PC5228 Quantum Information and Computation

Dag Kaszlikowski, Zhian Jia

October 30, 2025

Contents

Chapter 1

Brief Introduction to Discrete Quantum Mechanics

1.1 Classical Interference: the Mach-Zehnder Interferometry

In classical physics, light is an electromagnetic wave made of oscillating electric and magnetic fields. Here, we simplify things to the max, so if you're interested in details, ask me.

For our purposes[1], let us assume the light is a monochromatic (single frequency) plane wave:

$$E(t) = E_0 \exp\{(i\omega t)\}, \tag{1.1}$$

where $\omega = \frac{2\pi}{T}$ is the wave's frequency. Here, t stands for time, and the real number E_0 is the electric field amplitude. Note that we don't bother with the magnetic field, and we assume that the electric field's direction is fixed. Of course, you have to take the real or imaginary part of $E(t)$ for this to make sense, but it's convenient to keep it as a complex number as you will see.

Exercise 1. *Explain why we can forget about the magnetic field in this description.*

The energy of the plane wave is proportional to $|E(t)|^2 = |E_0|^2$.

We're now going to build a Mach-Zehnder interferometer (MZI) and discuss it in reasonable details. It's a beautiful piece of physics, so let's get down to it.

[1] We make lots of simplifications here! For instance, we totally omit the wave's propagation direction.

Consider a piece of glass called a *beamsplitter*. It splits $E(t)$ into two waves: the reflected and transmitted one. The conservation of energy requires that $|E(t)|^2 = |E_T(t)|^2 + |E_R(t)|^2$, and we assume that our beamsplitter splits the incoming wave equally, i.e., $|E_T(t)|^2 = |E_R(t)|^2$. Thus, $E_T(t) = E_R(t) = \frac{1}{\sqrt{2}}E(t)$. Since reflection and transmission can introduce phases to the corresponding waves, we have:

$$E_X(t) = \frac{1}{\sqrt{2}} E_0 \exp\{[i(\omega t + \phi_X)]\}, \tag{1.2}$$

where $X = R, T$. Interestingly, using an argument based on symmetry, you can show that you must have $\phi_T - \phi_R = \frac{\pi}{2}$. A hint to prove it is to send two waves with the same energy at the beamsplitter from the top and bottom (see Figure 1.1). If you can't prove it, you can consult this paper: V. Degiorgio, "Phase shift between the transmitted and reflected optical field of a semireflecting lossless mirror is $\frac{\pi}{2}$", Am. J. Phys. 48 (1980) 81–82. You can access it through the NUS online library (and many other papers and books, so knock yourself out!).

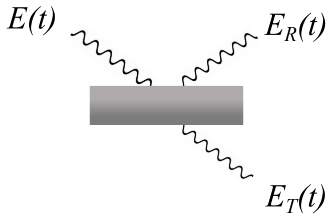


Figure 1.1: A 'caricature' of a beamsplitter as a piece of glass

Exercise 2. *How does the magnetic field contribute to the electromagnetic wave's energy?*

We now need a mirror, which is a beamsplitter that only reflects incoming light and, for convenience, doesn't introduce any phase shift, see Fig. (1.1).

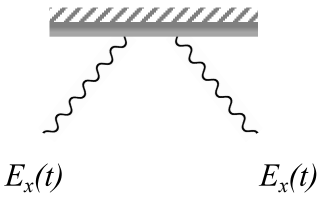


Figure 1.2: Mirror

The last piece of gear is a phase shifter, Fig. (1.1). As the name suggests, it shifts the wave's phase.

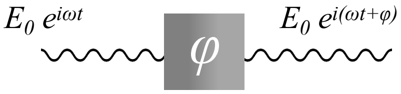


Figure 1.3: Phase shifter

Exercise 3. *How would you build a phase shifter? Think of the most primitive solution.*

We're now ready to assemble the MZI interferometer. As you'll see in Section 2, this is one of the most bizarre devices in the whole visible Universe when you feed it with quantum particles of light.

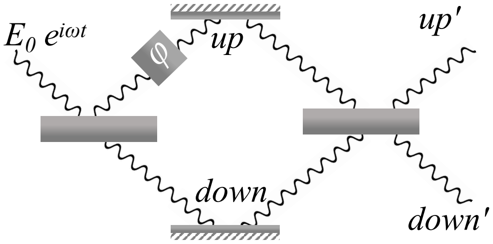


Figure 1.4: The Mach-Zehnder interferometer

What comes out at up'?

1. After the first beamsplitter: $\frac{1}{\sqrt{2}}E_0 \exp\{[i(\omega t + \phi_R)]\} + \frac{1}{\sqrt{2}}E_0 \exp\{[i(\omega t + \phi_T)]\}$
2. After the phase shifter: $\frac{1}{\sqrt{2}}E_0 \exp\{[i(\omega t + \phi_R + \phi)]\} + \frac{1}{\sqrt{2}}E_0 \exp\{[i(\omega t + \phi_T)]\}$
3. After the second beamsplitter, in the up' output: $\frac{1}{2}E_0 \exp\{[i(\omega t + \phi_R + \phi + \phi_R)]\} + \frac{1}{2}E_0 \exp\{[i(\omega t + \phi_T + \phi_T)]\}$.

Thus, the light's energy in up' output of the interferometer is:

$$\frac{1}{2}|E_0|^2 \left[1 + \cos(\phi + 2\phi_R - 2\phi_T)\right].$$

Before we go any further let's fix ϕ_R and ϕ_T. Their absolute values depend on the beamsplitter's design but their difference is fixed (see above) so we make them $\phi_R = 0$ and $\phi_T = \frac{\pi}{2}$ for convenience. This gives us a simple expression for the energy in up':

$$\frac{1}{2}|E_0|^2 \left(1 - \cos(\phi)\right).$$

You can see that the energy oscillates between zero and $|E_0|^2$. This oscillation is a direct consequence of the nature of classical light, see Fig. (1.1). Waves, unlike classical particles, can interfere.

Exercise 4. *This is a* **difficult** *question: Suppose you can only set a phase shift ϕ in the Mach-Zehnder interferometer to a certain precision. How does this affect the observed interference?*

Very often it is good to have means to measure a given physical phenomenon. For instance, you can measure how much electric current flows thru a circuit or how much energy an electromagnetic wave has etc. How do we measure interference strength?

Interference strength is quantified via the so-called *visibility* V, defined like this:

$$V := \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{1.3}$$

where I_{max} is the maximum energy in the output of your choice and I_{min} is the minimal energy in the same output.

Exercise 5. *Show that:*

1. $0 \leq V \leq 1$
2. *Visibility for the experiment described in this chapter is* $V = 1$.

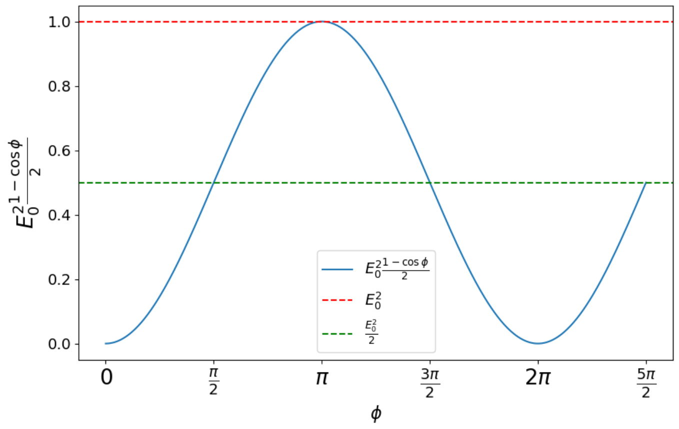


Figure 1.5: Classical interference in the Mach-Zehnder interferometer

Summary of 1.1:

- Elementary description of electromagnetic waves.

- Basic theory of classical interference in Mach-Zehnder interferometer.
- Measure of interference strength.

1.2 Quantum Interference: the Mach-Zehnder Interferometry

Max Planck introduced energy quanta to explain black-body radiation that classical theory of light failed to explain. Albert Einstein took Planck's idea and explained the photoelectric effect, assuming that light is made of *particles* called photons. If you try to explain the photoelectric effect using classical electromagnetic waves, you get predictions that are 'opposite'[2] to what you observe. No matter how beautiful your theory is, if it doesn't explain experiments, you must trash it.

Current technologies make it possible to create single photons in the lab, something people couldn't do in the early 30s when quantum theory was incepted. So, let's go to the lab and inject a single photon into the MZI.

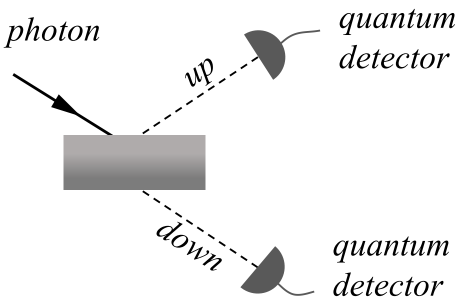


Figure 1.6: Photon on beamsplitter

We start with a single beamsplitter and a photon impinging on it 'from above' (see the drawing above). When the photon emerges on the other side, we catch it with quantum detectors[3]. The experiment shows that every single photon behaves randomly, i.e., you never know which detector will capture it. Let's remember this and move on to see what happens to the photon once it passes through the interferometer with the phase shifter set

[2] The phrase 'opposite prediction' is an oxymoron, but it works if you revisit the history of the photoelectric effect.

[3] They work because of the photoelectric effect!

to $\phi = \pi$. The experiment is clear about what happens:

$$\begin{aligned} p_{exp}(up'|\phi = \pi) &= 1 \\ p_{exp}(down'|\phi = \pi) &= 0. \end{aligned} \tag{1.4}$$

Can we reconcile it with the fact that a beamsplitter distributes the photon randomly as we just verified it, i.e.,

$$\begin{aligned} p_{exp}(up'|up) &= \frac{1}{2} \\ p_{exp}(up'|down) &= \frac{1}{2} \\ p_{exp}(down'|up) &= \frac{1}{2} \\ p_{exp}(down'|down) &= \frac{1}{2}? \end{aligned} \tag{1.5}$$

Note that we don't show dependence on ϕ because it doesn't show in the lab results[4]. A simple Bayesian calculus gives us

$$p_{Bayes}(up') = p_{exp}(up'|up)p_{exp}(up)+p_{exp}(up'|down)p_{exp}(down) = \frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{2} = \frac{1}{2}, \tag{1.6}$$

which is totally different than what you see in the lab, i.e., $p_{Bayes}(\cdot) \neq p_{exp}(\cdot)$! It's perfect predictability (experiment) vs. perfect randomness (common sense mathematical reasoning).

Let's do it again but this time for an arbitrary ϕ. Experiments unanimously show that

$$p_{exp}(up'|\phi) = \frac{1}{2}(1 - \cos(\phi)). \tag{1.7}$$

However, the Bayesian calculus tells us that

$$p_{Bayes}(up'|\phi) = \frac{1}{2}\left(p_{\mathrm{exp}}(up'|up, \phi) + p_{exp}(up'|down)\right). \tag{1.8}$$

There is no way to make it fit the experimental result $\frac{1}{2}(1-\cos(\phi))$ unless you use negative probabilities or some other exotic stuff[5].

All of this is rather bizarre. It seems that photons somehow know what experiment you are going to perform and behave accordingly. If you test them with a single beamsplitter, they act like particles, but if you choose to put them into the MZI, they behave like waves,

[4] This is a statement that could be challenged in the following way: what if, after passing thru the phase shifter, the photon remembers the phase? This is tackled later on.

[5] Anyone interested in exotic stuff, please ask me

i.e., they interfere. Some people call it the wave-particle duality, but there is no need for such confusing names – quantum mechanics has a neat mathematical solution we discuss in the next section. However, the mystery still remains, and you should spend some time thinking about it. Just don't think too much, or else you may go crazy;-)

Summary of 1.2:

- Mach-Zehnder experiment with a single photon behaves similarly to its classical counterpart but there are fundamental differences that cannot be reconciled.

1.3 Quantum formalism of the Mach-Zehnder Interferometry

Let's revisit the classical interference picture. This will allow us to 'derive' quantum formalism. I will resort to hand-waving arguments but they work pretty well in physics, less so in mathematics.

A beam-splitter has two entries: up and down. If an EM wave $E_0 \exp(i\omega t)$ comes from the above we can encode it as

$$\begin{bmatrix} E_0 \exp(i\omega t) \\ 0 \end{bmatrix} \tag{1.9}$$

and when it comes from below as

$$\begin{bmatrix} 0 \\ E_0 \exp(i\omega t) \end{bmatrix} \tag{1.10}$$

The beam-splitter transforms (1.9) it to

$$\frac{1}{\sqrt{2}} \begin{bmatrix} E_0 \exp(i\omega t) \\ \\ E_0 \exp\left(i\omega t + \frac{\pi}{2}\right) \end{bmatrix} \tag{1.11}$$

because the wave has been split to two components. We see that before and after the beamsplitter $E_0 \exp(i\omega t)$ doesn't change so we can forget it to make the formulas look neater[6].

Now, how to describe this vector transformation from (1.9) to (1.11)? We need to use a matrix, which we can guess by comparing (1.9) and (1.11):

[6] It's always a good idea to remove clutter in life's every aspect. Japanese call it *danshari*.

$$B = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & ? \\ \exp\{(i\frac{\pi}{2})\} & ?? \end{bmatrix}.$$

The entries in the 2nd column can be fixed if we let the EM wave come from the below and repeat the whole reasoning.

Exercise 6. *Convince yourself that* $? = \exp\{(i\frac{\pi}{2})\}$ *and* $?? = 1$.

How about the phase-shifter? It's a piece of cake because it does nothing if the photon travels 'down' path, so:

$$\Phi = \begin{bmatrix} \exp\{(i\phi)\} & 0 \\ 0 & 1 \end{bmatrix}.$$

We now have a description of the MZI with vectors and matrices.

Exercise 7. *Verify that $B\Phi B$ acting on the vector (1.9) reproduces the correct MZI's outputs up' and down'.*

Now let's make a giant step, not very different than the step made by Werner Heisenberg, Erwin Schroedinger and Niels Bohr and the other uncles[7], and say this:

1. If a photon enters MZI from above (*state up*) we ascribe to this piece of physical information – a vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and a vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (*state down*) if it comes from below.

2. Optical devices in MZI that affect the photon's state are now represented as 2D matrices B and Φ.

3. To calculate the probability of the photon to be in up/down state we take the modulus square of the vector's first/second entry. This is called *Born* rule.

[7] We're in Singapore!

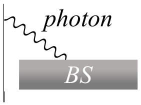
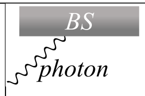
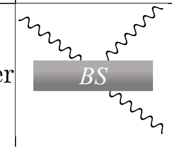
Up-input photon	 $\equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
Down-input photon	 $\equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
Beam-splitter	$\equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$
Phase-shifter	$\equiv \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix}$
Action of Beam-splitter	 $\equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Table 1.1: "Giant" step

Remark 1. *Note that 1. and 2., and analogies with classical light, imply that photon's state can also be* $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$ *after the first beamsplitter if it entered MZI from above (state up). Unlike up and down states, this one's rather bizarre because it looks as if the photon exists*[8] *up and down simultaneously. Of course, 3. gets rid of any need to dwell on it because it reduces the information encoded in this vector, via the Born rule, to equal chances of finding the photon up and down at the output of the beamsplitter.*

Exercise 8. *Replace the beamsplitters in MZI with asymmetric ones that transmit t% of light and reflect r%=100%-t%. Input a photon in 'up' state and calculate the probability it will come out in the MZI's 'down' exit. What's the photon's state after the 1st beamsplitter?*

The MZI's vector-matrix formalism explains all the lab's results. This is pretty remarkable but what has happened here? A clue comes from two simple observations: (i) the matrices B and Φ are unitary, i.e., $BB^\dagger = I$, and $\Phi\Phi^\dagger = I$. Here I is the identity matrix and $\dagger$ is transposition, and complex conjugation combined (ii) vectors describing photon's states have complex entries $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$ (you will understand it better if you finish Exercise (8)).

Remark 2. *Note that any unitary matrix* U *(*$UU^\dagger = U^\dagger U = I$*), not only* B *and* Φ*, transforms* $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ *with* $|\alpha_0|^2 + |\alpha_1|^2 = 1$ *to* $\begin{bmatrix} \alpha_0' \\ \alpha_1' \end{bmatrix}$ *with* $|\alpha_0'|^2 + |\alpha_1'|^2 = 1$.

[8] What is the meaning of "exist" now?

Algebra of vectors and unitary matrices is embedded in a mathematical structure called *Hilbert space*. In physics, a system described by a 2D Hilbert space (like our photon in the MZI) is called a *qubit*. Qubit is the simplest, non-trivial quantum system in the Universe.

Summary of 1.3:

- Vector and matrix representation of Mach-Zehnder interferometry as a step towards discrete quantum mechanics.
- Unitary matrices.
- Born rule.
- Concept of qubit.

1.4 2D Hilbert space

2D Hilbert space is effectively[9] a set of two-dimensional vectors with complex coefficients

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix},$$

where $\alpha_0, \alpha_1 \in \mathbb{C}$. Vectors can be multiplied by complex numbers λ

$$\lambda \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \lambda\alpha_0 \\ \lambda\alpha_1 \end{bmatrix},$$

added

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} + \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 + \beta_0 \\ \alpha_1 + \beta_1 \end{bmatrix}$$

and both

$$\lambda \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} + \mu \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \lambda\alpha_0 + \mu\beta_0 \\ \lambda\alpha_1 + \mu\beta_1 \end{bmatrix}.$$

Each vector $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ has its Hermitian conjugate defined as $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}^\dagger := [\alpha_0^*, \alpha_1^*]$. This let us define a *scalar product* of two vectors as

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}^\dagger \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \alpha_0^*\beta_0 + \alpha_1^*\beta_1.$$

[9] A proper mathematical definition is more general but we shouldn't care for now.

Exercise 9. *Recall the definition of a scalar product of two three-dimensional real vectors $\vec{a}$ and $\vec{b}$: $\vec{a}\cdot\vec{b}$. What are the physical quantities you remember that use such a scalar product?*

Two vectors are called orthogonal if their scalar product is zero. A vector such that its scalar product with itself equals to one is *normalised.* Two vectors that are normalised and orthogonal are called *orthonormal.* Each vector from 2D Hilbert space can be decomposed like this

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \alpha_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The vectors $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ form an *orthonormal basis.* There are infinitely many orthonormal bases, for instance, $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix}$ and $\begin{bmatrix} \frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$.

Vectors can be transformed into another vectors via 2×2 matrices with complex entries

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} := \begin{bmatrix} a_{00}\alpha_0 + a_{01}\alpha_1 \\ a_{10}\alpha_0 + a_{11}\alpha_1 \end{bmatrix}.$$

In 2D Hilbert space, matrices $A, B, C, \ldots$ are called *linear operators.* Each operator A has its Hermitian conjugation $A^\dagger$:

$$A^\dagger = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}^\dagger := \begin{bmatrix} a_{00}^* & a_{10}^* \\ a_{01}^* & a_{11}^* \end{bmatrix}.$$

In quantum mechanics we use operators that are normal, i.e., such that $AA^\dagger = A^\dagger A$. Examples of normal operators:

1. Unitary operators U: $UU^\dagger = I$, where I is an identity operator (matrix with diagonal entries equal to 1 and off diagonal ones equal to zero)

2. Hermitian (also called self-adjoint) operators A: $A = A^\dagger$

3. Orthogonal projectors P: $P = P^\dagger$ and $P^2 = P$.

Exercise 10. *Find some explicit examples of unitary, hermitian and orthogonal projector operators.*

All the above must be translated to <u>*Dirac's bra and ket notation*</u> if we want to understand any physics.

1.4.1 Dirac's bras and kets

We encode vectors $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ as *kets* $|0\rangle$ and $|1\rangle$, also known as the *computational basis*. Their Hermitian conjugations are the so-called *bras* $\langle 0|$ and $\langle 1|$. Any vector $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ is now a new ket, say, $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ with its corresponding bra $\langle\alpha| = \alpha_0^*\langle 0| + \alpha_1^*\langle 1|$.

Scalar product of $|\alpha\rangle$ and $|\beta\rangle$ is now a *bracket*

$$\begin{aligned} \langle\alpha|\beta\rangle &= (\alpha_0^*\langle 0| + \alpha_1^*\langle 1|)(\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0^*\beta_0\langle 0|0\rangle + \alpha_0^*\beta_1\langle 0|1\rangle + \alpha_1^*\beta_0\langle 1|0\rangle + \alpha_1^*\beta_1\langle 1|1\rangle \\ &= \alpha_0^*\beta_0 + \alpha_1^*\beta_1 \end{aligned} \tag{1.12}$$

because $\langle 0|1\rangle = \langle 1|0\rangle = 0$ and $\langle 0|0\rangle = \langle 1|1\rangle = 1$.

Exercise 11. *Show that* $\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^*$.

Exercise 12. *Show that* $|k\rangle\langle l|$ *(*$k, l = 0, 1$*) is a* 2×2 *matrix of all zeros except for the intersection of the* k*th row and* l*th column equal to* 1 *and that* $(\lambda|k\rangle\langle l|)^\dagger = \lambda^*|l\rangle\langle k|$. *Show that this implies* $(|\alpha\rangle\langle\beta|)^\dagger = |\beta\rangle\langle\alpha|$.

If you solved Ex. (12) you will immediately see that any linear operator A can be written as

$$A = \sum_{k,l=0,1} a_{kl}|k\rangle\langle l|.$$

You can prove that

1. Normal: $N = n_0|\eta_0\rangle\langle\eta_0| + n_1|\eta_1\rangle\langle\eta_1|$ where $n_k \in \mathbb{C}$ and $\langle\eta_k|\eta_l\rangle = \delta_{kl}$.
2. Unitary: $U = u_0|\psi_0\rangle\langle\psi_0| + u_1|\psi_1\rangle\langle\psi_1|$ where $u_k \in \mathbb{C}$, $|u_k| = 1$ and $\langle\psi_k|\psi_l\rangle = \delta_{kl}$.
3. Hermitian: $A = a_0|\phi_0\rangle\langle\phi_0| + a_1|\phi_1\rangle\langle\phi_1|$ where a_k are real and $\langle\phi_k|\phi_l\rangle = \delta_{kl}$.
4. Orthogonal projectors[10]: $P = |\chi\rangle\langle\chi|$ where $\langle\chi|\chi\rangle = 1$.

