1 . Hilbert Spaces

1.1 Linear Maps

Example 1
Below we define the *two-point butterfly* operation.

$$(a,b,w) \mapsto (a+wb, a-wb)$$

Show that this is a *linear* map from $\mathbb{C}^3 \to \mathbb{C}^2$.

Dual: The dual of a vector is a linear map that performs the inner product

---

Bilinear map: $B(u,v)$, fixing u gives a linear map $B_u(v)$ and fixing v gives a linear map $B_v(u)$

Bilinear maps will become useful later when we look at *tensor products*.

Example 2
Let $B(a1,a2)$ denote the determinant of the 2x2 matrix with columns a1 and a2. Prove that B is a bilinear map.

Example 3
Show that the *covariance* of two random variables $cov(X,Y)$ is a bilinear map.

1.2 Inner Products

Hermitian Inner Product:

.

This is often called the *complex inner product*.

$\langle u|\,|v\rangle$ is treated the same as $\langle u|v\rangle$

---

which is treated the same as $\langle u,v\rangle$. In this text, they *all* denote the Hermitian inner product.

One might think that $\langle u|\,|v\rangle$ is the multiplication of u with v. After all, we have a row vector next to a column vector. However this operation is not useful in Hilbert spaces. Instead we treat $\langle u|\,|v\rangle$ as $\langle u|v\rangle$. The complex inner product of two vectors is used far more and we will be seeing it often. The multiplication of u with v using normal vector dot products is not seen in Quantum Mechanics and none of our notations will ever denote this kind of product.

We join the two vertical lines of $\langle u|\,|v\rangle$ to form one vertical line. in $\langle u|v\rangle$. We will rarely write $\langle u|\,|v\rangle$ because it is slightly shorter to write $\langle u|v\rangle$.

2 . Qubits

2.1 Bloch Sphere

A circle has one degree of freedom, a sphere has *two degrees of freedom*

The *area* of a circle has two degrees of freedom. We can therefore find a mapping between points on the *surface* of the Bloch Sphere and points within an *area* of two dimensional unit circle.

Every point on

Sphere Equation: $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + e^{i\phi}\sin(\frac{\theta}{2})|1\rangle$

The $\frac{\theta}{2}$ converts the *angle associcated with perpendicularity*. $|0\rangle$ and $|1\rangle$ are always orthogonal. Geometrically, we are used to using 90 degrees but, *on the block sphere*, we use 180 degrees to express perpendicularity. It turns out that in four dimensional space, 180 degrees is gives a more natural angle for perpendicularity. The same idea comes up when dealing with *Quaternions*. Quaternions are a useful mathematical abstraction for expressing rotations using a four dimensional space.

Example 4
Plot the following qubit on the Bloch Sphere $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$

We have $\cos(\frac{\theta}{2}) = \sin(\frac{\theta}{2})$ which only occurs when $\frac{\theta}{2} = 45$. Hence $\theta = 90$. And solving $e^{i\phi} = i$ gives $\phi = 90$.

Example 5
Plot the following qubit on the Bloch Sphere $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$

We have $\cos(\frac{\theta}{2}) = \sin(\frac{\theta}{2})$ which only occurs when $\frac{\theta}{2} = 45$. Hence $\theta = 90$. And solving $e^{i\phi} = -i$ gives $\phi = 270$.

2.2 Density Matrices

Previously, in order to find a point on the Bloch sphere, we had to solve simulatenous equations to find angles θ and ϕ. We look at a new way to find points on a Bloch sphere by introducing the *Density Matrix* of a state, and the concept of a *Bloch vector*.

$$\rho = \tfrac{1}{2}(I + r_x X + r_y Y + r_z Z)$$

where X, Y, Z are the Pauli matrices.

The vector (r_x, r_y, r_z) is called the Bloch vector. The elements are real numbers and locate a qubit on the Bloch sphere. They can be determined from the density matrix of a state using: $r_x = 2Re[\rho_{10}]$ $r_y = 2Im[\rho_{10}]$ $r_z = \rho_{00} - \rho_{11}$

Example 6
Locate on the Bloch sphere, the location of the qubit $\frac{1}{2}(1+i, 1-i)$ by determining its density matrix.

Rather than finding ρ, we can just directly use the following equations

$r_x = 2Re[\alpha^*\beta]$ $r_y = 2Im[\alpha^*\beta]$ $r_z = |\alpha|^2 - |\beta|^2$

2.3 Measurement

Albert Einstein once said:

> I think that matter must have a separate reality indepenent of the measurements. That is an electron has spin, location and so forth even when it is not being measured, I like to think that the moon is there even if I am not looking at it.

A quantum state is in a *superposition state* when one is not looking at it. And when we look at the state or *measure* does a different system form with physical quantities of interest.

Erwin Schrödinger's cat: $|\text{cat}\rangle = \frac{1}{\sqrt{2}}(|\text{dead}\rangle + |\text{alive}\rangle)$

3 . Quantum Circuits

3.1 Introducing Tensor Products

Example 7
Show that the tensor product distributes over vector addition.

This is a special case of the bilinearity property with the scalars set to 1 and $v' = 0$.

3.2 Matrix Tensor Products

Example 8
An *outer product* is an intersting case where normal *matrix multiplication* gives the same operation the same as a tensor product. An *outer product* is normal matrix multiplication of a column vector on the left multiplying a row vector on the right.
(a) Determine $(4,2,7)(1\ 2\ 6)$ and $(1,2,6) \otimes (4,2,7)$
(b) Show that $v_1 v_2 = v_1 \otimes v_2$ for any two vectors in $\mathbb{C}^2$
(c) Prove that $v_1 v_2 = v_1 \otimes v_2$ for general vectors.

Multiplication Property: $(M \otimes N)(M' \otimes N') = (MM') \otimes (NN')$

Using the Dirac bra-ket nototation, the previous result can be written as: $|\psi_1\rangle \langle\psi_2| = |\psi_1\rangle \otimes \langle\psi_2|$

3.3 Composite Systems

$n =$ number of qubits, $N =$ number of basis vectors

Example 9
For one system with state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and second system with state $\frac{1}{\sqrt{2}}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ find the state of the *composite system*.

$$\boxed{\frac{1}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{1}{2\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|11\rangle}$$

The fact that we combined together *more than one* tensor product and use *different levels of nesting* causes us to lose the top-level factorisation.

Example 10
Prove that the following two-qubit system *cannot be factorised* into two qubits.

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

Assume there is a factorisation. We can therefore express it in the following form.

$$\begin{aligned} |\psi\rangle = \tfrac{1}{\sqrt{2}}|00\rangle + \tfrac{1}{\sqrt{2}}|11\rangle &= (\alpha_0|0\rangle + \alpha_1|1\rangle)(\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \end{aligned}$$

Comparing coefficients we have the following:
$\alpha_0\beta_0 = \frac{1}{\sqrt{2}}$, $\alpha_0\beta_1 = 0$, $\alpha_1\beta_0 = 0$ and $\alpha_1\beta_1 = \frac{1}{\sqrt{2}}$.

