eigenvector* of $A$ associated to $c$.

Every eigenvector determines a complex vector subspace of the vector space. This space is known as the *eigenspace* associated with the given eigenvector.

**Definition 15.** Let $A$ be an $n \times n$ matrix. Define the *adjoin* of $A$ to be $A^\dagger = (A^T)^*$, where $A^*$ denotes the complex conjugate of $A$. The matrix $A$ is called *unitary* if $AA^\dagger = A^\dagger A = I_n$, where $I_n$ is the $n \times n$ identity matrix.

### 1.4.1 Combining Systems with Tensor Product

We will often deal with multiple systems and we will want to combine them as a whole single compound system. Tensor product is used to combine two or more systems into one. We will be using tensor products to represent combined quantum systems.

**Definition 16.** The *tensor product* of $r \times s$ and $t \times u$ matrices

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \dots & a_{rs} \end{bmatrix} \text{ and } B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1u} \\ b_{21} & b_{22} & \dots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & \dots & b_{tu} \end{bmatrix},$$

is an $rt \times su$ matrix defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1s}B \\ a_{21}B & a_{22}B & \dots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}B & a_{r2}B & \dots & a_{rs}B \end{bmatrix}$$

To give an example how a combined system works, consider the marble example we gave in the beginning of this chapter. Imagine now a blue marble placed on a graph whose vertices are labeled as $a, b, c$ and whose adjacency matrix is given as

$$M = \begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}$$

Imagine also a red marble placed on another separate graph whose vertices are labeled as $0, 1$ and whose adjacency matrix is defined as

$$N = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

Suppose that we want to combine these two systems. Then,

$$M \otimes N = \begin{bmatrix} 0 \cdot N & \frac{1}{6} \cdot N & \frac{5}{6} \cdot N \\ \frac{1}{3} \cdot N & \frac{1}{2} \cdot N & \frac{1}{6} \cdot N \\ \frac{2}{3} \cdot N & \frac{1}{3} \cdot N & 0 \cdot N \end{bmatrix}$$

would be the dynamics of the combined system represented by a $6 \times 6$ matrix.

A state in the combined two-marble system would be the tensor product of the state of the blue marble system and the state of the red marble system. Since the blue marble can be in one of three possible state and the red marble can be in one of two possible states, the tensor product of the combined system has six possible states. Let

$$B = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

be a state in a blue marbled system and let

$$R = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

be a state in a red marbled system. The state for the combined system would look like as follows:

$$X = \begin{matrix} a0 \\ a1 \\ b0 \\ b1 \\ c0 \\ c1 \end{matrix} \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix},$$

which corresponds to the fact that there is a

$\frac{1}{6}$ chance of the blue marble being on vertex $a$ and the red marble being on vertex 0.

$\frac{1}{6}$ chance of the blue marble being on vertex $a$ and the red marble being on vertex 1.

$\frac{1}{6}$ chance of the blue marble being on vertex $b$ and the red marble being on vertex 0.

$\frac{1}{6}$ chance of the blue marble being on vertex $b$ and the red marble being on vertex 1.

$\frac{1}{6}$ chance of the blue marble being on vertex $c$ and the red marble being on vertex 0.

$\frac{1}{6}$ chance of the blue marble being on vertex $c$ and the red marble being on vertex 1.

# 2 Basic Quantum Theory

We shall cover in this section how quantum states are formally represented and what kind of postulates quantum systems have.

## 2.1 Qubits

From classical computing we know that a *bit* is a boolean value that can either be in the basis state 0 or 1, but not both. A *quantum bit (qubit)* is the basic information content for an intended quantum system. A qubit can be in the basis state 0 or 1 or simultaneously both. We denote these *basis states* by $|0\rangle$ and $|1\rangle$.[1] This is called the *ket* notation, invented by the famous physicist Paul Dirac.[2] We shall denote the quantum states by lowercase Greek letters like $\psi, \phi, \varphi$.

A general state of a qubit $|\psi\rangle$ is a linear combination (superposition) of states in the canonical basis. That is,

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

where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$.

The coefficients $\alpha$ and $\beta$ are called *amplitudes*. So $|\psi\rangle$ can be denoted by a two dimensional complex vector

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

such that $|\alpha|^2 + |\beta|^2 = 1$. Note that when $\alpha = 1$ and $\beta = 0$, we have the basis state

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

When $\alpha = 0$ and $\beta = 1$, we get the basis state

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

So essentially a qubit can be viewed as a unit length vector in a two dimensional Hilbert space. We will see shortly, though, whenever we measure a qubit, it immediately collapses into a classical bit.

[1] We call this set of basis states, the *canonical basis*.

[2] It is also called the *Dirac notation* for this reason.

How are qubits implemented? There are many ways to achieve this. A qubit can be represented by one of the following phenomena.

1. Two possible states of an electron around the nucleus, being in the *ground state* or the *excited state*.
2. A photon having two different polarizations, i.e. *horizontal* and *vertical* polarizations.
3. A subatomic particle having two energy levels.

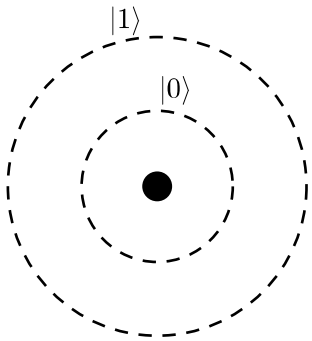



Figure 6: The state of an electron. We may let $|0\rangle$ denote the ground state and let $|1\rangle$ denote the excited state.

The examples of course are not limited to those we listed above.

**Geometric representation.** A nice way to represent qubits is by the polar form. Recall that given a complex number $c$,

$$|c|^2 = c \times c^* = (a + bi) \times (a - bi) = a^2 + b^2.$$

Given a complex number $c = a + bi$ of modulus 1, we can visualize it as an arrow of unit length from the origin of the circle of unit radius.

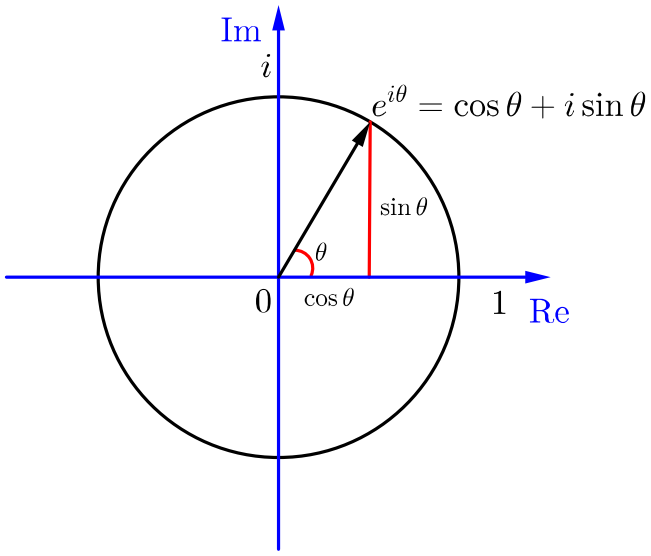



Figure 7: Trigonometric representation of a complex number.

As seen in Figure 7, every complex number can be identified by an angle $\theta$ that the vector makes with the positive $x$ axis and a magnitude $\rho$ (we may omit $\rho$ if we only consider vectors in the unit circle). A complex number $c$ in its general form, using polar coordinates, therefore can be written as

$$c = \rho(\cos\theta + i\sin\theta).$$

The following formula is known as *Euler's formula* and it is expressed as

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

Note that when $\theta = \pi$, it follows that $e^{\pi i} = -1$. So using Euler's formula we can write $c$ simply as

$$c = \rho e^{i\theta}.$$

There is an analogous representation of a qubit as an arrow from the origin to a three dimentional ball. Now a generic qubit is of the form

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle,$$

where $|c_0|^2 + |c_1|^2 = 1$. Although it may look like there are four real number parameters in this equation, it turns out that there are only two parameters. Since

$$c_0 = r_0 e^{i\phi_0}$$

and

$$c_1 = r_1 e^{i\phi_1},$$

the generic form of a qubit can be rewritten as

$$|\psi\rangle = r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle.$$

Defining $r_0$ as $\cos\theta$ and $r_1$ as $\sin\theta$, and using the fact that the squared modulus of the amplitudes add up to 1, any qubit $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ can be then represented as

$$|\psi\rangle = \cos(\theta)|0\rangle + e^{i\phi}\sin(\theta)|1\rangle.$$

Since only the relative phase between the coefficients of the two basis vectors has any physical meaning, we can take the coefficient of $|0\rangle$ to be real and non-negative representation. When we only care about the relative phases of the states but not their actual phases, this gives rise to *Bloch sphere*.

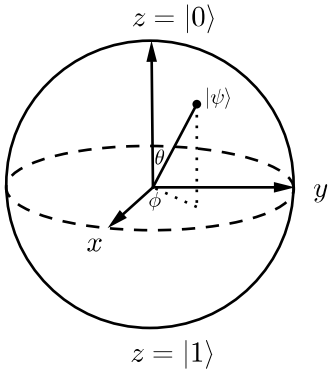



Figure 8: Bloch sphere representation of a single qubit quantum system $|\psi\rangle$.

A qubit can be represented by a unit length vector in a *Bloch sphere* as shown in Figure 8. We just need two angles that describe such a vector in the Bloch sphere. A qubit $|\phi\rangle$ then can be written as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle.$$

The angle $\theta$ shows the *latitude* and $\phi$ shows the *longitude (phase)*. The precise meaning of the above equation is the following: $\phi$ is the angle that

$|\psi\rangle$ makes from $x$ axis along the equator and $\theta$ is the angle that $|\psi\rangle$ makes with the $z$ axis. When a qubit is measured in the standard basis, it collapses to a bit, or equivalently, to the north or south pole of the Bloch sphere.

Rotating a vector around the $z$ axis is changing its phase. Notice that the probability of which classical state it will collapse to is not affected. Such transformation is called a *phase change*. In the equation given above, it corresponds to altering the phase parameter $e^{i\phi}$.

Single qubit systems may not be very practical. It is not possible to solve a reasonable problem with single qubit systems. We will need quantum computing systems involving multiple qubits. This is obtained by tensor products which was introduced in the last section.

**Notation.** Considering the definition of tensor products, let $|\psi\rangle \in \mathbb{C}^n$, $|\psi'\rangle \in \mathbb{C}^m$. Then $\mathbb{C}^n \otimes \mathbb{C}^m = \mathbb{C}^{n\times m}$ and $|\psi\rangle \otimes |\psi'\rangle = |\psi\rangle|\psi'\rangle = |\psi, \psi'\rangle = |\psi\psi'\rangle$. It is worth noting that the tensor product of vectors does not commute. For example, $|1\rangle|0\rangle \neq |0\rangle|1\rangle$.

As an example to a tensor product, suppose that we are given a state $|\varphi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, and a state $|\psi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. The tensor product of these two single qubit systems is

$$|\varphi\rangle \otimes |\psi\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{3}}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{3}}{2\sqrt{2}} \end{bmatrix}.$$

So the tensor product is a four dimensional (two qubit) system with basis states $00, 01, 10, 11$. Then, the two qubit system above is

$$|\varphi\rangle \otimes |\psi\rangle = \frac{1}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{1}{2\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|11\rangle.$$

Then, a 2-qubit system would have $2^2 = 4$ states in total, i.e. $00, 01, 10, 11$. These canonical basis states can be represented by the vectors as, respec-

tively,

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

A state of a 2-qubit system in a four dimensional Hilbert space $H^4$ has the form

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle = \begin{bmatrix}c_0\\c_1\\c_2\\c_3\end{bmatrix},$$

where $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$.

Generally, we may have $n$-dimensional quantum systems, generalizing our earlier implementation example:

1. A particle that can be in one of $n$ positions.
2. A system that might have one of $n$ energy levels.
3. A photon might have one of $n$ polarization directions.

A state of an $n$-dimensional quantum system $|\psi\rangle$ is represented by a unit length vector in an $n$-dimensional Hilbert space $H^n$

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + \ldots + c_{n-1}|n-1\rangle = \begin{bmatrix}c_0\\c_1\\\vdots\\c_{n-1}\end{bmatrix}$$

as a linear combination of $n$ many mutually orthogonal basis states $|0\rangle, |1\rangle, \ldots, |n-1\rangle$ such that

$$\sum_{i=0}^{n-1} |c_i|^2 = 1.$$

**Entanglement.** Multiple qubit systems exhibit a very fundamental quantum mechanical phenomenon called *entanglement* that was not present in the double-slit experiment. Consider a two qubit system

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$$

such that $\sum |\alpha_i|^2 = 1$. The probability to see $|\psi\rangle$ in state $|00\rangle$ is $|\alpha_0|^2$, to see in state $|01\rangle$ is $|\alpha_1|^2$, and so on. What about if we only measure the first qubit? This is called a *partial measurement*. What is the result of measuring just the first qubit? The probablity of seeing $|0\rangle$ in the first qubit is simply $|\alpha_0|^2 + |\alpha_1|^2$. A more interesting question is the following: If the outcome of the first qubit is 0, what is the new state after the measurement? Well, if the outcome of the first qubit is 0, it is certainly not $|10\rangle$ and $|11\rangle$ because these states are inconsistent with the outcome. What is left is the state $|00\rangle + |01\rangle$. Of course this state is not normalized now so we need to normalize it. The new state then becomes

$$|\psi'\rangle = \frac{\alpha_0|00\rangle + \alpha_1|01\rangle}{\sqrt{|\alpha_0|^2 + |\alpha_1|^2}}.$$

Let us look at the state

$$|\psi\rangle = \left(\frac{1}{2} + \frac{i}{2}\right)|00\rangle + \frac{1}{2}|01\rangle + \frac{i}{2}|11\rangle.$$

What is the probability of seeing $|0\rangle$ in the first qubit? It is simply $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$. The new state is

$$|\psi'\rangle = \frac{\left(\frac{1}{2} + \frac{i}{2}\right)|00\rangle + \frac{1}{2}|01\rangle}{\sqrt{\frac{3}{4}}}.$$

Recall that to combine two systems we use the tensor product. Given two qubits, we apply tensor product to get a combined 2-qubit system. What about the inverse process? Given a 2-qubit system, can we always factorize it into two systems of single qubit so that their tensor product is the composite 2-qubit system? It turns out that this is not always the case.

A state that is not the tensor product of the smaller states is called an *entangled state*. Consider 2-qubit systems. A state $z \in H^4$ of a two qubit system is *decomposable* if $z$ can be written as a product of states in $H_2$, $z = x \otimes y$. A state that is not decomposable is *entangled*, and in this case the qubits are entangled to each other.

**Example.** The state $\dfrac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$ is decomposable, since

$$\tfrac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

On the other hand the state $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ is entangled.[3] Suppose the contrary that

$$\begin{aligned}\tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) &= (a_0|0\rangle + a_1|1\rangle) \otimes (b_0|0\rangle + b_1|1\rangle) \\ &= a_0b_0|00\rangle + a_0b_1|01\rangle + a_1b_0|10\rangle + a_1b_1|11\rangle\end{aligned}$$

for some complex numbers $a_0, a_1, b_0, b_1$. But then

$$\begin{aligned}a_0b_0 &= \tfrac{1}{\sqrt{2}} \\ a_0b_1 &= 0 \\ a_1b_0 &= 0 \\ a_1b_1 &= \tfrac{1}{\sqrt{2}}.\end{aligned}$$

So $a_0b_0$ and $a_1b_1$ are non-zero. Then, $a_0, b_0, a_1, b_1$ are all non-zero. But then neither $a_0b_1$ nor $a_1b_0$ can be zero. A contradiction.

What happens when we measure the Bell state? We will see the state $|00\rangle$ with probability $\frac{1}{2}$ and see $|11\rangle$ with the same probability. What is important here is that if we measure the first qubit, the second qubit turns out exactly the same with the first qubit, and vice versa. Imagine we prepare a quantum state in the Bell state and send one qubit to Mars and leave the other in Earth. Since two qubits are entangled, whenever we apply an operation on one qubit, the same operation will be automatically applied to the other.

## 2.2 State evolution

We said that quantum systems are probabilistic. A state in a two dimensional quantum system, say, will look like

$$|\psi\rangle = \begin{bmatrix}\alpha_0 \\ \alpha_1\end{bmatrix}$$

First of all let us take a look at the state $|\psi\rangle$. We should not interpret $|\psi\rangle$ like a state of a probabilistic system. It is incorrect to say by looking at $|\psi\rangle$ that the probability of a particle being in position $k$ is $|\alpha_k|^2$. In a quantum system, to be in state $|\psi\rangle$ means that the particle is in *all* positions simultaneously. It is in a *superposition* of basis states. Now if we perform a *measurement* the superposition collapses to a single classical state. $|\psi\rangle$ says,

[3] This state is called a *Bell state* and it will be widely used in quantum information exchange protocols. We will also call it an *EPR pair*.

*after* measuring the system, the particle will be found in position $k$ with probability $|\alpha_k|^2$.

Geometrically, state evolution can be seen as a rotation of the vector space. This rotation is represented by a matrix. However, we require the transformation to be unitary.

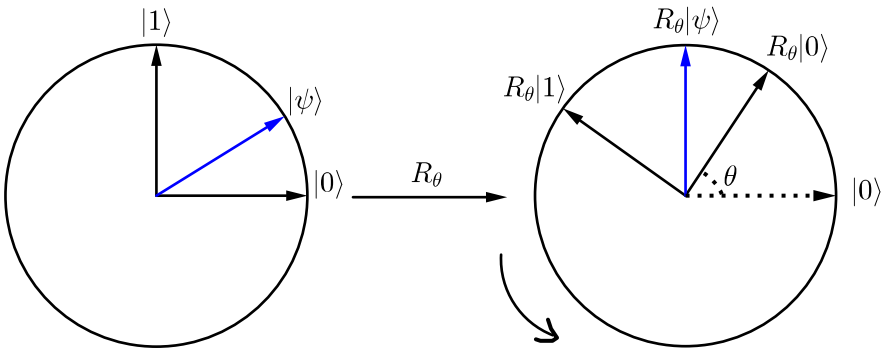



Figure 9: State evolution is a rotation of the vector space.

In the figure above, the rotation matrix maps the state $|0\rangle$ to state $\cos\theta|0\rangle + \sin\theta|1\rangle$, and it maps the state $|1\rangle$ to state $-\sin\theta|0\rangle + \cos\theta|1\rangle$. This rotation matrix can be then defined as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

The inverse of this rotation is just the transpose of $R_\theta$. That is,

$$R_\theta^{-1} = R^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Note that $R_\theta R_\theta^{-1} = R_\theta^{-1} R_\theta = I$.

In quantum computing, we require that all transformations are unitary. The first reason why transformations must be unitary is because quantum mechanical processes are *reversible*. Reversible in the sense that we are able to uniquely obtain the input from the output. Unitary operators are reversible since their inverse is simply their adjoin. Given a unitary operator $U$ and a vector $V$, we have that

$$V \to UV \to U^\dagger UV \to IV = V.$$

Second reason for using unitary matrices is that they preserve inner products and the norm of the vector.

**Proposition 17.** Unitary operators preserve the inner products. That is, if $U$ is a unitary matrix, then for any $V$ and $V'$ in $\mathbb{C}^n$ we have $\langle UV, UV' \rangle = \langle V, V' \rangle$.