Summary of 1.4:

- Mathematical description of two dimensional Hilbert space: vectors and linear operators.
- Dirac bra and ket formalism.

[10] Rank one to be exact but for a 2D Hilbert space to mention this is an overkill.

- Definitions of normal, unitary, hermitian and orthogonal projectors.

1.5 Physics of Qubit

A qubit is the simplest quantum system. It has only two physical states, one more than a one state system, which is too trivial to consider (one state always remains one state so nothing interesting can happen). Before inception of quantum information, qubit was rife in everyday quantum physics but nobody called it that:

1. A photon in Mach-Zehnder interferometer: $|up\rangle, |down\rangle$ states or as we denoted them earlier, $|0\rangle$ and $|1\rangle$.

2. A two level atom with its ground $|ground\rangle$ and excited $|excited\rangle$ state. All atoms such as, hydrogen, oxygen, polonium etc., have more than two levels. However, we can make artificial atoms in the lab (they're called superconducting qubits) and they're used in today's quantum computers.

3. Electron's spin: spin up along z direction $|\uparrow_z\rangle$ and down along z direction $|\downarrow_z\rangle$.

4. Photon's polarisation (more about it later): $|vertical\rangle, |horizontal\rangle$.

Dirac's notation isn't just a fancy way to represent Hilbert space. Kets and bras have a deep physical meaning, which you will gradually discover as you go deeper into quantum information theory. As of now I'd like to mention one important feature of kets. Two orthonormal kets, say those that form the computational basis $|0\rangle$ and $|1\rangle$ represent two physically distinguishable situations. For instance, if an electron's spin is up along z direction, $|0\rangle = |\uparrow_z\rangle$, it can be experimentally 100% distinguished from spin down along z direction, $|1\rangle = |\downarrow_z\rangle$. How? You can use a gradient of magnetic field along z direction that will deflect spin up if the spin is up and down if the spin is down. It's as simple as this.

Exercise 13. *Let's discuss some classical physics of Stern-Gerlach experiment. In this seminal experiment you send particles having a magnetic moment $\vec{\mu}$ and a constant velocity $\vec{v}$ to a finite size region filled up with a magnetic field $\vec{B}(\vec{r})$, where $\vec{r}$ describes a point in a 3D space we inhabit. What happens to the particles after they finished their interaction with the magnetic field?*

1.5.1 Pure and Mixed States of Qubit

Consider a qubit described by a ket $|\psi\rangle$

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{1.13}$$

where $\langle\psi|\psi\rangle = |\alpha_0|^2 + |\alpha_1|^2 = 1$ (we call it normalisation of the ket $\psi\rangle$). All kets in quantum physics must be normalised (see below the Born rule) and because of this physical requirement we can write

$$|\psi\rangle = e^{i\phi_0}\cos(\theta)|0\rangle + e^{i\phi_1}\sin(\theta)|1\rangle = e^{i\phi_0}\left(\cos(\theta)|0\rangle + e^{i(\phi_1-\phi_0)}\sin(\theta)|1\rangle\right), \tag{1.14}$$

where $e^{i\phi_0}$ is the so-called *global phase.* All kets that differ by a global phase are physically equivalent (more about it soon) so we can simply remove it. This leaves us with two parameters describing the qubit: θ and $\phi = \phi_1 - \phi_0$.

We mentioned before the Born rule quite a few times. This rule is of a paramount importance to link kets with experiment because we do not live in Hilbert space but in a physical, 4D universe. The Born rule tells us that $|\alpha_i|^2$ $(i = 0, 1)$ is the probability $p(i|\psi)$ of measuring the outcome i for a qubit described by the ket $|\psi\rangle$. This is mathematically equivalent to

$$p(0|\psi) = |\langle 0|\psi\rangle|^2 = \cos^2(\theta)$$

and

$$p(1|\psi) = |\langle 1|\psi\rangle|^2 = \sin^2(\theta).$$

We can neatly write it as

$$p(a|\psi) = \langle\psi|P(a)|\psi\rangle,$$

where $P(a) = |a\rangle\langle a|$ $(a = 0, 1)$ is the orthogonal projector on the ket $|a\rangle$. To move forward, let's introduce an operation that takes operators to numbers, called *trace*, and defined like this

$$Tr(|\psi\rangle\langle\phi|) := \langle\phi|\psi\rangle$$

for arbitrary kets $|\psi\rangle$ and $|\phi\rangle$. It's now easy to see the Born reads:

$$p(a|\psi) = Tr(P(a)|\psi\rangle\langle\psi|).$$

Since the probabilities of outcomes 0 and 1 are the only things we observe in the lab, the projector on $|\psi\rangle$ is fundamental. Let's massage this projector a little:

$$|\psi\rangle\langle\psi| = \cos^2(\theta)|0\rangle\langle 0| + \sin^2(\theta)|1\rangle\langle 1| + \frac{1}{2}\sin(2\theta)e^{-i\phi}|0\rangle\langle 1| + \frac{1}{2}\sin(2\theta)e^{i\phi}|1\rangle\langle 0|. \tag{1.15}$$

With the help of simple trigonometry we can write it like this

$$\begin{aligned} |\psi\rangle\langle\psi| &= \frac{1}{2}(1+\cos(2\theta)(|0\rangle\langle 0| - |1\rangle\langle 1|) + \sin(2\theta)\cos(\phi)(|0\rangle\langle 1| + |1\rangle\langle 0|) \\ &+ \sin(2\theta)\sin(\phi)(-i|0\rangle\langle 1| + i|1\rangle\langle 0|)). \end{aligned} \tag{1.16}$$

Let's introduce Pauli operators $X := |0\rangle\langle 1| + |1\rangle\langle 0|$, $Y := -i|0\rangle\langle 1| + i|1\rangle\langle 0|$ and $Z := |0\rangle\langle 0| - |1\rangle\langle 1|$. We can then write

$$|\psi\rangle\langle\psi| = \frac{1}{2}(1 + \hat{n}\cdot\vec{\sigma}) \tag{1.17}$$

where $\hat{n} = (\sin(2\theta)\cos(\phi), \sin(2\theta)\sin(\phi), \cos(2\theta))$ and $\vec{\sigma} = (X, Y, Z)$. The meaning of $\hat{n}\cdot\vec{\sigma}$ is this $\hat{n}\cdot\vec{\sigma} := n_x X + n_y Y + n_z Z$. Sometimes it is convenient to use a different notation $\sigma_1 = X, \sigma_2 = Y$ and $\sigma_3 = Z$ because then we can write $\hat{n}\cdot\vec{\sigma} = \sum_{k=1}^{3} n_k \sigma_k$.

It's easy to see that $\hat{n}$ is a 3D, normalised vector that we call *Bloch* vector. The significance of Bloch vector is enormous. We just showed that any qubit ket $|\psi\rangle$ is mapped to a unique Bloch vector $\hat{n}$. Tips of Bloch vectors lie on the surface of the unit Bloch sphere, i.e., a sphere with the radius length one, see Fig. 8.

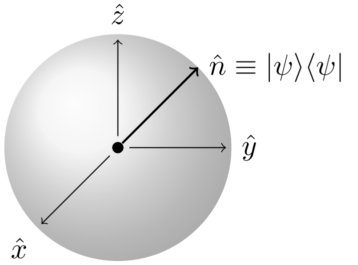


Figure 1.7: Bloch sphere of radius one. $\hat{n}$ represents some ket $|\psi\rangle$. Every point on the sphere corresponds to a unique ket.

Let's discuss the Born rule using the concept of the Bloch vector. First, we notice that

$$P(a) = |a\rangle\langle a| = \frac{1}{2}(1 + (-1)^a \hat{z}\cdot\vec{\sigma}).$$

To see this, remember that we encoded $|0\rangle$ as a column vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle\langle 1|$ with $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Now,

$$p(a|\psi) = Tr(P(a)|\psi\rangle\psi|) = \frac{1}{2}(1 + (-1)^a \hat{z}\cdot\hat{n}).$$

This is a very nice formula because it depends on the scalar product between two Bloch vectors: one corresponding to the projector P_a and the other one to the vector $\hat{n}$ representing the state $|\psi\rangle\langle\psi|$. But the scalar product $\hat{z}\cdot\hat{n}$ is symmetric, i.e., $\hat{z}\cdot\hat{n}=\hat{n}\cdot\hat{z}$. What is a physical meaning of this symmetry? We can definitely write that

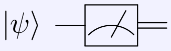
$$p(a|\psi)=p(\psi|a)$$

and this suggests that $p(\psi|a)$ is the probability of measuring outcome ψ for the ket $|a\rangle$. But what is outcome ψ? Outcome a in $p(a|\psi)$ has two values $a=0,1$ but we have only one ψ. To solve this puzzle, note that if the ket $|\psi\rangle$ corresponds to the Bloch vector $\hat{n}$, the Bloch vector $-\hat{n}$, corresponds to the ket $|\psi^{\perp}\rangle$ such that $\langle\psi|\psi^{\perp}\rangle=0$. $|\psi\rangle$ and $\psi^{\perp}\rangle$ form an orthonormal basis just like the kets $|0\rangle,|1\rangle$ form the computational basis. To sum it up, we see that the Born rule works for an arbitrary basis and any pair of antipodal vectors on the Bloch sphere corresponds to some basis in Hilbert space.

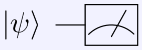
A remark about notation: Orthogonal projectors in the computational basis $|0\rangle,|1\rangle$ are written as $P(a)=|a\rangle\langle a|$ where $a=0,1$. If we change the basis, to a basis given by vectors $\hat{n},-\hat{n}$, we use $P(a|\hat{n})$ such that $P(0|\hat{n})$ corresponds to $\hat{n}$ and $P(1|\hat{n})$ to $-\hat{n}$.

Measurement over arbitrary basis

In this box, we illustrate how to measurement the qubit state over arbitrary chosen basis. For a quantum circuit, we usually assume that our measurement is over the Pauli Z basis $|0\rangle,|1\rangle$, and we use the following notation:



Here we have used double line to represent the classical outcome, which is also a standard notation in quantum circuit, the classical wires are represented as double-line. Notice that in many situation, we omit the classical double-line corresponding to the classical outcomes and simply denote the measurement as



This means that we measure state $|\psi\rangle$ over $|0\rangle,|1\rangle$ basis and obtain

$$p(0)=|\langle 0|\psi\rangle|^2,\quad p(1)=|\langle 1|\psi\rangle|^2. \tag{1.18}$$

But how can we measure over an arbitrary orthonormal basis $|\chi_+\rangle$ and $|\chi_-\rangle$? That is, how to obtain

$$p(\chi_+)=|\langle\chi_+|\psi\rangle|^2,\quad p(\chi_-)=|\langle\chi_-|\psi\rangle|^2. \tag{1.19}$$

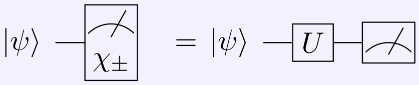
To achieve this goal, we can use a unitary U to rotate the basis from $|\chi_+\rangle$ to $|0\rangle$ and $|\chi_-\rangle$ to $|1\rangle$:

$$U=|0\rangle\langle\chi_+|+|1\rangle\langle\chi_-|. \tag{1.20}$$

We can rewrite Eq. (1.19) as

$$p(\chi_+) = |\langle 0|U|\psi\rangle|^2, \quad p(\chi_-) = |\langle 1|U|\psi\rangle|^2. \tag{1.21}$$

If you cannot figure this out directly, treat it as an exercise and check it in details. Now the measurement over the $|\chi_+\rangle$ and $|\chi_-\rangle$ can be implemented in quantum circuit as



Exercise 14. *1. Consider a basis made of two kets $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$.*

(a) Find Bloch vectors $\hat{n}_\pm$ for the kets $|\pm\rangle$ and show that $\hat{n}_+ = -\hat{n}_-$.

(b) Write the projectors $P(a|\hat{n}_+)$ using the Bloch vectors from (a).

(c) What are the probabilities of measuring projectors $P(a|\hat{n}_+)$ for a ket given by some arbitrary Bloch vector $\hat{n}$?

Exercise 15. *Show that:*

1. $X^2 = Y^2 = Z^2 = 1$.
2. $X = X^\dagger, Y = Y^\dagger, Z = Z^\dagger$
3. $XY = iZ$, $YZ = iX$ *and* $ZX = iY$.
4. $XY + YX = 0$, $YZ + ZY = 0$ *and* $XZ + ZX = 0$
5. $Tr(X) = Tr(Y) = Tr(Z) = 0$

Exercise 16. *Prove that:*

1. *The* $50-50$ *beam splitter in the Mach-Zehnder interferometer rotates the Bloch vector $\hat{n}$ corresponding to an arbitrary photon's polarisation state $|\psi\rangle$ around the x axis clockwise by 90^o degrees.*
2. *The Mach-Zehnder phase shifter Φ rotates the Bloch vector $\hat{n}$ corresponding to an arbitrary photon's state $|\psi\rangle$ around the z axis clockwise by the angle of ϕ.*

What is the physical meaning of a pure state of a qubit? So far we only played with mathematics but physics is something more than mathematical formulas. One way to look at it is this: if you give me a pure state $|\psi\rangle$ I can always find a unitary operation U such that $U|\psi\rangle = |0\rangle$. In the lab, it means that I can always find a physical operation on the qubit such that after it's done I always find the qubit in the state $|0\rangle$, i.e., if I measure it in the computational basis I get outcome '0' with probability one. This means that I can have the most complete information about my qubit.

Example 1. *I can transform the pure state* $|i_+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ *to* $|0\rangle$ *with a* $50-50$ *beamsplitter* B*.*

Now, I'll show you that you can have different kind of states in the lab. We call them *mixed states*, *density matrices* or *density operators*.

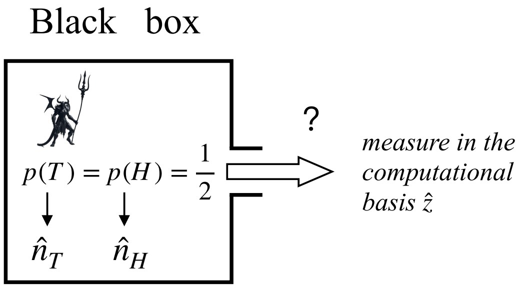

Black box
$p(T) = p(H) = \frac{1}{2}$
$\hat{n}_T$
$\hat{n}_H$
?
measure in the computational basis $\hat{z}$

Figure 1.8: Black box experiment

Let's play a game. Someone in a black box throws a two-sided unbiased coin. When she gets *tails*, she prepares a qubit in a pure state given by a Bloch vector $\hat{n}^T$ and if she gets *heads* she prepares a pure state with a Bloch vector $\hat{n}^H$. Each qubit she prepared is sent to you and you measure it in the computational basis. You get the following conditional probabilities:

$$\begin{aligned} p(k|\hat{n}^T) &= \frac{1}{2}\left(1+(-1)^k\hat{n}^T\cdot\hat{z}\right) \\ p(k|\hat{n}^H) &= \frac{1}{2}\left(1+(-1)^k\hat{n}^H\cdot\hat{z}\right). \end{aligned} \tag{1.22}$$

Using old good Bayes we get that the total probability of getting outcome k is

$$p(k) = p(T)p(k|\hat{n}^T) + p(H)p(k|\hat{n}^H),$$

for the unbiased coin, i.e., $p(T) = p(H) = \frac{1}{2}$, we get

$$p(k) = \frac{1}{2}\left(1+(-1)^k\frac{\hat{n}^T+\hat{n}^H}{2}\cdot\hat{z}\right).$$

Let's put $\frac{\hat{n}^T+\hat{n}^H}{2} = \vec{m}$. Obviously, the length of this new vector is less than one unless $\hat{n}^T = \hat{n}^H$. Can you find a basis $\hat{b}$ such that $p(0|\hat{b}) = 1$? The best you can do is to chose

$\hat{b} = \frac{\vec{m}}{|\vec{m}|}$ for which you get

$$p(k|\hat{b}) = \frac{1}{2}\left(1 + \frac{(-1)^k}{\sqrt{2}}\sqrt{1 + \hat{n}^T \cdot \hat{n}^H}\right).$$

It's easy to check that if $\hat{n}^T$ and $\hat{n}^H$ are not parallel this probability is less than one. This means that what comes out of the box isn't a pure state. It's a statistical mixture of two pure states and thus I lost my complete information about two states prepared in the box.

We observe that if we define an operator $\rho := \frac{1}{2}(1 + \vec{m} \cdot \vec{\sigma})$ then $Tr(\rho P(k|\hat{b})) = p(k|\hat{b}, \hat{m})$ (note that I indicated in this probability the basis of measurement and the vector pertaining to the ρ to indicate its dependence on those variables). If the coin is biased, the vector $\vec{m} = p(T)\hat{n}^T + p(H)\hat{n}^H$ and the rest follows. Therefore, what comes out of the box can be described by the operator ρ characterised by a Bloch vector $\vec{m}$ lying anywhere inside the Bloch sphere if we chose $p(T), p(H), \hat{n}^T, \hat{n}^H$ accordingly. So we can define a new class of states ρ that are called mixed states

$$\rho := \frac{1}{2}(1 + \vec{m} \cdot \vec{\sigma})$$

with $|\vec{m}| \leq 1$. If $|\vec{m}| = 1$ we recover pure states. Mathematical properties of mixed states are

1. $Tr(\rho) = 1$.
2. $\rho \geq 0$ meaning that for every pure state $|\psi\rangle$, $\langle\psi|\rho|\psi\rangle \geq 0$.

The above is equivalent to saying that measurements in any basis yield positive probabilities, i.e., numbers that sum up to one 1 and are positive 2. Born rule remains the same, i.e., $p(k|\hat{b}, \rho) = Tr(P(k|\hat{b})\rho)$.

Remark 3. *Note that $\vec{m}$ can be written in infinitely many ways as $\sum_l p(l)\hat{n}^l$. This means that any mixed state can be prepared in infinitely many ways and we can't tell unless we are told how it was prepared. For instance, let's take $\vec{m} = 0$. This can be prepared by mixing two pure states corresponding to an arbitrary basis or by randomly mixing infinitely many pure states with their Bloch vectors summing up to the zero vector.*

Remark 4. *If A is positive, i.e., $A \geq 0$ then it is automatically Hermitian. So all mixed states are Hermitian.*

Remark 5. *Mixed states in higher dimensional Hilbert spaces are any positive operators with trace equal to one.*

What happens to ρ after you measure it in an arbitrary basis given by a Bloch vector $\hat{b}$? Quantum theory requires a change in the state after the measurement, called *state collapse.*

If you get outcome k your state changes to

$$\frac{P(k|\hat{b})\rho P(k|\hat{b})}{Tr(P(k|\hat{b})\rho)}.$$

State collapse is something that has always bothered physicists because before a measurement states evolve unitarily and we know exactly what the state is at any time in the future. Collapse 'destroys' this clear picture. But that's how nature behaves, end of story.

Exercise 17. *Prove that* $\frac{P(k|\hat{b})\rho P(k|\hat{b})}{Tr(P(k|\hat{b})\rho)}$ *is a pure state.*

Exercise 18. *Consider the following experiment:*

1. *In stage one, prepare a mixed qubit state described by a Bloch vector* $\vec{m}$*.*

2. *In stage two, measure this qubit in the basis described by a Bloch vector* $\hat{n}_1$*. If you get the outcome* -1*, throw away the qubit and keep it if you get the outcome* $+1$*.*

3. *In stage three, take the 'surviving' qubit and subject it to a measurement in the basis described by a Bloch vector* $\hat{n}_2$*. Calculate the probability distribution of the last measurement.*

4. *Now, perform the same sequence of measurements but this time do not reject anything. Calculate the probability of getting the sequence of outcomes* $n_1 = \pm 1$ *followed by* $n_2 = \pm 1$*, i.e.,* $p(n_2, n_1|\vec{m})$*.*

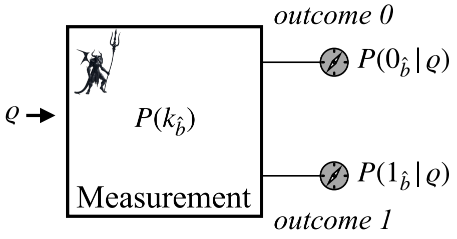
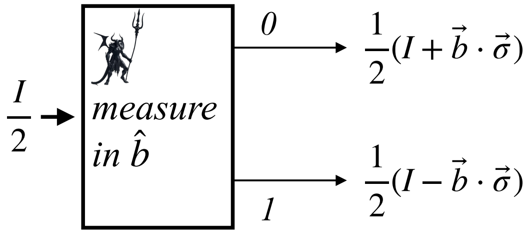


Figure 1.9: State collapse

We close this section with a remark: a measurement in basis $\hat{b}$ can serve as a preparation procedure for a pure state with the Bloch vector $\pm\hat{b}$. Let's take an extreme example of a measurement on the white noise $\rho = \frac{1}{2}$. If you get outcome k in the basis $\hat{b}$ you collapsed white noise into the pure state with the Bloch vector $(-1)^k\hat{b}$. Each outcome happens with probability $\frac{1}{2}$ so you can prepare your desired state in half of the cases.

Summary of 1.5:



$\frac{I}{2}$
measure in $\hat{b}$
0
1
$\frac{1}{2}(I + \vec{b} \cdot \vec{\sigma})$
$\frac{1}{2}(I - \vec{b} \cdot \vec{\sigma})$

Figure 1.10: State preparation

- Physical interpretation of kets as pieces of information about a qubit
- Concept of Bloch vector and Bloch sphere
- Projective measurements in different bases
- Qubit pure and mixed states
- Collapse of quantum state after measurement

1.6 Two qubits

One qubit is cool but you can't do much with it. Really interesting things happen for two or more qubits. Let's learn how to handle such cases.