If we multiply eqn1 and eqn4 we get $\alpha_0\alpha_1\beta_0\beta_1 = \frac{1}{2}$. If we multiply eqn2 and eqn3 we get $\alpha_0\alpha_1\beta_0\beta_1 = 0$. And yet, $0 \neq \frac{1}{2}$. We have a contradiction and so the system has no factorisation.

Example 11
For a general two-qubit state, $|\psi\rangle$, define the terms:
(a) *separable state* **(b)** *entangled state*

separable state: $\|\psi\rangle$ can be written as $\|x\rangle\|y\rangle$ entangled state: $\|\psi\rangle$ *cannot* written as such

Asking ourselves whether a state is entangled is as simple as algebraic exercise of proving whether or not the factorisation exists.

Entangled State cannot be written as a product state

However entanglement has a much richer meaning when we take this the definition and apply it at the atomic level to discover important physical consequences. In particular the discussion of *Principle of Locality* and *Local Realism*. In Quantum Computing

entanglement is not seen as a *problem* but rather as a useful *resource*

Later we will look at *Quantum Encryption* where one can find out if somebody is evesdropping on a transmission via entanglement. Classical channels *cannot* do this. This gives a much more reliable test over how secure the channel is.

3.4 Measurement Revisited

Example 12
For three-qubit state, $|\psi\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle)$ the first qubit is measured. Write down the possible outcomes, and whether they are separable or entangled.

separable state: $\|\psi\rangle$ can be written as $\|x\rangle\|y\rangle$ entangled state: $\|\psi\rangle$ *cannot* written as such

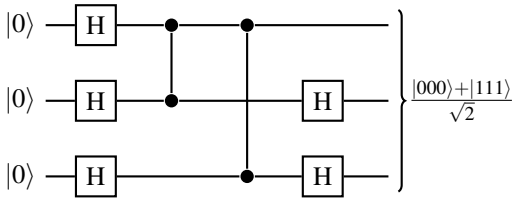


Figure 3.1: A quantum circuit for producing a GHZ state using Hadamard gates and controlled phase gates.

General Measurement Principle: $\sum_{j=1}^{k} M_j^H M_j = I$

3.5 Introducing Quantum Circuits

3.6 Universal Sets

Recall from Computational Techniques, the definition of the *Euclidean Norm*.

Every unitary operator U, on the space $\mathbb{C}^n$ can be represented by a product of $C(n,2)$ *two-level unitary matrices*

Every unitary matrix has at most $C(n,2)$ degrees of freedom. This is how many elements are in upper triangle plus the diagonal. So the idea is to decompose the matrix into the product of $C(n,2)$ matrices where each degree of freedom turns into the multiplication of rank one matrix.

Example 13
Find a quantum circuit that uses single qubit transforms and CNOTs to implement the transform on three qubits.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & b & 0 & 0 & 0 & 0 & d \end{bmatrix}$$

3.7 Reversible Computation

Reversible Computation: *Every* quantum circuit is *reversible*

3.8 Implementing Classical Circuits

Is there a method for converting classical algorithms into quantum algorithms? A classical algorithm or *classical circuit* is $f : \{0,1\}^n \to \{0,1\}$ Thinking of classical algorithms as being simply a function is perfectly acceptable according to the *Church-Turing Thesis*, a thesis which gives suggests that Turing machines form the definition for what an algorithm actually is.

Quantum computers attack the Church-Turing thesis (any algorithmic process can be efciently simulated using a Turing machine).

A quantum computer can implement mimic the possibly *several* classical steps for computing f by using just *one unitary operator*

In principle, quantum computation is able to subsume classical computation.

Classical circuits are, in general, *irreversible*

. There are classical models of computation that can be reversed, but we will ignore them.

This is a consequence of the *pidgeon-hole principle*. We have a surjective function that is not an injection. So there is no chance that f is invertible. Although, NOT gate is an example of an invertible (reversible) gate, there are many classical gates such as AND, XOR, NAND that are *irreverisble* (non-invertible).

We *can* implement classical circuits with quantum circuits, but quantum gates are always unitary and therefore reversible. The result is that we have *ancilla bits* and *garbage bits*. The ancilla and garbage are additional bits that are only important for making the computation reversible.

Example 14
A quantum circuit consisting of a *controlled swap* can be used to implement the classical circuit consisting of an AND gate.

Suppose use inputs a, b and 0. Here 0 is an *ancilla bit*. We want to know the outputs from this circuit. To do this, let's draw a truth table for all the possible input values.

Hence The outputs from the circuit are ..., ... and .....

$\boxed{a,\ \neg a \wedge b \text{ and } a \wedge b}$

The third input qubit is an *ancilla bit* and the second output bit is a *garbage bit*.

We want to remove $|junk(x)\rangle$ so that *interference isn't prevented*

We erase the junk (replace with zero), replace the output $f(x)$ with the input x and add $y \oplus f(x)$.

There is no junk associated with the input, $|junk(x)\rangle$. Instead with have $y \oplus f(x)$. Now interference isn't prevented.

---

4 . Quantum Algorithms I

4.1 Parallel Hadamard Gates

Parity Inner Product: $[\mathsf{x},\ \mathsf{y}] = \bigoplus_{i=1}^{n} \mathsf{x}_i \wedge \mathsf{y}_i$

In other words, we component-wise, take the AND operation, and then XOR these together.

$$[\mathsf{x},\ \mathsf{y}] = (x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \oplus \cdots \oplus (x_n \wedge y_n)$$

We call this an *inner product* because we are taking the inner product of two vectors and outputting a number. Here our vectors our strings of binary digits and the number outputting is $\{0,1\}$. In fact, we will show soon that this inner product behaves quite similarly with the inner product used for Euclidean spaces (dot product) and the inner product used for Hilbert spaces (Hermitian inner product) .

We call this the *parity* inner product because we will soon that this product outputs one if and only if the *product string* has an odd number of ones. By *product string*, we mean the the string that results from doing a component-wise AND operation.

Example 15
Determine $[0011,\ 0101]$

0

This may seem like a useless way to write the Hadamard operation but it was done in preparation for understanding how *parallel* Hadamard gates operate.

$$H|x\rangle = \tfrac{1}{\sqrt{2}}[(-1)^{[x,0]}|0\rangle + (-1)^{[x,1]}|1\rangle]$$

Example 16
Show that $(-1)^{[u,v]}|i\rangle \otimes (-1)^{[x,y]}|j\rangle = (-1)^{[ux,vy]}|ij\rangle$ where u,v,x,y,i,j are bits.