**Proof.** The proof follows directly from the definition of unitary operators.

$$\langle UV, UV' \rangle = (UV)^\dagger (UV') = (V^\dagger U^\dagger)(UV') = V^\dagger V' = \langle V, V' \rangle.$$

The first and fourth equalities follow from the property of inner product, the second equality follows from the adjoin operator over matrices, the third equality follows from the fact that $U^\dagger U$ gives the identity matrix since $U$ is unitary. □

**Corollary 18.** Unitary operators preserve norms, i.e., $||UV|| = ||V||$.

**Proof.** It is easy to observe that

$$||UV|| = \sqrt{\langle UV, UV \rangle} = \sqrt{\langle V, V \rangle} = ||V||.$$

The first and third equalities follow from the definition of the norm, the second equality follows from the previous proposition. □

In this course we consider discrete time evolution of quantum systems. However in a typical quantum mechanics course, one learns that the continuous time evolution of a closed quantum system (ignoring special relativity) follows the *Schrödinger equation*

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = \hat{H}(t)|\psi(t)\rangle,$$

where $\hbar$ is a physical constant known as *Planck's constant* and $\hat{H}(t)$ is a Hermitian operator known as the *Hamiltonian* of the system. The Hamiltonian is an operator which represents the total energy function for the system. It may be a function of time but it may also be taken as constant.

## 2.3 Observables and measurement

In order to extract quantum information from a quantum system we have to observe the system, i.e. perform a *measurement*. We shall only consider projection measurements in our study for now.

Suppose that we are given a qubit

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle,$$

where $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Any measurement on $|\psi\rangle$ will cause the qubit to decide which one of the basis states it will remain in.

A measurement is therefore a *projection* onto one of the basis states $|i\rangle$ with probability $|\alpha_i|^2$, where $i \in \{0, 1\}$.

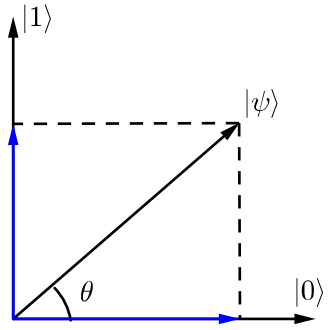



Figure 10: Measuring a qubit is a projection onto one of the basis states.

When we measure the state $|\psi\rangle$ we get $|0\rangle$ with probability $\cos^2\theta$ or get $|1\rangle$ with probability $\sin^2\theta$.

We may in fact measure a state in an any set of basis states which are mutually orthogonal. Take for example the basis states

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

$$|u^\perp\rangle = -\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

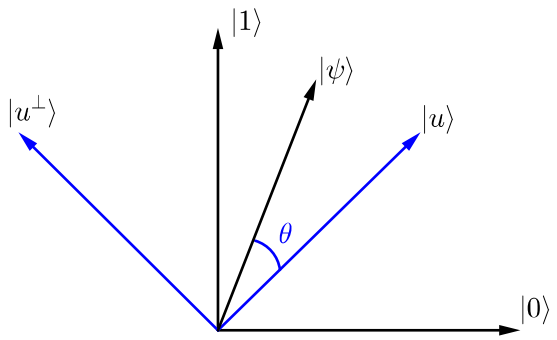



Figure 11: Change of basis.

If we measure $|\psi\rangle$ in the $\{|u\rangle, |u^\perp\rangle\}$ basis, we get $|u\rangle$ with probability $\cos^2\theta$, or get $|u^\perp\rangle$ with probability $\sin^2\theta$. In fact, the probability of seeing $|\psi\rangle$ in the $|u\rangle$ basis state is determined by the square of the inner product of $|\psi\rangle$ and $|u\rangle$.

Measuring in an arbitrary basis provides us to write any state $|\psi\rangle$ in that basis. Consider the *sign basis* which we shall define as

$$|+\rangle = \tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle,$$
$$|-\rangle = \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle.$$

This is a basis where $|+\rangle$ is the vector with a $\frac{\pi}{4}$ degrees angle with the state $|0\rangle$, and $|-\rangle$ is the vector which is the reflection of $|+\rangle$ around the state $|0\rangle$. So it should look like as follows

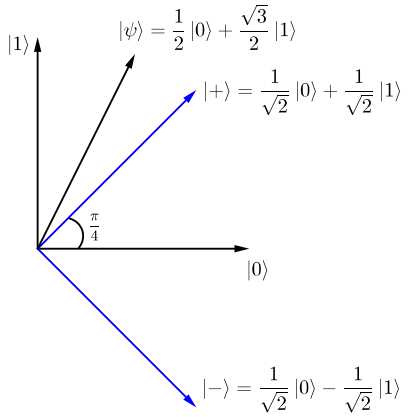



Figure 12: Representation of the sign basis.

Notice that these vectors are normalized and they are orthogonal to each other as their inner product is zero. Now after measuring, the probability of seeing $|\psi\rangle$ in state $|+\rangle$ is basically the square of the inner product of $|\psi\rangle$ and $|+\rangle$, which is

$$\left(\frac{1}{2}\cdot\frac{1}{\sqrt{2}} + \frac{\sqrt{3}}{2}\cdot\frac{1}{\sqrt{2}}\right)^2 = \frac{2+\sqrt{3}}{4}.$$

How do we write $|\psi\rangle$ in the sign basis?

$$\begin{aligned}
|\psi\rangle &= \alpha|+\rangle + \beta|-\rangle \\
|0\rangle &= \tfrac{1}{\sqrt{2}}|+\rangle + \tfrac{1}{\sqrt{2}}|-\rangle \\
|1\rangle &= \tfrac{1}{\sqrt{2}}|+\rangle - \tfrac{1}{\sqrt{2}}|-\rangle \\
|\psi\rangle &= \tfrac{1}{2}\left(\tfrac{1}{\sqrt{2}}|+\rangle + \tfrac{1}{\sqrt{2}}|-\rangle\right) + \tfrac{\sqrt{3}}{2}\left(\tfrac{1}{\sqrt{2}}|+\rangle - \tfrac{1}{\sqrt{2}}|-\rangle\right) \\
&= \left(\tfrac{1+\sqrt{3}}{2\sqrt{2}}\right)|+\rangle + \left(\tfrac{1-\sqrt{3}}{2\sqrt{2}}\right)|-\rangle
\end{aligned}$$

**Observables.** We now turn to question, what do we measure? We measure observables. An *observable* is a quantity like energy, momentum, position.

**Definition 19.** An $n \times n$ matrix $A$ is called *Hermitian* if $A^\dagger = A$. If $A$ is a Hermitian matrix then the operator that it represents is called *self-adjoint*.

**Example.** The matrix

$$\begin{bmatrix} 5 & 4+5i & 6-16i \\ 4-5i & 13 & 7 \\ 6+16i & 7 & -2.1 \end{bmatrix}$$

is Hermitian.

In quantum mechanics, given an $n$-dimensional quantum system, an observable is an $n \times n$ Hermitian operator.

By definition, the diagonal entries of a Hermitian operator must be real. This is due to the fact that if the diagonal entries were complex, then it would have to be $c = c^*$. But if $c = c^*$, then $c$ must be a real number.

Let us remind that the following property about the inner product that if $V, V' \in \mathbb{C}^n$, then

$$\langle V, V' \rangle = \langle V', V \rangle^*.$$

**Proposition 20.** If $A$ is an $n \times n$ Hermitian matrix, then for all $V$ and $V'$ in $\mathbb{C}^n$, we have that

$$\langle AV, V' \rangle = \langle V, AV' \rangle.$$

**Proof.** The proof is easy. We have

$$\langle AV, V' \rangle = (AV)^\dagger V' = V^\dagger A^\dagger V' = V^\dagger A V' = \langle V, AV' \rangle$$

The first and fourth equalities follow from the definition of an inner product. The second equality is from the property of $\dagger$ in matrix operations. The third equality follows from the fact that $A$ is a Hermitian matrix, i.e., $A^\dagger = A$. □

It is imporant to note that the eigenvalues of a Hermitian matrix are all real numbers.

**Proposition 21.** If $A$ is a Hermitian matrix, then all eigenvalues of $A$ are real.

**Proof.** Suppose that $A$ is a Hermitian matrix with eigenvalue $c \in \mathbb{C}$ and eigenvector $V$. We have the following equalities:

$$c\langle V, V \rangle = \langle cV, V \rangle = \langle AV, V \rangle = \langle V, AV \rangle = \langle V, cV \rangle = c^* \langle V, V \rangle.$$

The first and fifth equalities follow from the property of the inner product. The second and fourth properties follow from the definition of eigenvalue. The third equality follows from the last proposition. Now since $c$ and $V$ are non-zero and $c = c^*$, then $c$ must be a real number. □

Moreover, we have the following proposition.

**Proposition 22.** For a given Hermitian matrix $A$, distinct eigenvectors of $A$ which have distinct eigenvalues are orthogonal.

**Proof.** Let $V_1$ and $V_2$ be distinct eigenvectors of a Hermitian matrix $A$.

$$AV_1 = c_1V_1 \quad \text{and} \quad AV_2 = c_2V_2.$$

Then we have the following equalities:

$$c_1\langle V_1, V_2\rangle = \langle c_1V_1, V_2\rangle = \langle AV_1, V_2\rangle = \langle V_1, AV_2\rangle$$

$$= \langle V_1, c_2V_2\rangle = c_2^*\langle V_1, V_2\rangle = c_2\langle V_1, V_2\rangle.$$

The first and fifth equalities are from the properties of inner product, the second and fourth equalities are by definition of eigenvector, the third equality follows from the fact that $H$ is Hermitian, and the last equality is from the fact that eigenvalues of Hermitian matrices are real. As the left side is equal to the right side, we may subtract one from the other to get 0. That is,

$$c_1\langle V_1, V_2\rangle - c_2\langle V_1, V_2\rangle = (c_1 - c_2)\langle V_1, V_2\rangle = 0.$$

Because $c_1$ and $c_2$ are distinct, $c_1 - c_2 \neq 0$. Hence, it follows that $\langle V_1, V_2\rangle = 0$. Therefore, $V_1$ and $V_2$ are orthogonal since their inner product is 0. □

The orthogonality has a physical meaning. By *Heisenberg's uncertainity principle*, we can only measure either the position or the momentum of an electron of, say, a hydrogen atom. We cannot measure both at the same time.

**Definition 23.** A *diagonal matrix* is a square matrix whose only non-zero entries are on the diagonal.

**Theorem 24** (Spectral Theorem)**.** Every self-adjoint operator $A$ on a finite-dimensional complex vector space $V$ can be represented by a diagonal matrix whose diagonal entries are the eigenvalues of $A$, and whose eigenvectors form an orthonormal basis for $V$ which we shall call it an *eigenbasis*.

Spectral theorem says that every Hermitian operator $A$ has orthonormal eigenvectors $|\phi_1\rangle \ldots, |\phi_n\rangle$ with real eigenvalues $\lambda_1 \ldots, \lambda_n$ such that $A|\phi_i\rangle = \lambda_i|\phi_i\rangle$. So Hermitian operators are really just a mathematical formalism for specifying orthonormal basis for measurements. If we have a general state

$$|\psi\rangle = \alpha_1|\phi_1\rangle + \cdots + \alpha_n|\phi_n\rangle,$$

measurement on $|\psi\rangle$ is a projection onto one of the eigenvectors and the outcome will be the corresponding eigenvalue $\lambda_j$ (a real number) with probability $|\alpha_j|^2$. After the measurement the new state will be $|\phi_j\rangle$. How we read the eigenvalue $\lambda_j$ can be imagined as having a needle on a measurement device deflecting the number $\lambda_j$.

**Example.** Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ be a single qubit quantum state and let

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

be an observable. Suppose that we want to make a measurement with respect to the observable $X$. We need to determine the eigenvalues and eigenvectors of $X$. Fortunately, they are quite simple. The eigenvectors are $|+\rangle$ and $|-\rangle$. The corresponding eigenvalues are 1 and $-1$, respectively. We can verify this easily.

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

For the other eigenvector,

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

.

So how do we make the measurement? We want to write $|\psi\rangle$ in the basis of the eigenvectors. In this case, in the sign basis. Remembering what $|0\rangle$ and $|1\rangle$ are in the sign basis, we get

$$|\psi\rangle = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|+\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|-\rangle.$$

Now if we make a measurement on $|\psi\rangle$, we get the eigenvalue 1 with probability $\left|\frac{\alpha+\beta}{\sqrt{2}}\right|^2$ and the new state in this case will be $|+\rangle$; Or we get the

eigenvalue $-1$ with probability $\left|\frac{\alpha-\beta}{\sqrt{2}}\right|^2$ and the new state in this case will be $|-\rangle$.

We may also ask the *expectation value*. The expectation value gives the average outcome, and in this case it is

$$1 \cdot \left|\frac{\alpha+\beta}{\sqrt{2}}\right|^2 + (-1) \cdot \left|\frac{\alpha-\beta}{\sqrt{2}}\right|^2.$$

What if we have repeated eigenvalues? It may be the case that we have the same eigenvalues for different eigenvectors. Let us have a state $|\psi\rangle$ in a 3-dimensional space with a given observable, say a Hermitian operator $A$, with eigenvalues $\lambda_1, \lambda_2, \lambda_3$ with the corresponding orthonormal eigenvectors $|\phi_1\rangle, |\phi_2\rangle, |\phi_3\rangle$.

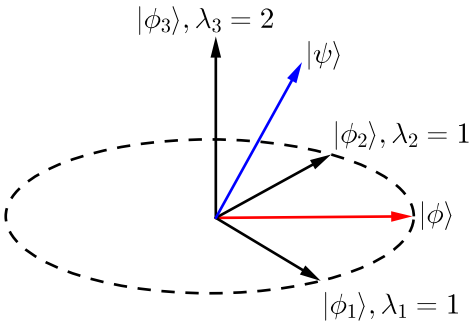



Figure 13: Repeated eigenvalues.

If we consider any vector $|\phi\rangle$ which is a linear combination of $|\phi_1\rangle$ and $|\phi_2\rangle$, it will also be an eigenvector with the same eigenvalue 1. That is,

$|\phi\rangle = \alpha|\phi_1\rangle + \beta|\phi_2\rangle$. If we apply $A$ on $|\phi\rangle$, we get
$A|\phi\rangle = \alpha A|\phi_1\rangle + \beta A|\phi_2\rangle$
$= \alpha \cdot 1|\phi_1\rangle + \beta \cdot 1|\phi_2\rangle$
$= \alpha|\phi_1\rangle + \beta|\phi_2\rangle$
$= |\phi\rangle$.

We now want to generalize the definition of observables for measuring in an arbitrary basis $|\phi_1\rangle, \ldots |\phi_n\rangle$ with arbitrary real outcomes $\lambda_1, \ldots, \lambda_n$. We

can define an observable with corresponding eigenvectors and eigenvalues. Let us demonstrate this with an example.

Suppose that the eigenvectors are

$$|\phi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle, \qquad |\phi_2\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle.$$

with corresponding eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -1$ respectively. We now define a Hermitian operator $A$ having these eigenvectors and eigenvalues. When we are given a state $|\psi\rangle$ we want to project it onto a state $|\phi\rangle$. The way we do this is by the *projection matrix* $P$ defined as

$$P = |\phi\rangle\langle\phi|$$

So $P$ projects a given state onto $|\phi\rangle$. It is easy to verify this as we can check that

$$P|\psi\rangle = |\phi\rangle\langle\phi||\psi\rangle = |\phi\rangle(\langle\phi||\psi\rangle)$$

The paranthesis is just the inner product of $|\phi\rangle$ and $\psi\rangle$ so it gives us a complex number as a coefficient of $|\phi\rangle$. We may then write the last expression as

$$P|\psi\rangle = (\langle\phi||\psi\rangle)|\phi\rangle$$

Now we claim that we can define the observable $A$ as follows

$$A = \lambda_1|\phi_1\rangle\langle\phi_1| + \lambda_2|\phi_2\rangle\langle\phi_2|$$

$$= 1 \cdot |\phi_1\rangle\langle\phi_1| + (-1) \cdot |\phi_2\rangle\langle\phi_2|$$

It can be verified that $A$ is a Hermitian matrix with eigenvectors $|\phi_1\rangle$ and $|\phi_2\rangle$ with eigenvalues $\lambda_1$ and $\lambda_2$, respectively associated with the eigenvectors. Let us verify that $A|\phi_1\rangle = \lambda_1|\phi_1\rangle$.

$$\begin{aligned} A|\phi_1\rangle &= (|\phi_1\rangle\langle\phi_1| - |\phi_2\rangle\langle\phi_2|)|\phi_1\rangle \\ &= |\phi_1\rangle\langle\phi_1||\phi_1\rangle - |\phi_2\rangle\langle\phi_2||\phi_1\rangle \\ &= |\phi_1\rangle(\langle\phi_1||\phi_1\rangle) - |\phi_2\rangle(\langle\phi_2||\phi_1\rangle) \\ &= |\phi_1\rangle \cdot 1 - |\phi_2\rangle \cdot 0 \\ &= |\phi_1\rangle \end{aligned}$$

So $A|\phi_1\rangle = \lambda_1|\phi_1\rangle$. We leave the reader to verify that $A|\phi_2\rangle = \lambda_2|\phi_2\rangle$. We can also verify this by explicitly writing the matrix $A$.

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \end{bmatrix} + (-1) \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{2} & -\frac{i}{2} \\ \frac{i}{2} & \frac{1}{2} \end{bmatrix} + (-1) \begin{bmatrix} \frac{1}{2} & \frac{i}{2} \\ -\frac{i}{2} & \frac{1}{2} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{2} & -\frac{i}{2} \\ \frac{i}{2} & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} & -\frac{i}{2} \\ \frac{i}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

From this we can show that $A|\phi_2\rangle = \lambda_2|\phi_2\rangle$.

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix} = (-1) \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} \end{bmatrix}$$

So to generalize all this, given an orthonormal set of vectors $\{|\phi_1\rangle, \ldots, |\phi_n\rangle\}$ and set of eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$, the corresponding observable is

$$A = \sum \lambda_i |\phi_i\rangle\langle\phi_i|$$

In summary, the two notions of measuring quantum systems are equivalent.

Now that we know the properties of Hermitian operators, we can give the following postulates of quantum computing.

**Postulate.** To each physical observable of a quantum system there corresponds a Hermitian operator.

**Example.** Let $|\psi\rangle = [-1, -1-i]^T$ be the start state in the two-dimensional state space. Let

$$\Omega = \begin{bmatrix} -1 & -i \\ i & 1 \end{bmatrix}$$

This matrix acts as an operator on $\mathbb{C}^2$. Therefore, we can apply it to $|\psi\rangle$. The result is the vector $\Omega|\psi\rangle = [i, -1-2i]^T$. Observe that $|\psi\rangle$ and $\Omega|\psi\rangle$ are *not* scalar multiples of one another, and thus they do not represent the same state: $\Omega$ has modified the state of the system.

**Postulate.** The eigenvalues of a Hermitian operator $\Omega$ associated with a physical observable are the only possible values the observable can take as

a result of measuring it on any given state. Furthermore, eigenvectors of $\Omega$ form a basis for the state space.

Observables can be thought of as legitimate questions as we can pose to quantum systems. Each question admits a set of answers: The eigenvalues of the observable.