In our mathematical description of bipartite systems we need to account for a situation where each qubit can be accessed individually. Imagine Alice and Bob, each in their own laboratory located in different places. Both of them have their computational basis $|a\rangle_A$ ($a = 0, 1$) and $|b\rangle_B$ ($b = 0, 1$). A joint Hilbert space is a tensor product of Alice's and Bob's Hilbert spaces, spanned by the orthonormal basis $|a\rangle_A \otimes |b\rangle_B = |a\rangle_A |b\rangle_B = |a, b\rangle_{AB}$ or simply $|a, b\rangle$. The pure state $|a, b\rangle$ means that Alice's qubit is in the state $|a\rangle_A$ and Bob's qubit in the state $|b\rangle_B$[11]. A corresponding bra is $\langle a, b|$ – pay attention to the order of a and b here!

[11] Prepare for a different notation where sometimes we attach subscripts indicating Alice and Bob and sometimes we don't. As long as we know which qubit is which, we are good.

The basis states can be written as 4 dimensional vectors if you wish.

$$|a,b\rangle = \begin{bmatrix} \delta_{a,0} \\ \delta_{a,1} \end{bmatrix} \otimes \begin{bmatrix} \delta_{b,0} \\ \delta_{b,1} \end{bmatrix} = \begin{bmatrix} \delta_{a,0}\delta_{b,0} \\ \delta_{a,0}\delta_{b,1} \\ \delta_{a,1}\delta_{b,0} \\ \delta_{a,1}\delta_{b,1} \end{bmatrix}.$$

Any ket can be decomposed in this basis $|\psi\rangle = \sum_{a,b} \psi_{a,b}|a,b\rangle$ where $\sum_{a,b} |\psi_{a,b}|^2 = 1$, which is the manifestation of Born rule: a probability of finding Alice's qubit in the state $|a\rangle$ and Bob's in the state $|b\rangle$ is $|\psi_{ab}|^2$. As before we can write Born rule as $p(a,b||\psi\rangle) = Tr(P(a) \otimes P(b)|\psi\rangle\langle\psi|)$.

Remark 6. *We define two-qubit trace in complete analogy with the single qubit trace, i.e., $Tr(|\psi\rangle\langle\phi|) := \langle\phi|\psi\rangle$. For instance $Tr(|a,b\rangle\langle c,d|) = \langle c,d|a,b\rangle = \langle c|a\rangle\langle d|b\rangle$.*[12]

Exercise 19. *Show that $Tr(P(a) \otimes P(b)|\psi\rangle\langle\psi|) = |\langle a,b|\psi\rangle|^2$ for an arbitrary $|\psi\rangle$.*

Two qubit states evolve via unitary operations in a similar fashion as single qubit states but there are more, physically different, scenarios.

1. Alice evolves her qubit and Bob doesn't: $U_A \otimes I_B|\psi\rangle_{AB}$.
2. Alice doesn't evolve her qubit and Bob does: $1_A \otimes U_B|\psi\rangle_{AB}$.
3. Both evolve their qubits separately: $U_A \otimes U_B|\psi\rangle_{AB}$.
4. If Alice and Bob share the same laboratory they can make their qubits interact and then they evolve them globally: $U_{AB}|\psi\rangle_{AB}$.

Example 2. *Consider the following global unitary transformation*

$$U_{AB} = |\psi_-\rangle\langle 0,0| + |\psi_+\rangle\langle 0,1| + |\phi_-\rangle\langle 1,0| + |\phi_+\rangle\langle 1,1|.$$

It takes the computational basis to the basis made of the so called Bell state: $|\psi_\pm\rangle = \frac{1}{\sqrt{2}}(|0,1\rangle \pm |1,0\rangle), |\phi\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle \pm |1,1\rangle)$.

Exercise 20. *Prove that the above U_{AB} is unitary.*

Example 3. *Consider an experiment where spatially separated Alice (Singapore) and Bob (Shanghai) feed their qubits in the state $|0,0\rangle$ into their respective Mach-Zehnder interfer-*

[12] We aren't pedantic with the notation here because it's clear which state describes Alice's and Bob's qubit.

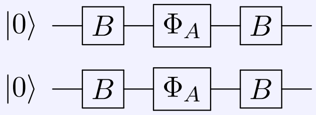
ometers. We have the following unitary evolution

$$\begin{aligned} |0,0\rangle \to B\otimes B|0,0\rangle \quad &\to \\ \Phi_A\otimes\Phi_B B\otimes B|0,0\rangle = \Phi_A B\otimes\Phi_B B|0,0\rangle \quad &\to \\ B\otimes B\Phi_A B\otimes\Phi_B B|0,0\rangle = B\Phi_A B\otimes B\Phi_B B|0,0\rangle \quad &= \\ B\Phi_A B|0\rangle B\Phi_B B|0\rangle. & \end{aligned} \tag{1.23}$$

It's obvious that $p(a,b||0,0\rangle) = p(a||0\rangle)p(b||0\rangle)$*, i.e., Alice's and Bob's measurement outcomes aren't correlated.*

Quantum circuit for Example 3

The operation in Example 3 can be represented in a quantum circuit if we treat B*,* Φ_A *and* Φ_B *as quantum gates.*



Let's repeat this calculation for $|\Phi_+\rangle$*. We have*

$$p(a,b|\Phi_+) = |\langle a,b|B\Phi_A B\otimes B\Phi_B B|\Phi_+\rangle|^2 = \frac{1}{4}\Big(1+(-1)^{a+b}\cos(\phi_A-\phi_B)\Big).$$

A quick glance reveals that now the measurement outcomes are correlated because cosine function of the local phases doesn't factorize. If you wish you can calculate the correlations function between Alice's and Bob's measurements to see this more clearly:

$$C = \sum_{a,b=0,1}(-1)^{a+b}p(a,b|\Phi_+) = \cos(\phi_A-\phi_B).$$

For a suitable choice of their local phases the outcomes can be perfectly correlated $C=1$ *or perfectly anti-correlated* $C=-1$[13].

There is a neat way to represent two-qubit pure states. We write $|\psi\rangle_{AB}$ as a projector $|\psi\rangle\langle\psi|_{AB}$ and expand it in the operator basis made of Pauli operators: $1\otimes 1, 1\otimes X, 1\otimes Y, 1\otimes Z, X\otimes 1, Y\otimes 1, Z\otimes 1, X\otimes X, X\otimes Y \ldots Z\otimes Z$.

I don't want to spend too much time on linear algebra here but you can show that this is the maximal set of linearly independent operators and thus any other operator can be written as their linear combination $A=\frac{1}{4}\sum_{k,l=0}^{3}A_{k,l}\sigma_k\otimes\sigma_l$, where $\sigma_0=1, \sigma_1=X, \sigma_2=Y, \sigma_3=Z$. The expansion coefficients are given by $A_{k,l}=Tr(A\sigma_k\otimes\sigma_l)$[14].

[13] Exercise caution here. If $C=1$ it could be that Alice and Bob always get outcome 0. But it also could be that they both get 0 with probability $\frac{1}{2}$ and 1 with the same probability. Only the latter is an example of perfect correlations while the former isn't.

[14] We need $\frac{1}{4}$ because trace of any σ_k^2 is $Tr(\sigma_k^2)=2$.

Exercise 21. *Prove that the set of* $\sigma_k \otimes \sigma_l$ *is linearly independent, i.e.,* $\sum_{k,l=0}^{3} \lambda_{kl} \sigma_k \otimes \sigma_l = 0$ *if and only if all* $\lambda_{k,l} = 0$.

If you want to expand $|\psi\rangle\langle\psi|_{AB}$ as a linear combination of $\sigma_k \otimes \sigma_l$ you need to ensure that $Tr(|\psi\rangle\langle\psi|_{AB}) = 1$ and thus $A_{00} = 1$. Then this follows

$$|\psi\rangle\langle\psi|_{AB} = \frac{1}{4}\left(1 \otimes 1 + \vec{a} \cdot \vec{\sigma} \otimes 1 + 1 \otimes \vec{b} \cdot \vec{\sigma} + \sum_{k,l=1}^{3} T_{k,l} \sigma_k \otimes \sigma_l\right).$$

The components of $\vec{a}$ and $\vec{b}$ are $a_i = Tr(\sigma_i \otimes 1 |\psi\rangle\langle\psi|_{AB}), b_j = Tr(1 \otimes \sigma_j |\psi\rangle\langle\psi|_{AB})$ and $T_{kl} = Tr(\sigma_k \otimes \sigma_l |\psi\rangle\langle\psi|_{AB})$.

Let's analyze this more to get some physics out of this mathematical jungle.

Suppose Alice measures her qubit in the computational basis and Bob does nothing. Alice's measurement outcomes a appear with probabilities

$$p(a|\psi_{AB}) = Tr(P(a) \otimes 1 |\psi\rangle\langle\psi|_{AB}) = \frac{1}{2}(1 + (-1)^a \hat{z} \cdot \vec{a}).$$

Exercise 22. *Prove that* $p(a||\psi\rangle\langle\psi|) = Tr(P(a) \otimes 1 |\psi\rangle\langle\psi|_{AB}) = \frac{1}{2}(1 + (-1)^a \hat{z} \cdot \vec{a})$.

This result hints at $\vec{a}$ being a local Bloch vector for Alice's part of the bipartite state $|\psi\rangle_{AB}$. If we repeat this calculation for Bob, i.e., calculate $p(b||\psi\rangle\langle\psi|) = Tr(1 \otimes P(b)|\psi\rangle\langle\psi|_{AB})$, we will see that $\vec{b}$ plays a similar role. Can we understand it better? Yes, but we need another mathematical operation called *partial trace*. It's simple

$$\begin{aligned} Tr_A(|a,b\rangle\langle c,d|) &:= \langle c|a\rangle |b\rangle\langle d| \\ Tr_B(|a,b\rangle\langle c,d|) &:= \langle d|b\rangle |a\rangle\langle c|. \end{aligned} \tag{1.24}$$

We call Tr_A partial trace over Alice's qubit and Tr_B partial trace over Bob's qubit. Obviously we have $Tr_A(O_A \otimes O_B) = Tr(O_A)O_B$ and $Tr_B(O_A \otimes O_B) = Tr(O_B)O_A$ for arbitrary operators O_A, O_B.

Let's calculate these partial traces for $|\psi\rangle\langle\psi|_{AB}$.

$$\begin{aligned} Tr_A(|\psi\rangle\langle\psi|_{AB}) &= \frac{1}{2}\left(1 + \vec{b} \cdot \vec{\sigma}\right) \\ Tr_B(|\psi\rangle\langle\psi|_{AB}) &= \frac{1}{2}\left(1 + \vec{a} \cdot \vec{\sigma}\right). \end{aligned} \tag{1.25}$$

We have now enough evidence to claim that if I do a partial trace over Alice (Bob) I get Bob's (Alice's) density matrix.

Example 4. $Tr_{A(B)}(|\psi_-\rangle\langle\psi_-|_{AB}) = \frac{1_{B(A)}}{2}$. *The Bell state $|\psi_-\rangle_{AB}$ represents the maximal knowledge about Alice's and Bob's qubits but their respective qubits are in the maximal disorder.*

Exercise 23. *Show that*

1. $|\psi_-\rangle\langle\psi_-|_{AB} = \frac{1}{4}\left(1\otimes 1 - X\otimes X - Y\otimes Y - Z\otimes Z\right).$
2. $p(a,b|\hat{a},\hat{b},\psi_-^{AB}) = \frac{1}{4}\left(1-(-1)^{a+b}\hat{a}\cdot\hat{b}\right).$
3. $C_{\psi_-} = -\hat{a}\cdot\hat{b}..$

Summary of 1.6:

- Computational basis for two qubits as a tensor product of individual computational bases
- Two qubit operators
- Born rule for two qubits
- Partial trace and its physical meaning

1.7 Mach-Zehnder revisited

We are now equipped with enough quantum engineering know-how to look inside Mach-Zehnder interferometer and gain more intuitions about weird aspects of quantum superposition. Let's first design a quantum circuit to implement the interferometer. Here, we will use a different representation of a $50-50$ beamsplitter that is often used in quantum information theory and can be easily implemented on a quantum computer.

Mach-Zehnder consists of two single-qubit Hadamard gates

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

and one phase gate

$$\Phi = \exp\{(i\phi)\}|0\rangle\langle 0| + |1\rangle\langle 1|.$$

If we input to the circuit a qubit in the state $|0\rangle$, the state becomes a superposition state $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. This is a rather weird state because (i) if you measure the qubit in the computational basis you find it randomly distributed in "up" and "down" arms of the interferometer (ii) it enables interference fringes because we can "imprint" in it phase ϕ and observe it after the second Hadamard gate.

Quantum circuit for Mach-Zehnder interferometer

Let us consider how to using quantum circuit to represent the Mach-Zehnder interferometer. As we have mentioned before, there are two Hadamard gates and one phase gate.
The Hadamard gate can be represented as

$$-\boxed{H}-$$

The phase gate can be represented as

$$-\boxed{\Phi}-$$

The quantum circuit for Mach-Zehnder interferometer is the following

$$-\boxed{H}-\boxed{\Phi}-\boxed{H}-$$

If we input to the circuit a qubit in the state $|0\rangle$, we can check that after first Hadamard gate we obtain

$$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{1.26}$$

Then applying phase gate Φ we obtain

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto \frac{1}{\sqrt{2}}(e^{i\phi}|0\rangle + |1\rangle). \tag{1.27}$$

Finally, applying the second Hadamard gives

$$\frac{1}{\sqrt{2}}(e^{i\phi}|0\rangle + |1\rangle) \mapsto \frac{1}{2}[(e^{i\phi}+1)|0\rangle + (e^{i\phi}-1)|1\rangle]. \tag{1.28}$$

What does this mean? This means that, if we measure over $|0\rangle$, the probability is

$$p(0) = |\frac{1}{2}(e^{i\phi}+1)|^2 = \frac{1}{2}(1+\cos\phi). \tag{1.29}$$

Similarly, when we measure over $|1\rangle$, the probability is

$$p(0) = |\frac{1}{2}(e^{i\phi}-1)|^2 = \frac{1}{2}(1-\cos\phi). \tag{1.30}$$

It enables interference fringes.

Suppose, you measured the qubit after the first Hadamard gate and you got the outcome 0 thus collapsing $H|0\rangle$ back to $|0\rangle$. This completely destroys interference because you can't imprint phase ϕ anymore onto your qubit. Since the collapse is acquisition of information- you now know the qubit is in the ”up” arm - you suspect that this acquisition has something to do with interference destruction. Can we investigate it further?

Yes, we can :-) Let’s attempt to acquire information about which arm the qubit took in a different, perhaps more subtle, manner. Since we are dealing with quantum phenomena, we need to employ quantum mechanics. The idea is to introduce a method of marking the path within the interferometer, allowing us to ’observe’ this marker and thereby determine the qubit’s position. Here’s one way: let’s introduce another qubit (a marker) and make it interact with the qubit inside the interferometer via the following two-qubit unitary operation

$$C[V(\theta)] = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes V(\theta),$$

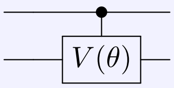
where $V(\theta) = \cos(\theta)I + i\sin(\theta)X$. Notice $[C[V(\theta)]$ can be interpreted as a conditional quantum gate: if the qubit inside the interferometer is in the ”up” arm, do nothing on the marker qubit, else apply $V(\theta)$ to the marker qubit.

If we start with the marker qubit in the state $|0\rangle$, after the firstly apply Hadamard on the first qubit and then apply $C[V(\theta)]$ to both qubits, we get

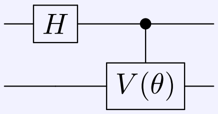
$$|0,0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|0,0\rangle + |1\rangle V(\theta)|0\rangle). \tag{1.31}$$

Quantum circuit of two-qubit Mach–Zehnder interferometer

Represented as quantum circuit, the controlled operation is represented as

$C[V(\theta)] =$ 

The circuit corresponds to Eq. (1.31) is of the following form



Since after this step the marker qubit never interacts with the interfering qubit we can now look at its quantum state:

$$\rho_{marker} = \frac{1}{2}(|0\rangle\langle 0| + V(\theta)|0\rangle\langle 0|V(\theta)^{\dagger}). \tag{1.32}$$

It has a clear interpretation: you receive the states $|0\rangle$ and $V(\theta)|0\rangle$ with equal probabilities. Take $V(\theta = \frac{\pi}{2})|0\rangle = i|1\rangle$, in which case you get two orthogonal quantum states you can experimentally distinguish and thus you know where the interference qubit is with 100% accuracy. And you get this information without measuring the qubit in the interferometer! But what happens then to the interference? To answer this we need to trace out the marker qubit, arriving at the state ρ_{int} of the qubit in the interferometer.

$$\rho_{int} = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1| + \langle 0|V^{\dagger}(\frac{\pi}{2})|0\rangle|0\rangle\langle 1| + h.c.). \tag{1.33}$$

But $V(\frac{\pi}{2})|0\rangle = i|1\rangle$ so ρ_{int} becomes pure noise and the interference is gone.

This is not the end of the story. If θ in V is not 0 or $\frac{\pi}{2}$ our marker is in two non-orthogonal states $|0\rangle$ and $V(\theta)|1\rangle$. Non-orthogonal quantum states are not perfectly distinguishable and thus our marker becomes ambiguous. We will discuss this situation in the next chapter.

Exercise 24.

1. *Show that $C[V(\theta)]$ is unitary*
2. *Show that $V(\theta) = e^{i\theta X}$*
3. *Derive ρ_{int} and ρ_{marker}*
4. *Show that there is no interference for $V(\frac{\pi}{2})$ and full interference when $V(0) = I$.*

Summary of 1.7:

- Hadamard gate as a different representation of a $50 - 50$ beamsplitter
- Mach-Zehnder as a quantum circuit
- Quantum interference depends on what knowledge we possess about interfering system

Chapter 2

Elements of Quantum Information

2.1 Bell Theorem

Consider a simple experiment[1] involving two spatially separated parties, Alice and Bob, who share a singlet state $|\psi_-\rangle$. Alice (Bob) measures her (his) qubit in one of two randomly chosen bases, defined by the Bloch vectors $\hat{a}, \hat{a}'$ $(\hat{b}, \hat{b}')$. This results in four sets of four-outcome probability distributions.

$$\begin{aligned}
p(a,b|\psi_-) &= \frac{1}{4}\left(1-(-1)^{a+b}\hat{a}\cdot\hat{b}\right) \\
p(a,b'|\psi_-) &= \frac{1}{4}\left(1-(-1)^{a+b'}\hat{a}\cdot\hat{b}'\right) \\
p(a',b|\psi_-) &= \frac{1}{4}\left(1-(-1)^{a'+b}\hat{a}'\cdot\hat{b}\right) \\
p(a',b'|\psi_-) &= \frac{1}{4}\left(1-(-1)^{a'+b'}\hat{a}'\cdot\hat{b}'\right). \qquad (2.1)
\end{aligned}$$

Note that we mapped $0,1$ outcomes to ± 1 outcomes via the mapping $(-1)^k$, where $k = 0, 1$. This is for convenience only and it isn't in any way fundamental.

In 1935, Einstein, Podolsky, and Rosen asked if these sets of probabilities can be explained within the framework of *local realism*.[2] What is local realism?

[1] Simple conceptually, though it requires considerable skill to execute in the lab.

[2] They asked a slightly different question. For details, check out their (in)famous paper.

- **Realism:** physical quantities, i.e., parameters you can measure in experiments, should exist independently from an observer.
- **Locality:** information cannot be transferred faster than the speed of light.

Realism + *Locality* = *Local Realism.* However philosophical it sounds, all classical theories of matter are local and realistic. Local realism also agrees with common sense. If I don't look at the Moon now, I'm sure it's still there.

How to apply local realism to our experiment?

Realism: We encode local measurement outcomes via functions $I_A(\hat{x}, \lambda) = \pm 1, I_B(\hat{y}, \lambda) = \pm 1$ where $\hat{x} = \hat{a}, \hat{a}'$, $\hat{y} = \hat{b}, \hat{b}'$, and λ is some 'hidden' parameter carried by qubits that determines the measurement outcomes. The parameter λ is distributed with some probability $p(\lambda)$ to account for the inherent randomness we observe in the atomic world. In fact, we can always assume that there are 16 different λs that serve as instructions for what Alice's and Bob's measurements should give:

$$\begin{aligned} \lambda_1 &= (0,0|0,0) \\ \lambda_2 &= (0,0|0,1) \\ \lambda_3 &= (0,0|1,0) \\ \lambda_{15} &= (0,1|1,1) \\ \lambda_{16} &= (1,1|1,1), \end{aligned} \tag{2.2}$$

where $(a, a'|b, b')$ means give outcome a if Alice measures in $\hat{a}$ basis, outcome a' if she measures in $\hat{a}'$ basis, outcome b if Bob measures in $\hat{b}$ basis and outcome b' if he measures in $\hat{b}'$ basis. For instance $I_A(\hat{a}, \lambda_3) = (-1)^0, I_B(\hat{b}, \lambda_3) = (-1)^1$.

Locality: Alice's measurement outcomes don't depend on Bob's settings and vice versa: $I_A(\hat{x}, \lambda)$ $(I_B(\hat{y}, \lambda))$ is only a function of local setting $\hat{x}(\hat{y})$. This is ensured by random choices of Alice's and Bob's measurement settings. If they choose their setting perfectly at random before every measurement, the finite speed of information transfer guarantees that they can't know what the other party have chosen to measure so we **cannot** have $I_A(\hat{x}, \hat{y}, \lambda)$ and $I_B(\hat{y}, \hat{x}, \lambda)$.

We can now calculate a correlation function between Alice and Bob if local realism holds:

$$C_{LR}(\hat{x}, \hat{y}) = \sum_{\lambda} p(\lambda) I_A(\hat{x}, \lambda) I_B(\hat{y}, \lambda).$$

A necessary condition to reproduce the quantum mechanical results is to have:

$$C_{LR}(\hat{x}, \hat{y}) = C_{QM}(\hat{x}, \hat{y}) = -\hat{x} \cdot \hat{y}.$$

Can we have it?