$(-1)^{u\wedge v}(-1)^{x\wedge y}|i\rangle|j\rangle =$
$(-1)^{u\wedge v+x\wedge y}|ij\rangle =$
$(-1)^{[u\wedge v+x\wedge y]\bmod 2}|ij\rangle =$
$(-1)^{u\wedge v\oplus x\wedge y}|ij\rangle = (-1)^{[ux,vy]}|ij\rangle$

Example 17
Show that $(-1)^{[ux,vy]}|ij\rangle \otimes (-1)^{[s,t]}|k\rangle = (-1)^{[uxs,vyt]}|ijk\rangle$

After doing the previous examples, it's now not too hard to what happens in the general case.

$(-1)^{x_1 \wedge y_1} |i_1\rangle \otimes (-1)^{x_2 \wedge y_2} |i_2\rangle \otimes \cdots \otimes (-1)^{x_n \wedge y_n} |i_n\rangle =$

$(-1)^{x_1 \wedge y_1 + x_2 \wedge y_2 + \cdots + x_n \wedge y_n} |i_1 i_2 \cdots i_n\rangle =$

$(-1)^{[x_1 \wedge y_1 + x_2 \wedge y_2 + \cdots + x_n \wedge y_n] \bmod 2} |i_1 i_2 \cdots i_n\rangle$

$(-1)^{(x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \oplus \cdots \oplus (x_n \wedge y_n)} |i_1 i_2 \cdots i_n\rangle =$

$(-1)^{[x_1 x_2 \cdots x_n,\, y_1 y_2 \cdots y_n]} |i_1 i_2 \cdots i_n\rangle$

We will use this next.

Example 18
Determine $H^{\otimes n} |x\rangle$.

$H^{\otimes n} |x\rangle$

$= \frac{1}{\sqrt{2}} \left[(-1)^{x_1 \wedge 0} |0\rangle + (-1)^{x_1 \wedge 1} |1\rangle\right] \otimes \frac{1}{\sqrt{2}} \left[(-1)^{x_2 \wedge 0} |0\rangle + (-1)^{x_2 \wedge 1} |1\rangle\right] \otimes \cdots$
$\otimes \frac{1}{\sqrt{2}} \left[(-1)^{x_n \wedge 0} |0\rangle + (-1)^{x_n \wedge 1} |1\rangle\right]$

$= \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} \frac{1}{\sqrt{2}} [(-1)^{x_1 \wedge y_1} |y_1\rangle] \otimes \frac{1}{\sqrt{2}} [(-1)^{x_2 \wedge y_2} |y_2\rangle] \otimes \cdots \otimes \frac{1}{\sqrt{2}} [(-1)^{x_n \wedge y_n} |y_n\rangle]$

$= \frac{1}{\sqrt{2^n}} \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} (-1)^{x_1 \wedge y_1 + x_2 \wedge y_2 + \cdots + x_n \wedge y_n} |y_1 y_2 \cdots y_n\rangle$

$= \frac{1}{\sqrt{2^n}} \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} (-1)^{[x_1 \wedge y_1 + x_2 \wedge y_2 + \cdots + x_n \wedge y_n] \bmod 2} |y_1 y_2 \cdots y_n\rangle$

$= \frac{1}{\sqrt{2^n}} \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} (-1)^{(x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \oplus \cdots \oplus (x_n \wedge y_n)} |y_1 y_2 \cdots y_n\rangle$

$= \frac{1}{\sqrt{2^n}} \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} (-1)^{[x_1 x_2 \cdots x_n,\, y_1 y_2 \cdots y_n]} |y_1 y_2 \cdots y_n\rangle$

Parallel Hadamard: $H^{\otimes n} |\mathsf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathsf{y} \in \{0,1\}^n} (-1)^{[\mathsf{x},\, \mathsf{y}]} |\mathsf{y}\rangle$

This is known as *Fourier Samping*.

$$h_{(i,j)} = \frac{1}{\sqrt{N}} (-1)^{[i,j]}$$

---

where $h_{(i,j)}$ denotes the element of $H^{\otimes n}$ at row i and column j, where we number row/columns from 0 to $N-1$.

Example 19
Write out explicitly $H^{\otimes 3}$

Example 20
Write out explicitly the 7th column of $H^{\otimes 4}$

Example 21

Prove by induction over N that $h_{(i,j)} = \frac{1}{\sqrt{N}}(-1)^{[i,j]}$

Example 22
Using the previous result, what is $H^{\otimes n}|00..0\rangle$?

We insert $\mathsf{x} = 00..0$ into the Parallel Hadamard equation to get:
$H^{\otimes n}|00..0\rangle = \frac{1}{\sqrt{2^n}}\sum_{\mathsf{y}\in\{0,1\}^n}(-1)^{[00..0,\,\mathsf{y}]}|\mathsf{y}\rangle = \frac{1}{\sqrt{2^n}}\sum_{\mathsf{y}\in\{0,1\}^n}|\mathsf{y}\rangle$

This is a special case of Fourier Sampling where here, instead of using a general $|\mathsf{x}\rangle$, we are using $\mathsf{x} = 00..0$. Notice how this gives a superposition of all 2^n computational basis vectors for an n qubit system.

An n-qubit **Uniform Superposition** is given by $H^{\otimes n}|00..0\rangle$

In other words, by connecting n Hadamards in parallel each with input $|0\rangle$.

What happens if we give $H^{\otimes n}$ a string that is not all zeros. Well, we actually still get an n-qubit superposition, but there will be some *minus signs*.

Example 23
Given that H is its own inverse, prove that $H^{\otimes n}$ is its own inverse.

Because $H^{\otimes n}$ is it's own inverse, we have the following result.

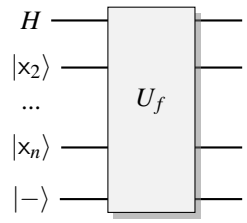
Undo Superposition: $H^{\otimes n}(\frac{1}{\sqrt{2^n}}\sum_{\mathsf{y}\in\{0,1\}^n}(-1)^{[\mathsf{x},\,\mathsf{y}]}|\mathsf{y}\rangle) = |\mathsf{x}\rangle$

This is a special case where the superposition is undone and $|\mathsf{x}\rangle$ is recoverd. It is not always the case that taking the Hadamard of a superposition performs this undoing action. It just so happened to work in this case because the superposition had a special structure having $(-1)^{[\mathsf{x},\,\mathsf{y}]}$ and using all the computational basis vectors, $\mathsf{y}\in\{0,1\}^n$.

We can consider a more general case of taking $H^{\otimes n}$ for a *general* superposition of $\sum_x \alpha_x H^{\otimes n}|x\rangle$. The coefficients of this superposition are a general α_x and we do not have to use the n computational basis vectors.