In classical physics we assume that measuring a system would leave the system in the same state. This was shown to be wrong in quantum physics. The observable can only assume one of its eigenvalues as the result of an observation. Nothing tells us how frequently we are going to see a specific eigenvalue $\lambda$. Moreover our framework does not tell us yet what happens to the state vector if $\lambda$ is actually observed. We introduce the following postulate.

**Postulate.** Let $\Omega$ be an observable and $|\psi\rangle$ be a state. If the result of measuring $\Omega$ on state $|\psi\rangle$ is the eigenvalue $\lambda$, the state after measurement will always be an eigenvector corresponding to $\lambda$.

**Example.** Consider the same observable $\Omega$ given in the previous example. It is easy to check that the eigenvalues of $\Omega$ are $\lambda_1 = -\sqrt{2}$ and $\lambda_2 = \sqrt{2}$, and the corresponding normalized eigenvectors are $|e_1\rangle = [-0.923i, -0.382]^T$ and $|e_2\rangle = [-0.382i, 0.923]^T$.

Now, let us suppose that after an observation of $\Omega$ on a state $|\psi\rangle = \frac{1}{2}[1, 1]^T$, the actual value observed is $\lambda_1$. The system has "collapsed" from $|\psi\rangle$ to $|e_1\rangle$.

Recall that if $A|\phi\rangle = c|\phi\rangle$ for some $c$ then $c|\phi\rangle$ represents the same state as $|\phi\rangle$. So if the system is in an eigenvector of the basis, then the system will not change.

## 2.4 EPR Paradox

Suppose that we are given a Bell state

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

that we want to measure in the sign basis. We want to know the probability of seeing the state $|+\rangle$ when measuring the first qubit and if we see a plus, we want to know the probability of seeing $|+\rangle$ in the second qubit. To know

this we need to write the Bell state in the sign basis. We claim that the Bell state in the sign basis is actually again a Bell state. That is, we claim that

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle = \frac{1}{\sqrt{2}}|++\rangle + \frac{1}{\sqrt{2}}|--\rangle.$$

To see this, write the right handside as

$$\tfrac{1}{\sqrt{2}}\left((\tfrac{1}{\sqrt{2}}|0\rangle + |1\rangle)(\tfrac{1}{\sqrt{2}}|0\rangle + |1\rangle) + (\tfrac{1}{\sqrt{2}}|0\rangle - |1\rangle)(\tfrac{1}{\sqrt{2}}|0\rangle - |1\rangle)\right)$$

which can be written as

$$\tfrac{1}{\sqrt{2}}\left((\tfrac{1}{2}|00\rangle + \tfrac{1}{2}|01\rangle + \tfrac{1}{2}|10\rangle + \tfrac{1}{2}|11\rangle) + (\tfrac{1}{2}|00\rangle - \tfrac{1}{2}|01\rangle - \tfrac{1}{2}|10\rangle + \tfrac{1}{2}|11\rangle)\right)$$

If we cancel the terms, we get

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

So the probability we see a plus in the first qubit is just $\frac{1}{2}$. The new state in this case will become $|++\rangle$. So the probability of seeing a plus in the second qubit after measuring the first qubit is 1. The EPR (Einstein-Podolsky-Rosen) paradox is that, unlike Heisenberg's uncertainity principle, we may know the bit value and the sign value of the first qubit at the same time. What we do is we prepare and entangled state and move the second qubit to a distant galaxy or planet. We measure the first qubit in the standard canonical basis. At the same time we also measure the second qubit in the sign basis. Since these qubits are physically separated from each other and since the light does not have a chance to travel from the first qubit to the second qubit, measuring the second qubit will not disturb the state of the first qubit. Measuring the second qubit we get the sign value. This contradicts Heisenberg's uncertainity principle. The way quantum mechanics explains this paradox is that as soon as we measure the first qubit, the state will not be entangled anymore. If we, say, measure the first qubit as $|0\rangle$ in the canonical basis, the new state will be $|00\rangle$. But then this new state is not entangled anymore since it can be written as the tensor product of two single-qubit states $|0\rangle|0\rangle$.

In fact given any orthonormal basis states $|u\rangle$ and $|u^\perp\rangle$, the Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ in the $\{|u\rangle, |u^\perp\rangle\}$ basis is still a Bell state $\frac{1}{\sqrt{2}}|uu\rangle + \frac{1}{\sqrt{2}}|u^\perp u^\perp\rangle$. This property is called the *rotational invariance of Bell state*.

**Exercise.** Show that the rotational invariance property holds for the basis states

$$|u\rangle = a|0\rangle + b|1\rangle, \quad |u^{\perp}\rangle = -a|0\rangle + b|1\rangle,$$

where assume for simplicity $a, b \in \mathbb{R}$ and $a^2 + b^2 = 1$.

Continuing our discussion, let us suppose that we want to measure the second qubit in a different orthonormal basis $\{|v\rangle, |v^{\perp}\rangle\}$.

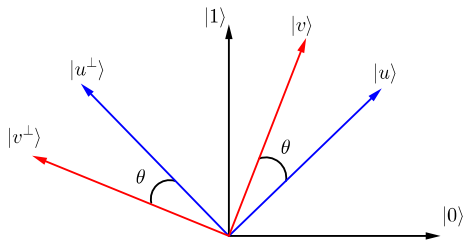



Figure 14: Rotational invariance of Bell state.

If the outcome of the first qubit is $|u\rangle$ then the new state of the system will be $|uu\rangle$. Now if we measure the second qubit in basis $\{|v\rangle, |v^{\perp}\rangle\}$, the probability of seeing $|v\rangle$ is simply $\cos^2\theta$. Similarly, if it if the outcome of measuring the first qubit was $|u^{\perp}\rangle$, the new state would be $|u^{\perp}u^{\perp}\rangle$ since we are working with the Bell state, and the probability of seeing $|v^{\perp}\rangle$ would be again $\cos^2\theta$. So the probability of seeing orthogonal complements is preserved through when working with the entangled Bell state.

## 2.5 Classical gates and quantum gates

Now we look at how we implement classical and quantum gates via matrices. An gate with $n$ input bits and $m$ output bits is represeted by an $2^m \times 2^n$ matrix. We represent $n$ input bits as a $2^n \times 1$ matrix and $m$ output bits as a $2^m \times 1$ matrix. So a $2^m \times 2^n$ matrix takes a $2^n \times 1$ matrix and outputs a $2^m \times 1$ matrix. Consider the NOT gate. Classical NOT gate take a single bit and outputs a single bit. Fortunately, it is same for qubits. NOT of $|0\rangle$ equals $|1\rangle$. NOT of $|1\rangle$ equals $|0\rangle$.

$$NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The first column shows what the output will be for the first input case (for $|0\rangle$ that is), and the second column shows what the output will be for the second input case (for $|1\rangle$). This matrix satisfies

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

and also

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

What about other classical gates? Here are the descriptions of other gates used

$$AND = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, NAND = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$OR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}, NOR = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

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

### 2.5.1 Reversible computation.

Quantum computation is heavily related to reversible computation since the time evolution of states in quantum mechanics is unitary and so it is inherently reversible. This reversibility is a consequence of restricting attention to closed quantum systems (ignoring special relativity). A computation is *reversible* if the input can be uniquely recovered from the output. The NOT gate for example is reversible since it is its own inverse. On the other hand, AND gate is not reversible.

In non-reversible gates, for example in AND gate, we lose information. Given the output 0, we may not know what the original input was. So we lose that information. Interestingly, quantum computation was first studied to try to understand whether or not the reversibility requirement of quantum mechanics impose any limit on what we can compute. It was not clear in the 1970's if we could compute anything in classical computing quantumly. Every step of a computation must dissapate some amount of energy. But this

energy is not due to producing information but it is due to losing information. This was came to known as *Landauer's principle*. Fortunately enough, it was shown that any classical computation can be done reversibly and so, in theory, no energy or heat would be produced. So any non-reversible gate can be transformed into a reversible gate, computing the same function (we will introduce Toffoli gates in a moment for this purpose).

We want our quantum gates to be reversible. Moreover, we do not want any *junk* information in the output of our quantum circuit for reasons which will be clear in a moment. Given a classical function $f$, we want to define it quantumly. At any stage of the computation we should be able to recover the computation and compute the inverse function. But one problem is the case when we try to find a reversible counterpart for functions $f : \{0,1\}^n \to \{0,1\}^m$ such that $m < n$. For example the AND gate. But that is just one example. How can we recover the computation if we lose all the information?

To do a computation reversibly we need to introduce additional auxiliary input and output wires if necessary, preserving equal number of input and output wires. By simply replacing all the non-reversible components with their reversible counterparts, we get a reversible version of the circuit. If we start with the output, and run the circuit backwards (replacing each gate by its inverse), we obtain the input again. So the reversible version of a non-reversible gate might require constant numbers of additional wires. Furthermore, the additional *junk* information generated by making each gate reversible can also be erased at the end of the computation by first copying the output, and then running the inverse of the circuit to obtain the start state again. The general schema of the reversible counterpart $C_f$ of a non-reversible function $f$ will look as in Figure 15.

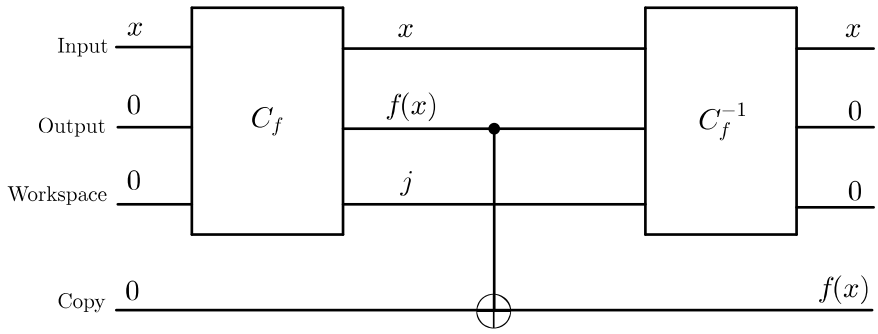



Figure 15: General scheme of a reversible circuit for quantum computation.

The above scheme is for reversibly computing a given function $f$. The block labelled $C_f$ represents a circuit composed of reversible gates. Then copy the output $f(x)$ to another register. Finally run the circuit $C_f^{-1}$ (inverse of $C_f$) to erase the contents of the output and the junk information. Notice that if we did not copy the output bit to another register, we would not have the output in our hand. Having the above entire circuit as our reversible version of the function $f$ that can be thought of as a quantum circuit. The output of this circuit is $x$, $f(x)$ and a bunch of workspace qubits. We got rid of the junk information as we promised.

Erasing the junk information was essential because junk output prevents interference.

Let us introduce the controlled-NOT (CNOT) gate which is defined as follows.

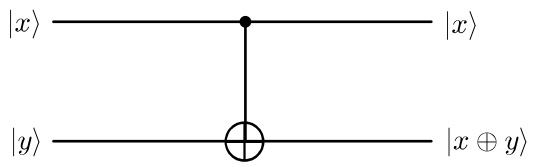



Figure 16: CNOT gate.

The CNOT gate has two inputs and two outputs. The top input is the

control bit. It controls the output. It inverts the second input only when the control bit is active. So if $|x\rangle = |0\rangle$, then $|y\rangle$ remains the same. It inverts $|y\rangle$ only when $|x\rangle = |1\rangle$. Then CNOT gate takes $|x, y\rangle$ to $|x, x \oplus y\rangle$, where $\oplus$ is the binary *exclusive-or* (XOR) operation. The truth table of $CNOT$ is given as follows.

| $x$ | $y$ | $CNOT(x,y)$ | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

The matrix representation of CNOT is

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

It can be easily verified that CNOT is unitary. Hence, the CNOT matrix is reversible. Observe that

$$CNOT(CNOT(|A, B\rangle)) = |A, B\rangle.$$

Another reversible gate is called the *Toffoli gate* which is defined as follows.

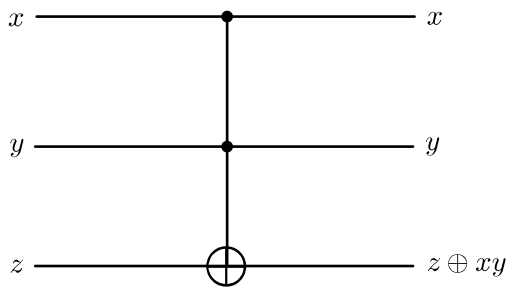



Figure 17: Toffoli gate.

The first two bits are used for controlling the target bit. So, in the boolean algebraic form, we have

$$T(x_1, x_2, x_3) = (x_1, x_2, x_1x_2 - x_3).$$

Toffoli gate is a reversible gate. Moreover, it is *universal* for classical computation. Another universal gate for classical computation is the NAND gate which is defined as $NOT(x \wedge y)$. Any classical logic gate can be simulated solely by using Toffoli gates. For example we can simulate the AND gate using only Toffoli gates. Notice that $T(x_1, x_2, 0) = (x_1, x_2, x_1x_2)$. So using 0 in the third input, Toffoli gates computes the logical AND of $x_1$ and $x_2$. Now about simulating other gates, since $x_1 \vee x_2 = \neg(\neg x_1 \wedge \neg x_2)$, we can replace all OR gates by AND and NOT gates. We know that NOT gate is reversible and we just showed that the AND gate can also be reversible using Toffoli gate. So in principle, classical computation can be solely made reversible.

**Quantum gates.** There are also gates which are used exclusively for quantum computing. One of the most important is called the *Hadamard transform*

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

Hadamard transformation basically creates a superposition of basis states. For example, when we apply $H$ on the basis state $|0\rangle$, we get

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

Verify that the Hadamard transform is unitary. Note that it is its own inverse. So if we apply it twice we get the original input. Therefore, the Hadamard gate is reversible.

What Hadamard transform does is that it rotates the vector space around the $\pi/8$ axis 180 degrees. So, the basis state $|0\rangle$ gets mapped to $|+\rangle$, and $|1\rangle$ gets mapped to $|-\rangle$.

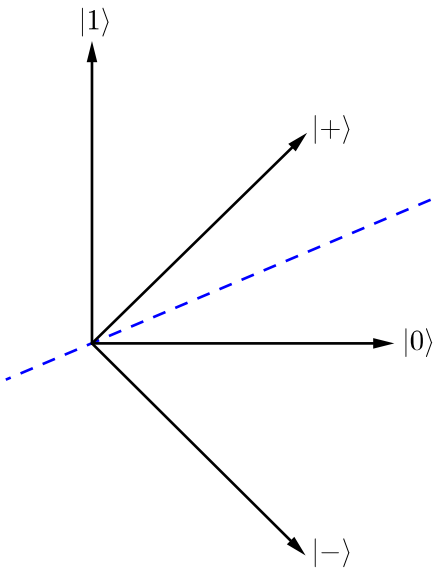



Figure 18: Hadamard transform is a rotation of the vector space around the $\pi/8$ axis.

Another important one bit gates are called *Pauli gates* which correspond to rotation around the $x$, $y$ and $z$ axes of the Bloch sphere. Pauli gates are defined as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Here, note that $X$ is nothing different than the NOT gate. Also note that the $X$, $Y$, $Z$ Pauli gates rotate the Bloch sphere 180° about the $x$, $y$, $z$ axes respectively. $Z$ is also known as the *phase flip* gate. Sometimes we may want to turn vector not by 180° around an axis but just about $\theta$ degrees along a particular direction. When applied for the $Z$-gate, such transformation would become a *phase shift* gate defined as

$$R(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix},$$

for some real $\theta$.

What if we want to apply the Hadamard transform to two qubit system.

In this case, we apply the matrix

$$H^{\otimes_2} = H \otimes H.$$

So for an $n$-qubit system we take the tensor product of the gate $n$ times and apply the resulting matrix to the system. What if we want to apply different gates to the qubits? If we want to apply, say, the NOT gate and the Hadamard gate to the first and second qubits, respectively, we take the tensor product of the NOT gate and the Hadamard gate $X \otimes H$ and apply this matrix to our system.

We spoke a lot about the Bell state so we should also give the Bell state circuit and how to create it. To create a Bell state all we need is a Hadamard transform and a CNOT gate providing the input $|00\rangle$.

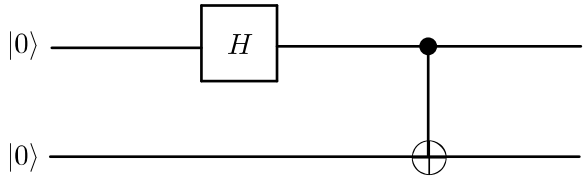



Figure 19: Bell state circuit.

We start with the state $|00\rangle$. We apply the Hadamard transform on the first qubit and get

$$(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle)|0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

Now if we apply the CNOT gate taking the second qubit as our target qubit and taking the first qubit as our control qubit, we get

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

which is how we get the Bell state. In fact, we get all the orthonormal set of Bell states when given different inputs $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. As an exercise we leave the reader to work with these inputs.

Let us review the postulates of quantum systems. We summarize the postulates as follows.

**State Space Postulate.** The state of a quantum system is described by a unit length vector in a Hilbert space.

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

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

State evolution can be seen as a rotation of the vector space.

**Composition of Systems Postulate.** When two physical systems, say $H_1$ and $H_2$, are treated as one combined system, the state space of the combined physical system is the tensor product space $H_1 \otimes H_2$ of the state spaces $H_1$ and $H_2$ of the component subsystems. If the first system is in the state $|\psi_1\rangle$ and the second system in the state $|\psi_2\rangle$, then the state of the combined system is

$$|\psi_1\rangle \otimes |\psi_2\rangle.$$

**Measurement Postulate.** For a given orthonormal basis $B = \{|\phi_i\rangle\}$ of a state space $H_A$ for a system $A$, it is possible to perform a *measurement* on system $H_A$ with respect to basis $B$ that, given a state

$$|\psi\rangle = \sum_i \alpha_i |\phi_i\rangle,$$

outputs a label $i$ with probability $|\alpha_i|^2$ and leaves the system in state $|\phi_i\rangle$. Furthermore, given a state $|\psi\rangle = \sum_i \alpha_i |\phi_i\rangle|\gamma_i\rangle$ from a bipartite state space $H_A \otimes H_B$ (every $|\phi_i\rangle$ is mutually orthogonal; the $|\gamma_i\rangle$ have unit norm but are not necessarily orthogonal), then performing a measurment on system $A$ will yield outcome $i$ with probability $|\alpha_i|^2$ and leave the bipartite system in state $|\phi_i\rangle|\gamma_i\rangle$.

# 3 Quantum Teleportation

Quantum teleportation is the process by which the state of an arbitrary qubit is transferred from one location to another. As we shall see however, when the state of a qubit is teleported to another location, the state of the original will have to be necessarily destroyed. The teleportation protocol will be heavily based on the use of entanglement EPR pairs.

Note that if two qubits are in an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, then observing one of them will give 0 or 1, both with probability of $\frac{1}{2}$, but it is not possible to observe different values of the qubits. This correlation can remain even if the qubits are seperated from each other light years away. This correlation plays an important role in quantum communication protocols. A pair of qubits in state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is called en *EPR pair*. Note that if we apply the Hadamard matrix on an EPR pair, the result is again EPR pair.