Let's consider the following inequality that any local realistic theory must obey. It was first found by Bell and then slightly improved by Clauser-Horne-Shimony-Holt (CHSH). We call it Bell-CHSH inequality and it reads:

$$-2 \leq C_{LR}(\hat{a},\hat{b}) + C_{LR}(\hat{a},\hat{b}') + C_{LR}(\hat{a}',\hat{b}) - C_{LR}(\hat{a}',\hat{b}') \leq 2.$$

It's easy to prove it. We have

$$\begin{gathered} C_{LR}(\hat{a},\hat{b}) + C_{LR}(\hat{a},\hat{b}') + C_{LR}(\hat{a}',\hat{b}) - C_{LR}(\hat{a}',\hat{b}') = \\ \sum_{\lambda} p(\lambda) \Big(I_A(\hat{a},\lambda)(I_B(\hat{b},\lambda) + I_B(\hat{b}',\lambda)) + I_A(\hat{a},\lambda)(I_B(\hat{b},\lambda) - I_B(\hat{b}',\lambda)) \Big) . \end{gathered} \tag{2.3}$$

If $I_B(\hat{b},\lambda)+I_B(\hat{b}',\lambda) = \pm 2$ then $I_B(\hat{b},\lambda)-I_B(\hat{b}',\lambda) = 0$ and if $I_B(\hat{b},\lambda)-I_B(\hat{b}',\lambda) = \pm 2$ then $I_B(\hat{b},\lambda) + I_B(\hat{b}',\lambda) = 0$ so $C_{LR}(\hat{a},\hat{b}) + C_{LR}(\hat{a},\hat{b}') + C_{LR}(\hat{a}',\hat{b}) - C_{LR}(\hat{a}',\hat{b}')$ cannot exceed 2 and be less than -2 because the sum $\sum_{\lambda} p(\lambda)$ is convex. This ends the proof.

Remark 7. *Why is there one minus in the Bell-CHSH inequality, i.e., the minus in front of $C_{LR}(\hat{a}',\hat{b}')$? It's there to get ±2 bound. If it was $+C_{LR}(\hat{a}',\hat{b}')$, the bound would be ±4. Of course, we can put this minus in front of a different correlation function and the bound still is ±2.*

How about quantum mechanics? For the singlet state $|\psi_-\rangle$ the correlation function reads $-\hat{x}\cdot\hat{y}$ and if Alice choses two orthogonal directions $\hat{a}$ and $\hat{a}'$ and Bob choses $\hat{b} = \frac{1}{\sqrt{2}}(\hat{a}+\hat{a}')$ and $\hat{b}' = \frac{1}{\sqrt{2}}(\hat{a}-\hat{a}')$ we get:

$$C_{QM}(\hat{a},\hat{b}) + C_{QM}(\hat{a},\hat{b}') + C_{QM}(\hat{a}',\hat{b}) - C_{QM}(\hat{a}',\hat{b}') = 2\sqrt{2}. \tag{2.4}$$

This proves that local realism isn't compatible with quantum mechanics. Correlations generated by the singlet $|\psi_-\rangle$ are stronger than any local realistic correlations.

Remark 8. *If all the terms in the Bell-CHSH inequality had plus sign, we wouldn't be able to demonstrate that local realism isn't compatible with quantum mechanics. It was a genius of Bell to put that minus. Some people are suspicious about it. They think that it's cheating. It's not, it's the same as proving that an apple isn't an orange. If you only look at the shape of both fruits, they look the same but if you inspect their surface, you'll see a big difference.*

What does this mean that quantum mechanics violates the Bell-CHSH inequality? It means that either locality or realism or both are not true in the atomic world. We haven't seen any evidence that the locality isn't obeyed by nature so it seems that realism doesn't hold. However innocent this sounds, it's a truly bizarre conclusion if you think about it carefully.

If realism doesn't hold, the Moon isn't there if you don't look at it[3]! More seriously, lack of realism means that if you measure a position of an atom, you cannot infer the the atoms has been at this position before you measured it. Somehow, you measurement creates the atom at the measured position. Crazy, right?

Remark 9. *Quantum mechanics is local. The proof is rather simple. Consider a bipartite pure state*[4] $|\psi\rangle_{AB}$*. We can always write it in the Schmidt form*[5] $|\psi\rangle_{AB} = \sum_k \sqrt{\mu_k}|k\rangle_A|k\rangle_B$*. Suppose Bob performs a measurement in his lab in some basis* $|\beta\rangle_B$*. If he doesn't communicate his measurement results to Alice, she will never know he did any measurement. This is because after he gets the outcome* β *he collapses the state* $|\psi\rangle$ *into* $|\psi_\beta\rangle = \frac{1}{\sqrt{p_B(\beta)}}\sum_k \sqrt{\mu_k}|k\rangle_A\langle\beta|k\rangle_B$ *where* $p_B(\beta)$ *is the probability of getting the outcome* β*. But without communicating the outcome, Alice sees the state*

$$\sum_\beta p_B(\beta)|\psi_\beta\rangle\langle\psi_\beta| = \sum_k \mu_k|k\rangle\langle k|,$$

which doesn't depend on β*.*

2.1.1 Popescu-Rorlich Boxes

Consider Bell inequality in a more abstract framework. In quantum theory, Alice and Bob perform von Neumann measurements with two possible settings each: $\hat{a}, \hat{a}'$ for Alice and $\hat{b}, \hat{b}'$ for Bob, yielding probabilistic outcomes a, a' and b, b'. This scenario can be generalized as boxes with binary inputs $a = 0, 1$ for Alice and $b = 0, 1$ for Bob, producing probabilistic outputs A and B. The system is fully characterized by a set of conditional probabilities:

$$\begin{aligned} &p(A|a) \\ &p(B|b) \\ &p(A, B|a, b). \end{aligned} \tag{2.5}$$

This captures the essence of the Bell experiment, where Bloch vectors are removed as inputs to extend the analysis beyond quantum theory. The original Bell experiment can be described using such boxes. Locality is encoded in the first two conditional probabilities: $p(A|a) = \sum_B p(A, B|a, b)$ and $p(B|b) = \sum_A p(A, B|a, b)$.

Remark 10. *Non-locality, or signalling, would result in* $\sum_B p(A, B|a, b) = p(A|a, b)$ *and* $\sum_A p(A, B|a, b) = p(B|a, b)$*. In this case, Bob (or Alice) could signal to Alice (or Bob) by altering the input* b *(or* a*) to his (or her) box.*

[3] Of course, the Moon isn't an atom but... it's made of many atoms:-)

[4] Prove to yourself that if this is a mixed state, the conclusion is the same.

[5] We will prove it soon.

Now, let's examine the probability distribution generated by the so-called Popescu-Rohrlich (PR) boxes:

$$p(A,B|a,b) = \frac{1}{2}\delta_{A\oplus B,ab},$$

where $\oplus$ denotes sum modulo 2.

Exercise 25. *Prove that PR boxes are local, i.e.,* $\sum_B \frac{1}{2}\delta_{A\oplus B,ab}$ *doesn't depend on* b *and* $\sum_A \frac{1}{2}\delta_{A\oplus B,ab}$ *doesn't depend on* a.

Let's now calculate the PR box correlation function $C_{PR}(a,b)$[6]:

$$C_{PR}(a,b) = \sum_{A,B}(-1)^{A+B}\frac{1}{2}\delta_{A\oplus B,ab} = \delta_{ab,0} - \delta_{ab,1}.$$

If you substitute these PR box correlation functions into the Bell-CHSH inequality, you'll find a value of 4, representing the maximum algebraic violation of local realism. This indicates that, without breaking locality, one could achieve much stronger correlations than those observed in nature. This intriguing observation has drawn significant interest in the study of quantum theory's foundations.

If PR box correlations existed in nature, we would be living in a very peculiar universe. Here's why.

2.1.2 Van Dam Protocol

This is a mind-boggling protocol that might just be the case against PR boxes existing in nature. Plus, it's a very cool thing worth knowing just for the heck of it.

Alice and Bob share PR boxes. Alice has two random bits, a_0 and a_1, and Bob wants to know the value of one of them—either a_0 or a_1. However, Alice can only send Bob one bit. Is this possible?

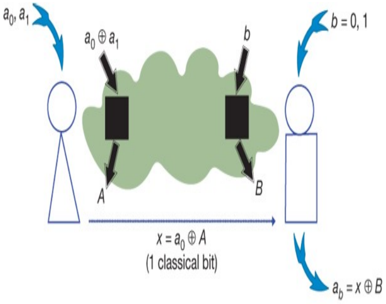
Let's break down some basic protocols: 1) Alice always sends Bob a_0. But if Bob wants a_1, he's stuck with a 50% guess. Not great. 2) What if Alice sends a_0 half the time and a_1 the other half? Nope, that's no better than option 1.

Enter van Dam!

1. Alice calculates $a = a_0 \oplus a_1$.
2. Alice inputs the resulting bit a into her PR box and gets some outcome A.
3. She then calculates a bit $x = a_0 \oplus A$ and sends it Bob.

[6] Once could claim that PR boxes resuscitated a rather stale Bell theorem research.

Figure 2.1: van Dam protocol



4. If Bob wants to learn a_0 he inputs $b = 0$ to his PR box and $b = 1$ if he wants to know a_1.

5. Bob calculates the value β of the bit he wants to know as $\beta = x \oplus B = a_0 \oplus A \oplus B$.

Does it work? If it does then we must have

$$\begin{aligned} p(\beta = a_0 | b = 0) &= 1 \\ p(\beta = a_1 | b = 1) &= 1, \end{aligned} \tag{2.6}$$

i.e., Bob is never wrong. Let's calculate $p(\beta = a_0 | b = 0)$. $\beta = a_0 \oplus A \oplus B$ and it's equal to a_0 only if $A \oplus B = 0$, which happens if $ab = 0$. But the condition is $b = 0$ so it's true. How about $p(\beta = a_1 | b = 1)$? For $\beta = a_1$ we must have $A \oplus B = a_0 \oplus a_1$. But $a = a_0 \oplus a_1$ and thus $ab = a$ (because Bob put $b = 1$), which is true because $a = a_0 \oplus a_1$. This ends the proof.

The van Dam protocol is clever, very clever. It shows that PR boxes, while not breaking any fundamental physical principles, are almost too good to be true. If they existed, we'd be able to compress information in ways that defy reason. And as we all know, there's no free lunch. But in the end, the van Dam protocol doesn't disprove PR boxes, leaving the mystery of why quantum mechanics maxes out at $2\sqrt{2}$ instead of 4 unsolved.

Summary of 2.1:

- **Local Realism:**
  - **Realism:** Physical quantities exist independently of observation.
  - **Locality:** No faster-than-light information transfer.
- **Bell-CHSH Inequality:**
  - Local realism leads to Bell-CHSH inequality - bounded by 2.

  - Quantum mechanics violates Bell-CHSH inequality - gives $2\sqrt{2}$.
  - Quantum mechanical violation of Bell-CHSH inequality means that locality or realism, or both, do not hold.

- **PR Boxes:**
  - Are no-signalling.
  - Maximize Bell-CHSH violation - gives 4 (absolute maximum).

- **Van Dam Protocol:**
  - Uses PR boxes to grant access to two Alice's random bits with one bit of communication only.
  - Seems to be against common sense intuitions.

2.2 Entanglement

Entanglement is a cornerstone of quantum information theory. Discovered by Erwin Schrödinger in the late '30s and later spotlighted in the EPR paper we've discussed, it's central to the field. We'll touch on key aspects of entanglement in our lectures since it's a complex topic that deserves focused attention.

2.2.1 Pure state entanglement

What's so special about the Bell states? We've shown that the singlet $|\psi_-\rangle$ maximally violates the Bell-CHSH inequality[7], and we'll see even more weirdness from these states.

Mathematically, the singlet state (and any Bell state, in fact) can't be written as a product of local states for Alice and Bob, i.e.,

$$|\psi_-\rangle \neq |A\rangle|B\rangle,$$

where $|A\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|B\rangle = \beta_0|0\rangle + \beta_1|1\rangle$ belong to Alice and Bob. States like $|A\rangle|B\rangle$ are called *product states.*

Exercise 26. *Prove that $|\psi_-\rangle \neq |A\rangle|B\rangle$.*

This observation is the basis for the definition of quantum entanglement for pure states: *Any pure bipartite state*[8] *that isn't a product state is called entangled.*

[7] All Bell states violate the Bell-CHSH inequality maximally, but with different measurement settings.

[8] Of any dimension.

There's a handy tool to check if a bipartite state is entangled or not. It's called Schmidt decomposition, and it works like this: Take any bipartite state of dimension D, $|\psi\rangle = \sum_{a,b=1}^{D} \psi_{ab}|a\rangle|b\rangle$. You can always find local orthonormal bases $|\alpha_A\rangle, |\beta_B\rangle$[9] for Alice and Bob such that

$$|\psi\rangle = \sum_{\alpha=1}^{d} \sqrt{\lambda_\alpha}|\alpha\rangle_A|\alpha\rangle_B$$

where $d \leq D$. The proof is straightforward. Let's calculate the reduced density matrix for Alice:

$$\rho_A = \mathrm{Tr}_B(|\psi\rangle\langle\psi|),$$

and find its eigenvectors $|\alpha_A\rangle$ and eigenvalues λ_α. Now, we expand Alice's basis in terms of $|\alpha_A\rangle$[10]:

$$|a\rangle = \sum_{\alpha} |\alpha_A\rangle\langle\alpha_A|a\rangle,$$

and substitute it into $|\psi\rangle$:

$$|\psi\rangle = \sum_{\alpha} |\alpha_A\rangle \left(\sum_{a,b} \psi_{ab}\langle\alpha_A|a\rangle|b\rangle \right).$$

Let's call the states $\sum_{a,b} \psi_{ab}\langle\alpha_A|a\rangle|b\rangle$ by $|\tilde{\alpha}_A\rangle$.

Exercise 27. *Assuming that ρ_A is full rank*[11]*, prove the following:*

1. *Show that $\langle\tilde{\alpha}_A|\tilde{\alpha}'_A\rangle = \lambda_{\alpha_A}\delta_{\alpha_A\alpha'_A}$.*
2. *Show that the eigenvalues of ρ_B are equal to the eigenvalues of ρ_A.*

This wraps up the proof.

The number d is called the Schmidt number, and if $d > 1$, $|\psi\rangle$ is entangled.

2.2.2 Mixed state entanglement

Okay, but what about mixed states? Can we define entanglement for those?

First, let's check if mixed states can violate the Bell-CHSH inequality, since product states definitely don't.

Exercise 28. *Show that the Bell-CHSH inequality can't be violated by product states.*

[9] Figure out why I labeled the states like this.

[10] For simplicity, we assume ρ_A has full rank, but I'll leave that for the mathematically curious.

[11] Check the meaning of full rank in books, online, or whatever source you prefer.

For a warm-up, let's investigate the Werner state:

$$\rho_W = V|\psi_-\rangle\langle\psi_-| + \frac{1-V}{4},$$

with $-\frac{1}{3} \leq V \leq 1$.

Exercise 29. *Show that the Werner state is a valid density matrix for* $-\frac{1}{3} \leq V \leq 1$*. Specifically, show that its trace equals* 1 *and it's a positive operator.*

The correlation function for the Bell-CHSH inequality becomes

$$C_W(\hat{a}, \hat{b}) = VC_{\psi_-}(\hat{a}, \hat{b}).$$

You don't need to fully calculate it, because the white noise for two qubits, $\frac{1}{4}$, has no correlations. Think about it—it's pretty intuitive. :-)

So, as long as $V > \frac{1}{\sqrt{2}}$, you'll get a violation of the Bell-CHSH inequality by $2\sqrt{2}V$. This shows that mixed states can also be entangled in the sense that they can violate local realism, just like pure entangled states. Be cautious, though—I'm not saying entanglement is defined as violating local realism. We didn't define it like that for pure states, and we won't for mixed states. However, if local realism is violated, entanglement is likely in play.

Now, for bipartite product states like $|A\rangle|B\rangle$, these can be prepared locally by Alice and Bob—Alice prepares $|A\rangle$ in her lab, Bob prepares $|B\rangle$ in his. In contrast, bipartite pure entangled states can't be locally prepared. So, how does this extend to mixed states?

There's a class of mixed states that Alice and Bob can prepare locally with classical communication (e.g., using a phone line), which look like this:

$$\sum_i p_i \rho_i^A \otimes \rho_i^B,$$

where p_i is some probability distribution and ρ_i^A, ρ_i^B are arbitrary mixed states. Here's how they prepare them: Alice generates i with probability p_i, prepares her state ρ_i^A, and sends Bob the information so he can prepare his state ρ_i^B. We call such states separable, and if you can't make the state this way, it's entangled. This process—using local operations and classical communication—is called **LOCC**.

So, how do you check if a given bipartite mixed state is entangled? It's a tricky problem because you need to show that you can't decompose the state into a convex combination of tensor products, and this decomposition isn't unique. In fact, there are infinitely many ways to do it! But here's a neat tool from Asher Peres—the positive partial transposition (PPT) criterion.

The partial transposition of a bipartite state, $\rho_{AB}^{T_B}$[12], is defined like this:

$$\rho_{AB}^{T_B} := \left(\sum_{a,b,a',b'} \rho_{ab,a'b'} |a,b\rangle\langle a',b'| \right)^{T_B} = \sum_{a,b,a',b'} \rho_{ab,a'b'} |a,b'\rangle\langle a',b|.$$

Notice how b and b' are swapped.

Peres noticed that if ρ_{AB} is separable, after PT it remains a valid density matrix, which means it's positive. That's because

$$\rho_{sep}^{T_B} = \sum_i p_i \rho_i^A \otimes (\rho_i^B)^T,$$

and for each i, $(\rho_i^B)^T$ is still a valid mixed state. To see this, write any mixed state as $\rho = \sum_k q_k |\psi_k\rangle\langle\psi_k|$ and note that $|\psi_k\rangle\langle\psi_k|^T = |\psi_k\rangle\langle\psi_k|$.

Exercise 30. *Show that for any mixed state ρ, its transposition ρ^T is also a mixed state using Hermitian conjugation.* **Hint:** *Hermitian conjugation combines transposition and complex conjugation.*

Now, let's apply Peres' PPT criterion to the Werner state. First, observe that

$$|\psi_-\rangle\langle\psi_-|^{T_B} = \frac{1}{2} - |\phi_+\rangle\langle\phi_+|,$$

and the two-qubit white noise is invariant under partial transposition[13]. So, we get

$$\rho_V^{T_B} = V \left(\frac{1}{2} - |\phi_+\rangle\langle\phi_+| \right) + \frac{1-V}{4} = -V|\phi_+\rangle\langle\phi_+| + \frac{1+V}{4}.$$

If you decompose the white noise in the Bell basis, you'll get

$$\frac{1}{4} = \frac{1}{4} \left(|\psi_-\rangle\langle\psi_-| + |\psi_+\rangle\langle\psi_+| + |\phi_-\rangle\langle\phi_-| + |\phi_+\rangle\langle\phi_+| \right),$$

and you can read off the eigenvalues: $\frac{1-3V}{4}$ and a 3x degenerate one, $\frac{1+V}{4}$. Only the non-degenerate one can be negative when $V > \frac{1}{3}$, meaning the Werner state is entangled for $V > \frac{1}{3}$. Interestingly, for $\frac{1}{3} < V < \frac{1}{\sqrt{2}}$, the Werner state is entangled **and** has a local realistic description. This remains an unsolved mystery[14].

[12] Similarly for T_A.

[13] This is trivial, no need to show it explicitly.

[14] What are you waiting for? Solve it and celebrate!

2.3 How to measure entanglement?

In the next section, we'll explore how entanglement enables tasks that would be impossible otherwise, making it a valuable resource. To quantify this resource, we need a numerical measure. For simplicity, let's start with two qubits.

A good measure should be zero for separable states (the minimum) and one for all four Bell states (the maximum). While there are other possible criteria, we'll focus on this for now.

One such measure is *concurrence*. It's defined for a bipartite state ρ_{AB} as:

$$C(\rho_{AB}) = \max{(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4)}, \tag{2.7}$$

where the λ's are the eigenvalues, in decreasing order, of the Hermitian matrix:

$$R = \sqrt{\sqrt{\rho_{AB}}\sigma_{AB}\sqrt{\rho_{AB}}}. \tag{2.8}$$

The matrix σ_{AB} is defined as:

$$\sigma_{AB} = Y \otimes Y \rho_{AB}^{*} Y \otimes Y, \tag{2.9}$$

where the complex conjugation $*$ is taken in the eigenbasis of the Z operator, i.e., in the computational basis.

I know this looks convoluted, but hang in there. Let's go through the calculations step by step. First, let's calculate the concurrence for the singlet state $|\psi_-\rangle$. Notice that the complex conjugation in the computational basis doesn't change the singlet state, and $Y \otimes Y|\psi_-\rangle = |\psi_-\rangle$. So, $\sigma_{AB} = |\psi_-\rangle\langle\psi_-|$. Now, clearly, $\rho_{AB} = \sigma_{AB}$, which means:

$$\sqrt{\rho_{AB}} = \rho_{AB},$$

and:

$$\sqrt{\sqrt{\rho_{AB}}\sigma_{AB}\sqrt{\rho_{AB}}} = \rho_{AB} = R.$$

The eigenvalues of R are 0 and 1, since it's an orthogonal projector. It's straightforward to see that $C(|\psi_-\rangle\langle\psi_-|) = 1$. You can check that concurrence for all Bell states is equal to 1 using the same reasoning.