Hadamard of Superposition: $H^{\otimes n}(\sum_x \alpha_x|x\rangle) = \sum_x\sum_{y\in\{0,1\}^n}\alpha_x(-1)^{[x,y]}|y\rangle$

Example 24
By substituting the superposition $\sum_x \alpha_x|x\rangle = \sum_{\mathsf{x}\in\{0,1\}^n}(-1)^{[\mathsf{x},\,\mathsf{z}]}|\mathsf{x}\rangle$ into the formula above, show that the result is $|\mathsf{z}\rangle$



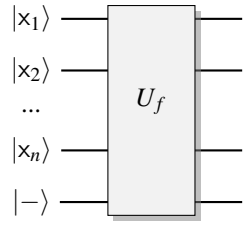
4.2 Bernstein-Vazirani Algorithm

The *parity problem*.

Parity Problem: Input a function $f : \{0,1\}^n \to \{0,1\}$ where $f(x) = [\mathsf{u}, \mathsf{x}]$ for some hidden $\mathsf{u} \in \{0,1\}^n$, output u

The **Bernstein-Vazirani Algorithm** solves the *Parity Problem*

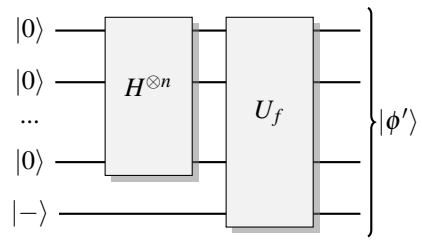
The definition of f is that it acts as a *parity mask* (similar to a bit mask except that we are outputting parities rather than bits). $f(x) = 0$ only when x We want to use as few queries to f as possible.



$$U_f |\mathsf{x}\rangle |-\rangle = (-1)^{f(\mathsf{x})} |\mathsf{x}\rangle |-\rangle$$

The output is the same except that whenever $f(\mathsf{x}) = 1$, the output picks up a relative phase of minus one.

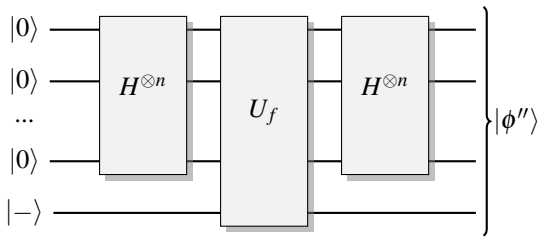
Now suppose we add a parallel Hadamard before U_f. Instead of inputting $|x\rangle$, we input $H^{\otimes n} |00..0\rangle$. Recall from our previous work on parallel Hadamard gates that: $H^{\otimes n} |00..0\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathsf{y} \in \{0,1\}^n} |\mathsf{y}\rangle$



The output is now the following.

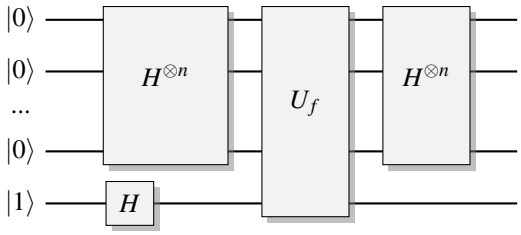
$$
\begin{aligned}
|\phi'\rangle &= U_f[(H^{\otimes n}\otimes I)(|0^n\rangle)\otimes|-\rangle)] && \text{from circuit}\\
&= U_f[(H^{\otimes n}|0^n\rangle)\otimes(I|-\rangle)] && \text{multiplication property}\\
&= U_f[H^{\otimes n}|0^n\rangle|-\rangle] && \text{identity matrix}\\
&= U_f[(\tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}|x\rangle)|-\rangle] && \text{Hadamard superposition}\\
&= U_f[\tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}|x\rangle|-\rangle] && \text{distributivity of } \otimes \text{ over } +\\
&= \tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}U_f|x\rangle|-\rangle && \text{linearity of } U_f\\
&= \tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle|-\rangle && U_f \text{ with } |x\rangle|-\rangle \text{ result}\\
&= \tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}(-1)^{[u,\,x]}|x\rangle|-\rangle && \text{def of } f(x)\\
&= (\tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}(-1)^{[u,\,x]}|x\rangle)|-\rangle && \text{distributivity of } \otimes \text{ over } +
\end{aligned}
$$

Recall that $H^{\otimes n}$ is its own inverse. Now if we feed the first n qubits through a parallel Hadamard then we undo the superposition and get u. We simply measure this and we have our output.



$$
\begin{aligned}
|\phi''\rangle &= (H^{\otimes n}\otimes I)([\tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}(-1)^{[u,\,x]}|x\rangle]\otimes|-\rangle) && \text{from circuit}\\
&= (H^{\otimes n}[\tfrac{1}{\sqrt{2^n}}\textstyle\sum_{x\in\{0,1\}^n}(-1)^{[u,\,x]}|x\rangle])(I|-\rangle) && \text{multiplication property}\\
&= |u\rangle\,(I|-\rangle) && \text{undo superposition}
\end{aligned}
$$

We can add some final touches by using $H|1\rangle = |-\rangle$ and adding the measurement to the circuit. Hence our final circuit for the Bernstein-Vazirani Algorithm is the following.

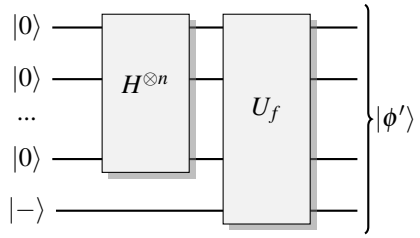


4.3 Deutsch-Jorza Algorithm

Balanced Problem: Can we determine whether a function $f:\{0,1\}^n\to\{0,1\}$ is *balanced or not* using U_f only once in a quantum circuit

The **Deutsch-Jorza Algorithm** solves the *Balanced Problem*

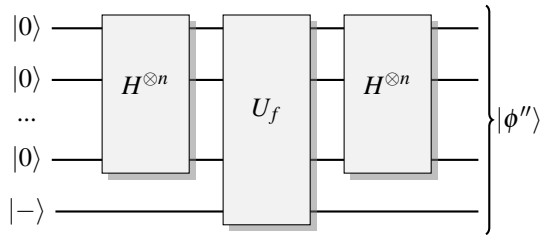
In turns out that the Deutsch-Jorza Algorithm uses the *same circuit* as the Bernstein-Vazirani Algorithm. But notice that now, we don't have $f(x) = [u,\ x]$ We instead assuming that $f(x)$ is either *balanced* or *constant*. We no longer assume it's a parity function as was the case in the parity problem. There is no longer a u to output, so the *last part* of the Bernstein-Vazirani Algorithm will be slightly different. We will deviate from the last part of Bernstein-Vazirani Algorithm derivation to, instead, create a circuit that solves the Balanced Problem.