## 3.1 No Cloning Theorem

**Definition 25.** Let $\{|a_1\rangle, \ldots, |a_n\rangle\}$ be a set of basis states. Let us denote the state space by $H_n$ and specify that the state $|a_1\rangle$ is a "blank sheet state". A unitary mapping in $H_n \otimes H_n$ is called a *quantum copy machine*, if for any state $|\psi\rangle \in H_n$,

$$U(|\psi\rangle|a_1\rangle) = |\psi\rangle|\psi\rangle.$$

**Theorem 26** (No Cloning Theorem)**.** For $n > 1$, there is no quantum copy machine.

**Proof.** Assume that a quantum copy machine $U$ exists even if $n > 1$. Because $n > 1$, there are two orthogonal states $|a_1\rangle$ and $|a_2\rangle$. We should have $U(|a_1\rangle|a_1\rangle) = |a_1\rangle|a_1\rangle$ and $U(|a_2\rangle|a_1\rangle) = |a_2\rangle|a_2\rangle$, and also

$$U(\tfrac{1}{\sqrt{2}}(|a_1\rangle + |a_2\rangle)|a_1\rangle) = (\tfrac{1}{\sqrt{2}}(|a_1\rangle + |a_2\rangle))(\tfrac{1}{\sqrt{2}}(|a_1 + |a_2\rangle))$$

$$= \frac{1}{2}(|a_1\rangle|a_1\rangle + |a_1\rangle|a_2\rangle + |a_2\rangle|a_1\rangle + |a_2\rangle|a_2\rangle).$$

But since $U$ is linear,

$$U(\tfrac{1}{\sqrt{2}}(|a_1\rangle + |a_2\rangle)|a_1\rangle) = \tfrac{1}{\sqrt{2}}U(|a_1\rangle|a_1\rangle) + \tfrac{1}{\sqrt{2}}U(|a_2\rangle|a_1\rangle)$$

$$= \tfrac{1}{\sqrt{2}}|a_1\rangle|a_1\rangle + \tfrac{1}{\sqrt{2}}|a_2\rangle|a_2\rangle.$$

The two representations above for $U(\frac{1}{\sqrt{2}}(|a_1\rangle + |a_2\rangle)|a_1\rangle)$ do not coincide by the very definition of a tensor product. A contradiction. □

So copying an arbitrary quantum state is not possible unless we destroy it, i.e. only cut-and-paste is allowed.

### 3.2 Teleportation protocol

Consider two communication parties Alice (denoted by A) and Bob (denoted by B). Suppose that Alice has a single qubit in state

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

which is actually unknown to Alice. Suppose that Alice wishes to send this state to Bob. We say that there is a *quantum channel* from Alice to Bob if it is possible to send Bob the whole two-state quantum system. Similarly, if Alice can send classical bits to bob, we say that there is a classical channel from Alice to Bob.

We assume that there is no quantum channel from Alice to Bob, but a classical one exists. Alice cannot measure her state since it would disturb her qubit. She cannot make copies of her qubit either for many observations since copying an arbitrary quantum state is not possible due to No Cloning Theorem. It seems that it is possible for Alice to send her quantum bit to bob by only using a classical channel.

On the other hand, if Alice and Bob initially share an EPR pair, there is a way to execute the required task. This protocol is called *quantum teleportation*. Suppose that Alice and Bob have two qubits in EPR state

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

Assume that the first qubit belongs to Alice and the second qubit belongs to Bob. In addition, Alice has her qubit to be teleported in state

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

The compound state of all of the qubits will be denoted as

$(a|0\rangle + b|1\rangle)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$= \frac{a}{\sqrt{2}}|0\rangle|00\rangle + \frac{a}{\sqrt{2}}|0\rangle|11\rangle + \frac{b}{\sqrt{2}}|1\rangle|00\rangle + \frac{b}{\sqrt{2}}|1\rangle|11\rangle$

$= \frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|100\rangle + \frac{b}{\sqrt{2}}|111\rangle.$

Now recall that only the third qubit belongs to Bob. Alice can access the first two qubits.

**Teleportation protocol.**

1. Alice performs the CNOT matrix on her qubits, using the first qubit as the control qubit. The 3-qubit compound state then becomes

$$\frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|110\rangle + \frac{b}{\sqrt{2}}|101\rangle$$

Next, Alice applies the Hadamard matrix on the first qubit. The result is

$\frac{a}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|00\rangle + \frac{a}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|11\rangle$
$+\frac{b}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|10\rangle + \frac{b}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|01\rangle$

which can be written as

$\frac{a}{2}|000\rangle + \frac{a}{2}|100\rangle + \frac{a}{2}|011\rangle + \frac{a}{2}|111\rangle$
$+\frac{b}{2}|010\rangle - \frac{b}{2}|110\rangle + \frac{b}{2}|001\rangle - \frac{b}{2}|101\rangle$

Taking apart Bob's qubit, last state can be rewritten as

$\frac{1}{2}|00\rangle a|0\rangle + \frac{1}{2}|10\rangle a|0\rangle + \frac{1}{2}|01\rangle a|1\rangle + \frac{1}{2}|11\rangle a|1\rangle$
$+\frac{1}{2}|01\rangle b|0\rangle - \frac{1}{2}|11\rangle b|0\rangle + \frac{1}{2}|00\rangle b|1\rangle - \frac{1}{2}|10\rangle b|1\rangle.$

and, moreover as,

$\frac{1}{2}|00\rangle(a|0\rangle + b|1\rangle) + \frac{1}{2}|01\rangle(a|1\rangle + b|0\rangle)$
$+\frac{1}{2}|10\rangle(a|0\rangle - b|1\rangle) + \frac{1}{2}|11\rangle(a|1\rangle - b|0\rangle).$

Now Alice observes her two qubits. As the outcome, she sees $00, 01, 10, 11$, each with probability of $\frac{1}{4}$.

| Alice's obervation | Post-observation state |
|---|---|
| 00 | $\|00\rangle(a\|0\rangle + b\|1\rangle)$ |
| 01 | $\|01\rangle(a\|1\rangle + b\|0\rangle)$ |
| 10 | $\|10\rangle(a\|0\rangle - b\|1\rangle)$ |
| 11 | $\|11\rangle(a\|1\rangle - b\|0\rangle)$ |

Recall that the first two qubits belong to Alice, and the third qubit belongs to Bob.

4. Alice sends Bob her observation result (two classical bits).
5. Bob performs the following

- If Alice's bits are 00, Bob makes nothing. His qubit is already in state $a|0\rangle + b|1\rangle$.
- If Alice sent 01, Bob performs NOT gate on his qubit $a|1\rangle + b|0\rangle$ and get the desired state $a|0\rangle + b|1\rangle$.
- If Alice sent 10, Bob performs the *phase-change* matrix

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

  on his qubit to get $a|0\rangle + b|1\rangle$.
- If Alice sent 11, Bob first performs NOT gate and then phase-change gate.

Remark: It is worth noting that in quantum teleportation, no physical qubit is transmitted, just the state of the qubit. The quantum state is transmitted, not copies. So the original state held by Alice is destroyed in the protocol.

### 3.3 Superdense Coding

Superdense coding is the complementary action to quantum teleportation. Initially, Alice and Bob share an EPR pair and there is *quantum channel* from Alice to Bob. Now Alice wants to send *classical bits* to Bob. Superdense coding is a protocol where Alice sends one qubit to Bob, but the amount of transmitted information is two classical bits. This is done as follows: Alice has two classical bits, $b_1$ and $b_2$, and Alice shares an EPR pair, with Bob,

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

We assume that the first qubit belongs to Alice, and the other one belongs to Bob.

**Superdense coding protocol**

1. If $b_1 = 1$, then Alice performs the phase-change gate on her qubit. If also $b_2 = 1$, then she also performs the NOT gate on her qubit.

| $b_1$ $b_2$ | State after Alice's operation |
|---|---|
| 0 0 | $\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$ |
| 0 1 | $\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$ |
| 1 0 | $\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$ |
| 1 1 | $\frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$ |

2. Alice sends her qubit to Bob.

3. Now that Bob has access to both qubits, he runs them both through a two-qubit gate $B$ defined as

$$B = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

Note that $B$ is a unitary matrix. It is easy then to verify the following table

| $b_1$ $b_2$ | State after Alice's operation | State after Bob's operation |
|---|---|---|
| 0 0 | $\frac{1}{\sqrt{2}}\|00\rangle + \|11\rangle$ | $\|00\rangle$ |
| 0 1 | $\frac{1}{\sqrt{2}}\|10\rangle + \|01\rangle$ | $\|01\rangle$ |
| 1 0 | $\frac{1}{\sqrt{2}}\|00\rangle - \|11\rangle$ | $\|10\rangle$ |
| 1 1 | $\frac{1}{\sqrt{2}}\|10\rangle - \|01\rangle$ | $\|11\rangle$ |

4. Bob observes his qubits. The table shows that the bits $b_1$ and $b_2$ are recovered correctly.

Remark: Regarding Alice's action in the beginning, it is sufficient and necessary to force the qubits into orthonormal states.

# 4 Quantum Algorithms

In this section we will finally introduce quantum algorithms. But first let us discuss why quantum computing gives us an incredible extreme speed-up over classical computers. Given an $n$-bit classical computer, the number of states is just $2^n$. The computer will be in one of the $2^n$ possible states. Given an $n$-qubit quantum system, the quantum computer can be in a superposition of all possible $2^n$ states. For example, if we have a 3 qubit system, the set of basis states will be $\{|000\rangle, |001\rangle, \ldots, |111\rangle\}$. So in total there are $2^3 = 8$ basis states. In the general form, an $n$-qubit quantum system will have $2^n$ many basis states and the general state of the system will look like

$$|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle$$

such that $\alpha_n \in \mathbb{C}$ and $\sum |\alpha_x|^2 = 1$. For the reader to get used to the notation, we encourage the reader to explicitly write the general state of an 3-qubit system.

As the number of bits increase, the number of states grow exponentially. Exponential functions grow extremely fast. For moderate values of $n$, $2^n$ is even larger than the number of atoms in the universe. What is the source of the computational power of quantum computers? Suppose we have a $k$-level quantum system having $k$ many basis states, and suppose we also have an $l$-level system having $l$ many basis states. The composite system, using tensor product, will need $k \cdot l$ parameters to be defined. Assume we bought 16MB and 32MB of classical memory for our computer. In total we have 48MB of memory. If this was a quantum system, the composite system's memory would not be the sum but the *product* of the memory of two systems. A qubit lies in the vector space $\mathbb{C}^2$. If we had an $n$-qubit system, we would have to take the tensor product of the vector spaces

$$\underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ times}} = \mathbb{C}^{2^n}.$$

So we have an exponentially large vector space. But can we manipulate the state of an $n$-qubit system and the amplitudes efficiently? Even if we have a superposition of states, we should manipulate these states in a clever way so that when we measure the qubits we get a desired outcome with a high probability. We will be starting with some initial qubits set to a fixed state and then apply a sequence of unitary transformations to qubits.

This sequence of unitary transformations will form a quantum algorithm. Defining a quantum algorithm is an art. This will be the aim of this chapter, to see how we can make use of the nature's most powerful computer to solve problems.

## 4.1 Deutsch's Algorithm

We now look at perhaps the simplest quantum algorithm which is known as *Deutsch's algorithm*. This algorithm is to solve a problem about finding the character of a function.

**Definition 27.** Let $f : \{0,1\} \to \{0,1\}$ be a function. If $f(0) \neq f(1)$, we say that $f$ is *balanced*. If $f(0) = f(1)$, then $f$ is *constant*.

**Problem.** Given a function $f : \{0,1\} \to \{0,1\}$ as a black-box (we can evaluate but cannot look inside the function), tell if the function is constant or balanced.

A classical computer can solve this problem by evaluating $f$ on both inputs, and then compare them to see whether $f$ is balanced or constant. Is there a better solution? With a classical algorithm, no. However with quantum computing, this is possible indeed. A quantum computer can be in a superposition of all states. We shall use this property to evaluate both inputs at one time.

Consider a function $f$

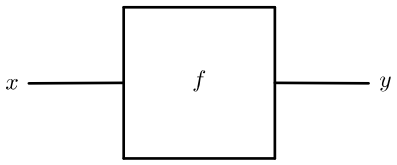



Figure 20: Classical representation of a function.

Such function could be thought of as a matrix. For example, the function $f$, where $f(0) = 0$, $f(1) = 0$ is represented as

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

Multiplying this with state $|0\rangle$ or with state $|1\rangle$ would result in state $|0\rangle$. For a quantum system we need the transformation to be unitary (hence reversible). If $f$ is the name of the function, then the following black-box (oracle) $U_f$ will be the quantum counterpart that we shall employ to evaluate input.

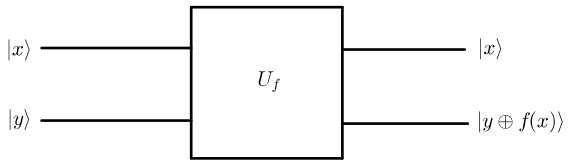



Figure 21: Quantum counterpart of the function $f$ as an oracle.

The top input $|x\rangle$ will be the qubit we want to evaluate and $|y\rangle$ controls the input. The function $U_f$ takes the state $|x, y\rangle$ to state $|x, y \oplus f(x)\rangle$. If $y = 0$, this simplifies to

$$|x, y\rangle = |x, 0 \oplus f(x)\rangle = |x, f(x)\rangle.$$

Clearly $U_f$ is reversible. That is, if we give $|x, y \oplus f(x)\rangle$ as an input to $U_f$ it will output $|x, y\rangle$. State $|x, y\rangle$ goes to $|x, y \oplus f(x)\rangle$ which further goes to $U_f$ once more. That is,

$$|x, (y \oplus f(x)) \oplus f(x)\rangle = |x, y \oplus 0\rangle = |x, y\rangle$$

For the function $f$ such that $f(0) = 1$ and $f(1) = 0$, the corresponding $U_f$ is

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

Each column represents the output when given the input $|x, y\rangle$.

So we are given some $U_f$ as an oracle and we want to determine whether the function expressed by $U_f$ is constant or balanced.

**Notation.** We shall denote a measurement by the "meter" figure

Figure 22: Measurement symbol.

**First attempt:**

Rather than evaluating $f$ twice, we try to put the states into superposition. Instead of having the top input to be either in state $|0\rangle$ or $|1\rangle$, we put the top input in state

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

So we are applying the Hadamard matrix $H$ on state $|0\rangle$.

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

Although it might not be correct, let us try to put the second qubit to state $|0\rangle$.

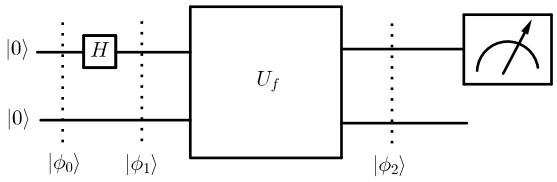



Figure 23: First attempt for Deutsch's problem.

So we have $U_f(H|0\rangle \otimes |0\rangle)$.
Let us examine each state.
$|\phi_0\rangle = |00\rangle$.
$|\phi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)|0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$

After multiplying with $U_f$, we have

$$|\phi_2\rangle = \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}$$

For the function $f$ such that $f(0) = 1$ and $f(1) = 0$, the state $|\phi_2\rangle$ would be

$$|\phi_2\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}.$$

If we measure the first qubit we get $\frac{1}{2}$ chance of finding it in state $|0\rangle$ and $\frac{1}{2}$ chance in $|1\rangle$. Similarly for the second qubit. So this solution gives us no information.

**Second attempt.**

Rather than leaving the second qubit in state $|0\rangle$, let us put it in the superposition state

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

This is still a superposition of states but notice the phase change. We can obtain this state by multiplying the Hadamard matrix with the state $|1\rangle$. Let us leave the first qubit as an ambiguous $|x\rangle$.

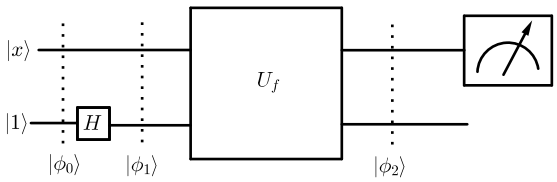



Figure 24: Second attempt for Deutsch's problem.

So here we have $U_f(|x, H(|1\rangle)\rangle)$. Then the states are as follows:

$$|\phi_0\rangle = |x, 1\rangle$$

$$|\phi_1\rangle = |x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \frac{|x0\rangle - |x1\rangle}{\sqrt{2}}$$

Applying $U_f$, we get

$$|\phi_2\rangle = \frac{|x, 0 \oplus f(x)\rangle - |x, 1 \oplus f(x)\rangle}{\sqrt{2}} = |x\rangle \left( \frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}} \right),$$

where $\overline{f(x)}$ denotes the complement of $f(x)$.
So we have

$$|\phi_2\rangle = \begin{cases} |x\rangle \left( \dfrac{|0\rangle - |1\rangle}{\sqrt{2}} \right), & \text{if } f(x) = 0 \\ \\ |x\rangle \left( \dfrac{|1\rangle - |0\rangle}{\sqrt{2}} \right), & \text{if } f(x) = 1 \end{cases}$$

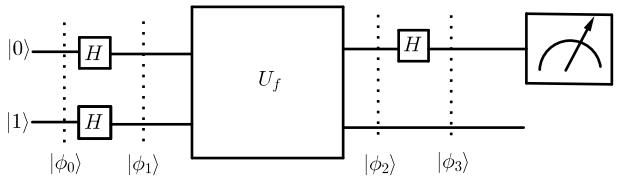
Recall that $a - b = (-1)(b - a)$. So we might write $|\phi_2\rangle$ as

$$|\phi_2\rangle = (-1)^{f(x)} |x\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

What would happen if we measure either the first or the second state? Again, this does not give any information. The first qubit will be in state $|x\rangle$ and the second qubit will be either in state $|0\rangle$ or $|1\rangle$, with equal probability. We need something more.

**Solution.**

We put both first and second qubits into superposition. We shall also put the outcome of the first qubit through a Hadamard transformation, causing an interference.




Let us examine the states.

$|\phi_0\rangle = |01\rangle$

$$|\phi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

We saw from the last attempt at solving this problem, that when we put the second qubit into superposition and then multiply by $U_f$ we will be in a superposition

$$(-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}.\right)$$

Note that $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ is the eigenvector of $U_f$ with the eigenvalue $(-1)^{f(x)}$. Now with $|x\rangle$ in a superposition, we have

$$|\phi_2\rangle = \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Let us carefully look at

$$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$

If $f$ is constant, this becomes either

$$+1(|0\rangle + |1\rangle) \text{ or } -1(|0\rangle + |1\rangle)$$

If $f$ is balanced, it becomes either

$$+1(|0\rangle - |1\rangle) \text{ or } -1(|0\rangle - |1\rangle)$$

Summing up, we have that

$$|\phi_2\rangle = \begin{cases} (\pm 1)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right), & \text{if } f \text{ is constant} \\ (\pm 1)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right), & \text{if } f \text{ is balanced} \end{cases}$$

Remember that the Hadamard matrix is its own inverse and takes $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$ to $|0\rangle$ and takes $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ to $|1\rangle$. We apply the Hadamard matrix to the first qubit to get

$$|\phi_3\rangle = \begin{cases} (\pm 1)|0\rangle \left( \dfrac{|0\rangle - |1\rangle}{\sqrt{2}} \right), & \text{if } f \text{ is constant} \\ \\ (\pm 1)|1\rangle \left( \dfrac{|0\rangle - |1\rangle}{\sqrt{2}} \right), & \text{if } f \text{ is balanced} \end{cases}$$

Now we simply measure the first qubit. If it is in state $|0\rangle$, then we know that $f$ is constant. Otherwise it is balanced. We did this with just one evaluation!