Now, let's calculate the concurrence for the Werner state ρ_V. The operator $Y \otimes Y$ does nothing to it, but we still need to calculate $\sqrt{\rho_V}$. To do that, we need the eigenvalues and eigenvectors of ρ_V:

$$\rho_V = \left(V + \frac{1-V}{4}\right)|\psi_-\rangle\langle\psi_-| + \frac{1-V}{4}|\psi_+\rangle\langle\psi_+| + \frac{1-V}{4}|\phi_+\rangle\langle\phi_+| + \frac{1-V}{4}|\phi_-\rangle\langle\phi_-|.$$

Using this, we get:

$$\sqrt{\rho_V} = \sqrt{\frac{1+3V}{4}}|\psi_-\rangle\langle\psi_-| + \sqrt{\frac{1-V}{4}}|\psi_+\rangle\langle\psi_+| + \sqrt{\frac{1-V}{4}}|\phi_+\rangle\langle\phi_+| + \sqrt{\frac{1-V}{4}}|\phi_-\rangle\langle\phi_-|.$$

Now it's easy to calculate R:

$$R = \frac{1+3V}{4}|\psi_-\rangle\langle\psi_-| + \frac{1-V}{4}|\psi_+\rangle\langle\psi_+| + \frac{1-V}{4}|\phi_+\rangle\langle\phi_+| + \frac{1-V}{4}|\phi_-\rangle\langle\phi_-|.$$

Finally, we find that:

$$C(\rho_V) = \max\left(0, \frac{3V-1}{2}\right).$$

For $V = 1$, we get the maximal value of concurrence, as we should, and for $V = \frac{1}{3}$ (i.e., in the regime of separability), concurrence is zero, which is exactly what we expect.

What other criteria should a good entanglement measure meet? The most important one is that it shouldn't increase under LOCC. This is because entanglement is a resource that can't be created by LOCC. Proving that concurrence follows this rule is beyond the scope of this lecture (but it does).

Summary of 2.3:

- Pure State Entanglement
  - Entangled states, such as the Bell states, cannot be written as a product of local states for two parties, Alice and Bob.
  - The Schmidt decomposition is a tool used to check whether a bipartite state is entangled by analyzing its reduced density matrix. If the Schmidt number is greater than 1, the state is entangled.
  - The singlet state maximally violates the Bell-CHSH inequality, showing how entanglement can lead to non-classical correlations.
- Mixed State Entanglement
  - Mixed states can also be entangled, but detecting and defining it is more complex compared to pure states.
  - The Werner state serves as an example of a mixed state that violates the Bell-CHSH inequality, demonstrating that even noisy states can exhibit entanglement.
  - The Peres criterion, known as the positive partial transposition (PPT) test, helps determine if a mixed state is entangled by checking whether the partial transpose of the state remains a valid density matrix.

- Measuring Entanglement
  - A numerical measure of entanglement called *concurrence* was introduced, which quantifies the degree of entanglement for bipartite states.
  - Concurrence is 0 for separable states and 1 for maximally entangled states, like the Bell states.
  - The concurrence measure does not increase under local operations and classical communication (LOCC), making it a valid resource measure for entanglement.

2.4 Simple quantum information protocols with entanglement

In this section we will discuss a bunch of interesting, yet reasonably simple quantum information protocols that use quantum entanglement. These protocols can't be done without it and thus they can't be done using classical systems.

2.4.1 Quantum Teleportation

In sci-fi literature and films, teleportation is sending objects through space without physically moving them around. The vague ideas is to get a parcel in New York, put it inside a teleporter and press a button. The receiver, say in Tokyo[15], gets the parcel instantaneously[16] and the parcel never moves thru the physical space between these two cities. I'll now show you how to do it for real:-)

In this scenario, Alice gets a qubit HA_0 in some unknown to her state $|\phi\rangle_{A_0} = \alpha_0|0\rangle_{A_0} + \alpha_1|1\rangle_{A_0}$ and her task is to teleport it to Bob. Alice and Bob share the singlet $|\psi_-\rangle_{A_1B}$ and a classical communication line like a telephone line or connected computer terminals[17]. The protocols is as follows:

1. Alice measures her qubits A_0 and A_1 in the Bell basis. She gets four measurement outcomes:
   - $|\psi_-\rangle \to 00$
   - $|\psi_+\rangle \to 01$

[15] Speaking of Tokyo, I made a feature film about quantum Zeno effect in Tokyo: https://www.zenoeffectmovie.com

[16] This isn't crucial.

[17] They could also send letters but that's awfully tacky.

- $|\phi_-\rangle \to 10$
- $|\phi_+\rangle \to 11$.

2. She sends Bob her measurement outcome 00, 01, 10 or 11 using the classical communication line.

3. Bob, upon receiving two bits from Alice, performs a suitable unitary operation on his qubit B:

   - $00 \to 1$
   - $01 \to Z$
   - $10 \to X$
   - $11 \to ZX$.

How the protocol works can be seen easily by rewriting the three qubits' initial state $|\phi_{A_0}\rangle|\psi_-\rangle_{A_1B}$ like this:

$$\begin{aligned}
&|\phi_{A_0}\rangle|\psi_-\rangle_{A_1B} = \\
&\frac{1}{2}(-|\psi_-\rangle_{A_0A_1}|\phi\rangle_B - |\psi_+\rangle_{A_0A_1}Z|\phi\rangle_B + |\phi_-\rangle_{A_0A_1}X|\phi\rangle_B + \\
&|\phi_+\rangle_{A_0A_1}ZX|\phi\rangle_B). \qquad (2.10)
\end{aligned}$$

The above equation is just a different way to write the initial state so nothing magical[18] has happened. Something beautiful happens when Bob receives the outcome of Alice's measurement. If he receives, say, 01, the three qubit state collapsed to $|\psi_+\rangle_{A_0A_1}Z|\phi\rangle_B$ and if he now applies Z to his state he gets $ZZ|\phi\rangle_B = |\phi\rangle_B$! Alice's state $|\phi\rangle$ appears in his laboratory. The same happens for all the other Alice's measurement outcomes.

Exercise 31. *Show that:*

1. *Each of Alice's measurement outcomes happens with probability $\frac{1}{4}$.*
2. *After each of Alice's measurements her qubit A_0 becomes white noise.*

What's so cool about quantum teleportation? First of all, you need infinitely many bits to describe the state $|\phi\rangle_{A_0}$ yet you can transfer it to Bob without sending it thru space with only two bits of classical information. Moreover, Alice doesn't have to know what her state $|\phi\rangle_{A_0}$ is! Perhaps, the ultimate cool factor of this protocol is that you can't do it without quantum mechanics. Why? Because you need quantum entanglement to do this.

[18] Of course, magic doesn't exist.

2.4.2 Super-dense Coding

This protocol is a sort of reversal of quantum teleportation. Alice's taks is now to send Bob two bits of classical information with a qubit. Again, both guys share an entangled state, this time (for fun) it's $|\phi_+\rangle_{AB}$. Unlike in the teleportation protocol, there are only two qubits involved, both coming from $|\phi_+\rangle_{AB}$.

It goes like this:

1. To send
   - 00 Alice simply sends her qubit A to Bob
   - 01 Alice does X on her qubit A and sends it to Bob
   - 10 Alice does Z on her qubit A and sends it to Bob
   - 11 Alice does ZX on her qubit A and sends it to Bob.
2. Bob does the CNOT unitary operation with Alice's qubit A as the controlled qubit and his qubit B as the target qubit.
3. Bob does Hadamard unitary operation on Alice's qubit.

Before we see how the protocol works, we need to talk about CNOT and Hadamard unitaries:

$$\begin{aligned} CNOT &= |00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11| \\ H &= \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| - |1\rangle\langle + |0\rangle\langle 1| + |1\rangle\langle 0|\right). \end{aligned} \tag{2.11}$$

What CNOT does is this: $CNOT|a, b\rangle = |a, a \oplus b\rangle$. The first qubit is the control qubit and the second one is the target qubit. The Hadamard is a version of a $50 - 50$ beamsplitter: $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$. Both unitaries are their own reversals, i.e., $CNOT^{-1} = CNOT$ and $H = H^{-1}$.

Let's illustrate the workings of the protocol when Alice sends 10

$$\begin{aligned} &|\phi_+\rangle_{AB} \to Z \otimes 1|\phi_+\rangle_{B_A B} \to CNOT(Z \otimes 1|\phi_+\rangle_{B_A B}) \\ &\to H \otimes 1 CNOT(Z \otimes 1|\phi_+\rangle_{B_A B}) \to |01\rangle_{B_A B}. \end{aligned} \tag{2.12}$$

Note how $A \to B_A$ to indicate that Alice's qubit A is now in Bob's lab.

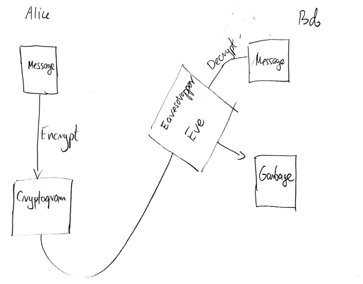
It's straightforward to calculate the other possibilities.

2.4.3 Quantum Cryptography

In this section we will discuss an idea that originated from Stephen Wiesner in 1983. Later, in 1984, Gill Brassard and Charles Bennett took it and adapted it to one of the most celebrated ideas in quantum information processing field. But not before Artur Ekert's spin on it, published in a seminal paper in 1992. As fate would have it, it was the Ekert's paper that brought it into the mainstream physics dragging with it the previous efforts. The idea is commonly known as 'quantum cryptography' but it's correct technical name is 'quantum key distribution', QKD for short. Before I'll explain it, let me first talk about classical cryptography.

Alice wants to send a secret message to Bob such that Eve, a malicious eavesdropper, can't read it even if she intercepts it. This is probably as old problem as humanity itself and lots of efforts have been done to make it work. The general idea is like this: Alice *encrypts* her *message* creating a *cryptogram* that she sends to Bob via a *public channel.* Eve tries to read the cryptogram but she gets garbage. Bob, on the other hand, *decrypts* the message and gets what Alice wanted to tell him. Easier said than done!

Figure 2.2: General idea of cryptography



The gist of the idea lies in *encryption.* What's worse, an encryption protocol must be known to everyone in the universe – including Eve that is. Cryptography and secrecy are two different things. So, it looks like a really serious problem and it is but there is a beautiful solution found by Gilbert Vernam[19]. It utilises a simple principle of 'garbage in → garbage out' and it works like this:

VERNAM CIPHER

1. Convert letters to bits: $a = 00000, b = 00001, c = 00010, \cdots$. You need 5 bits for English alphabet but your mileage may vary from language to language.

2. Now, your message is a bit string $\bar{\mathbf{m}}$ of length N (this of course depends on the message). Add to it, modulo 2, a completely random and completely secure bit string

[19] Read about the guy in Wikipedia, he was quite an interesting character.

$\bar{\mathbf{k}}$ of the same length as the message. The bit string $\bar{\mathbf{k}}$ is called the cryptographic key.

3. Create a cryptogram $\bar{\mathbf{c}} = \bar{\mathbf{m}} \oplus \bar{\mathbf{k}}$ and send it to Bob via a public channel: letter, email a newspaper add or you can even ask Eve to bring it to Bob.

4. To read the message, Bob needs the same key $\bar{\mathbf{k}}$ – we assume he has it and Eve doesn't. More about this innocent but deadly detail later. He simply adds the key to the cryptogram modulo 2: $\bar{\mathbf{c}} \oplus \bar{\mathbf{k}} = \bar{\mathbf{m}} \oplus \bar{\mathbf{k}} \oplus \bar{\mathbf{k}} = \bar{\mathbf{m}}$ because $\bar{\mathbf{k}} \oplus \bar{\mathbf{k}} = \bar{0}$.

What if Eve has the cryptogram $\bar{\mathbf{c}}$ and she doesn't have the key $\bar{\mathbf{k}}$? Well, then she sees a completely random bit string $\bar{\mathbf{c}}$ that can mean anything and thus it means nothing. Whatever message Alice had is now completely garbled[20].

When is Vernam Cipher secure?

1. Encryption keys are perfectly random and never re-used.

2. Encryption keys are of the same length as the message they encrypt.

3. Encryption keys are kept **secret at all times**.

The above conditions are the only but pretty serious drawbacks of the Vernam cipher.

Enter Ekert91 protocol!

In this protocol, Alice and Bob use an entangled source of two-qubits generated by someone else, even by Eve herself! As you will see later, in spite of Eve holding the source, Alice and Bob are perfectly safe if they follow the protocol.

To start let's first assume that the source produces Alice and Bob's qubits in the singlet state $|\psi_-\rangle$. Later we will relax this assumption.

EKERT 91 QKD PROTOCOL

1. Alice randomly chooses to measure her qubits along three directions on the Bloch sphere: $\hat{a}_0 = \hat{z}$, $\hat{a}_1 = \frac{1}{\sqrt{2}}(\hat{x} + \hat{z})$ and $\hat{a}_2 = \hat{x}$.

2. Bob randomly chooses to measure his qubits along three directions on the Bloch sphere: $\hat{b}_0 = \frac{1}{\sqrt{2}}(\hat{x} + \hat{z})$, $\hat{b}_1 = \hat{x}$ and $\hat{b}_2 = \frac{1}{\sqrt{2}}(\hat{x} - \hat{z})$.

3. After they collected a sufficient number of samples to properly evaluate their measurement probabilities, they publish over a public channel accessible to Eve (a newspaper, television, podcast, internet etc.) what measurement setting they chose for each pair of measured qubits.

[20] Here's a message encrypted by me: IUYJTUD. I used this webpage algorithm to encrypt https://www.boxentriq.com/code-breaking/one-time-pad. Please decrypt it, he he he.

4. For each pair when they **did not chose** the following pairs of settings $(\hat{a}_1, \hat{b}_0)$, $(\hat{a}_2, \hat{b}_1)$ they publish the outcomes of their measurements.
5. Using the published data, they calculate the following correlation functions: $C(\hat{a}_0, \hat{b}_0)$, $C(\hat{a}_0, \hat{b}_2)$, $C(\hat{a}_2, \hat{b}_0)$ and $C(\hat{a}_2, \hat{b}_2)$ and check for a violation of the Bell-CHSH inequality.
6. If they violate the Bell-CHSH inequality, they use the measurement outcomes for directions $(\hat{a}_1, \hat{b}_0)$ and $(\hat{a}_2, \hat{b}_1)$ or else abort the protocol and move on to another available source of qubits.

Remarks:

1. Note that $C(\hat{a}_1, \hat{b}_0) = C(\hat{a}_2, \hat{b}_1) = -1$ (see the calculations we did before). This means that Alice and Bob's bits for these measurement settings are anti-correlated. If Bob flips his bit they become perfectly correlated and these bits can be used as a cryptographic key in the Vernam cipher.
2. Some measurement data is never used, for instance date for the measurement settings $(\hat{a}_1, \hat{b}_1)$.

Why is Ekert91 protocol safe from Eve?

The most general proof is quite difficult to prove but there is a great intuition behind it. We discuss it first and after that I'll sketch a limited but still quite a powerful proof under some reasonable assumptions.

The idea behind the security lies in Einstein-Podolsky-Rosen (EPR) infamous paper. If Alice and Bob violate the Bell-CHSH inequality they are sure that their correlations are not local and realistic. This means that outcome measurements they observe do not exist before they measure them! The very act of quantum measurement creates their reality locally. If this is the case, how can Eve know Alice and Bob's outcomes and thus acquire any information about the cryptographic key? There is nothing there for Eve!

Now, let's analyse some aspects of Ekert 91's protocol security . To this end, we assume that Eve holds the source and generates a tripartite state $|\Psi\rangle_{ABE}$, sending two qubits A and B to Alice and Bob and keeping her quantum system E. The dimension of her system can be arbitrary. We can write

$$\begin{aligned}
|\Psi\rangle_{ABE} &= \sum_{a,b=0}^{1} |a,b\rangle_{AB} |\tilde{E}_{ab}\rangle_E \\
&= |\psi_+\rangle_{AB} \frac{1}{\sqrt{2}}(|\tilde{E}_{01}\rangle_E + |\tilde{E}_{10}\rangle_E) + |\psi_-\rangle_{AB} \frac{1}{\sqrt{2}}(|\tilde{E}_{01}\rangle_E - |\tilde{E}_{10}\rangle_E) + \\
&+ |\phi_+\rangle_{AB} \frac{1}{\sqrt{2}}(|\tilde{E}_{00}\rangle_E + |\tilde{E}_{11}\rangle_E) + |\phi_-\rangle_{AB} \frac{1}{\sqrt{2}}(|\tilde{E}_{00}\rangle_E - |\tilde{E}_{11}\rangle_E) \qquad (2.13)
\end{aligned}$$

where $|a,b\rangle$ are computational basis kets and $|\tilde{E}_{ab}\rangle_E$ are Eve's, unnormalised kets. However, the total state $|\Psi\rangle_{ABE}\rangle$ is, of course, normalised, i.e.,

$$\sum_{a,b=0}^{1}\langle\tilde{E}_{a,b}|\tilde{E}_{a,b}\rangle_E = 1. \tag{2.14}$$

Keeping Eve's state unnormalised at this stage will simplify calculations as you will see.

What's the physical meaning of Eve's states? They carry quantum information about Alice and Bob's qubits. Each state $|\tilde{E}_{a,b}\rangle$ tells Eve that Alice and Bob's measurement outcomes in the computational basis are a and b respectively. This information can be potentially used for eavesdropping. How Alice and Bob protect themselves? To see it we need to calculate Alice and Bob's state:

$$\rho_{AB} = \sum_{a,b=0}^{1}\sum_{a',b'=0}^{1}\langle\tilde{E}_{a',b'}|\tilde{E}_{a,b}\rangle_E|a,b\rangle\langle a',b'|_{AB}. \tag{2.15}$$

Since Alice and Bob will abort unless they violate the Bell-CHSH inequality, Eve has to arrange her states in such a way that this doesn't happen. First let's see what Eve has to do if she wants Alice and Bob to reach the maximal violation of local realism, i.e., let them see the singlet state. This only happens if

$$\langle\psi_-|\rho_{AB}|\psi_-\rangle_{AB} = 1,$$

i.e., $|\tilde{E}_{10}\rangle_E = -|\tilde{E}_{01}\rangle_E$ and all the other Eve's states vanish. But then Eve loses all quantum information about Alice and Bob's measurement outcomes: she has only one state $|\tilde{E}_{01}\rangle$ that is insensitive to any possible correlations ocuring between Alice and Bob. This already tells you that if Eve wants to eavesdrop Alice and Bob will never maximally violate the Bell-CHSH inequality. But we know they won't abort the protocol as long as they see any violation. This gives Eve some room to manoeuvre. Let's have a closer look.

We know that the Bell-CHSH inequality can be violated with Werner state $\rho_V = V|\psi_-\rangle\langle\psi_-|_{AB} + \frac{1-V}{4}$ as long as there is enough singlet in it, i.e., ${}_{AB}\langle\psi_-|\rho_V|\psi_-\rangle_{AB} > \frac{1+\frac{3}{\sqrt{2}}}{4}$ (this happens for $V > \frac{1}{\sqrt{2}}$). Eve, certainly, can manufacture Werner state with an arbitrary V. We have

$$\begin{aligned}
\frac{1+3V}{4} &= \frac{1}{2}\Big(\langle\tilde{E}_{01}|\tilde{E}_{01}\rangle_E + \langle\tilde{E}_{10}|\tilde{E}_{10}\rangle_E - \langle\tilde{E}_{01}|\tilde{E}_{10}\rangle_E - \langle\tilde{E}_{10}|\tilde{E}_{01}\rangle_E\Big)\\
\frac{1-V}{4} &= \frac{1}{2}\Big(\langle\tilde{E}_{01}|\tilde{E}_{01}\rangle_E + \langle\tilde{E}_{10}|\tilde{E}_{10}\rangle_E + \langle\tilde{E}_{01}|\tilde{E}_{10}\rangle_E + \langle\tilde{E}_{10}|\tilde{E}_{01}\rangle_E\Big)\\
\frac{1-V}{4} &= \frac{1}{2}\Big(\langle\tilde{E}_{00}|\tilde{E}_{00}\rangle_E + \langle\tilde{E}_{11}|\tilde{E}_{11}\rangle_E + \langle\tilde{E}_{00}|\tilde{E}_{11}\rangle_E + \langle\tilde{E}_{00}|\tilde{E}_{11}\rangle_E\Big)\\
\frac{1-V}{4} &= \frac{1}{2}\Big(\langle\tilde{E}_{00}|\tilde{E}_{00}\rangle_E + \langle\tilde{E}_{11}|\tilde{E}_{11}\rangle_E - \langle\tilde{E}_{00}|\tilde{E}_{11}\rangle_E - \langle\tilde{E}_{11}|\tilde{E}\rangle_E\Big).
\end{aligned}$$

Those equations can be solved, giving us normalized Eve's states and angles between them.

$$\begin{aligned}
|\tilde{E}_{01}\rangle_E &= \sqrt{\frac{1+V}{4}}\cos\phi|E_{01}\rangle_E \\
|\tilde{E}_{10}\rangle_E &= \sqrt{\frac{1+V}{4}}\sin\phi|E_{10}\rangle_E \\
|\tilde{E}_{00}\rangle_E &= \sqrt{\frac{1-V}{4}}|E_{00}\rangle_E \\
|\tilde{E}_{11}\rangle_E &= \sqrt{\frac{1-V}{4}}|E_{11}\rangle_E \\
\langle E_{01}|E_{10}\rangle_E &= -\frac{2V}{(1+V)\sin 2\phi} \\
\langle E_{00}|E_{11}\rangle_E &= 0 \qquad (2.16)
\end{aligned}$$

where the angle ϕ is a parameter to be adjusted for now. In the above calculations I assumed, without losing generality, that the scalar products are real numbers. It can be shown that Eve gains nothing if they are complex numbers. I also assumed that the group of states $|E_{00}\rangle_E, |E_{11}\rangle_E$ is orthogonal to the group $|E_{01}\rangle_E, |E_{10}\rangle_E$. I'll explain the reason for this assumption below. Now let's discuss the physics behind this messy math.