The output is now the following.

$$
\begin{array}{lll}
|\phi'\rangle & = U_f[(H^{\otimes n} \otimes I)(|0^n\rangle) \otimes |-\rangle)] & \text{from circuit} \\
& = U_f[(H^{\otimes n} |0^n\rangle) \otimes (I|-\rangle)] & \text{multiplication property} \\
& = U_f[H^{\otimes n} |0^n\rangle |-\rangle] & \text{identity matrix} \\
& = U_f[(\frac{1}{\sqrt{2}^n} \sum_{x\in\{0,1\}^n} |x\rangle) |-\rangle] & \text{Hadamard superposition} \\
& = U_f[\frac{1}{\sqrt{2}^n} \sum_{x\in\{0,1\}^n} |x\rangle |-\rangle] & \text{distributivity of } \otimes \text{ over } + \\
& = \frac{1}{\sqrt{2}^n} \sum_{x\in\{0,1\}^n} U_f |x\rangle |-\rangle & \text{linearity of } U_f \\
& = \frac{1}{\sqrt{2}^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle |-\rangle & U_f \text{ with } |x\rangle|-\rangle \text{ result}
\end{array}
$$

We now feed the first n bits through an parallel Hadamard. From our previous work on parallel Hadamards, recall the formula for the Hadamard of a general superposition: $H^{\otimes n}(\sum_x \alpha_x |x\rangle) = \sum_x \sum_{y\in\{0,1\}^n} \alpha_x (-1)^{[x,y]} |y\rangle$. We will use this result here, substituting $\sum_x \alpha_x |x\rangle$ for the superposition we current have in the circuit of $\sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle$.



$$
\begin{array}{lll}
|\phi''\rangle & = (H^{\otimes n} \otimes I)([\frac{1}{\sqrt{2}^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle] \otimes |-\rangle) & \text{from circuit} \\
& = (\frac{1}{\sqrt{2}^n} H^{\otimes n} [\sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle]) \otimes (I|-\rangle) & \text{mult. property} \\
& = (\frac{1}{\sqrt{2}^n} H^{\otimes n} [\sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle]) \otimes |-\rangle & \text{identity matrix} \\
& = \frac{1}{\sqrt{2}^n} \frac{1}{\sqrt{2}^n} [\sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} (-1)^{f(x)} (-1)^{[x,y]} |y\rangle] \otimes |-\rangle & H^{\otimes n} \text{ of superpos.} \\
& = \frac{1}{2^n} [\sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} (-1)^{f(x)+[x,y]} |y\rangle] \otimes |-\rangle & \text{simplify}
\end{array}
$$

We have a sum of y-qubits $\sum_y$. Notice that there are no more x-qubits being summed up. The next step is to *measure*. It we measure the first n qubits, there are 2^n states that it can collapse to. There are 2^n bit strings that can be output.

It turns out that whether or not we see $|0^n\rangle$ alone, is enough to tell us whether $f(x)$ is constant or balanced (recall our assumption is that the $f(x)$ inputted *must be* either constant or balanced)

Example 25
What is the probability that $|\phi''\rangle$ collapses $|0^n\rangle$?

We find the amplitude (coefficient) of $|0^n\rangle$ and square it. The amplitude of $|0^n\rangle$ in $|\phi''\rangle$ is found by removing the summation and fixing $y = 0^n$. This gives us the amplitude .....

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

Because the amplitude of a general y is $\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+[x,\,y]}$. Setting $y = 0^n$ gives $\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+[x,\,0^n]}$ which simplifies using $[x,\,0^n] = 0$.

- ☐ **Balanced** If f is balanced, $\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = 0$. This is because half the outputs are 0, half the outputs are 1 (by def. of *balanced*). So half the summation terms are -1, half the terms are +1. They cancel out in the sum to give zero. Hence the amplitude of $|0^n\rangle$ is $\frac{1}{2^n}(0) = 0$. So the probability of seeing $|0^n\rangle$ after measurement is $0^2 = 0$.
- ☐ **Constant with Zero** If f outputs constantly zero, $f(x) = 0$ for all x, and so the summation simplifies to $\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = \sum_{x\in\{0,1\}^n} 1 = 2^n$. Hence the amplitude of $|0^n\rangle$ is $\frac{1}{2^n}(2^n) = 1$. So the probability of seeing $|0^n\rangle$ after measurement is $1^2 = 1$.
- ☐ **Constant with One** If f is constantly one, $f(x) = 1$ for all x, and so the summation $\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = \sum_{x\in\{0,1\}^n}(-1) = -2^n$. Hence the amplitude of $|0^n\rangle$ is $\frac{1}{2^n}(-2^n) = -1$. So the probability of seeing $|0^n\rangle$ after measurement is $(-1)^2 = 1$.

After measuring the first n quibits, if we observe $|0^n\rangle$, f is *definitely* constant, if we don't observe $|0^n\rangle$, f is *definitely* balanced

---

The natural question is what happens is what happens when we input a function that isn't balanced or constant into this circuit. If we do this, then the amplitude of $|0^n\rangle$ will be between zero and one. So the probability of seeing $|0^n\rangle$ is $0 < p < 1$, and so sometimes the circuit outputs *constant* (with probability p), and sometimes the circuit will output *balanced* (with probability $1 - p$).

However constant and balanced functions are special because they conveniently cause the amplitude for $|0^n\rangle$ to be 1 and 0 respectively. By using a single measurement, we can see whether f is constant or balanced. This gives an exponential reduction in the complexity of the classical algorithm.

4.4 Simon's Algorithm

4.5 Grover's Algorithm

Averaging Operator: finds the mean amplitude of *phi*

Example 26
Show that the averaging operator is *unitary*.

About-the-Mean Inversion: $v_i' = \frac{2}{N}\sum_j v_j - v_i$

Example 27
Prove that *about-the-mean inversion* does not change the mean.

Increase Amplitude: combine *phase inversion* with *about-the-mean inversion*

5 . Quantum Algorithms II

5.1 Fourier Transform

Root of Unity: $\omega_n^x = e^{2\pi i \frac{x}{n}}$

We can also think of this as a function $\omega_n(x) = e^{2\pi i \frac{x}{n}}$. But, what is nice about putting the x on top as the superscript of omega, is that the right hand side of omega looks like the fraction that multiplies $2\pi i$ in the exponent of e.

Phasor: $w_n^x = (\omega_n^0, \omega_n^x, \omega_n^{2x}, ..., \omega_n^{(n-1)x}) = \sum_{k=0}^{n-1} \omega_n^{kx} |k\rangle$

Again, this can be thought of as a function: $w_n(x) = (\omega_n^0, \omega_n^x, \omega_n^{2x}, ..., \omega_n^{(n-1)x})$. This function outputs a vector whose roots of unity form a sequence that rotate with frequency x.

Vandermonde Matrix: $V_{n\times n} = [w_n^0 w_n^1 \cdots w_n^{n-1}]$

Example 28
Determine the inner product of w^{j_1} and w^{j_2} for $j_1 \neq j_2$.

Example 29
Determine the inner product of w^{j_1} and w^{j_2} for $j_1 = j_2$.

Example 30
Prove that $V_{n\times n}$ is unitary.