## 4.2 Deutsch-Jozsa Algorithm

We now give a straight forward generalization of the Deutsch's Problem. Instead of the single bit domain function, consider the function

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

given as a black-box. We assume that $f$ is either constant ($f(x)$ is the same for all $x$) or balanced ($f(x) = 0$ for exactly half of the input strings $x$, and $f(x) = 1$ for the other half of the inputs). We are asking the same question as before, whether $f$ is constant or balanced.

Classically, the solution requires $2^{n-1} + 1$ queries in the worst case scenario. The quantum algorithm we present here will solve the problem by making only one query!

Similar to what we did in the Deutsch's algorithm, we define the unitary quantum oracle

$$U_f : |\mathbf{x}\rangle|y\rangle \to |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$$

We are writing $\mathbf{x}$ in boldface since it refers to an $n$-bit string. $U_{f(\mathbf{x})}$ is now controlled by the register of $n$ qubits in the state $|\mathbf{x}\rangle$. It is easy again to see that $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ is an eigenvector of $U_{f(\mathbf{x})}$ with the eigenvalue $(-1)^{f(\mathbf{x})}$.

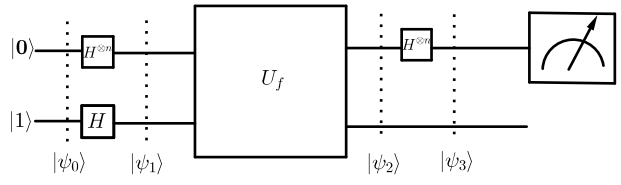



Figure 25: Circuit diagram for Deutsch-Jozsa algorithm.

Notice that instead of using 1 qubit Hadamard gate, we are using a Hadamard gate $H^{\otimes n}$ of $n$ qubits. That is,

$$H^{\otimes n} = \underbrace{H \otimes \cdots \otimes H}_{n \text{ times}}.$$

We shall use $|0\rangle^{\otimes n}$, or $|\mathbf{0}\rangle$ to denote the state that is the tensor product of $n$ qubits, each in state $|0\rangle$.

First step is to initialize the control register to prepare in state

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle.$$

Consider the action of an $n$-qubit Hadamard transformation on the state $|0\rangle^{\otimes n}$.

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \cdots (|0\rangle + |1\rangle). \quad (n \text{ times})$$

Expanding the tensor product, this can be written as

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle.$$

So the state immediately after the first Hadamard transformation is

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].$$

Note that the query register is now in an equally weighted superposition of all the possible $n$-bit input strings. Next we have the state after $U_{f(\mathbf{x})}$ gate.

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{\sqrt{2^n}} U_f \left( \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]\end{aligned}$$

Consider the action of the $n$-qubit Hadamard gate on an $n$-qubit basis state $|\mathbf{x}\rangle$. But first let us note that it is easy to see that the effect of the 1-qubit Hadamard gate on a 1-qubit basis state $|x\rangle$ can be written as

$$H|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle)$$

$$= \frac{1}{\sqrt{2}} \sum_{z \in \{0,1\}} (-1)^{xz} |z\rangle.$$

We can see that the action of the Hadamard transformation on an $n$-qubit basis state $|\mathbf{x}\rangle = |x_1\rangle|x_2\rangle \ldots |x_n\rangle$ is given by

$$H^{\otimes n}|\mathbf{x}\rangle = H^{\otimes n}(|x_1\rangle|x_2\rangle \ldots |x_n\rangle)$$

$$= H|x_1\rangle H|x_2\rangle \ldots H|x_n\rangle$$

$$= \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)\tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_2}|1\rangle) \cdots \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_n}|1\rangle)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{z_1 z_2 \ldots z_n \in \{0,1\}^n} (-1)^{x_1 z_1 + x_2 z_2 + \cdots x_n z_n} |z_1\rangle|z_2\rangle \cdots |z_n\rangle.$$

The above equation can be then written as

$$H^{\otimes n}|\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{z} \in \{0,1\}^n} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z}\rangle,$$

where $\mathbf{x} \cdot \mathbf{z}$ denotes the bitwise inner product of $\mathbf{x}$ and $\mathbf{z}$, modulo 2. We shall use this equation often, so it is important for the reader to keep this convention in mind. The state after the final $n$-qubit Hadamard transformation is

$$|\psi_3\rangle = \left( \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x})} \frac{1}{\sqrt{2^n}} \sum_{\mathbf{z} \in \{0,1\}^n} (-1)^{\mathbf{x} \cdot \mathbf{z}} |\mathbf{z}\rangle \right) \left[ \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

$$= \frac{1}{2^n} \sum_{\mathbf{z} \in \{0,1\}^n} \left( \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{z}} \right) |\mathbf{z}\rangle \left[ \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right].$$

At the end of the algorithm a measurement of the first register is made in the computational basis. To see what happens, consider the total amplitude of $|\mathbf{z}\rangle = |0\rangle^{\otimes n}$ in the first register of state $|\psi_3\rangle$. This amplitude is

$$\frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x})}.$$

Consider this amplitude in the two cases.

Case 1: $f$ is constant. If $f$ is constant, the amplitude of $|0\rangle^{\otimes n}$ is either 1 or $-1$, depending on which value $f(\mathbf{x})$ takes. So if $f$ is constant, a measurement of the first register is certain to return $|0\rangle^{\otimes n}$.

Case 2: $f$ is balanced. If $f$ is balanced, then it is easy to see that the positive and negative amplitudes interfere (cancel) each other since half of the amplitudes would be positive and the other half would be negative. In this case, the amplitude of $|0\rangle^{\otimes n}$ is 0. So if $f$ is balanced, a measurement of the first register is certain not to return $|0\rangle^{\otimes n}$.

To determine whether $f$ is balanced or constant, it is sufficient to measure the first register. If the result is $|0\rangle^{\otimes n}$, then $f$ is constant. Otherwise $f$ is balanced. Again, notice that we determined the characteristic of $f$ by making a single query by giving the register just the state $|0\rangle^{\otimes n}$.

### 4.3 Simon's Algorithm

Consider a function $f : \{0,1\}^n \to \{0,1\}^n$ such that $f(x) = f(y)$ if and only if $x = y$ or $y = x \oplus s$ for some hidden string $s \in \{0,1\}^n$.

In other words, the values of $f$ repeat themselves in some pattern and the pattern is determined by $s$. We shall call $s$ the *period* of $f$. *Simon's Problem* is to find the period $s$.

Note that if $s = 0^n$, then the function is one-to-one. Otherwise the function is two-to-one.

How can we solve this problem using a classical algorithm? We would have to evaluate $f$ on different binary strings. After each evaluation, we check to see if that output has already been found. If we find two inputs $x_1$ and $x_2$ such that $f(x_1) = f(x_2)$, then we are assured that

$$x_1 = x_2 \oplus s$$

and we can obtain $s$ by $\oplus$-ing both sides with $x_2$:

$$x_1 \oplus x_2 = x_2 \oplus s \oplus x_2 = s$$

If the function is a two-to-one function, then we will not have to evaluate more than half the inputs before we get a repeat. If we evaluate more than half the strings and still cannot find a match, then we know that $f$ is one-to-one and that $s = 0^n$. Hence, in the worst case scenario, we need $2^{n-1} + 1$ evaluations.

The quantum part of Simon's algorithm is to evaluate the following several times:

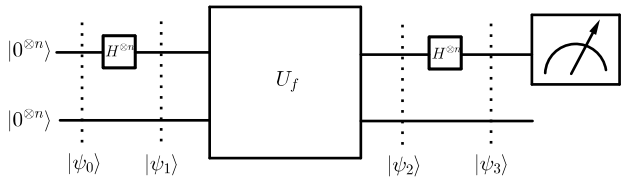



Figure 26: Quantum part of the Simon's algorithm.

Let us look at each stage. It is clear that

$$|\psi_0\rangle = |\mathbf{0}, \mathbf{0}\rangle.$$

We then place the first register in a superposition of all possible strings of $\{0,1\}^n$. From the previous examples, we know that

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}, \mathbf{0}\rangle.$$

Evaluation of $U_f$ on gives us

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}, f(\mathbf{x})\rangle.$$

Finally we apply $H^{\otimes n}$ to the first register and get

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^n} \sum_{\mathbf{z}\in\{0,1\}^n} (-1)^{\mathbf{x}\cdot\mathbf{z}} |\mathbf{z}, f(\mathbf{x})\rangle.$$

Note that there are exactly two $\mathbf{x}$'s, call them $\mathbf{x}_1$ and $\mathbf{x}_2$, for which $f(\mathbf{x}_1) = f(\mathbf{x}_2)$. Since $\mathbf{x}_2 = \mathbf{x}_1 \oplus \mathbf{s}$, we can rewrite $|\psi_3\rangle$ as

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{\mathbf{x}_1,\mathbf{x}_2\in\{0,1\}^n} \sum_{\mathbf{z}\in\{0,1\}^n} (-1)^{\mathbf{z}\cdot\mathbf{x_1}} + (-1)^{\mathbf{z}\cdot\mathbf{x_2}} |\mathbf{z}, f(\mathbf{x}_1)\rangle,$$

where $\mathbf{x}_2 = \mathbf{x}_1 \oplus \mathbf{s}$. Let us look at the coefficient of $|\mathbf{z}, f(\mathbf{x}_1)\rangle$.

The coefficient of $|\mathbf{z}, f(\mathbf{x}_1)\rangle$ is 0 when $(-1)^{\mathbf{z}\cdot\mathbf{x_1}} + (-1)^{\mathbf{z}\cdot\mathbf{x_2}}$ is 0. Given that $\mathbf{x}_2 = \mathbf{x}_1 \oplus \mathbf{s}$, this means that the coefficient of $|\mathbf{z}, f(\mathbf{x}_1)\rangle$ is 0 when

$(-1)^{\mathbf{z}\cdot\mathbf{x_1}} + (-1)^{\mathbf{z}\cdot\mathbf{x_1}\oplus\mathbf{s}} = (-1)^{\mathbf{z}\cdot\mathbf{x_1}} + (-1)^{\mathbf{z}\cdot\mathbf{x_1}\oplus\mathbf{z}\cdot\mathbf{s}} = (-1)^{\mathbf{z}\cdot\mathbf{x_1}} + (-1)^{\mathbf{z}\cdot\mathbf{x_1}}(-1)^{\mathbf{z}\cdot\mathbf{s}}$ is 0.

So when $\mathbf{z} \cdot \mathbf{s} = 1$, the coefficient of $|\mathbf{z}, f(\mathbf{x}_1)\rangle$ is 0. Hence, upon measuring the first register, we will only find those $\mathbf{z}$ which are orthogonal to $\mathbf{s}$, i.e., those which satisfy $\mathbf{z} \cdot \mathbf{s} = 0$.

This determines a subspace which $\mathbf{s}$ must lie on. Given $n$ many samples of $\mathbf{z}$, $\mathbf{s}$ is constrained to a 0-dimensional subspace and we can solve the $n$ linear equations using classical methods, Gaussian elimination for example, to determine $\mathbf{s}$.

Simon's algorithm requires $O(n)$ queries to the black-box, whereas a classical algorithm would need at least $\Omega(2^{n/2})$ queries. We should also note that Simon's algorithm is optimal in the sense that any quantum algorithm to solve this problem requires $\Omega(n)$ queries.

## 4.4 Grover's Search Algorithm

How do you find needle in a haystack? We have to look at each piece of hay separately and check each one to see if it is the desired needle. We may also convert this problem to finding a special element, or the index of the special entry, in an unstructured list or database as Lov Grover himself puts it.

**Problem.** Given a blackbox for computing the function $f : \{0,1\}^n \to \{0,1\}$ such that $f(x^*) = 1$; $f(x) = 0$ if $x \neq x^*$. Find the key value $x^*$.

Let us assume for convenience that there are $N = 2^n$ elements to be looked at. Classically, we are required to check $N - 1$ elements, provided that the element is in the list. We do not need to check the last element as it will be the desired entry. So that means, using classical algorithm, we need $O(N)$ steps to find the key value. Of course if the list were ordered, say in an increasing order, we could apply *binary search* for instance and obtain a solution with logarithmic complexity. With quantum computing the unstructured search can be done in $O(\sqrt{N})$ steps using *Grover's algorithm*. So we have a quadratic speed-up over the classical algorithm. Let us assume that, for convenience, there is a unique $x^*$ such that $f(x^*) = 1$. So by definition, $f(x) = 1$ if $x$ is the key value (i.e. the solution), and $f(x) = 0$ if $x$ is not a solution to the search problem. Suppose also that we are provided with an *oracle*, whose internal workings is not important for us now, with the ability to *recognize* solution to the search problem.

The discussion of the oracle without seeing how it works in practice is rather abstract and maybe even puzzling. It seems as if the solution is kept in a register in the oracle and it *knows* the answer. But then why bother with a search algorithm if there is an oracle already to consult? There is

a difference between *knowing* the solution and being able to *recognize* the solution when given. A simple example to this is the distinction between proof checking and finding a proof. The former problem is fairly easy as the proof is already given and all we need to do is to check if the proof is logically correct. The latter problem is actually very hard, and in fact, unsolvable in some cases. Finding a proof of a sentence, say in first-order logic, is known to be a hard problem. The former problem is Turing-computable, the latter is Turing-recognizable.

The quantum oracle, similar to that in the previous algorithms, is a unitary operator $O$ defined as

$$O : |x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle,$$

where $|x\rangle$ is the index register, such that $x \in \{0,1\}^n$, $\oplus$ denotes addition modulo 2 (i.e. bitwise XOR), and $|q\rangle$ is the oracle single qubit.

First let us try solving this problem by placing $|x\rangle$ into an equally weighted superposition of all possible strings and then evaluating the oracle $O$ on state

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle|0\rangle.$$

After applying the oracle, we get the state

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle.$$

Measuring the first register will give one of the $N$ strings with equal probability. Measuring the second register will give $|0\rangle$ with probability $\frac{N-1}{N}$, and give $|1\rangle$ with probability $\frac{1}{N}$ since there is only one solution $x$ such that $f(x) = 1$ by definition of the problem. If we get lucky enough to measure $|1\rangle$ on the second qubit, then the first qubit will have the correct answer since the first and second registers are entangled. However, it is highly improbable that we get so lucky. So we need something more innovative.

Grover's algorithm uses two tricks. First trick is called the *phase inversion*, which "marks" the key value by changing its phase. It works as follows: We set the oracle qubit in the state $\dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$ which is basically equal to $H|1\rangle$. The action of the oracle is thus

$$O : |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \rightarrow (-1)^{f(x)}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Now notice that $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ is an eigenvector, so we can omit this qubit and the action of the oracle may be written as

$$O : |x\rangle \rightarrow (-1)^{f(x)}|x\rangle.$$

Now if we look carefully, the oracle *marks* the key value by inverting its phase.

So how does the phase inversion act on states? First let us give a small illustration of what we mean by that. If $|x\rangle$ starts in an equal superposition of four different states, i.e. $[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]^T$, and assume that the key value is the computational basis state 10. After performing the phase change, the state will become $[\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}]^T$. Measuring $|x\rangle$ does not give enough information unfortunately. All states have equal probabilities. Changing the phase has no effect on the outcome. What we need is *amplifying* the amplitude of the key value.

The second trick of Grover's algorithm is called the *inversion about the mean*. As an example, consider a sequence of integers: $53, 38, 17, 23, 79$. The average of these numbers is $\mu = 42$. If we want to invert an element $\alpha$ around the mean $\mu$, we get $\alpha' = \mu + (\mu - \alpha)$ or simply

$$\alpha' = 2\mu - \alpha.$$

So given a general state

$$\sum_k \alpha_k |k\rangle,$$

the inversion about the mean will produce the state

$$\sum_k (2\mu - \alpha_k)|k\rangle,$$

where $\mu = \frac{1}{N}\sum_k \alpha_k$.

How do we define the inversion about the mean in terms of matrices then? Let us again first try to describe it by giving an example and then later define it more precisely. Consider the vector $V = [53, 38, 17, 23, 79]^T$.

Consider the matrix

$$A = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

It is easy to see that $A$ is a matrix that finds the average of the elements of the vector $V$.

$$AV = [42, 42, 42, 42, 42]^T$$

In terms of matrices, the formula $\alpha' = 2\mu - \alpha$ becomes

$$V' = 2AV - V = (2A - I)V.$$

In general, if we were to deal with $2^n$ possible states, then

$$A = \begin{bmatrix} \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \end{bmatrix}$$

So the operator $(2A - I)$ inverts the given vector about its mean. But how do we actually implement this? We will need to define an $n$-qubit operator $U_0^{\perp}$ acting as follows

$$U_0^{\perp} = \begin{cases} |x\rangle \to -|x\rangle, & x \neq 0 \\ |0\rangle \to |0\rangle \end{cases}$$

This operator applies a phase inversion to all $n$-qubit states that are orthogonal to the state $|0\rangle$. In terms of matrices

$$U_0^{\perp} = (2|0\rangle\langle 0| - I) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}$$

This operator inverts the phase of the states that are orthogonal to the state $|0\rangle$. But we are not done quite yet. We want to invert the phase of the states that are orthogonal to the general superposition of states.