Let's first look at the group of states $|E_{01}\rangle_E, |E_{10}\rangle_E$. This group of states appears to Eve with probability $\frac{1+V}{2}$ while the other, orthogonal one, with probability $\frac{1-V}{2}$. Suppose Eve gets the first group of states. Within this group she gets the state $|E_{01}\rangle_E$ with probability $\cos^2\phi$ and the state $|E_{10}\rangle_E$ with probability $\sin^2\phi$. As we will show later, she benefits the most if these states appear with equal probabilities, which happens for $\phi = \frac{\pi}{4}$. The states she now gets are not orthogonal:

$$\langle E_{01}|E_{10}\rangle_E = -\frac{2V}{1+V}$$

and thus she can't tell one from another with 100% accuracy! If she could she would know that Alice gets 0 and Bob 1 when she detects the state $|E_{01}\rangle_E$ and vice versa, she would know Alice gets 1 and Bob 0 if she detects the state $|E_{10}\rangle_E$. The situation is different for the other group of states $|E_{00}\rangle_E, |E_{11}\rangle_E$. These states automatically appear with equal probabilities and are orthonormal. This counterintuitive assymmetry arises because Eve reproduces the singlet state with the highest fidelity, not the other Bell states - remember Alice and Bob expect to see the singlet state in the ideal situation and thus gear their measurements for it. This also explains why the states in the group $|E_{00}\rangle_E, |E_{11}\rangle_E$ are orthogonal. Surely, Eve can perfectly know when Alice and Bob's measurements are correlated but Alice and Bob rely on anti-correlated events for the key generation. Thus as long as anti-correlated events are the majority of the events contributing to the quantum key generation, Alice and Bob have some advantage.

Now, note that if $V = 1$, the states in the first group become anti-parallel (we derived it before!) and Eve can't extract any information whatsoever while if $V = 0$, the states in the first group become orthonormal and Eve knows everything. Clearly, there must be a critical V_0 above which Eve can't eavesdrop and below which she can. So when Alice and Bob are safe from Eve? According to Ekert 91 the critical $V_0 = \frac{1}{\sqrt{2}}$ but to prove it rigorously we need to wait a little and learn some new stuff in the next section.

2.4.4 Generalised measurement

In this chapter I'll show you how to extract different information from a quantum system than you thought you could we the von Neumann type measurement. You need this to take full advantage of quantum information processing.

Consider again a single qubit in a state $\rho = \frac{1}{2}(1 + \vec{n} \cdot \vec{\sigma})$. If you measure it in a basis $\pm\vec{m}$ you get probabilities $p(k_{\hat{m}}) = \frac{1}{2}(1 \pm \hat{m} \cdot \vec{n})$, which is your information about the qubit. But you can do something else. Here's how it goes.

Attach a D dimensional system, an ancilla[21], in the state $|\Omega_0\rangle$. This state can be a part of the ancilla's orthonormal basis $|\Omega_\alpha\rangle$, $\alpha = 0, 1, \ldots, D-1$. Prepare the qubit in a pure state $|0\rangle$ and evolve it together with the ancilla via a unitary operation U[22], i.e.,

$$\begin{aligned} |0, \Omega_0\rangle \to U|0, \Omega_0\rangle &= \sum_{a,\alpha} |a, \alpha\rangle\langle a, \alpha|U|0, \Omega_0\rangle \\ &= \sum_{a,\alpha} \langle a, \alpha|U|0, \Omega_0\rangle |a, \alpha\rangle. \end{aligned} \tag{2.17}$$

Now, measure the ancilla (von Neumann) in the basis $|\Omega_\alpha\rangle$ but do nothing to the qubit:

$$\begin{aligned} p(\alpha) &= Tr_{Q\Omega}(1 \otimes |\Omega_\alpha\rangle\langle\Omega_\alpha|U|0, \Omega_0\rangle\langle 0, \Omega_0|U^\dagger) \\ &= Tr_{Q\Omega}\left(\sum_b |b\rangle\langle b| \otimes |\Omega_\alpha\rangle\langle\Omega_\alpha|U|0, \Omega_0\rangle\langle 0, \Omega_0|U^\dagger\right) \\ &= \sum_b \langle b, \Omega_\alpha|U|0, \Omega_0\rangle\langle 0, \Omega_0|U^\dagger|b, \Omega_\alpha\rangle. \end{aligned} \tag{2.18}$$

Let's define a qubit operator

$$E_\alpha := \langle\Omega_\alpha|U|\Omega_0\rangle.$$

[21] Servant in Spanish.

[22] This can be done by introducing a physical interaction between the systems or, as you'll see later, by looking at different degrees of freedom of a single system.

Although it looks as if it's a scalar product and thus a number, it isn't because

$$\begin{aligned} E_\alpha &= \langle\Omega_\alpha|\sum_{b,\gamma}|b,\Omega_\gamma\rangle\langle b,\Omega_\gamma|U|\sum_{b',\gamma'}|b',\Omega_{\gamma'}\rangle\langle b',\Omega_{\gamma'}|\Omega_0\rangle \\ &= \sum_{b,b'}\langle b,\Omega_\alpha|U|b',\Omega_0\rangle|b\rangle\langle b'|, \end{aligned} \tag{2.19}$$

where I used a partial scalar product on the ancilla Hilbert space $\langle\Omega_\alpha|b,\Omega_\gamma\rangle := \langle\Omega_\alpha|\Omega_\gamma\rangle|b\rangle$[23].

Exercise 32. *Prove that* $p(\alpha) = \langle 0|E_\alpha^\dagger E_\alpha|0\rangle$.

With this in mind, we have now

$$p(\alpha) = \langle 0|E_\alpha^\dagger E_\alpha|0\rangle.$$

This is remarkable because we expressed probabilities $p(\alpha)$ solely in the Hilbert space of the qubit. Of course, the ancilla's Hilbert space is 'hidden' in E_α.

The set of operators E_α is called a generalised measurement or POVM[24]. These operators can be anything as long as $\sum_\alpha E_\alpha^\dagger E_\alpha = 1$, which is easy to prove from the above calculations.

Let's now see what they are good for. First, we discuss *Helström measurement*. It goes like this: Alice randomly sends to Bob two, generally, non-orthogonal, photonic polarization states $|\psi_\pm\rangle = \cos\theta|H\rangle \pm \sin\theta|V\rangle$, where $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$, and Bob's task is to tell which state Alice sent. If he uses a von-Neumann measurement, the best he can do[25] is to chose the basis $|\phi_\pm\rangle$ such that $|\langle\phi_+|\psi_+\rangle| = |\langle\phi_-|\psi_-\rangle|$ and bet that if the outcome corresponding to the $|\phi_+\rangle$ state occurs Alice sent $|\psi_+\rangle$ and $|\psi_-\rangle$ if he gets the outcome corresponding to the state $|\phi_-\rangle$. He won't get it always right and the probability of this happening is $P(error) = \frac{1}{2}\left(1 - \sqrt{1 - |\langle\psi_+|\psi_-\rangle|^2}\right)$.

Exercise 33. *Prove the Helström formula* $P(error) = \frac{1}{2}\left(1 - \sqrt{1 - |\langle\psi_+|\psi_-\rangle|^2}\right)$.

We see that as long as the states Alice sends aren't orthogonal, Bob is never really sure about his guess. But we can do the following generalised measurement that changes this.

Consider an interferometer shown in Fig. 3.1.

[23] Definition is general, please think it thru.

[24] Positive Operator Valued Measure in mathematics.

[25] Proven by Helström.

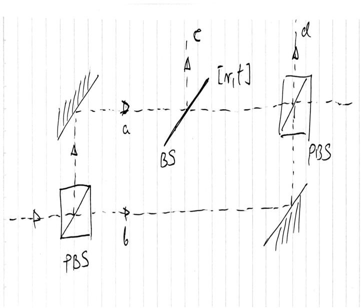


Figure 2.3: Generalised measurement: PBS stands for Polarizing Beamsplitter. It's a device that lets vertical polarisation V through and reflects horizontal polarisation H. The BS in path 'a' transmits t^2% of light and reflects r^2%.

Bob lets in Alices states through the input port 'in' denoted with a dashed arrow, i.e., the initial states are $|\psi_\pm\rangle|in\rangle$. They evolve like this:

$$\begin{aligned}&|\psi_\pm\rangle|in\rangle \to \cos\theta|H,a\rangle \pm \sin\theta|V,b\rangle \to \\ &t\cos\theta|H,a\rangle \pm \sin\theta|V,b\rangle + r\cos\theta|H,c\rangle \to \\ &(t\cos\theta|H\rangle \pm \sin\theta|V\rangle)\,|d\rangle + r\cos\theta|H,c\rangle. \end{aligned} \tag{2.20}$$

Choosing $t = \tan\theta$ Bob makes the states coming out in the path 'd' orthogonal and thus perfectly distinguishable. The states in 'd' are $|\pm\rangle = \frac{1}{\sqrt{2}}(|H\rangle \pm |V\rangle)$ and they correspond to the Alices states $|\psi_\pm\rangle$. To help himself he puts the third polarising beamsplitter that sends the state $|+\rangle$ to the new path 'd_+' and the state $|-\rangle$ to another path 'd_-'. This distinguishable situation happens with probability $1 - \langle\psi_-|\psi_+\rangle$[26]. However, the price Bob pays is that with the probability $\langle\psi_-|\psi_+\rangle$ he has completely no idea which state Alice sent him because this corresponds to photons coming out in the output 'c' where all information about the states' polarisation is lost.

What are operators E_α corresponding to this generalised measurement? To answer this question let's look at the evolution generated by the interferometer for any input state $|\phi, in\rangle = \alpha|H,in\rangle + \beta|V,in\rangle$:

$$\begin{aligned}&\alpha|H,in\rangle + \beta|V,in\rangle \to \frac{\alpha t + \beta}{\sqrt{2}}|+\rangle|d_+\rangle + \frac{\alpha t - \beta}{\sqrt{2}}|-\rangle|d_-\rangle + \\ &\alpha r|H,c\rangle. \end{aligned} \tag{2.21}$$

A little thinking brings us to

$$|\phi, in\rangle = \alpha|H,in\rangle + \beta|V,in\rangle \to E_+|\phi\rangle|d_+\rangle + E_-|\phi\rangle|d_-\rangle + E_0|\phi\rangle|c\rangle, \tag{2.22}$$

[26] Here this expression makes sense because of how we defined the states and the range of θ. Generally, it doesn't make sense because the scalar product could be negative or even complex!.

where

$$\begin{aligned} E_+ &= \frac{1}{\sqrt{2}}|+\rangle \left(t\langle H| + \langle V|\right) \\ E_- &= \frac{1}{\sqrt{2}}|-\rangle \left(t\langle H| - \langle V|\right) \\ E_0 &= r|H\rangle\langle H|. \end{aligned} \tag{2.23}$$

It's trivial to check that $E_+^\dagger E_+ + E_-^\dagger E_- + E_0^\dagger E_0 = 1$. Note that tracing out the ancilla (which is the path in the interferometer) from the equation (2.22) gives nicely the POVM measurement as we discussed before.

2.5 Shannon entropy, von Neumann entropy and Holevo Bound

The *von Neumann entropy* is a fundamental concept in quantum information theory, analogous to the Shannon entropy in classical information theory. It quantifies the uncertainty or randomness of a quantum system's state.

The *Holevo bound*, also known as the *Holevo χ quantity*, sets an upper limit on the amount of classical information that can be transmitted using a quantum channel. Specifically, it provides a bound on the accessible information from a quantum ensemble. Even though quantum states can encode vast amounts of information, the Holevo bound indicates that the amount of retrievable classical information is limited. The bound is crucial in determining the efficiency of quantum communication systems and highlights the gap between the potential quantum and classical information capacities of quantum states.

Together, these concepts are pivotal in understanding quantum information, communication limits, and the interplay between quantum and classical information theory. In this section, we will introduce their definitions and discuss their properties.

2.5.1 Shannon entropy

Shannon entropy, introduced by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication," is a foundational concept in information theory. It quantifies the uncertainty or unpredictability of a random variable, often used to measure the amount of information in a source or the efficiency of a communication system. The greater the entropy, the higher the uncertainty or the amount of information required to describe the system.

Given a discrete random variable X that can take on values $x_1, x_2, \ldots, x_n$ with correspond-

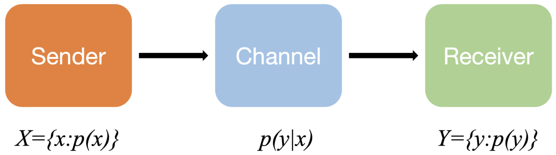


Figure 2.4: Shannon-Weaver model for communication, we have omitted the encoding and decoding part.

ing probabilities $p(x_1), p(x_2), \ldots, p(x_n)$, the Shannon entropy $S(X)$ is defined as:

$$S(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) = -\sum_{x} p(x) \log p(x).$$

In this formula:

- $p(x_i)$ is the probability of the random variable X taking the value x_i,
- $-\log_2 p(x_i)$ represents the information content of the outcome x_i, and
- the sum over all possible outcomes gives the expected value of the information content, or the entropy.

The Shannon entropy can be understood from many different aspects, from the communication perspective, it measure the uncertainty of a information source.

The classical channel is characterized by conditional probability $p(y|x)$. Given the senders message $X = x$, the receiver have its distribution of Y as $p(Y = y|X = x)$, from which we can calculate the information

$$S(Y|X = x) = -\sum_{y} p(y|x) \log p(y|x). \tag{2.24}$$

This quantifies the receiver's uncertainty about Y after receiving the message $X = x$. On average, we have the following quantity called conditional entropy

$$S(Y|X) = \sum_{x} p(x) S(Y|X = x). \tag{2.25}$$

Notice that before receiver receive the message from the sender, his uncertainty about Y is $S(Y)$, after receiving the message X, his uncertainty becomes $s(Y|X)$. This means that the information transmitted by the channel is

$$I(X;Y) = H(Y) - H(Y|X), \tag{2.26}$$

which is called mutual information. We can check that (as we have show during the class)

$$I(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P_{(X,Y)}(x,y)\log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x)\,P_Y(y)}\right). \tag{2.27}$$

From this it is clear that $I(X;Y) = I(Y;X)$. We also have

$$\begin{aligned}
\mathrm{I}(X;Y) &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_{(X,Y)}(x,y)\log\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)} \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_{(X,Y)}(x,y)\log\frac{p_{(X,Y)}(x,y)}{p_X(x)} - \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_{(X,Y)}(x,y)\log p_Y(y) \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_X(x)p_{Y|X=x}(y)\log p_{Y|X=x}(y) - \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p_{(X,Y)}(x,y)\log p_Y(y) \\
&= \sum_{x\in\mathcal{X}} p_X(x)\left(\sum_{y\in\mathcal{Y}} p_{Y|X=x}(y)\log p_{Y|X=x}(y)\right) - \sum_{y\in\mathcal{Y}}\left(\sum_{x\in\mathcal{X}} p_{(X,Y)}(x,y)\right)\log p_Y(y) \\
&= -\sum_{x\in\mathcal{X}} p_X(x)S(Y|X=x) - \sum_{y\in\mathcal{Y}} p_Y(y)\log p_Y(y) \\
&= S(Y) - S(Y|X) \\
&= S(Y) - S(Y|X).
\end{aligned} \tag{2.28}$$

The relation of $S(X), S(Y), S(XY), S(X|Y), S(Y|X)$ can be summarized as a Venn diagram, where the area represent the value of these quantities, see Fig. 2.5. Besides what we have derived before, we also have

$$I(X;Y) = S(X) + S(Y) - S(XY), \tag{2.29}$$

$$S(X|Y) = S(XY) - S(Y), S(Y|X) = S(XY) - S(X). \tag{2.30}$$

Shannon entropy has several important properties that make it a useful measure of uncertainty and information:

1. **Non-negativity:** The entropy is always non-negative, $S(X) \geq 0$, since probabilities are positive and the logarithm of a number between 0 and 1 is negative, thus negating it gives a positive value.

2. **Maximum entropy:** The entropy is maximized when all outcomes of the random variable are equally likely, i.e., $p(x_i) = \frac{1}{n}$ for all i. In this case, the entropy is $S(X) = \log_2 n$, indicating maximum uncertainty when all outcomes are equally probable.

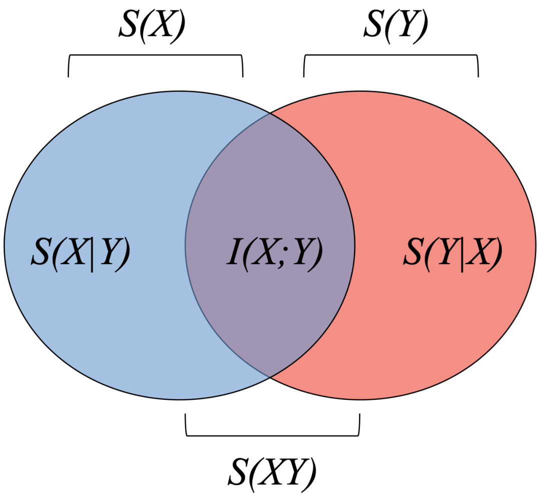


Figure 2.5: Venn diagram for entropy.

3. **Minimum entropy:** The entropy is zero when the outcome is certain, meaning one of the $p(x_i) = 1$ and all others are 0. This indicates no uncertainty in the system.

4. **Concavity:** For two random variable (information sources) X_1, X_2 with probability distributions $p_{X_1}(x_1), p_{X_2}(x_2)$, we have

$$S(\lambda X_1 + (1-\lambda)X_2) \geq \lambda S(X_1) + (1-\lambda)S(X_2). \tag{2.31}$$

5. **Mutual and conditional information entropies are non-negativity**: $I(X;Y) \geq 0$, $S(X|Y), S(Y|X) \geq 0$.

6. **Additivity:** If two independent random variables X and Y are considered, the entropy of the joint distribution $S(XY)$ is the sum of the individual entropies:

$$S(XY) = S(X) + S(Y).$$

   This property generalizes to conditional entropy, providing a way to measure how much additional information is needed to describe one variable given knowledge of another.

Shannon entropy can be interpreted as the average amount of information, surprise, or uncertainty inherent in the outcomes of a random process. It measures the efficiency of a communication system by indicating how many bits on average are required to encode a message from a given source.

Key applications of Shannon entropy include:

- **Data Compression:** In lossless compression algorithms, Shannon entropy provides a lower bound on the average number of bits needed to encode a source without loss of information. The entropy represents the ideal limit for compressing data.
- **Communication Theory:** In communication systems, Shannon entropy is used to measure the efficiency of encoding schemes and to understand how much information can be transmitted over noisy channels (Shannon's Channel Capacity Theorem).
- **Information Gain and Decision Trees:** In machine learning, Shannon entropy is used to construct decision trees by selecting splits that maximize the information gain, or reduction in entropy.
- **Cryptography:** Entropy measures the unpredictability of a cryptographic key, where high entropy corresponds to stronger security due to less predictability.

2.5.2 von Neumann entropy

Now let us replace the classical source X with a quantum source: $\rho_X = \{\rho_x, p(x)\}$, that is, we send ρ_x with probability $p(x)$. In this case, classical Shannon entropy has its quantum equivalent called von Neumann entropy.

For a density operator ρ, the von Neumann entropy is defined as

$$S(\rho) = -\operatorname{Tr} \rho \log \rho. \tag{2.32}$$

How to calculate this? We can calculate the eigenvalues of ρ first, these eigenvalues form a probability distribution $\vec{\lambda} = (\lambda_1, \cdots, \lambda_n)$, then the von Neumann entropy is just the Shannon entropy for

$$S(\rho) = S(\vec{\lambda}) = -\sum_i \lambda_i \log \lambda_i. \tag{2.33}$$

For pure state $\rho_\psi = |\psi\rangle\langle\psi|$, since the only eigen value is 1, this means that

$$S(\rho_\psi) = 0. \tag{2.34}$$

In face, a state ρ is a pure state if and only if $S(\rho) = 0$.

Example 5. *Take a qubit mixed state described by a Bloch vector $\vec{a}$. As we already know, it has eigenvalues $\frac{1}{2}(1 \pm |\vec{a}|)$ corresponding to eigenkets $\frac{1}{2}(1 \pm \frac{\vec{a}}{|\vec{s}|} \cdot \vec{\sigma})$. Von Neumann entropy is then defined as $S(\rho) := -\frac{1}{2}(1 + |\vec{a}|) \log_2(\frac{1}{2}(1 + |\vec{a}|)) - \frac{1}{2}(1 - |\vec{a}|) \log_2(\frac{1}{2}(1 - |\vec{a}|))$, which is equivalent to $S(\rho) = -Tr(\rho \log_2(\rho))$[27].*

[27] Purists will insist on natural logarithm but it is not important for us.

What is a meaning of von Neumann entropy? If you read about Shannon classical entropy you learnt that one of its meanings is how many classical bits you need to describe a random source producing them. For instance, a purely random source of bits, distributing 0's and 1's with equal probability, requires $-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$ bit to describe it. This is because each bit appears totally randomly and thus it is always uncertain. But a source, that produces 0's with probability p and 1's with probability $1-p$ requires $-p\log_2 p - (1-p)\log_2(1-p) < 1$ bits to fully characterize it. If $p \neq \frac{1}{2}$ each bit produced by the source is partially predictable and my uncertainty is not maximal. Von Neumann entropy of ρ has analogous meaning - it tells us how many qubits we need to encode information contained in ρ. A full exploration of this concept is beyond the scope of this course and interested readers are encouraged to approach me for explanations or study Schumacher compression theorem.

Similar as classical Shannon entropy, for bipartite system ρ_{AB}, we have reduced density matrices $\rho_A = \mathrm{Tr}_B\,\rho_{AB}$ and $\rho_B = \mathrm{Tr}_A\,\rho_{AB}$, then we can define

$$S(A) = S(\rho_A), S(B) = S(\rho_B), S(AB) = S(\rho_{AB}). \tag{2.35}$$

Then conditional entropy and mutual entropy can be defined similarly to classical entropy (See Fig. 2.5)

Let ρ be a density matrix. The von Neumann entropy $S(\rho)$ is defined as:

$$S(\rho) = -\mathrm{Tr}(\rho\log\rho)$$

The following are key properties of von Neumann entropy:

1. **Non-negativity**:

$$S(\rho) \geq 0 \tag{2.36}$$

Von Neumann entropy is always non-negative.

2. **Purity**:

$$S(\rho) = 0 \quad \text{if and only if} \quad \rho \text{ is a pure state.} \tag{2.37}$$

For a pure state, the entropy is zero.