Example 31
Prove that $V_{n\times n}$ is unitary.

Quantum Fourier Transform: $F = \frac{1}{\sqrt{2^n}} V_{2^n \times 2^n}$

Example 32
What is the matrix for F for $n = 1, 2, 3$?

Example 33
Prove that F is unitary.

Example 34
What is $F|00..0\rangle$? How does this relate to $H^{\otimes n}|00..0\rangle$?

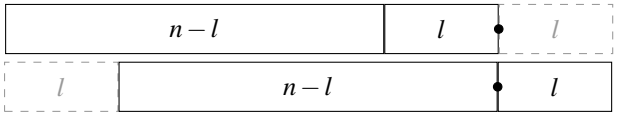
Example 35
Determine $\sum_{t_1 t_2 \cdots t_n \in \{0,1\}^n} c^{t_1 t_2 \cdots t_n} |t_1 t_2 \cdots t_n\rangle = ...$ written in the form of a tensor product with n factors.

$$\boxed{\bigotimes_{l=1}^{n} (|0\rangle + c^{2^{n-l}} |1\rangle)}$$

Example 36
Determine $\sum_{t_1 t_2 \cdots t_n \in \{0,1\}^n} c^{d t_1 t_2 \cdots t_n} |t_1 t_2 \cdots t_n\rangle = ...$ written in the form of a tensor product with n factors.

$$\boxed{\bigotimes_{l=1}^{n}(|0\rangle + c^{d2^{n-l}}|1\rangle)}$$

$$\begin{aligned}
F|j\rangle &= \tfrac{1}{\sqrt{2^n}} V_{2^n\times 2^n}|j\rangle && \text{def of } F\\
&= \tfrac{1}{\sqrt{2^n}} w_{2^n}^{j} && \text{def of } V_{2^n\times 2^n};j\text{th col}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{k=0}^{2^n-1} \omega_{2^n}^{kj}|k\rangle && \text{def of } w_n^j\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{k=0}^{2^n-1} e^{2\pi i\frac{kj}{2^n}}|k\rangle && \text{def of } \omega_n^j\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{\mathsf{k}_1\mathsf{k}_2\cdots\mathsf{k}_n\in\{0,1\}^n} e^{\frac{2\pi ij}{2^n}\times \mathsf{k}_1\mathsf{k}_2\cdots\mathsf{k}_n}|\mathsf{k}_1\mathsf{k}_2\cdots\mathsf{k}_n\rangle && \text{writing } k \text{ in binary}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{\frac{2\pi ij}{2^n}2^{n-l}}|1\rangle) && \text{prev ex with } c{=}e, d{=}\tfrac{2\pi ij}{2^n}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{\frac{2\pi ij}{2^l}}|1\rangle) && \text{cancelling } 2^n
\end{aligned}$$



$$\begin{aligned}
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{\frac{2\pi i}{2^l}\sum_{r=1}^{n}2^{n-r}\mathsf{j}_r}|1\rangle) && \text{writing } j \text{ in binary}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{\frac{2\pi i}{2^l}[\sum_{r=1}^{n-l}2^{n-r}\mathsf{j}_r+\sum_{r=n-l+1}^{n}2^{n-r}\mathsf{j}_r]}|1\rangle) && \text{break up least sig } l \text{ bits}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{2\pi i[\sum_{r=1}^{n-l}2^{n-r-l}\mathsf{j}_r+\sum_{r=n-l+1}^{n}2^{n-r-l}\mathsf{j}_r]}|1\rangle) && \text{binary shift right by } l \text{ bits}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{2\pi i[K+\sum_{r=n-l+1}^{n}2^{n-r-l}\mathsf{j}_r]}|1\rangle) && \text{where } K \text{ is an integer}\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{2\pi i\sum_{r=n-l+1}^{n}2^{n-r-l}\mathsf{j}_r}|1\rangle) && e^{2\pi iK}{=}1 \text{ for any integer } K\\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\bigotimes_{l=1}^{n}(|0\rangle + e^{2\pi i\sum_{r=1}^{l}2^{-r}\mathsf{j}_{(n-l)+r}}|1\rangle) && \text{reindex summation}
\end{aligned}$$

QFT Output: $F|\mathsf{x}_1\mathsf{x}_2\cdots\mathsf{x}_n\rangle = \frac{1}{\sqrt{2^n}}\bigotimes_{l=1}^{n}(|0\rangle + e^{2\pi i\sum_{r=1}^{l}2^{-r}\mathsf{x}_{(n-l)+r}}|1\rangle)$

The index of the tensor product, l tells us how many binary places to the right we shift x. Once we have $\mathsf{x}\gg l$, we take the l bits that went past the binary point. These l bits represent the fraction of $2\pi i$ and determine the relative phase of $|1\rangle$.

Example 37
Suppose we have $\mathsf{x} = 10110$.

We have
$\mathsf{x}\gg 1 = 1011\cdot 0$
$\mathsf{x}\gg 2 = 101\cdot 10$
$\mathsf{x}\gg 3 = 10\cdot 110$
$\mathsf{x}\gg 4 = 1\cdot 0110$

$x \gg 5 = \cdot 10110$

So by taking the fractional parts we have:

$$\begin{aligned} F|10110\rangle = \quad & \tfrac{1}{\sqrt{2^n}}[(|0\rangle + e^{2\pi i \times \cdot 0}|1\rangle) \otimes (|0\rangle + e^{2\pi i \times \cdot 10}|1\rangle) \\ & \otimes (|0\rangle + e^{2\pi i \times \cdot 110}|1\rangle) \otimes (|0\rangle + e^{2\pi i \times \cdot 0110}|1\rangle) \\ & \otimes (|0\rangle + e^{2\pi i \times \cdot 10110}|1\rangle)] \end{aligned}$$

To implement the circuit we need a new gate, the *controlled phase shift* gate. This is a two qubit gate that adjusts the relative phase of $|1\rangle$ on the first qubit provided the second qubit is $|1\rangle$.

$$R_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{2\pi i \frac{1}{2^k}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the QFT, we are interested in adjusting the relative phase by factors of $e^{2\pi i \frac{1}{2^k}}$. In other words doing bit shift right by some k. The goal is to get $e^{2\pi i \sum_{r=1}^{l} 2^r \mathrm{j}_r}$ for each line in our parallel circuit using our gate R_k.

First line corresponds to $l = 1$. We need $|\mathrm{j}_1\rangle \mapsto e^{2\pi i \sum_{r=1}^{1} 2^{r-1} \mathrm{j}_1}$ which simplifies to $|\mathrm{j}_1\rangle \mapsto e^{2\pi i 2^{1-1} \mathrm{j}_1}$

Example 38
Given that F is unitary, what is the matrix for the inverse fourier transform.