Let us suppose that

$$|\psi\rangle = H|0\rangle = \frac{1}{\sqrt{N}} \sum_k |k\rangle.$$

What we can do is the following. We can first transform $|\psi\rangle$ into $|0 \dots 0\rangle$. Then we can do inversion about $|0 \dots 0\rangle$, and then we can transform $|0 \dots 0\rangle$ back into $|\psi\rangle$. For this, consider the action of the operator

$$HU_0^{\perp}H.$$

Now it is easy to observe that $HU_0^{\perp}H = (2|\psi\rangle\langle\psi| - I)$.[4] To see why this operator inverts about the mean, note that

$$HU_0^{\perp}H = H \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix} H$$

$$= H \left( \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} - I \right) H$$

$$= H \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} H - HIH$$

[4] By $H$, we of course mean $H^{\otimes n}$.

$$= \begin{bmatrix} \frac{2}{\sqrt{N}} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{\sqrt{N}} & 0 & \cdots & 0 \end{bmatrix} H - I$$

$$= \begin{bmatrix} \frac{2}{N} & \cdots & \frac{2}{N} \\ \vdots & \ddots & \vdots \\ \frac{2}{N} & \cdots & \frac{2}{N} \end{bmatrix} - I = \begin{bmatrix} \frac{2}{N} - 1 & & & \\ & \ddots & 2/\mathrm{N} & \\ & 2/\mathrm{N} & \ddots & \\ & & & \frac{2}{N} - 1 \end{bmatrix}.$$

Now why does this matrix invert a given vector about the mean? Let us take a look at what happens at the $x$th entry of the vector when given a state.

$$\begin{bmatrix} \frac{2}{N} - 1 & & & \\ & \ddots & 2/\mathrm{N} & \\ & & \ddots & \\ & 2/\mathrm{N} & & \frac{2}{N} - 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_x \\ \vdots \\ \alpha_{N-1} \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \frac{2}{N} \sum_{y=0}^{N-1} \alpha_y - \alpha_x \\ \vdots \\ \vdots \end{bmatrix}$$

Since $\frac{2}{N} \sum_y \alpha_y = 2\mu$, the $x$th entry of the resultant vector becomes $2\mu - \alpha_x$ which is exactly what we wanted to do. In other words, $HU_0^{\perp}H$ inverts the phase of the states that are orthogonal to $|\psi\rangle$. The set of states orthogonal to $|\psi\rangle$ is spanned by the states $H|x\rangle$ for $x \neq 0$. So for all $x \neq 0$,

$$(HU_0^{\perp}H)H|x\rangle = -H|x\rangle.$$

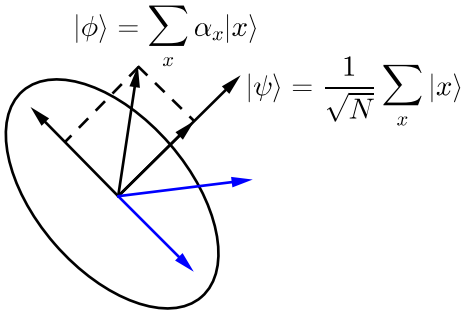



Figure 27: Reflection around $|\psi\rangle$ is actually a reflection around the mean. Given $|\phi\rangle$ to be inverted, we take the subspace of vectors that are orthogonal to $|\psi\rangle$, we take the orthogonal component of $|\phi\rangle$, invert it and we simply take the reflection of $|\phi\rangle$ in the direction of this inverted orthogonal component.

Now what happens if we use the phase inversion operator $O$ and the inversion about the mean together? Suppose that we prepared our qubits into an equal superposition $\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$.

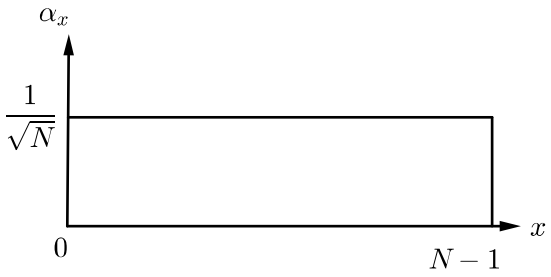



Figure 28: Putting states into an equal superposition.

If $x^*$ is the key value we are looking for, the phase inversion operator inverts the phase of $|x^*\rangle$.

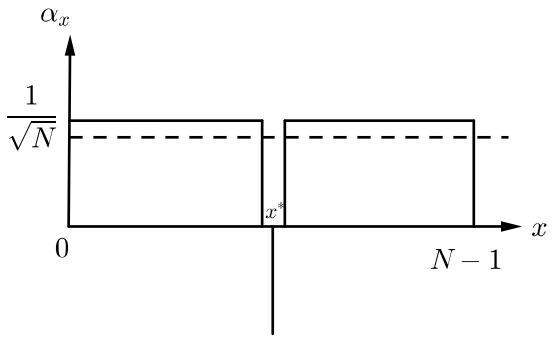



Figure 29: Phase inversion of the target value.

Since we invert the phase of $|x^*\rangle$, the average mean now drops down a little bit. What happens when we invert the phase of all the states around the mean? What happens is that the amplitude of $|x^*\rangle$ now goes up approximately to $\frac{3}{\sqrt{N}}$, while the amplitude of the non-key values goes down.

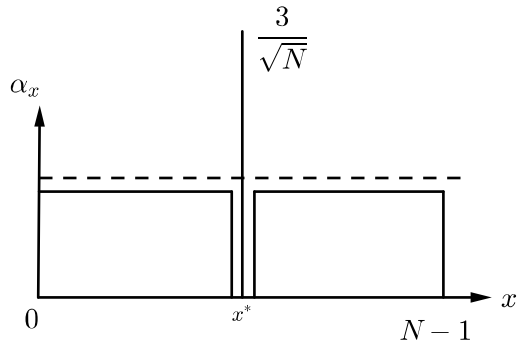



Figure 30: Inversion about the mean will amplify the the amplitude of the target value.

So if we iterate this process $O(\sqrt{N})$ times we get a sufficiently high probability of getting $|x^*\rangle$ when measuring the qubits. We call the iteration $HU_0^{\perp}HO$, *Grover's iteration.*

Hence, *Grover's algorithm* can be stated as follows.

**Step 1.** Start with a state $|\mathbf{0}\rangle$.
**Step 2.** Apply $H^{\otimes n}$.
**Step 3.** Repeat Grover's iteration $O(\sqrt{N})$ times. That is,

1. Apply the phase inversion operator $O$.
2. Apply the inversion about the mean operator $HU_0^{\perp}H$.

**Step 4.** Measure the qubits.

We might view this algorithm as follows.

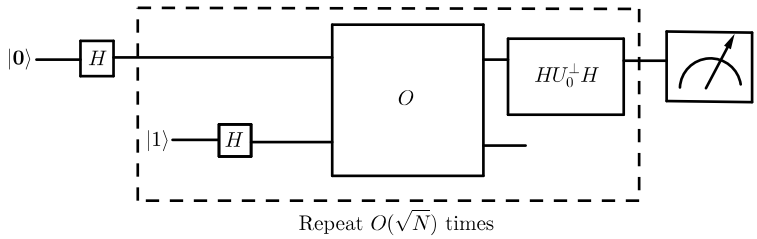



Figure 31: Diagram of Grover's algorithm.

There is a beautiful geometric interpretation of Grover's algorithm. In fact, Grover's iteration can be seen as a rotation in the two-dimensional space spanned by the starting vector $|\psi\rangle$ and the desired state $|x^*\rangle$. Actually, $|\psi\rangle$ and $|x^*\rangle$ are nearly orthogonal to each other for large $N$, but never entirely orthogonal since the inner product of $|\psi\rangle$ and $|x^*\rangle$ is $\frac{1}{\sqrt{N}}$. Let us define a state that is orthogonal to $|x^*\rangle$. Call this state $|x_{\perp}^*\rangle$.

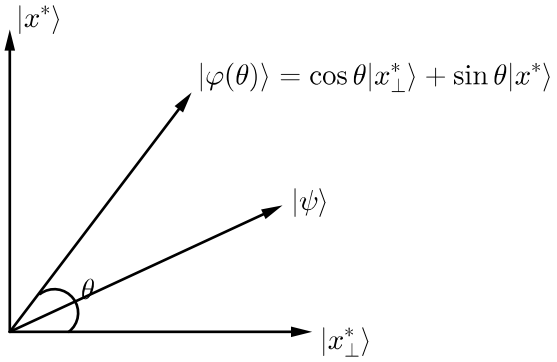



Figure 32: $|x^*\rangle$ denotes the solution space, $|x^*_\perp\rangle$ denotes the non-solution space, $|\psi\rangle$ is the initial equal superposition of states and $|\varphi(\theta)\rangle$ is the general trigonometric representation of any vectors lying on the plane spanned by $|x^*\rangle$ and $|x^*_\perp\rangle$.

Consider now a family of states all lying in this plane spanned by $|x^*\rangle$ and $|x^*_\perp\rangle$. The general form of a vector in this space has the form $\cos\theta|x^*_\perp\rangle + \sin\theta|x^*\rangle$, where $\theta$ is a real parameter. The desired state, namely $|x^*\rangle$, is in this parameterized family with $\theta = \frac{\pi}{2}$.

Before we start the Grover's iteration, we are in state $|\psi\rangle$. The state can also be written in the general trigonometric form with $\theta = \arcsin\frac{1}{\sqrt{N}}$. So what is the effect of one Grover iteration on $|\varphi(\theta)\rangle$? We begin with inverting the phase of the key state $|x^*\rangle$ which is just a reflection around $|x^*_\perp\rangle$ where we change the sign of the coefficient of $|x^*\rangle$. That is, $O|\varphi(\theta)\rangle = \cos\theta|x^*_\perp\rangle - \sin\theta|x^*\rangle$.

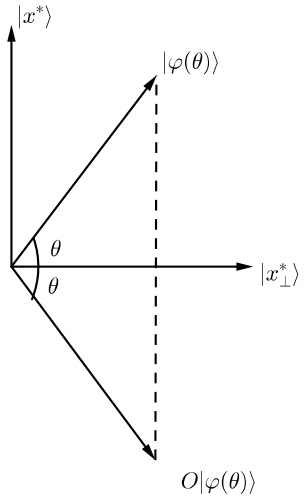



Figure 33: Geometric representation of phase inversion step.

Next we do $HU_0^\perp H$. As we noted earlier, this is a reflection around the state $|\psi\rangle$. The Grover iteration $HU_0^\perp HO$ is a product of two reflections which is a rotation twice the angle between $|\psi\rangle$ and $|x^*_\perp\rangle$, towards $|x^*\rangle$.

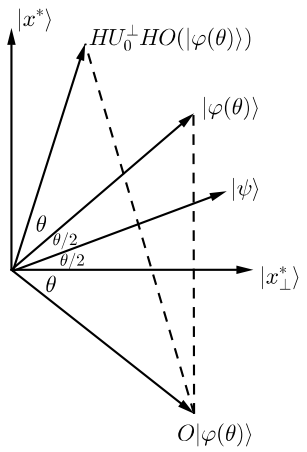



Figure 34: Inversion about the mean is just a reflection about $|\psi\rangle$. Combining phase inversion and the inversion about the mean operations, we end up with a rotation towards the target vector.

So each iteration rotates the state a little closer to the key state $|x^*\rangle$. But we have to be careful that if we over iterate the process, we pass the key state and get further away. So it is crucial to choose the right number of iteration, which is approximately $\sqrt{N}$. For $N$ item search problem with $M$ many key values, it turns out that we only need to apply the oracle $O\left(\sqrt{\frac{N}{M}}\right)$ times in order to obtain a solution on a quantum computer.

We should also note that the state does not hit the key state exactly but gets very near. Except for the special case when $N = 4$, where a single oracle call is enough to do the search with a correct answer with probability exactly 1. However, for the other cases, the output is a state that is not equal to but very close to the key state.

### 4.5 Shor's Factoring Algorithm

In this section, we introduce a polynomial time quantum algorithm for integer factorization. Given two integers, it is an easy task to multiply them together. It is however very hard in general to perform the inverse. That is, it is a harder problem to factorize a given integer $N$. The *Fundamental Theorem of Arithmetic* tells us that any natural number greater than 1 is either prime itself or is the product of prime numbers. Moreover, this product is unique.

There is an efficient algorithm, called Euclid's algorithm, for determining the greatest common divisor of two given numbers $a$ and $b$. Given two natural numbers $a$ and $b$, the function $GCD(a, b)$ does the following until $a = b$: If $a > b$ then define the new value of $a$ to be $a - b$, otherwise define the new value of $b$ to be $b - a$. If $a = b$, then stop and define $a$ to be the output of $GCD(a, b)$.

Integer factorization problem is to find the prime factorizations of a given integer $N$. Prime factorization of integers is commonly used in RSA cryptography. The reason why RSA cryptography systems use integer factorization is because there is no efficient classical algorithm known yet which factorizes large numbers. For the worst case scenario, let $N = pq$ for two large prime numbers. Given $N$, if we want to find its prime factors $p$ and $q$, we can do exhaustive search in the worse case. Since the input size is measured with two bits, the size of $N$ is approximately $w = \log_2 N$, i.e. size of the binary representation of $N$. Therefore, $O(N) = O(2^{\log_2 N}) = O(2^w)$. So the exhaustive search takes exponential time in the input size.

The problem of factorizing integers can be converted into the problem

of period finding using modular arithmetic. We say that $a \equiv x \mod N$ if $x$ is the remainder of $\frac{a}{N}$. The famous mathematician Leonard Euler discovered a very useful relationship between exponential functions and modular arithmetic. Let us examine the function $f(k) = 3^k$.

$3^1 = 3$
$3^2 = 9$
$3^3 = 27$
$3^4 = 81$
$3^5 = 243$
$3^6 = 729$
$3^7 = 2187$
$3^8 = 6561$

Mod10 of this function gives us
$3^1 \mod 10 = 3$
$3^2 \mod 10 = 9$
$3^3 \mod 10 = 7$
$3^4 \mod 10 = 1$
$3^5 \mod 10 = 3$
$3^6 \mod 10 = 9$
$3^7 \mod 10 = 7$
$3^8 \mod 10 = 1$
What we notice is a repeating sequence

$$3, 9, 7, 1, 3, 9, 7, 1 \ldots$$

Moreover the last element of the cycle is always 1. For a given function

$$f_{a,N}(x) = a^x \mod N,$$

as long as $a$ and $N$ are relatively prime, that is $GCD(a, N) = 1$, we have this cyclic property. The least number $r$ such that $a^r \mod N = 1$ is called the *period* of $f_{a,N}(x)$. The period of $f_{3,10}(x)$ is for instance 4, as it can be seen. Apparently integer factoriztion can be reduced to the problem of period finding.

The algorithm for integer factorization is as follows. Suppose that we are given $N = pq$ for two prime numbers $p$ and $q$. The aim is to find $p$ and $q$.

Step 1. Pick any integer $a < N$. See if $a$ and $N$ are relatively prime. If $GCD(a, N) = 1$, then the common factor must be a factor of $N$, so we found one of $p$ or $q$. In this case we can output the factors and we are done.

If $GCD(a, N) \neq 1$, continue with Step 2.

Step 2. Compute the period of $f_{a,N}(x) = a^x \mod N$ (This is a hard problem and it takes exponential time using classical algorithms). Call this period $r$. See if $r$ is even. If not, go to Step 1. If so, see if $a^{\frac{r}{2}} + 1 \equiv 0 \mod N$. If $a^{\frac{r}{2}} + 1 \equiv 0 \mod N$, then go to Step 1. If $a^{\frac{r}{2}} + 1 \not\equiv 0 \mod N$, then continue with Step 3.
(Note that $f_{a,N}(r+s) = f_{a,N}(s)$ as the sequence will simply repeat after the first stage the function outputs 1.)

Step 3. Rearrange with algebra. We know that $a^r \equiv 1 \mod N$. So $a^r - 1 \equiv 0 \mod N$. Then, there must exist some $k$ such that

$$a^r - 1 = k \cdot N$$

Since we assumed that $r$ was an even number,

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = k \cdot N = k \cdot p \cdot q \qquad (*)$$

Step 4. Solve $p$ and $q$ as follows.

Let $p = GCD(a^{\frac{r}{2}} - 1, N)$ and let
$q = GCD(a^{\frac{r}{2}} + 1, N)$.

So why does this algorithm work? In the equation $(*)$, $p$ must divide one of the factors on the left hand side of $(*)$ and $q$ must divide the other factor. However, they cannot divide the same factor since that factor would be divisible by $N$. But neither factor is divisible by $N$ since we assumed that $a^{\frac{r}{2}} + 1 \not\equiv 0 \mod N$ and we also assumed that $r$ is the least number such that $a^r \equiv 1 \mod N$.

Then $(a^{\frac{r}{2}} - 1) \not\equiv 0 \mod N$. Since $p$ and $q$ divides separate factors on the left hand side of $(*)$, we can assume $p$ divides $(a^{\frac{r}{2}} - 1)$, and $q$ divides $(a^{\frac{r}{2}} + 1)$.

**Example.** Let $N = 35$. Find the prime factors.

Choose $a = 8$. Since $GCD(8, 35) = 1$, we continue with the next step. The period of $f_{a,N}(x)$ is 4 since we observe that

Powers of 8: $8, 64, 512, 4096, \ldots$

$8^x \mod 35$: $8, 29, 22, 1, \ldots$

Now $r = 4$ is an even number. Next is to check if $8^{\frac{4}{2}} + 1 \not\equiv 0 \mod 35$. This clearly holds.

Therefore, $(8^2 - 1)(8^2 + 1) = k \cdot 35$ for some $k$.

So we let $p = GCD(63, 35) = 7$, and let $q = GCD(65, 35) = 5$.

The problem here is that period finding is an extremely difficult task for classical computers. It turns out very easy for quantum computers though. It is not known whether there exists a polynomial time classical algorithm for finding the period. On the other hand, Shor's algorithm can compute this in polynomial time. This is an exponential speed-up over the best known classical algorithms.

Shor's algorithm is really *the* quantum algorithm that attracted many scientists. At the heart of integer factorization is period finding. At the heart of period finding is phase estimation, which relies on quantum fourier transform (QFT).

Before explaning the quantum fourier transform we shall review $n$th roots of unity.

Recall that a complex number $c$ can be given in polar form $c = (\rho, \theta)$, where $\rho$ is the magnitude and $\theta$ is the phase. For unit length vectors, we simply take take $\rho = 1$. But suppose that we want to take the $n$th power of $c$. The $n$th power of $c$ is just

$$c^n = (\rho^n, n\theta).$$

Raising to the $n$th power is multiplying $n$ times. The figure below shows a complex number and its first, second and third powers.

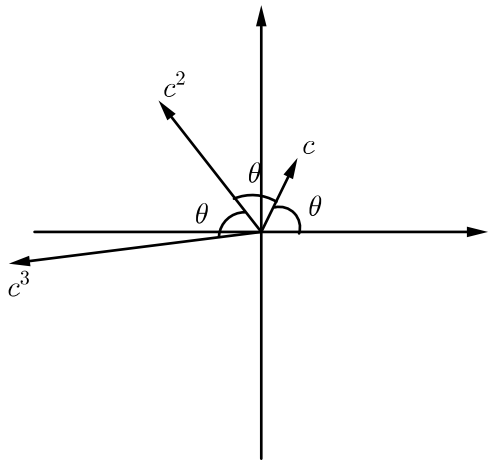



Figure 35: A complex number $c$ and its square and cube.

Similarly for multiplying two complex numbers. If we have two complex numbers defined on the unit circle as

$x = \cos\theta_1 + i\sin\theta_1 = e^{i\theta_1}$ and
$y = \cos\theta_2 + i\sin\theta_2 = e^{i\theta_2}$,

then $x \cdot y$ is

$x \cdot y = (\cos\theta_1 + i\sin\theta_1) + (\cos\theta_2 + i\sin\theta_2)$
$= \cos(\theta_1 + \theta_2) + i\sin(\theta_1 + \theta_2)$
$= e^{i(\theta_1+\theta_2)}$.