3. **Maximum Entropy for Maximally Mixed States**:

$$S\left(\frac{I}{d}\right) = \log d \tag{2.38}$$

where d is the dimension of the system and I is the identity matrix. This occurs for maximally mixed states.

4. **Invariance under Unitary Transformations**:

$$S(U\rho U^\dagger) = S(\rho) \tag{2.39}$$

for any unitary matrix U. This means entropy depends only on the eigenvalues of ρ.

5. **Concavity**:

$$S(p\rho_1 + (1-p)\rho_2) \geq pS(\rho_1) + (1-p)S(\rho_2) \tag{2.40}$$

for any density matrices ρ_1, ρ_2 and $0 \leq p \leq 1$.

6. **Subadditivity**:

$$S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B) \tag{2.41}$$

where $\rho_A = \mathrm{Tr}_B(\rho_{AB})$ and $\rho_B = \mathrm{Tr}_A(\rho_{AB})$ are the reduced density matrices for subsystems A and B.

7. **Strong Subadditivity**:

$$S(\rho_{ABC}) + S(\rho_B) \leq S(\rho_{AB}) + S(\rho_{BC}) \tag{2.42}$$

for a tripartite system with density matrix ρ_{ABC}.

8. **Triangle Inequality**:

$$|S(\rho_A) - S(\rho_B)| \leq S(\rho_{AB}) \tag{2.43}$$

for a bipartite system.

9. **Additivity for Tensor Products**:

$$S(\rho_A \otimes \rho_B) = S(\rho_A) + S(\rho_B) \tag{2.44}$$

if ρ_A and ρ_B are independent.

Notice that some properties of Shannon entropy holds also holds for von Neumann entropy, but some others are broken. Typical example is that $S(AB)$ is not necessarily greater than or equal to $S(A)$ or $S(B)$. This implies that conditional entropy is not necessarily nonnegative.

Exercise 34. *For Bell state $|\phi^+\rangle_{AB}$, show that $S(AB) = 0$ but $S(A) = S(B) = 1$.*

Exercise 35. *Show that for pure bipartite state $|\psi\rangle_{AB}$, it is entangled if and only if the entropy for reduced density matrix is nonzero: $S(A) = S(B) > 0$. [Hint: You need to use Schmidt decomposition.]*

Exercise 36. *Show that mutual information $I(A;B)$ is nonnegative. [Hint: this is a direct result of subadditivity.]*

2.5.3 Holevo theorem

Equipped with the knowledge about generalised measurement we are ready to discuss an important theorem about quantum information proven by Holevo.

Let us go back to Holevo bound. Consider Alice sending Bob a bunch of quantum states ρ_i, each with probability $p(a)$ $(a = 1, 2, \ldots, N)$. It is not difficult to realise that each state can be viewed as a piece of information Alice sends to Bob. Bob's task is to extract as much information as possible from this transmission. To do this Bob has to measure the states he has received from Alice. Since we need utmost generality Bob has to use a generalised measurement with as many outcomes as he needs, say, M. Such a measurement is given by a POVM E_b, $b = 1, 2, \cdots, M$.

We can now look at the Alice-Bob communication in terms of mutual information. Alice random variable A has N values, distributed with $p(a)$ and Bob's random variable M values, distributed with $p(b) = \sum_{a=1}^{N} Tr(E_b^\dagger E_b \rho_a) p(a)$. The classical channel describing this transmission is characterized by transition probabilities $p(b|a) = Tr(E_b^\dagger E_b \rho_a)$. We can now calculated the mutual information, $I(A : B)$ for this channel and maximize it over Bob's choice of POVMs E_b:

$$\max_{E_b} I(A : B). \tag{2.45}$$

This maximization is too difficult to calculate explicitly but Holevo proved that it is never greater than the Holevo quantity[28] χ, given by:

$$\chi = S\left(\sum_a p_a \rho_a\right) - \sum_a p_a S(\rho_a). \tag{2.46}$$

So, we have

$$\max_{E_b} I(A : B) \leq \chi. \tag{2.47}$$

Here, $S(\rho)$ is the Von Neumann entropy of density matrix ρ, $S(\rho) = -\operatorname{Tr}\{\rho \log \rho\}$.

Exercise 37. *Show that the Holevo bound when Alice sends to Bob two pure orthogonal states $|\psi\rangle$ and $|\psi^\perp\rangle$ ($\langle\psi^\perp|\psi\rangle = 0$), each with the same probability $\frac{1}{2}$, is $\chi = 1$. Does this bound depend on whether $|\psi\rangle$ and $|\psi^\perp\rangle$ are qubit states or arbitrary quantum states?*

Exercise 38. *Show that von Neumann entropy of an arbitrary qubit state $\rho = \frac{1}{2}(1 + \vec{s} \cdot \vec{\sigma})$ reads*

$$S(\rho) = -\frac{1}{2}(1 + |\vec{s}|) \log\left(\frac{1}{2}(1 + |\vec{s}|)\right) - \frac{1}{2}(1 - |\vec{s}|) \log\left(\frac{1}{2}(1 - |\vec{s}|)\right).$$

[28] Named after him- he didn't call it that!

If you solve the above exercise you'll see that von Neumann entropy of a pure qubit state[29] is zero. This observation tells you that it's better for Alice to send Bob pure states rather than mixed ones if they want to maximise their mutual information since in this case the second term in the χ quantity is zero.

Holevo bound plays an important role in quantum information science and if you go deeper into this branch of experimental and theoretical physics you will encounter it sooner or later.

Exercise 39. *Show that Alice, using n qubits, can't transfer more then n bit of classical information to Bob. This shows you that although n qubits can be in a quantum superposition, Bob can still only access n classical bits of information.*

2.6 Quantum Channels

The generalised measurement is in a very close relationship with a generalised evolution of quantum systems.

In the orthodox quantum mechanics quantum systems change in time via a unitary evolution. However, there are many situations where this is too restrictive and we will see this after a couple of derivations.

Let's look at the formula (2.17), trace out the ancilla and skip the measurement:

$$Tr_\Omega[U|0,\Omega_0\rangle\langle 0,\Omega_0|U^\dagger] = \sum_\alpha E_\alpha|0\rangle 0|E_\alpha^\dagger, \tag{2.48}$$

where E_α are exactly the same general measurement operators we discussed before. What happened? It looks like we measured the ancilla and 'forgot' the measurement outcomes. This sounds pretty wild but it's not and the best way to understand it is to look at some examples.

1. **Depolarizing channel.** Consider the following unitary operation on the qubit and ancilla

$$U|0,\Omega_0\rangle = \sqrt{p}|0,\Omega_0\rangle + \sqrt{\frac{1-p}{3}}\left(X|0\rangle|\Omega_1\rangle + Z|0\rangle|\Omega_2\rangle + Y|0\rangle|\Omega_3\rangle\right), \tag{2.49}$$

where $0 \leq p \leq 1$.

Exercise 40. *Argue/convince yourself that this is a unitary transformation.*

[29] This applies to any pure state of any quantum system, not only a qubit.

After tracing out the ancilla we arrive at

$$p|0\rangle\langle 0| + \frac{1-p}{3}(X|0\rangle\langle 0|X + Y|0\rangle\langle 0|Y + Z|0\rangle\langle 0|Z) \tag{2.50}$$

and from here we can infer the operators E_α

$$\begin{aligned} E_0 &= \sqrt{p}I \\ E_1 &= \sqrt{\frac{1-p}{3}}X \\ E_2 &= \sqrt{\frac{1-p}{3}}Z \\ E_3 &= \sqrt{\frac{1-p}{3}}Y. \end{aligned} \tag{2.51}$$

A quick check confirms that $\sum_{\alpha=0}^{3} E_\alpha^\dagger E_\alpha = I$. How to interpret this channel?

First of all, we call it a channel because it takes a quantum state as an input and produces another quantum states as an output. Symbolically we could write it like this: $\Lambda_D(\rho) = \sum_{\alpha=0}^{3} E_\alpha \rho E_\alpha^\dagger$. In this sense, a unitary operation is also a channel – a unitary channel!

The physical interpretation of this depolarizing channel is that with probability p the state is unaffected but with equal probabilities $\frac{1-p}{3}$ the state undergoes a *bit flip* X, a *phase flip* Z and the combination of both $iXZ = Y$ (the phase i is added for convenience). It is called a depolarizing channel because if you take a qubit with a Bloch vector $\vec{n}$, after the action of the channel it will shrink, i.e., $\vec{n} \to \frac{4p-1}{3}\vec{n}$. Hence the name of the channel.

Exercise 41. *Prove that the depolarizing channel shrinks an input state's Bloch vector:* $\vec{n} \to \frac{4p-1}{3}\vec{n}$.

2. **Dephasing channel.** This time we skip the ancilla and go straight to the set of operators describing the channel

$$\begin{aligned} E_0 &= \sqrt{1-p}I \\ E_1 &= \sqrt{\frac{p}{2}}(1+Z) \\ E_2 &= \sqrt{\frac{p}{2}}(1-Z), \end{aligned} \tag{2.52}$$

where, again, $0 \leq p \leq 1$.

2.7 Quantum error correction: 9-qubit Shor code

Consider a situation when Alice sends to Bob a quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}(a_0|0\rangle + a_1|1\rangle)$. Errors can happen during the transmission.

1. **Bit flip error channel:** $E_0 = \sqrt{p}, E_1 = \sqrt{1-p}X$, i.e., with probability p, the transmission is faithful, and with probability $1-p$, the state $|0\rangle$ flips to $|1\rangle$, and $|1\rangle$ flips to $|0\rangle$. The state Bob receives is:

$$p|\psi\rangle\langle\psi| + (1-p)X|\psi\rangle\langle\psi|X.$$

2. **Phase flip error channel:** $E_0 = \sqrt{p}, E_1 = \sqrt{1-p}Z$, i.e., with probability p, the transmission is faithful, and with probability $1-p$, the state $|0\rangle$ remains $|0\rangle$, but $|1\rangle$ changes to $-|1\rangle$. The state Bob receives is:

$$p|\psi\rangle\langle\psi| + (1-p)Z|\psi\rangle\langle\psi|Z.$$

3. **Bit flip and phase flip channel:** $E_0 = \sqrt{p}, E_1 = \sqrt{1-p}XZ$, etc.

There are many other possible errors that can corrupt the transmission, but for now, let's focus on the above three. How can Bob detect and correct these quantum errors?

First, note that the phase flip error becomes a bit flip error if you choose the $|\pm\rangle$ basis. This is because:

$$Z|+\rangle = |-\rangle \quad \text{and} \quad Z|-\rangle = |+\rangle.$$

This suggests that if we know how to deal with a bit flip error, we can apply the same methodology to handle the phase flip error.

To correct bit flip errors, we introduce redundancy by adding two more qubits, creating the so-called logical qubits:

$$|0\rangle_L = |0,0,0\rangle \quad \text{and} \quad |1\rangle_L = |1,1,1\rangle.$$

This can be done using a successive application of $CNOT$ gates:

$$CNOT_{13}CNOT_{12}|a\rangle|0\rangle|0\rangle = |a,a,a\rangle,$$

where $a = 0, 1$, and $CNOT_{12}$ is a $CNOT$ gate where the first qubit controls the second one, while $CNOT_{13}$ is a $CNOT$ where the first qubit controls the third qubit. We now have:

$$CNOT_{13}CNOT_{12}|\psi\rangle|0,0\rangle = a_0|0,0,0\rangle + a_1|1,1,1\rangle = a_0|0\rangle_L + a_1|1\rangle_L = |\psi\rangle_L.$$

Now, if $|\psi\rangle_L$ goes through our bit flip error channel, any of the three qubits can experience an error. It can also happen, albeit with a much smaller probability, that two or even three qubits flip. However, let's assume we can ignore the possibility of more than one bit flip error. In this case, we can detect which qubit flipped as follows:

First, we measure the operators Z_1Z_2 followed by Z_2Z_3. These sequential measurements are possible because Z_1Z_2 and Z_2Z_3 commute, i.e.,

$$[Z_1Z_2, Z_2Z_3] = 0.$$

These operators have eigenvalues ±1. They are called syndrome measurement operators.

Suppose that the bit flip error occurred on the first qubit, i.e., Bob received the state $X_1|\psi\rangle_L = a_0|1,0,0\rangle + a_1|0,1,1\rangle$. The measurement of Z_1Z_2 gives -1, and the measurement of Z_2Z_3 gives $+1$. Since the -1 from Z_1Z_2 could result from either $Z_1 = +1$ and $Z_2 = -1$, or from $Z_1 = -1$ and $Z_2 = +1$, we do not immediately know whether the error occurred on the first qubit or the second one. This is where the second measurement comes to the rescue. If the second qubit had flipped, Z_2Z_3 would have yielded -1 instead of $+1$, and thus it must have been the first qubit that flipped. Note, that the operators Z_1Z_2 and Z_2Z_3 don't affect the corrupted state except for introducing a global phase, i.e., $Z_1Z_2X_i|\psi\rangle_L = \pm X_i|\psi\rangle_L$ and $Z_2Z_3X_i|\psi\rangle_L = \pm X_i|\psi\rangle_L$. Once you know the error happened on the ith qubit, you can simply apply X_i to remove it.

The above procedure can be understood in a different way too. Consider the following projective measurement, given by the rank two projection operators:

$$\begin{aligned}
P_0 &= |0,0,0\rangle\langle 0,0,0| + |1,1,1\rangle\langle 1,1,1| \\
P_1 &= |1,0,0\rangle\langle 1,0,0| + |0,1,1\rangle\langle 0,1,1| \\
P_2 &= |0,1,0\rangle\langle 0,1,0| + |1,0,1\rangle\langle 1,0,1| \\
P_3 &= |0,0,1\rangle\langle 0,0,1| + |1,1,0\rangle\langle 1,1,0|. \qquad (2.53)
\end{aligned}$$

If the outcome 0 happens, we know there is no error. Outcome 1 means the first qubit flipped, etc. Importantly, this measurement doesn't destroy the corrupted state, i.e., $P_iX_i|\psi\rangle_L = X_i|\psi\rangle_L$. In other words, it's a clever measurement that locates error without destroying our precious quantum superposition.

These two ways of understanding bit flip correction code are linked by the following observation

$$\begin{aligned}
Z_1Z_2 &= P_0 + P_3 - P_1 - P_2 \\
Z_2Z_3 &= P_0 + P_1 - P_2 - P_3. \qquad (2.54)
\end{aligned}$$

We see how degeneracy of these operators, i.e., two dimensional subspaces corresponding to ±1 eigenvalues, relates to bit flip location when measuring them.

Exercise 42. *Using the fact that a phase flip error in the computational basis behaves like a bit flip error in the $|\pm\rangle$ basis, design a phase flip error detection and correction scheme in a similar way to how we handled the bit flip error. [Hint: Set $|+++\rangle$ as $|0\rangle_L$ and $|---\rangle$ as $|1\rangle_L$.]*

So, how to correct for both errors happening together? We need to use an error correcting code invented by Shor. This code combines the bit flip code and phase flip code in a clever way via using the block structure.

Shor code uses 9 qubits that encode logical qubits:

$$\begin{aligned}|0\rangle_L &= (\frac{1}{\sqrt{2}})^3(|0,0,0\rangle + |1,1,1\rangle)\otimes(|0,0,0\rangle + |1,1,1\rangle)\otimes(|0,0,0\rangle + |1,1,1\rangle)\\ |1\rangle_L &= (\frac{1}{\sqrt{2}})^3(|0,0,0\rangle - |1,1,1\rangle)\otimes(|0,0,0\rangle - |1,1,1\rangle)\otimes(|0,0,0\rangle - |1,1,1\rangle). \quad (2.55)\end{aligned}$$

Detection of bit flip error is achieved by measuring $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$. Phase flip location is determined by measuring $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$.

Let's first discuss how to detect and correct a phase flip error in this code. If a phase flip occurs on qubit 1, 2, or 3, the measurement of $X_1X_2X_3X_4X_5X_6$ will give -1, and the measurement of $X_4X_5X_6X_7X_8X_9$ will give $+1$. If the phase flip affects qubits 4, 5, or 6, the first measurement gives -1 and the second measurement gives -1. For a phase flip on qubits 7, 8, or 9, the results are $+1$ and -1, respectively. Although we don't know exactly which individual qubit within the three groups has experienced the phase flip, we can still correct the error by applying $Z_1Z_2Z_3$, $Z_4Z_5Z_6$, or $Z_7Z_8Z_9$, depending on which group has been affected.

We correct bit flip in a similar fashion and this time, by design, we locate which one flipped, say ith one, and we correct it by applying X_i.

Another way to look at the bit flip and phase flip detection and correction in the 9-qubit Shor code is to recognize that the operators used to locate the bit flip and those used to locate the phase flip commute, and thus share common degenerate orthogonal projectors. These projectors divide the entire 9-qubit Hilbert space into orthogonal subspaces, each corresponding to bit flips and phase flips on different qubits.[30] For instance, there is a subspace where a bit flip occurred on the first qubit, and a phase flip on qubits 1, 2, or 3. This perspective immediately explains why it is possible to correct simultaneous bit and phase flips occurring on the same qubit.

[30] The phase flip subspace has an additional degeneracy related to the fact that the phase flip can only be located within a block of qubits.

This leads to a beautiful conclusion: the 9-qubit Shor code corrects for an arbitrary error affecting a single qubit. How is that possible? An arbitrary error in this setting is represented by a quantum channel, given by a set of Krauss operators acting on the ith qubit:

$$E_\alpha^{(i)} = a_\alpha + b_\alpha X_i + c_\alpha Z_i + d_\alpha Z_i X_i,$$

where $a_\alpha, b_\alpha, c_\alpha, d_\alpha$ are arbitrary complex numbers, constrained only by the requirement that

$$\sum_\alpha \left(E_\alpha^{(i)}\right)^\dagger E_\alpha^{(i)} = 1.$$

Since the syndrome measurement operators in the Shor code project onto respective subspaces corresponding to different errors, a superposition of errors generated by a Krauss operator

$$E_\alpha^{(i)}|\psi\rangle_L = a_\alpha|\psi\rangle_L + b_\alpha X_i|\psi\rangle_L + c_\alpha Z_i|\psi\rangle_L + d_\alpha Z_i X_i|\psi\rangle_L,$$

will collapse into the corresponding subspace and can be corrected.

Exercise 43. *Show that any qubit Krauss operator can be written as $E_\alpha = a + bX + cZ + dZX$ with the constraints stated in the main text.*

Chapter 3

Elements of quantum computation

In this chapter we discuss some essentials of quantum computing that should give you solid foundations to launch your own studies on the topic.

Before we formally define what a quantum computer is, we first review some general concepts of classical computation. We then illustrate quantum computation using the quantum circuit model. Following that, we will present several key examples of quantum algorithms to provide a better understanding of quantum computation and its advantages over classical computation. However, their translation into quantum computer algorithms may not necessarily provide insights into how and why they work.

3.1 Basics of classical computation

Classical computers deal with classical bit strings, which consists of 0s and 1s. The computation is just the evaluation of a function: given an n-bit input, it produces an m-bit output uniquely determined by that input. In other words, it computes the value of the function[1]

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

for a specified n-bit argument x. A function with an m-bit output is equivalent to m functions, each producing a one-bit output. Therefore, we can say that the fundamental task performed by a computer is the evaluation of

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

[1] We will also use $\mathbb{B}$ to represent the binary set $\{0,1\}$ whenever convenient.

which is usually called *Boolean function*.

With this concept of computation in mind, we can consider how to model the computation process. There are numerous models for computation, including the Turing machine and the circuit model. In classical computation, we often focus on the Turing machine. However, for the convenience of generalizing to quantum computation, we will focus on the circuit model. It is important to note that different models of computation are equivalent. More crucially:

Theorem. 3.1: Turing machine

The circuit model and the Turing machine are equivalent for classical computation.

Although we won't discuss the Turing machine in detail, let us provide a rigorous definition for completeness. We won't use the Turing machine to discuss computation in these lecture notes, but it is important to stress that our computers are built based on the Turing machine rather than the circuit model. This is the central concept for classical computation.

Definition. 3.1: k-tape Turing machine

A k-tape Turing machine M is a triple $M = (\Gamma, Q, \delta)$ where

- $\Gamma = \{0, 1, \cdots, d-1, \triangleright, \Box\}$ a finite set called tape symbol alphabet of Turing machine, where $0, 1, \cdots, d-1$ are input symbols, $\triangleright \in \Gamma$ marks the left-hand end of the tape and $\Box$ is the blank symbol (the only symbol allowed occur infinitely often on tape at any step during computation);
- $Q = \{q_0, q_1, \cdots, q_n\}$ a finite set whose elements are called states of Turing machine, q_0 is a special element called initial state and there exist a subset $F \subset Q$ consists of the states accepted by the machine (called halt states, final states or accepting states) for which Turing machine will eventually halt in.
- $\delta : (Q \setminus F) \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{-1, 0, +1\}$ a partial function[a] called transition function, where -1 (reps. 0; resp. $+1$) represents the left shift (resp. stand still; resp. right shift) of the read-write tape-head of the Turing machine.

[a] The term *partial function* means that the domain of the function is actually a subset of the one written down.

For interested readers, please refer to Chapter 3 of Nielsen and Chuang's book for a brief introduction.

3.1.1 Classical logic gates and circuit model

Let us now introduce the circuit model for classical computation. A circuit consists of wires and gates, where wires represent classical bit strings and gates represent classical

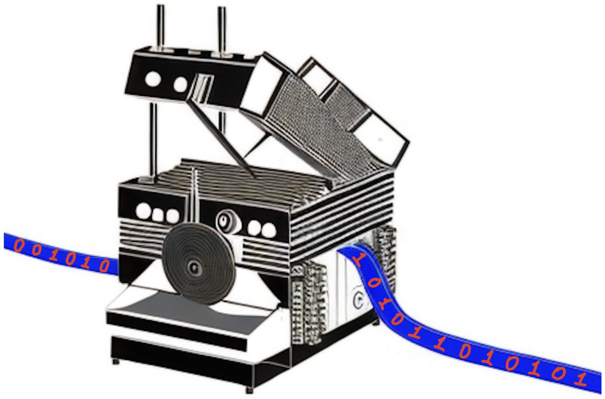
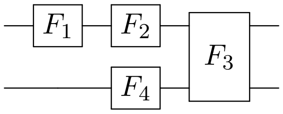


Figure 3.1: A schematic illustration of a Turing machine is shown. It features an infinite tape that records 0's and 1's, and the machine can read from and write to the tape. The figure was created with the help of AI.

operations. The gates are usually drawn as boxes, and the circuit is typically read from left to right. For example, a two-bit circuit is represented as



(3.1)

where two wires represent two bits and $F_1, \ldots, F_4$ are gates.