5.2 Period Finding

For M inputs, the number of times the function completes a whole repetition is given by:

Whole repetitions: $R = \lceil \frac{M}{r} \rceil$

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

We want to find the order of a, the smallest

---

f is one-to-one for a single period

5.3 Discrete Logarithms

A *Discrete logarithm*, *dlog*, is borrowing the the idea of logarithm with real numbers and applying them to *multiplicative cyclic groups*. They are sometimes called *indexes*.

Example 39
Consider the group $\mathbb{Z}_5^*$ that has generator 2.
The discrete log of 1, $dlog(1) = 4$ because $2^4 = 1 \text{mod } 5$.

5.4 Hidden Subgroup Problem

5.5 Shor's Algorithm

6 . Further Topics

6.1 Bell States

Bell states are *rotationally invariant*. Given $|\psi\rangle = \frac{1}{\sqrt{2}}[|uu\rangle + |u^{\perp}u^{\perp}\rangle]$ and a rotation of the bit basis: $|u\rangle = a|0\rangle + b|1\rangle$ with $|u^{\perp}\rangle = -b|0\rangle + a|1\rangle$

$$\begin{aligned}
|\psi\rangle &= \tfrac{1}{\sqrt{2}}[|uu\rangle + |u^{\perp}u^{\perp}\rangle] \\
&= \tfrac{1}{\sqrt{2}}[(a|0\rangle + b|1\rangle)(a|0\rangle + b|1\rangle) + (-b|0\rangle + a|1\rangle)(-b|0\rangle + a|1\rangle)] \\
&= \tfrac{1}{\sqrt{2}}[(a^2 + b^2)|00\rangle + (a^2 + b^2)|11\rangle \\
&= \tfrac{1}{\sqrt{2}}[|00\rangle + |11\rangle]
\end{aligned}$$

The previous bell state is only rotationlly invariant for real valued rotations. It is not invariant for complex rotations. However the bell state, psi-minus, $|\psi^{-}\rangle = \frac{1}{\sqrt{2}}[|01\rangle - |10\rangle]$ is invariant under all complex rotations.

Example 40
Prove that the bell state $|\psi^{-}\rangle$ is invariant under all complex rotations.

All four Bell States are *entangled*

6.2 Quantum Teleportation

Example 41
In order for quantum teleportation to work, Alice and Bob must share .....

6.3 Super-dense Coding

Super dense coding is the less popular sibling of teleportation. It can actually be viewed as the process in reverse. The idea is to send two classical bits of information by only sending one quantum bit. The process starts out with an EPR pair that is shared between the receiver and sender (the sender has one half and the receiver has the other).

6.4 Qutrits and Ququads

Qutrits are systems with 3 degrees of freedom. These are represented by vectors in $\mathbb{C}^3$ using base vectors $|0\rangle$, $|1\rangle$, and $|2\rangle$.

Qutrits are systems with 4 degrees of freedom. These are represented by vectors in $\mathbb{C}^4$ using base vectors $|0\rangle$, $|1\rangle$, $|2\rangle$ and $|2\rangle$.

Example 42
Give the coordinates of the base vectors $|0\rangle$, $|1\rangle$, $|2\rangle$ in $\mathbb{C}^2$

Example 43
Give the coordinates of the 2-qutrit states: $|00\rangle$, $|11\rangle$, $|22\rangle$.

Example 44
Show that $|\phi\rangle = \frac{1}{\sqrt{3}}[|000\rangle, |111\rangle, |222\rangle]$ is entangled.

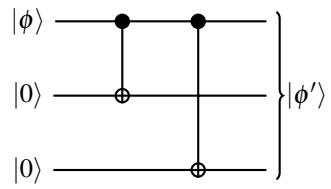
Example 45
Describe teleportation for a ququad. Sketch the appropriate quantum circuit and calculate the state before the measurement (the corrections needed can be ignored).

6.5 Quantum Cryptography

6.6 Quantum Error Correction

3-Qubit Bit-Flip Code:

We need to keep in mind that we have to measure to determine if there was an error.



The initial state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ is converted by the circuit into $|\phi'\rangle = \alpha|000\rangle + \beta|111\rangle$. Then each qubit of $|000\rangle$ is sent independently through the bit flip channel. Each qubit of $|111\rangle$ is sent independently through the bit flip channel. This gives us $|\phi''\rangle$.

We then apply a two-stage error correction method. We first make a measurement using the four measurement operators. Each measurement operator is gives a basis for the errors.

These operators are called *error syndromes*. Depending on which error syndrome is measured we then recover the original state.

$M_0 = |000\rangle\langle 000| + |111\rangle\langle 111|$
$M_1 = |100\rangle\langle 100| + |011\rangle\langle 011|$

$M_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$
$M_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$

Example 46
Suppose, for example, that a bit flip occurred, on the second qubit. Then $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ is converted by the circuit into $|\phi'\rangle = \alpha|000\rangle + \beta|111\rangle$. Then $|\phi'\rangle = \alpha|000\rangle + \beta|111\rangle$ is converted to $|\phi''\rangle = \alpha|010\rangle + \beta|101\rangle$.

Now we determine $p(2) = \langle\phi''|M_2^H M_2|\phi''\rangle = ...$

1

So it is *certain* that $j = 2$.

The state of the system will not change $M_2|\phi''\rangle = |\phi''\rangle$ Therefore, we finally flip the second qubit two to recover $|\phi'\rangle$.

Example 47
Show that these four measurement operators satisfy the *General Measurement Principle*: $\sum_{j=0}^{3} M_j^H M_j = I$

Example 48
Classical error correction uses repeated copies of the input bit(s); the no-cloning theorem forbids copying of qubits. Explain why the circuit for the 3-qubit bit-flip code does not violate the no-cloning theorem.

Example 49
Sketch how one can, in general, use an encoding circuit U_{enc} (like the 3-qubit bit-flip code circuit above) to recover a distorted qubit.

Example 50
Sketch how one can, in general, use an encoding circuit U_{enc} (like the 3-qubit bit-flip code circuit above) to recover a distorted qubit.

Example 51
Assume that at most one 1-qubit error occurs on a codeword for the 3-qubit bit-flip code. How can one recover the original state? Which recovery operation(s) are needed?

State all the intermediate states during an execution of the complete correction circuit encoding $|0\rangle$ for, (i) when no errors occur (ii) all cases where exactly one of the qubits flips.

Example 52

Sketch Shor's 9-qubit code circuit and calculate its output for general qubits

6.7 Teleportation Quantum Computation

6.8 One-way Quantum Computation

One-way Quantum Computing (1WQC) is a alternative method of thinking about implementing quantum algorithms using *graphs* rather than quantum circuits. Let us show the contrast between the circuit scheme of quantum computation and this new scheme that uses graphs.

Quantum circuits have a number of qubits which are initially in a product state and then are subjected to various one and two-qubit gates before being measured.

We can imagine the qubits as nodes of a graph and the graph has edges between two nodes iff the two qubits are entabled. In fact, we can have edges from many nodes to the same node (and vice-versa). These are special *multi-qubit* entangled state. The graph itself is a special multi-qubit entangled state.