So multiplying complex numbers is just adding their phases. Let us look at the roots now. Given a complex number $c$ in the polar form, its $n$th root is

$$c^{\frac{1}{n}} = \left(\rho^{\frac{1}{n}}, \frac{1}{n}\theta\right).$$

There is more to this however. Recall that the phase is defined only up to multiples of $2\pi$. Therefore, we can rewrite the equation above as

$$c^{\frac{1}{n}} = \left(\rho^{\frac{1}{n}}, \frac{1}{n}(\theta + 2k\pi)\right).$$

It appears that there are several $n$th roots of the same number. In fact, there are exactly $n$ many $n$th roots for a complex number. Taking the last equation, how many different solutions can we generate by varying $k$?

$$\begin{array}{c|c} k=0 & \frac{1}{n}\theta \\ k=1 & \frac{1}{n}\theta + \frac{1}{n}2\pi \\ \vdots & \vdots \\ k=n-1 & \frac{1}{n}\theta + \frac{n-1}{n}2\pi \end{array}$$

When $k = n$, we obtain the first solution, when $k = n + 1$ we obtain the second solution. So we get a cyclic sequence. Let us assume that we are looking for the $n$th roots of unity. That is, we want to obtain all complex solutions to the equation

$$x^n = 1.$$

Again, it turns out that there are $n$ many $n$th roots of unity. Setting $c = (1, 0)$ as its magnitude and phase, we have

$$c^{\frac{1}{n}} = (1,0)^{\frac{1}{n}} = \left(1^{\frac{1}{n}}, \frac{1}{n}(0 + k2\pi) = \left(1, \frac{2k\pi}{n}\right)\right).$$

By permitting $k = 0, 1, 2, \ldots, n - 1$, we get exactly $n$ different roots of unity. Note that when $k = n$, we get back to the first root. This is called the *primitive* $n$th root of unity. The $k$th $n$th root of unity in exponential form is $e^{\frac{2\pi ik}{n}}$. We denote these $n$ different roots of unity bye

$$\omega^0 = 1, \omega^1, \omega^2, \ldots, \omega^{n-1}.$$

We show this by dividing a unit circle to $n$ equal slices each of which is has an angle $\dfrac{2\pi}{n}$. For example the 8th roots of unity can be viewed as in the figure below.

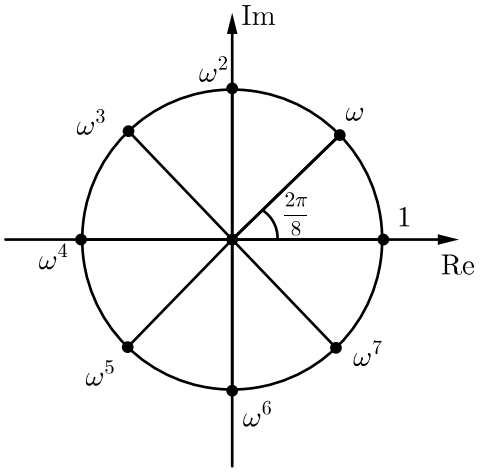



Figure 36: 8th roots of unity.

Note that

$$1 + \omega + \omega^2 + \ldots + \omega^{n-1} = 0.$$

since all vectors cancel each other. In general,

$$1 + \omega^j + \omega^{2j} + \ldots + \omega^{(n-1)j} = 0$$

provided that $j \neq 0$ since the expression is equal to the geometric series

$$\frac{\omega^{nj} - 1}{\omega^j - 1}.$$

As long as $j \neq 0$ the denominator is non-zero. But then since $\omega^n$ is equal to 1, the numerator becomes 0 so the sum of the geometric series is equal to 0 when $j \neq 0$. If $j = 0$, then clearly the sum is equal to $n$.

Another important thing to note about roots of unity is their conjugate. Note that conjugate of $\omega^j$ is $(\omega^j)^* = \omega^{n-j}$. Since $\omega^n = 1$, we have that

$$\omega^{n-j} = 1 \cdot \omega^{-j} = \omega^j.$$

So in the exponential notation, the $k$th $n$th root of unity is $\omega^k = e^{\frac{2\pi ik}{n}}$ and its conjugate is $(\omega^k)^* = e^{-\frac{2\pi ik}{n}}$.

### 4.5.1 Quantum Fourier Transform

The Fourier Transformation in classical computation is one of the most fundamental tools in science and applied mathematics. It maps time dependent functions to their frequency domains in such a way that a periodic function of period $r > 0$ is mapped to a function whose frequency amplitude is non-vanishing only at frequencies which are integer multiples of $1/r$. QFT is actually the quantum analogue of the discrete fourier transform in classical computation. Thq quantum fourier transform of dimension $N$, where $N$ is the power of 2, is defined as follows.

$$QFT_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$$

Then the matrix for the quantum fourier transform for an $N$-dimensional ($\log N$-qubit) quantum system has the $N$-th roots of unity as its entries. Recall that $\omega = e^{\frac{2\pi i}{N}}$ is the primitive $N$-th root of unity. In fact, there is an easy definition for the entries of the matrix. $QFT_N[j,k] = \omega^{jk \mod N}$. To understand what the matrix looks like, consider $QFT_8$ as an example.

$$QFT_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

Quantum Fourier transform is very related to the Hadamard transform. In fact, quantum Fourier transform is a generalization of the Hadamard transform. We leave the reader to write $QFT_2$ and compare it with the

Hadamard transform. It will be clear to see that the entries of the Hadamard transform concerns the 2nd roots of unity.

Let us give another example by defining $QFT_4$. What are the 4th roots of unity? In this case,

$\omega = e^{\frac{2\pi i}{4}} = e^{\frac{\pi i}{2}} = \cos\left(\frac{\pi}{2}\right) + i\sin\left(\frac{\pi}{2}\right) = i$
$\omega^2 = i^2 = -1$
$\omega^3 = i^3 = -i$
$\omega^4 = 1$

Then,

$$QFT_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

We might wonder how does this matrix act on a quantum state. It takes a general superposition state and outputs another superposition state.

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}.$$

Let us apply $QTF_4$ on the two qubit state $|10\rangle$.

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix} = \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle.$$

The classical discrete Fourier transform (DFT) takes $O(N^2)$ time for the fact that we need to make $N$ many multiplications and additions for $N$ many entries of the given input vector. The Fast Fourier Transform (FFT) algorithm gives the same output by performing $O(N \log N)$ steps which is nearly a quadratic improvement. When it comes to QTF, we have an even better improvement. Suppose for simplicity that $N = 2^n$ for some natural

number $n$. The input to QFT is a superposition of $n$-qubit states and it outputs another superposition of $n$-qubit states. It turns out that the complexity of the circuit for QFT can be made as small as $O(n^2)$. Considering that $n = \log N$, the complexity of QFT is $O(\log^2 N)$. This is an exponential improvement over the fast fourier transform algorithm. There is a catch though. Despite that the output of QFT is a superposition of states, we will not be able to access the superposition unfortunately, as quantum mechanics forbids us from measuring this enormous data. When we measure the qubits, we will just get to see an index $i$ with some probability for the basis state the superposition projected onto. We just see a *sample* of the output. Moreover, even though we have confined our attention to the case $M = 2^m$ for now, the algorithm can be implemented for arbitrary values of $M$, and can be summarized as follows:

1. Input: QFT takes as input a superposition of $m = \log M$ qubits $|\psi\rangle = \sum_{j=0}^{M-1} \alpha_j |j\rangle$.
2. Method: Using $O(m^2) = O(\log^2 M)$ quantum operations, perform the quantum Fourier transform to obtain the superposition $|\psi'\rangle = \sum_{j=0}^{M-1} \beta_j |j\rangle$.
3. Output: A random $m$-bit number $j$, where $0 \leq j \leq M-1$, with probability $|\beta_j|^2$.

**Properties of QFT.** We shall give two important properties of QFT. However, before explaning these properties we should first show that $QFT_M$ is a unitary operator. Note that there are exactly $M$ many $M$th roots of unity. Recall that $(\omega^j)^* = \omega^{-j}$. Note that $QFT_M$ is a symmetric matrix, so the adjoin of $QFT_M$ is simply

$$QFT_M^\dagger[j,k] = (\omega^{jk})^* = \omega^{-jk}.$$

**Claim 28.** $QFT_M$ is a unitary operation.

**Proof.** To show that $QFT_M$ is unitary let us multiply $QFT_M$ with $QFT_M^\dagger$.

$$(QFT_M \times QFT_M^\dagger)[j,k] = \frac{1}{M}\sum_{i=0}^{M-1} \omega^{ji}\omega^{-ik} = \sum_{i=0}^{M-1} \omega^{i(j-k)}.$$

If $j = k$, i.e. when we take the diagonal entries, this becomes

$$\frac{1}{M}\sum_{i=0}^{M-1} \omega^0 = \frac{1}{M}\sum_{i=0}^{M-1} 1 = 1.$$

If $j \neq k$, i.e. when we are off diagonal entries, we have a geometric series and the summation is equal to 0. This tells us that the resultant matrix has 1 on the diagonal entries and has 0 elsewhere. So $QFT_M \times QFT^\dagger = I$. Therefore, $QFT_M$ is a unitary matrix. □

Now we can look at the properties of the quantum Fourier transform. The first property is the *convolution multiplication* property. Suppose that we have an $N$-dimensional quantum state

$$|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$$

such that $\sum |\alpha_i|^2 = 1$. Suppose that we want to apply QFT to this state. The outcome will be another state

$$|\psi'\rangle = \sum_{i=0}^{N-1} \beta_i |i\rangle$$

Now suppose that we shift the input vector one row down as

$$\begin{bmatrix} \alpha_{N-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-2} \end{bmatrix}$$

Now if we apply QFT on this new vector, we have

$$QFT_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \alpha_{N-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-2} \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \omega \cdot \beta_1 \\ \omega^2 \cdot \beta_2 \\ \vdots \\ \omega^{(N-1)^2} \cdot \beta_{N-1} \end{bmatrix}.$$

The coefficients of the output vector are now multiples of the $N$-th roots of unity. But the complex roots are of unit length, so they do not change the probability of the outcome. The outcome will be the same as earlier. Shifting the input vector does not change the output probability distribution.

Second property of QFT is the way it treat periodic functions. Recall that a function $f$ is *periodic* if there exists some $r$ such that $f(x+r) = f(x)$. General behavior of QFT is that it transforms a given periodic superposition state of length $M$ with period $r$ into a superposition of the same length with period $\frac{M}{r}$.

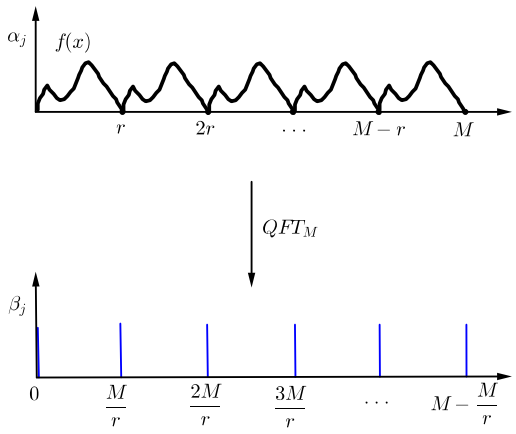
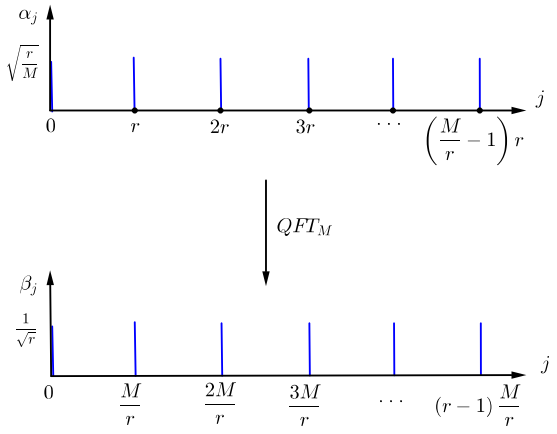



Figure 37: $QFT_M$ maps a perioid superposition with period $r$ to a periodic superposition of period $M/r$.

But let us suppose a special case. For simplicity we assume that the period $r$ divides $M$, dimension of the input vector which is of size at least $N^2$ where $N$ is the number we want to factor. Suppose that the input to the QFT, $|\psi\rangle = (\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$, is such that $\alpha_i = \alpha_j$ whenever $i \equiv j \mod r$, where $r$ is a particular integer that divides $M$. That is, the vector $|\psi\rangle$ consists of $M/r$ repetitions of some sequence $(\alpha_0, \alpha_1, \ldots, \alpha_{r-1})$ of length $r$. Moreover, suppose that exactly one of the $r$ numbers $\alpha_0, \alpha_1, \ldots, \alpha_{r1}$ is non-zero, say $\alpha_j$. Then we say that $|\psi\rangle$ is *periodic with period $r$ and offset $j$.*




If the input vector is an $M$-dimensional periodic superposition with period $r$, then using $QFT_M$ we can compute its period by the following fact we shall prove.

*QFT takes as input a superposition with period $r$ with some offset, for some $r$ that divides $M$. Then it outputs a superposition with period $M/r$ without an offset. The outcome of the measurement will be any of the $r$ multiples of $M/R$.*

Applying the QFT a few times and measuring different output indices, we can easily take the greatest common divisor of all the output values we have measured and with very high probability find the number $M/r$. From that, since we know $M$, we divide $M$ by $M/r$ and get $r$.

**Lemma 29.** Let $|\psi\rangle = (\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ be a periodic vector with period $r$ and with no offset, i.e. the only non-zero terms are $\alpha_0, \alpha_r, \alpha_{2r}, \ldots$. That is, let

$$|\psi\rangle = \sqrt{\frac{r}{M}} \sum_{j=0}^{M/r-1} |jr\rangle.$$

Then, the quantum Fourier transform of $|\psi\rangle$ is a periodic vector $|\psi'\rangle$ with period $\frac{M}{r}$ and with no offset. That is,

$$|\psi'\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \left|\frac{jM}{r}\right\rangle$$

**Proof.** The action of QFT on a given superposition can be described as

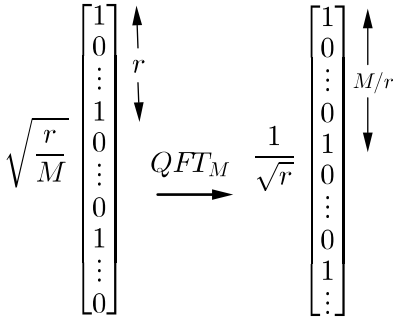



Figure 38: Action of $QFT_M$ on the amplitudes. Given a vector with only non-zero entries on the multiples of $r$, $QFT_M$ outputs a vector (superposition of states) whose only non-zero entries are the multiples of $M/r$.

The algorithm uses two registers. Suppose that $N$ is the integer we want to factorize. The first register stores the binary representation of an integer $M$ such that $M = N^2$. So it stores $m$-qubits where $m = \log_2 M$. The second register has at least $\log_2 N$ qubits for storing a number $\mod N$.

Suppose that $N$ is the number we want to factor. We take $M$ to be as large as $N^2$. The reason is the following. Here, $M > N^2$ values of the modular function $f(x)$ are computed in parallel. Since $r < N$, the period $r$ must manifest itself in the resulting sequence of $N^2$ function values now stored in the second register. So there can only be $r$ different function values. *Shor's algorithm* can be given as follows.

1. We start with initializing both registers to $|0\rangle$.

2. Apply Hadamard transform on both registers and obtain

$$\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle|0\rangle$$

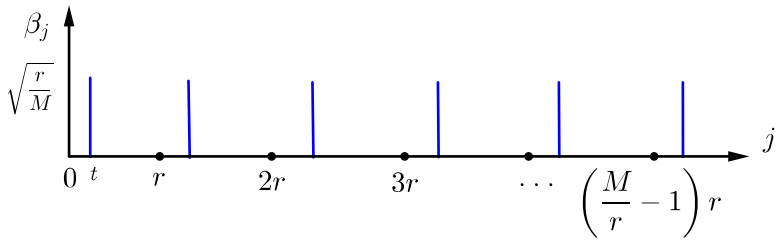
3. Next we apply a quantum gate $U_{f_a(x)}$ that implements the modular exponentiation function $f_a(x) = a^x \mod N$, where $a$ is randomly chosen

integer such that $GCD(a, N) = 1$. We get the state

$$\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |f(x)\rangle$$

A LITTLE EXPLANATION OF THE QUANTUM CIRCUIT FOR $U_{f_a(x)}$ COMES HERE..

4. We then measure the second register and obtain a random integer $a^x$ mod $N$. As soon as we measure it, since we are performing a partial measurement, everything in the superposition in the first register which is inconsistent with the measurement outcome will disappear. The only states left in the superposition, in the first register, are the integers $x$ whose $a^x$ mod $N$ is same as the measured value. Now the only non-zero values are multiples of $r$ plus some offset $t$ such that $t$ is a random integer between 0 and $r - 1$. For simplicity, we assume $r$ divides $M$. We shall investigate the general case later.




The state after measuring the second register is

$$\frac{1}{\sqrt{\frac{M}{r}}} \sum_{j=0}^{\frac{M}{r}-1} |jr + t\rangle$$

Notice that the distance between the non-zero values is exactly the period $r$, the value we are looking for. Can we get anywhere by measuring the first register? It's no good, because all we will get is a random point, with

no correlation across independent trials (because the offset $t$ is random). Here's what Shors algorithm does next. This will be the critical step in the algorithm.

5. Apply $QFT_M$ to the first register. The result of this step may not be immediately clear. So let us discuss what happens here.

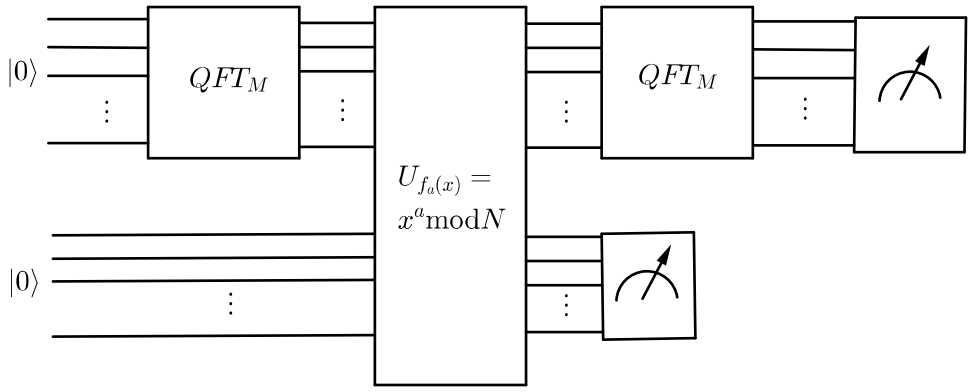



Figure 39: Circuit model of Shor's algorithm.

We may consider the action of $QFT_M$ as

$$\frac{1}{\sqrt{\frac{M}{r}}} \sum_{j=0}^{\frac{M}{r}-1} |jr+t\rangle \xrightarrow{QFT_M} \sum_{j=0}^{M-1} \beta_j |j\rangle$$

We have to figure out what $\beta_j$ looks like for every $j$. Let us examine the case when $j$ is a multiple of $\frac{M}{r}$. That is, we look at $\beta_{\frac{kM}{r}}$. It will get a contribution from every part of the input vector, the normalization factor of $QFT_M$ and a phase of the products of the $|jr\rangle$ and $\left|\frac{kM}{r}\right\rangle$.

$$
\begin{aligned}
\beta_{\frac{kM}{r}} &= \sum_{j=0}^{\frac{M}{r}-1} \frac{1}{\sqrt{\frac{M}{r}}} \cdot \frac{1}{\sqrt{M}} \cdot \omega^{jr \cdot \frac{kM}{r}} \\
&= \sum_{j=0}^{\frac{M}{r}-1} \sqrt{\frac{r}{M}} \cdot \frac{1}{\sqrt{M}} \cdot \omega^{jkM}. \text{ Since any multiple of } \omega^M \text{ is 1, we have} \\
&= \sum_{j=0}^{\frac{M}{r}-1} \sqrt{\frac{r}{M}} \cdot \frac{1}{\sqrt{M}} \cdot 1 \\
&= \frac{M}{r} \cdot \left( \sqrt{\frac{r}{M}} \cdot \frac{1}{\sqrt{M}} \right) \\
&= \frac{M}{r} \cdot \frac{\sqrt{r}}{M} = \frac{\sqrt{r}}{r} = \frac{1}{\sqrt{r}}
\end{aligned}
$$

Observe that $\beta_j = 0$ for $j \neq \frac{kM}{r}$. So we have a constructive interference at phase values $\beta_j$ for $j = \frac{kM}{r}$, i.e. multiples of $r$, and destructive interference at phase values $\beta_j$ for $j \neq \frac{kM}{r}$.

EXPLAIN THE PHASE SHIFTING A BIT MORE HERE...

6. Now that we have a superposition with period $\frac{kM}{r}$, i.e. multiples of $r$, we measure the first register and obtain a random multiple of $\frac{M}{r}$, say $\frac{kM}{r}$, where $k$ is a random number from 0 to $r-1$. It is easy to see that $GDC(k, \frac{M}{r}) = 1$. If so then computing $GDC(\frac{kM}{r}, M)$ will give us $\frac{M}{r}$. Since we know $M$, from $\frac{M}{r}$, we get $r$ by diving $M$ to $\frac{M}{r}$. Repeating the calculation $O(\log r) < O(\log N)$ times, one can amplify the success probability of finding $r$ to get as close to one as desired.

**Complexity analysis.** The best known classical algorithm for factorizing the prime numbers of a given integer is known as Field Sieve algorithm that works in time $2^{O(\sqrt[3]{\log n})}$. Let $n = \log N$ be the number of bits of the input integer $N$. The running time of the algorithm is dominated by the number of repetitions for steps given above, and the number of repetitions, as we noted, is $O(\log N) = n$. Since modular exponentiation takes $O(n^3)$ steps (see [**?**] Section 1.2.2), and the quantum Fourier transform takes $O(n^2)$ steps, the total running time of the algorithm is $O(n^3 \log n)$.

GENERAL CASE WILL COME HERE... WITH CONTINUED FRACTIONS.

**Phase estimation.** We shall now look at phase estimation. Let us first recall the action of Hadamard transform used in the previous algorithms. The Hadamard transform is used to get an information encoded in the relative phases of a state. The Hadamard gate is self-inverse so it does the opposite as well, i.e. encoding information into the phases. Let us consider the Hadamard transform $H$ on the basis state of one qubit. Recall that

$$\begin{aligned} H|x\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^x}{\sqrt{2}}|1\rangle \\ &= \frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}} (-1)^{xy}|y\rangle. \end{aligned}$$

We may think of the Hadamard transform as having encoded information about the value of $x$ into the phases between the basis states $|0\rangle$ and $|1\rangle$. Since the transform is self-inverse we have,

$$H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^x}{\sqrt{2}}|1\rangle\right) = |x\rangle.$$

Here, the Hadamard transform can be thought of as decoding the information about the value of $x$ that was encoded in the phases. The $n$-qubit Hadamard gate acting on $n$-qubit basis state $|\mathbf{x}\rangle$ gives the similar result as we saw earlier. That is,

$$H^{\otimes n}|\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y}\in\{\mathbf{0},\mathbf{1}\}^\mathbf{n}} (-1)^{\mathbf{x}\cdot\mathbf{y}}|\mathbf{y}\rangle.$$

Applying the same gate will give us back the original input state $|\mathbf{x}\rangle$. Unfortunately, $(-1)^{\mathbf{x}\cdot\mathbf{y}}$ are phases of a very particular form. In general, a phase is a complex number of the form $e^{2\pi i\omega}$, for any real number $\omega \in (0,1)$. The phase $(-1)$ actually corresponds to when $\omega = \frac{1}{2}$. The $n$-qubit Hadamard transformation is not able to fully access information that is encoded in more general ways. We should generalize this transformation to determine the information encoded in phases in the general form. Suppose that we are given a state

$$\frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \omega y} |y\rangle,$$

where $\omega \in (0, 1)$, and for simplicity suppose $N = 2^n$ for some natural number $n$.

### 4.5.2 Continued fractions

A *continued fraction* is an expression of the form

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \ddots}}}$$

where $a_i$ and $b_i$ are either rational numbers, real numbers or complex numbers. If $b_i = 1$ for all $i$, then the expression is called a *simple fraction*. If the expression contains a finite number of terms, it is called a *finite continued fraction*. If the expression contains an infinite number of terms, it is called an *infinite continued fraction*. For our purpose we will be working with finite continued fractions.

Continued fractions algorithm

# 5 Models of Computation and Complexity

Mathematical models of computation are abstract devices to study the limits of computability and computing power. As we shall see in the next chapter, although quantum algorithms can be constructed using circuit models, in this section we shall introduce another type of uniform approach which uses abstract models of computation. For this we take Turing machines as standard model of computation and define the quantum counterpart of this model.

First let us give some notions from formal language theory. An *alphabet* is a finite set of symbols. A *string* is a finite sequence of symbols. A *language* is a set of strings. Given two strings $w$ and $u$, $wu$ denotes the *concatenation* of $w$ and $u$. The *length* of a string $w$ is denoted by $|w|$. The *empty string* $\epsilon$ is the unique string of length 0. Given an alphabet $\Sigma$, $\Sigma^k$ denotes the set of strings of length $k$ over the alphabet $\Sigma$. We denote the set of all strings over $\Sigma$ by $\Sigma^*$ (this should not be confused with complex conjugate).

## 5.1 Turing machines

We now define an abstract model of computation called *Turing machine* which basically consists of a control unit having finitely many states and an infinite tape on which we write symbols, move around the tape cells, and carry the computation following the given transition rules. The formal definition is as follows.

**Definition 30.** A *(deterministic) Turing machine* $M$ is a 6-tuple

$$(Q, \Sigma, \delta, q_0, q_a, q_r)$$

such that $Q$ is a finite set of *states* where $q_0 \in Q$ is the *start state*, $q_a \in Q$ is an *accepting state* and $q_r \in Q$ is a *rejecting state*, $\Sigma$ is the alphabet, and $\delta$ is the *transition function* defined as

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, S, R\}$$

such that $L$ and $R$ denote one step *left (L)* , *right (R)*, $S$ denotes *stationary (S)* movement of the tape head.

The reason for calling the above model deterministic is due to the form of the transition function $\delta$. It is required that every pair $Q \times \Sigma$ is mapped to a unique triplet $Q \times \Sigma \times \{L, S, R\}$.

A *configuration* of a Turing machine is a triplet $(q, x, y)$, where $q \in Q$, $x, y \in \Sigma^*$. We interpret a configuration as follows. If $c = (q, x, y)$ is a configuration, we say that the Turing machine is in state $q$ and the tape contains the string $xy$. If the string $y$ begins with the symbol $a$, we say that the machine is *reading* the symbol $a$. To describe how the computation is carried out, we may write $x = w_1 a$ and $y = a_1 w_2$ such that $a, a_1 \in \Sigma$ and $w_1, w_2 \in \Sigma^*$. Therefore, $c$ can be written as $c = (q_1, w_1 a, a_1 w_2)$, and the transition function $\delta$ defines a transition rule from one configuration to another such that if $\delta(q_1, a_1) = (q_2, a_2, d)$ then a configuration $c = (q_1, w_1 a, a_1 w_2)$ is transformed to

$$c' = (q_2, w_1, a a_2 w_2), \quad c' = (q_2, w_1 a, a_2 w_2), \quad \text{or} \quad c' = (q_2, w_1 a a_2, w_2)$$

depending on if $d = L$, $d = S$ or $d = R$ respectively.

If $\delta$ defines a transition from $c$ to $c'$, then we say that $c$ *yields* $c'$ and we denote this by $c \vdash c'$. Every such yield defines a *computational step*. The two exceptional cases $c = (q_1, w, \epsilon)$ and $c = (q_1, \epsilon, w)$ can be handled by introducing the *blank symbol* $\#$ and so extending the definition of $\delta$ and replacing $(q_1, w, \epsilon)$ with $(q_1, w, \#)$, and replacing $(q_1, \epsilon, w)$ with $(q_1, \#, w)$.

A *computation* of a Turing machine with an input $w \in \Sigma^*$ is defined as a sequence of configurations $c_0, c_1, \ldots$ such that $c_0 = (q_0, \epsilon, w)$ and $c_i \vdash c_{i+1}$ for each $i$. We say that the computation *halts* if the state symbol of some configuration $c_i$ is either $q_a$ or $q_r$. In the former case, we say that the computation is *accepting*. Otherwise, we say that the computation is *rejecting*.

If a computation of a Turing machine $M$ starting with configuration $(q_0, \epsilon, w)$ ends up with a halting configuration $(q, w_1, w_2)$ in $t$ computational steps, we say that $T$ computes $w_1 w_2$ from the input $w$ in *time* $t$.

Since a Turing machine may not necessarily halt on a given input, it can also be seen as an *enumerator* classifying the set of strings accepted by the machine.

**Definition 31.** A language $L$ is called *recursively enumerable (computably enumerable)* if there is a Turing machine $T$ such that $T$ accepts $w$ if and only if $w \in L$. We say that $L$ is *recursive (computable)* such that each computation of $T$ halts, and $T$ accepts $w$ if and only if $w \in L$.

We denote the class of recursive languages by **R** and the class of recursively enumerable languages by **RE**. Clearly, by definition, $\mathbf{R} \subseteq \mathbf{RE}$. However, it is a known fact that $\mathbf{R} \neq \mathbf{RE}$.

It is widely accepted by the mathematical community that Turing machines capture the intuitive notion of effective computability. This belief is known as the *Church-Turing Thesis*. This makes it possible for us to use the intuitive algorithmic desrciption to write a solution for a problem instead of writing the complete and formal Turing mechanical description.

A *decision problem* is problem that has a yes or no answer. The decision problems in **R** are algorithmically *solvable*. All other problems are *unsolvable*. Recursively enumerable sets are also called *semi-decidable* as the Turing machine halts if there is a solution, but may not halt otherwise. We will not be interested in classifying unsolvable problems. We are more interested in the time complexity classification of solvable problems.

The set of problems that can be computed by Turing machines in polynomial time, i.e., in $n^k$ steps for some $k \in \mathbb{N}$ with respect to the given input of length $n$, is denoted by the class **P**.

A *non-deterministic Turing machine* is just like the deterministic counterpart except that the transition function in this case is allowed to map more than one triplets. That is for a non-deterministic Turing machine, we have that $\delta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{L, S, R\}$ which tells whether it is *possible* for a configuration $c$ to yield a configuration $c'$. So the fact that $(q_1, a_1, q_2, a_2, d) \in \delta$ means that, if the machine is in state $q_1$ reading the symbol $a_1$, it is possible that the machine will go into state $q_2$, write $a_2$ and move its tape head to the $d$ direction.

We denote the class of problems that are solvable by non-deterministic Turing machines in polynomial time by **NP**. Every deterministic Turing machine can simulate a non-deterministic Turing machines. So it is clear that $\mathbf{P} \subseteq \mathbf{NP}$. However, it is one of the millennial prize problems in theoretical computer science whether $\mathbf{P} \neq \mathbf{NP}$.

## 5.2 Probabilistic Turing machines

We now look at a new kind of Turing machine model which is not entirely deterministic or equally non-deterministic, but rather probabilistic.

**Definition 32.** A *probabilistic Turing machine* $M$ is a 6-tuple

$$(Q, \Sigma, \delta, q_0, q_a, q_r)$$

such that $Q$ is a finite set of *states* where $q_0 \in Q$ is the *start state*, $q_a \in Q$ is an *accepting state* and $q_r \in Q$ is a *rejecting state*, $\Sigma$ is the alphabet, and $\delta$ is the *transition function* with the requirement that for all $(q_1, a_1) \in Q \times \Sigma$

$$\sum_{(q_2,a_2,d)\in Q\times\Sigma\times\{L,S,R\}} \delta(q_1, a_1, q_2, a_2, d) = 1.$$

We may assume for convenience that the values of $\delta(q_1, a_1, q_2, a_2, d)$ are rational. If a configuration $c$ yields another configuration $c'$ with probability $p$, we denote it by $c \vdash_p c'$. Let $c_0, c_1, \ldots, c_t$ be a sequence of configurations such that $c_i \vdash_{p_{i+1}} c_{i+1}$ for each $i$. Then we say that $c_t$ is *computed* from $c_0$ in $t$ steps with probability $p_1 p_2 \cdots p_t$. If $p_1 p_2 \cdots p_t \neq 0$, we also say that $c_0 \vdash_{p_1} c_1 \vdash_{p_2} \ldots \vdash_{p_t} c_t$ is a *computation* of a probabilistic Turing machine. So unlike in a deterministic computation, a single configuration can now yield several different configurations with possibly different probabilities that add up to 1.

The notation

$$p_1[c_1] + p_2[c_2] + \ldots p_m[c_m]$$

such that $p_i \geq 0$ and $p_1 + \ldots + p_m = 1$ denotes that the system is in state $c_i$ with probability $p_i$. All computations can be expressed by the terms of a probabilistic system. Suppose that by the end of $t$ computational steps, a probabilistic Turing machines starting from an initial configuration computes configurations $c_1, \ldots c_m$ with probabilities $p_1, \ldots p_m$ such that $p_1 + \ldots p_m = 1$. We then say that the *total configuration* of a probabilistic machine at time $t$ is a probability distribution

$$p_1[c_1] + p_2[c_2] + \ldots p_m[c_m]$$

over the basis configuration $c_1, \ldots, c_m$. We can define a vector space, say the *configuration space*, that has all of the potential basis configurations as the basis vectors. A general configuration then can be considered as a linear combination in the configuration space with probability factors adding to 1.

## 5.3 Quantum Turing machines

We may want to generalize the notion of computability to the domain of quantum computers. Recall that the *Church-Turing Thesis* tells us that

any effectively computable function can also be computed by a classical computer. An extended version of this thesis would cover quantum models of computation

*Extended Church-Turing Thesis:* Any effectively computable function can also be computed by a quantum computer, i.e. a computer which is quantum mechanical.

We consider a straightforward generalization of a probabilistic Turing machine, replacing the probability factors with transition amplitudes. We define the *transition amplitude function* as

$$\delta : Q \times \Sigma \times Q \times \Sigma \times \{L, S, R\} \to \mathbb{C}$$

such that $\delta(q_1, a_1, q_2, a_2, d)$ gives the *amplitude* that whenever the machine is in state $q_1$ reading symbol $a_1$, it will write $a_2$, enter state $q_2$, and move the tape head in the direction of $d \in \{L, S, R\}$.

**Definition 33.** A *quantum Turing machine (QTM)* is a 6-tuple

$$(Q, \Sigma, \delta, q_0, q_a, q_r),$$

where $q_0, q_a, q_r \in Q$ are the initial, accepting, and rejecting states, respectively. The transition amplitude function satisfies that for any $(q_1, a_1) \in Q \times \Sigma$,

$$\sum_{(q_2,a_2,d)\in Q\times\Sigma\times\{L,S,R\}} |\delta(q_1, a_1, q_2, a_2, d)|^2 = 1.$$

As in any probabilistic computation, a general configuration of a quantum Turing machine is a linear combination

$$\alpha_1|c_1\rangle + \ldots + \alpha_n|c_n\rangle$$

of basis configurations. We take the basis configurations as an orthonormal basis of a Hilbert space, so we can see that the general configuration written above, which is a superposition, are merely unit length vectors in the configuration space. The transition amplitude function determines linear mapping $U_\delta$ in the state space which is required to be a unitary operator. It is known that the unitarity of $U_\delta$ can be determined by the local conditions (called *well-formedness conditions*) on the transition amplitude function. We will not give them here however.

Although QTMs are legitimate models of quantum computing, they are very impractical to define quantum algorithms. To define quantum algorithms we usually use the *circuit model* which will be introduced in the next chapter.

Let us discuss about some properties of QTMs. First thing to notice is that the transition function of QTM determines a unitary, hence reversible time evolution in the configuration space. Reversibility is not unique to QTMs. Standard Turing machines can be reversible as well.

Bennett [**?**] gave a reversible Turing machine model which uses a three-tape Turing machine with an *input tape*, *history tape*, and *output tape*. Reversibility is obtained by simulating the original machine on the input tape and at the same time writing the history of the computation on the history tape. Therefore, it is established that whatever is computable by a Turing machine is also computable by a reversible Turing machine, which can be seen as a quantum Turing machine as well. So QTMs are at least as powerful as standard Turing machines. What about the other way around? It is natural to ask whether or not standard Turing machines are as powerful as QTMs. The answer is positive for the same reason that standard Turing machines can simulate probabilistic Turing machines. However, it is very likely that QTMs cannot be efficiently simulated by standard Turing machines, not even by probabilistic ones. The reason why this is so is because of the algorithmic speedup in Shor's factoring algorithm. Although there is a polynomial time probabilistic quantum algorithm, no such polynomial time algorithm and not even a probabilistic algorithm has been found yet despite great efforts.

Another natural question to ask is whether or not there exists a *universal quantum Turing machine*. A Turing machine is *universal* if it can simulate the computation of any given Turing machine on a given input. Deutsch [**?**] proved the existence of a universal QTM which can simulate any other QTM with arbitrary precision.

We have the corresponding complexity classes for quantum Turing machines. The class **BQP** *(bounded error quantum polynomial time)* is the quantum counterpart of **BPP**. The class **EQP** *exact quantum polynomial time* is the quantum counterpart of **P** (and also **ZPP**). The quantum counterpart of **NP** is just denoted by **NQP**.

### 5.4 Complexity classes

There are a few options for classifying the time complexity of problems solvable by probabilistic Turing machines. We said that the class **NP** is the class of problems solvable by non-deterministic Turing machines in polynomial time. This is also the class of problems solvable by probabilistic Turing machines with non-zero probability in polynomial time.

The class **BPP** *(bounded error probability polynomial time)* denotes the set of problems solvable by probabilistic Turing machines with probability at least $\frac{2}{3}$.

The class **ZPP** *(zero error probability polynomial time)* denotes the set of problems solvable by probabilistic Turing machines with zero error.

As long as the probabilities are rational numbers (or at least computable numbers), we can always simulate probabilistic Turing machines with deterministic Turing machines by performing the possible transitions sequentially. The equivalency of the computational power of deterministic and probabilistic Turing machines entails an important fact that probabilistic Turing machines do not break the Church-Turing barrier. They compute the same class as that of deterministic Turing machines. Therefore, it is clear that we have the following relationships.

$$\mathbf{P} \subseteq \mathbf{ZPP}, \quad \text{and} \quad \mathbf{P} \subseteq \mathbf{BPP}.$$

Here are some basic relations between the classical, probabilistic and quantum complexity classes. The proof of these inclusions can be found in [**?**].

**Theorem 34.** $\mathbf{P} \subseteq \mathbf{EQP} \subseteq \mathbf{BQP}$.

**Theorem 35.** $\mathbf{BPP} \subseteq \mathbf{BQP}$.

**Theorem 36.** $\mathbf{BQP} \subseteq \mathbf{PSPACE}$.

# 6 Quantum Cryptology

# 7 Quantum Error Correction