Then *measure* qubits individually. We measure qubits one-by-one. Each measurement causes the state to be less entangled. It turns out that via a suitable choice of measurements, any quantum computation can be acheived using this process. However the result of each measurement is random, so we need to post-process or *correct*, the state so that we always get the exact computation we desire (regardless of which state a measurement could collapse to).

We treat entanglement as a limited resource (like space on a classical computer). We call the first state, the *resource state*. It is called *one-way* because the resource state is destroyed by the measurements.

Each measurement, M_j, is of the form: $\cos(\theta_j)|0\rangle + (-1)^{s_j}\sin(\theta_j)|1\rangle$

---

where $s_j \in \{0,1\}$.

Example 53
The following are Pauli measurements. For each state, θ_j and s_j.
(a) $|+\rangle, |-\rangle$
(b) $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$, $\frac{1}{\sqrt{2}}(i|0\rangle + |1\rangle)$
(c) $|0\rangle, |1\rangle$

6.9 Measurement Calculus I

The measurement calculus is a formalisation of *Measurement-Based Quantum Computing* (MBQC) and *one-way quantum computing* (1WQC). The measurement calculus formally describes these by using a *sequence of commands*. Each command is either *preparation*, *entanglement*, *measurement*, or *correction*. Each command is denoted with a capital letter (namely N, E, M, X, Z). Each command uses indicies $i, j \in \{1, ..., n\}$ indicate qubits to which the operations apply.

We informally describe the commands using in measurement calculus.

- **N - Prepare ith Qubit** The *preparation command* N_i prepares the ith qubit.
- **E - Entangle ith with jth Qubit** The *entanglement command*, $E_{i,j}$ entangles qubits numbered i and j. The entangled state is $|+\rangle = C(Z_{i,j})$. That is, a controlled $Z_{i,j}$ gate, where $Z_{i,j}$ is the matrix with $z_{ii} = 1$, $z_{ii} = 1$,
- **M - Measure ith Qubit** The *measurement command*, M_i measures the ith qubit. The measurement uses a specific basis that we will look at later. We will later extend the measurement command to specify this basis.
- **X, Z - Correct ith Qubit** The *correction commands*, X_i and Z_i are designed to correct the state of the system after qubits have been measured. The X_i command corrects the ith qubit by multiplication by the Pauli matrix X. Similarly, Z_i command corrects the ith qubit by multiplication by the Pauli matrix Z. These corrections are often called *Pauli corrections* because of the use of Pauli matrices Z and X.

Commands are read *right-to-left*

So if we have commands *ABCD*, it is command *D* that is done first.

Example 54
Complete the following:
(a) The command(s) that act on one qubit are: .....
(b) The command(s) that act on two qubits are: .....

We now define a *pattern*. Informally, a pattern consists of three things. Firstly, we need *qubits*. These will act as our resouce for computation. Secondly, we have a *command sequence*. This is a sequence of commands that we previously described (*preparation*, *entanglement*, *measurement*, or *correction*). They specify precisely how to perform the computation on our qubits. Thirdly, we need *rules*. We cannot just use any combination of commands. We need specific rules about what order commands can appear in sequences.

In general qubit i has a lifetime consisting of (1) *preparation*, then (2) *entanglement*, then (3) *measurement*, and finally *correction*. We want form rules so that our commands give qubits this lifetime.

$$\underset{N_i}{\text{prepare}} \longrightarrow \underset{E_{i,j}}{\text{entangle}} \longrightarrow \underset{M_i}{\text{measure}} \longrightarrow \underset{X_i,\, Z_i}{\text{correct}}$$

Furthermore, input qubits are output qubits have slightly different lifetimes and so our rules need to capture these differences. In particular, output qubits are never measured. Input/intermediate qubits must always be measured.

Input Qubits $\underset{N_i}{\text{prepare}} \longrightarrow \underset{E_{i,j}}{\text{entangle}} \longrightarrow \underset{M_i}{\text{measure}} \longrightarrow \underset{X_i,\, Z_i}{\text{correct}}$

Intermediate Qubits $\underset{N_i}{\text{prepare}} \longrightarrow \underset{E_{i,j}}{\text{entangle}} \longrightarrow \underset{M_i}{\text{measure}} \longrightarrow \underset{X_i,\, Z_i}{\text{correct}}$

Output Qubits $\underset{N_i}{\text{prepare}} \longrightarrow \underset{E_{i,j}}{\text{entangle}} \longrightarrow \underset{M_i}{\text{measure}} \longrightarrow \underset{X_i,\, Z_i}{\text{correct}}$

We specify the following rules for command sequences. Each command sequence must obey the following rules to be valid for a pattern.

- **D0** - no command depends on an outcome not yet measured
- **D1** - no command acts on a qubit already measured
- **D2** - no command acts on a qubit not yet prepared, except if is an input qubit
- **D3** - a qubit i is measured iff it is not an output qubit (input/intermediate)

Example 55
Assume we have a pattern $\mathcal{P}$ where 1 is an input qubit, and 2 is an intermediate qubit. Explain why the following command sequence is *invalid*.

$$A = X_1 M_1 E_{1,2}$$

Look at rules D0, D1, D2, D3 to see which is violated

D2 violated. $E_{1,2}$ acts on unprepared intermediate qubit: qubit 2

Example 56
Assume we have a pattern $\mathcal{P}$ where 1 is an input qubit, and 2 is an output qubit.
For the following command sequences explain why they are *invalid*.
(a) $M_1 X_1 E_{1,2} N_2$
(b) $M_2 M_1 N_2$
(c) N_2

Look at rules D0, D1, D2, D3 to see which is violated

(a) D0 violated. X_1 depends on qubit 1 but was not measured earlier in sequence
(b) D3 violated. output qubit is measured
(c) D3 violated. input qubit is never measured

We can also combine patterns using two methods, the *Composite Pattern* and the *Tensor Pattern*.

- **Composite Pattern** $\mathcal{P}_1\mathcal{P}_2$
- **Tensor Pattern** $\mathcal{P}_1 \otimes \mathcal{P}_2$

6.10 Measurement Calculus II

If there are m measurements then there are 2^m different executions. Each execution is described by a binary string s that has m bits.

Domain: The *domain* of a signal is the set of qubits on which it depends

---

The domain of the signals of a dependent command (measurement or correction), represents the set of measurements which one has to do before one can determine the actual value of the command.

Example 57
Describe the *standard form* (or *normal form*) of a pattern in the Measurement Calculus. Explain its parts.

Example 58
Transform the pattern $X_3^{s_2}X_2^xE_{23}X_2^{s_1}M_1^{-\alpha}E_{12}$ into the *standard form*.

6.11 Measurement Calculus III

$\mathbb{Z}_2^W$ denotes all total functions from W to $\mathbb{Z}_2$