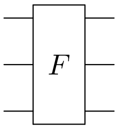
More formally, a gate can be defined as follows.

Definition. 3.2: Classical gate

An n-input, m-output **classical logic gate** (or simply, gate) is an m-component Boolean function

$$F : \{0,1\}^n \to \{0,1\}^m. \tag{3.2}$$

It can be represented by a box with m input wires and n output wires. For example:



(3.3)

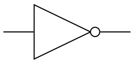
The study of these gates is crucial in mathematical logic and computation theory.

Single-bit gate

Notice that there are only 4 single-bit Boolean function $f : \{0,1\} \to \{0,1\}$[2]: (i) the identity function; (ii) $f \equiv 0$; (iii) $f \equiv 1$; (iv) the bit-flipping operation, which is the most interesting case. Thus it has its own name, NOT gate:

$$\mathsf{NOT} : a \to \bar{a} = 1 \oplus a. \tag{3.4}$$

Diagrammatically, it is denoted as follows (a notation commonly used in digital circuits if you are familar with that):



(3.5)

The little circle dot represents the NOT operation. If there is just a triangle, it represents the identity gate. For the NOT gate: when the input is 0, the output is 1; when the input is 1, the output is 0.

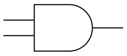
Two-bit input and single-bit output gate

For gates with a two-bit input and one-bit output, there are more possible gates. The number of such gates are the number of Boolean functions $f : \mathbb{B}^2 \to \mathbb{B}$ (simple calculation gives 2^4). Among these gates, there are two basic gates that play a crucial role in the study of logic and computation: the AND and OR gates:

- The operation of the AND gate is as follows:

  $$\mathsf{AND} : (a,b) \mapsto a \wedge b = ab. \tag{3.6}$$

  Diagrammatically, the AND gate is denoted as

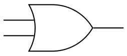
  (3.7)

  When both input bits are 1, it outputs 1; otherwise, it outputs 0.

- The operation of the OR gate is as follows:

  $$\mathsf{OR} : (a,b) \mapsto a \vee b = a \oplus b \oplus ab. \tag{3.8}$$

  Diagrammatically, the OR gate is denoted as

  (3.9)

  When both input bits are 0, it outputs 0; otherwise, it outputs 1.

[2] Since there are 2 possible outputs for each input 0 and 1, there are a total of $2 \times 2 = 4$ possibilities.

Notice that AND, OR, and NOT are independent of each other, meaning that no gate can be obtained by combining the other two (try to prove this yourself!). The composition of ANDwith NOT is called NAND. Similarly, the composition of OR with NOT is called NOR.

Other gates with two two-bit input and one-bit output that play crucial roles in studying circuit model are XOR and XNOR gates:

- For XOR gate, when two input values are different, it outputs 1; when two input values are the same, it output 0, i.e.,
$$\mathsf{XOR} : (a, b) \mapsto a \oplus b. \tag{3.10}$$
- For XNOR gate, when two input values are the same, it outputs 1; when two input values are different, it outputs 0, i.e.,
$$\mathsf{XNOR} : (a, b) \mapsto \overline{a \oplus b} = 1 \oplus a \oplus b. \tag{3.11}$$

Classical cloning operation

Besides the above 1-input 1-output or 2-input and 1-output gates, there are two other gates that we will use later: cloning gate and swap gate. The reason we introduce them is because that they play crucial roles in quantum computation. The swap gate in quantum computation play a similar role as that for classical swap, which just changes the orders of the inputs:
$$\mathsf{SWAP} : (a, b) \to (b, a). \tag{3.12}$$
A complicated swap operation can be decomposed into the composition of two-bit swap.

A more subtle gate is the cloning gate. In classical computation, this is usually called FANOUT (correspondingly, we have FANIN). Here, we will refer to it as the 1-to-n CLONE gate:
$$\mathsf{CLONE} : a \mapsto (a, \ldots, a). \tag{3.13}$$
The cloning gate essentially copies the input bit n times. However, in quantum mechanics, there is famous theorem call no-cloning theorem:

Theorem. 3.2: No-cloning theorem

In the quantum setting, there is no cloning gate that can copy an arbitrary unknown state perfectly.

Proof. There are two different types of proof. One is based on the assumption of unitarity of cloning gate another one is based on the linearity of the cloning gate. Here we take the first approach.

Assume we have a unitary gate U tha maps arbitrary $|\psi\rangle$ to $|\psi\rangle \otimes |\psi\rangle$. More rigorously,

$$U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle. \tag{3.14}$$

Now for two different state ψ_1, ψ_1, we have

$$U(|\psi_1\rangle \otimes |0\rangle) = |\psi_1\rangle \otimes |\psi_1\rangle, \tag{3.15}$$

$$U(|\psi_2\rangle \otimes |0\rangle) = |\psi_2\rangle \otimes |\psi_2\rangle. \tag{3.16}$$

Taking inner product for left hand sides of the above two expressions gives

$$\langle\psi_1|\langle 0|U^\dagger U|\psi_2\rangle|0\rangle = \langle\psi_1|\psi_2\rangle. \tag{3.17}$$

Taking inner product for right hand sides of the above two expressions gives

$$(\langle\psi_1|\psi_2\rangle)^2. \tag{3.18}$$

Thus we have

$$\langle\psi_1|\psi_2\rangle = (\langle\psi_1|\psi_2\rangle)^2. \tag{3.19}$$

This is only possible when $\langle\psi_1|\psi_2\rangle = 0, 1$, we thus obtain a contrary. □

This is a crucial difference between classical and quantum computation circuits. In classical computation, we usually implicitly assume the existence of a cloning gate.

To end this part, let us summarize the gates that we have learned:

- AND gate: $\mathsf{AND} : (a, b) \mapsto a \wedge b = \mathsf{NAND}$.
- OR gate: $\mathsf{OR} : (a, b) \mapsto a \vee b = a \oplus b \oplus ab$.
- NAND gate: $\mathsf{NAND} : (a, b) \mapsto 1 \oplus ab$.
- NOR gate: $\mathsf{NOR} : (a, b) \mapsto a \vee b = 1 \oplus a \oplus b \oplus ab$.
- XOR gate: $\mathsf{XOR} : (a, b) \mapsto a \oplus b$.
- NOR gate: $\mathsf{XNOR} : (a, b) \mapsto \overline{a \oplus b} = 1 \oplus a \oplus b$.
- SWAP gate: $\mathsf{SWAP} : (a, b) \to (b, a)$.
- CLONE gate: $\mathsf{CLONE} : a \mapsto (a, \ldots, a)$.

3.1.2 Universal gate set

Consider a general function

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

which can be regarded as m Boolean functions

$$f_i : \{0,1\}^n \to \{0,1\}, \quad \text{for } i = 1, 2, \ldots, m.$$

We say that a given set of gates is universal if any Boolean function can be computed by composing these gates.

We have the following result:

Theorem. 3.3: Universal gate

The gate set $\{\mathsf{NOT}, \mathsf{AND}, \mathsf{OR}, \mathsf{CLONE}\}$ is universal.

In classical computation textbooks, the cloning gate is often omitted, as cloning any state is straightforward. We explicitly mention this gate here to emphasize the difference in quantum computation.

Exercise 44. *Try to prove Theorem 3.3 yourself or consult a standard textbook on classical computation; it is not difficult.*

Hint: Examining examples of decomposing a general Boolean function into compositions of these gates will provide you with some intuition.

Theorem 3.3 guarantees that we can construct complex logic computations using basic operations. This is essentially how chips function! Also note that the universal gate set is not unique and there infinite sets of them. A standard approach to proving the universality of a set of gates is to demonstrate that it can construct all gates in a known universal gate set.

3.2 Quantum computation and quantum circuit model

We are now prepared to formulate a mathematical model of a quantum computer. We will extend the classical circuit model of computation to the quantum circuit model of quantum computation. Essentially, a quantum computer is a quantum circuit consisting of preparing the input state, applying the quantum circuit operations, and implementing measurements to obtain results. Quantum mechanics is inherently probabilistic, so we obtain results with their corresponding probabilities and select the most probable outcomes. To summarize, we have the following rough mathematical definition of a quantum computer:

Definition. 3.3: Quantum computer

A quantum computer is a physical realization of the quantum circuit model. To compute a function $f : \mathbb{B}^n \to \mathbb{B}^m$, there are three basic steps:

- Encode the classical information, such as bit strings, into quantum states and prepare this input state.
- Set up the quantum circuit that realizes the function f, and implement the quantum circuit on the input state.
- Implement quantum measurement to read out the results.

Note that, similar to classical computation, there exist several models for quantum computation, such as the quantum Turing machine, one-way quantum computation, magic state computation, and the quantum circuit model, but they all have the same computational power. In this lecture note, we will use the quantum circuit model for illustration, as it is more intuitive and convenient.

3.2.1 Encoding classical information into quantum states

Classical information is represented by bit strings such as

$$00100110,$$

but for each qubit, we use a Bloch sphere. This raises a natural question about how to encode classical information into quantum states. Typically, and more simply, we replace the classical bit 0 with the qubit state $|0\rangle$ and the classical bit 1 with the qubit state $|1\rangle$:

$$|00100110\rangle.$$

This is usually called *basis encoding.*

If we are given a more complicated data set, $\mathcal{X} \subset \{0,1\}^n$ (or $\mathcal{X} \subset \mathbb{R}^n$), it can be encoded in quantum states via a bijective map

$$\vec{x} \mapsto \psi_{\vec{x}}, \quad \forall \vec{x} \in \mathcal{X}, \tag{3.20}$$

where $\psi_{\vec{x}}$ is a pure state [3]. A natural requirement is that this should be a one-to-one map that $\psi_{\vec{x}}$ and $\psi_{\vec{y}}$ can be distinguished via quantum operations. This kind of problem plays a crucial role in quantum machine learning. Although we won't discuss much details in this direction. Here we will give two more approaches to encode the classical information into quantum states.

[3] You may ask, whether we can encode classical information into mixed states, the answer is yes. But we won't go that far in these lecture notes.

- The basis encoding can be rephrased as follows. For n-bit string $\vec{x} \in \mathcal{X}$, choose a n-qubit Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ and maps $\vec{x} = (x_1, x_2 \cdots, x_n) \in \mathcal{X}$ to the basis

  $$\vec{x} \mapsto |\vec{x}\rangle = \bigotimes_{i=1}^{n} (\cos(x_i)|0\rangle + \sin(x_i)|1\rangle). \tag{3.21}$$

  Notice that $x_i = 0, 1$, thus $(\cos(x_i)|0\rangle + \sin(x_i)|1\rangle) = |x_i\rangle$. The power of the above formula becomes evident when x_i are real numbers. This type of encoding is called angle encoding.

- We can encode the information into amplitudes and obtain amplitude encoding. By introducing a feature map $\vec{f} : \mathcal{X} \to \mathbb{R}^N$, we can encode classical data in an N-dimensional feature Hilbert space as

  $$\vec{x} \mapsto |\psi_{\vec{x}}\rangle = \frac{1}{\|\vec{f}(\vec{x})\|_2} \sum_i f_i(\vec{x})|i\rangle, \tag{3.22}$$

  where $\|\vec{f}(\vec{x})\|_2 = \left(\sum_i f_i(\vec{x})^2\right)^{1/2}$ and $i = 1, \cdots, N$. A frequently used example of a feature map $\vec{f}$ is defined as $f_i(\vec{x}) = x_i$, which means taking the i-th component of $\vec{x}$. In this case, the encoding is also referred to as wavefunction encoding.

3.2.2 Quantum circuit

A quantum circuit consists of quantum wires that represent quantum states and quantum gates that represent unitary operations. Additionally, there will also be measurements, which are usually performed at the final step. The quantum circuit model for quantum computation was proposed by David Deutsch in 1985[4]. The quantum circuit notation also appeared in Richard P. Feynman's paper[5].

We have the following definition of quantum gates. Essentially, you can regard quantum gates simply as unitary operations, but they can also be generalized to quantum channels (completely positive trace-preserving (CPTP) maps).

Definition. 3.4: Quantum gates

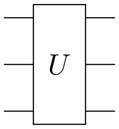
An n-qubit gate is a unitary operator

$$U : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n},$$

[4] David Deutsch. Quantum theory, the Church–Turing principle, and the universal quantum computer. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 400, pages 97–117, 1985.

[5] Richard Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6/7):467–488, 1982.

which maps an n-qubit state to a n-qubit state. It can be represented by a box with n input quantum wires and n output quantum wires. For example:



(3.23)

As we will see, the basic building blocks of a quantum circuit can be reduced to some single-qubit gates and two-qubit gates. Using these gates, we can compute any given function, performing universal quantum computation.

Single qubit gate

In the quantum information section, we became familiar with many single qubit gates. Let us quickly review what we learned and provide further discussion from the perspective of quantum computation.

The Pauli gates are three of most crucial gates ($X = \sigma_x$, $Y = \sigma_y$, $Z = \sigma_z$):

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -1 \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.24}$$

They form basis of all unitary matrices in the following sense:

Theorem. 3.4: $SU(2)$ group

A singe qubit gate is in general a 2×2 unitary matrix

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{3.25}$$

which satisfy $U^\dagger U = UU^\dagger = I$. All single qubit gates form a group called unitary group, denoted by $U(2)$. Since U is unitary, its two eigenvalues are of the form e^{ia}, e^{ib}, and the determinant satisfies

$$\det U = e^{i\alpha},$$

i.e., the determinant is a phase. If $\det U = e^{i\alpha}$, we define $V = e^{-i\alpha}U$, then $\det V = 1$. Any unitary U is related to a unitary V that has determinant 1 up to a phase factor. Notice that unitary matrices whose determinants are one form a group called special unitary group

$$SU(2) = \{U|U^\dagger U = I = UU^\dagger, \det U = 1\}. \tag{3.26}$$

Suppose that we have a matrix as in Eq. (3.25), since $U^{-1} = U^\dagger$, for which we must use the formula of inverse matrix ($A^{-1} = A^{\mathrm{ad}}/\det A$, where A^{ad} is the adjugate matrix of A)

$$U^{-1} = \frac{1}{\det U}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

From condition that $\det U = 1$ and by comparing it with $U^\dagger$, we obtain that a general element in $SU(2)$ is of the form

$$U = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}, \ a, b \in \mathbb{C}, \ aa^* + bb^* = 1. \tag{3.27}$$

From the general expression (3.27) and by setting $a = t + iz$ and $b = y + ix$, we see that

$$U = tI + ixX + iyY + izZ, \quad t^2 + x^2 + y^2 + z^2 = 1. \tag{3.28}$$

Every $U \in SU(2)$ can be decomposed as a linear combination of Pauli operators.

Using the correspondence between $SU(2)$ group and three dimensional rotation group $SO(3)$, we can construction rotation operator about direction $\vec{n}$ as (we assume $\|\vec{n}\| = 1$):

$$R_{\vec{n}}(\theta) = \exp\left(-i\theta\frac{\vec{n}\cdot\vec{\sigma}}{2}\right) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\vec{n}\cdot\vec{\sigma}, \tag{3.29}$$

where $\vec{\sigma} = (X, Y, Z)$ are vector of Pauli matrices.

Three special examples are rotation gates about $\hat{x}$, $\hat{y}$ and $\hat{z}$ axes, defined as follows:

$$R_x(\theta) = e^{-i\theta X/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{3.30}$$

$$R_y(\theta) = e^{-i\theta Y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{3.31}$$

$$R_z(\theta) = e^{-i\theta Z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{3.32}$$

Exercise 45. *(1) Using the Pauli relation $\sigma_i\sigma_j = \delta_{ij}I + i\varepsilon_{ijk}\sigma_k$ [6]to show*

$$(\vec{a}\cdot\vec{\sigma})(\vec{b}\cdot\vec{\sigma}) = \vec{a}\cdot\vec{b}I + i(\vec{a}\times\vec{b})\cdot\vec{\sigma}. \tag{3.33}$$

(2) From the above result, it is clear that

$$(\vec{n}\cdot\vec{\sigma})^2 = I,$$

[6] If you are unfamiliar with this, check this by yourself! Notice δ_{ij} is delta symbol, and ε_{ijk} is Levi-Civita symbol.

when $|\vec{n}| = 1$. Use this to show that

$$\exp\left(-i\theta\frac{\vec{n}\cdot\vec{\sigma}}{2}\right) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\vec{n}\cdot\vec{\sigma}.$$

Hint: *First, Taylor expand the left-hand side, then compare both sides.*

Exercise 46. *Using Bloch representation of the qubit state ψ, show that $R_{\vec{n}}(\theta)$ is a indeed a rotation of of the Bloch vector $\vec{r}$ of ψ along the direction $\vec{n}$ with the angle θ.*

The Hadamard gate is of the form

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.34}$$

Notice that $H = (Z + Z)/\sqrt{2}$. It is frequently used to transform Pauli-Z basis $|0\rangle$ and $|1\rangle$ to the Pauli-X basis $|+\rangle$ and $|-\rangle$:

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{3.35}$$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{3.36}$$

As we will see later, in quantum computation, the initial state is usually of the form

$$|0\rangle|0\rangle\dots|0\rangle. \tag{3.37}$$

Applying Hadamard gate on this state will give us an equally distributed state for all possible bit strings:

$$H^{\otimes n}|0\rangle|0\rangle\dots|0\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{x}\in\mathbb{B}^n}|\vec{x}\rangle, \tag{3.38}$$

where summation over all possible n-bit strings $\vec{x} \in \mathbb{B}^{n}$[7].

There are another two gates that is frequently use in quantum computation, mainly due to that fact they are in a universal set that is used commonly, they are two rotations along $\hat{z}$-axis. The first one is the phase gate

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{3.39}$$

[7] It is worth making some comments on notations, by $\vec{x} = (x_1, \dots, x_n) \in \mathbb{B}^n$ we mean a bit string with length n, where $x_i = 0, 1$ for all $i = 1, \dots, n$. $|\vec{x}\rangle = |x_1\rangle|x_2\rangle\dots|x_n\rangle$. In the later discussion, we will encounter a given set of bit strings $\vec{x}_1, \dots, \vec{x}_m$. We will omit the vector notation sometimes to avoid clustering of equation and just denote them as $x_1, \dots, x_m$. So be careful about the notation.

This can be regarded as a rotation along z-axis with angle $\frac{\pi}{2}$, since

$$e^{i\frac{\pi}{4}}R_z(\pi/2) = S. \tag{3.40}$$

The second one is the T-gate:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix}. \tag{3.41}$$

This can be regarded as a rotation along $\hat{z}$-axis with angle $\pi/4$, since

$$e^{i\pi/8}R_z(\pi/4) = T. \tag{3.42}$$

Due to $\pi/2$ prefactor, T-gate is also called $\pi/8$-gate.

Having introduced some basic single qubit gates, let us now provide some systematical result about how to decompose the general single qubit unitary gates into some basic building blocks.

Theorem. 3.5: Single-qubit unitary

An arbitrary single unitary gate can be in the form

$$U = e^{i\alpha}R_{\vec{n}}(\theta), \tag{3.43}$$

where α and θ are real parameters and $\vec{n}$ is a unit vector.

Proof. This is a direct result of Theorem 3.4. □

Theorem. 3.6: Single-qubit unitary

An arbitrary single unitary gate can be decomposed into the following form

$$U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\zeta) \tag{3.44}$$

where α, β, γ, ζ are real parameters.

Theorem. 3.7: Single-qubit unitary

If $\hat{m}$ and $\hat{n}$ are two non-parallel unit vectors, then an arbitrary single unitary gate can be decomposed into the following form

$$U = e^{i\alpha}R_{\hat{m}}(\beta)R_{\hat{n}}(\gamma)R_{\hat{m}}(\zeta) \tag{3.45}$$

where α, β, γ, ζ are real parameters.

Two qubit gate

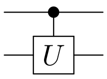
Two-qubit gates are fundamental in quantum information and quantum computation. Among these, the controlled operations are arguably the most important.

Consider a single qubit gate U, the controlled-U gate is defined as

$$C(U) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U = |0\rangle\langle 0| \otimes U^0 + |1\rangle\langle 1| \otimes U^1. \tag{3.46}$$

Represented diagrammatically, we have:

$C(U) =$



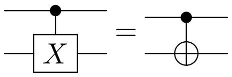
In the two-qubit circuit, the upper qubit is referred to as the control qubit, while the lower qubit is referred to as the target qubit. When the control qubit is $|0\rangle$ then we do nothing on target qubit, but when control qubit is $|1\rangle$ we apply the unitary gate U. If the control qubit is in a superposition state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and the target state is $|\varphi\rangle$, we have

$$C(U)|\psi\rangle \otimes |\varphi\rangle = \alpha|0\rangle \otimes |\varphi\rangle + \beta|1\rangle \otimes U|\psi\rangle. \tag{3.47}$$

Let us give some examples:

1. **CNOT gate**: When U is chosen as the bit-flip gate, represented by the Pauli X matrix, we obtain the famous CNOT gate. Due to its importance, the CNOT gate has its own circuit notation:

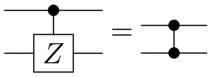
   CNOT =

   It maps $|a\rangle \otimes |b\rangle$ to $|a\rangle \otimes |a \oplus b\rangle$, with $a, b = 0, 1$.

2. **Controlled-Z gate**: When U is chosen as phase-flip gate represented by Pauli Z matrix, we obtain the controlled-Z gate. It is usually represented in quantum circuit as

   $C(Z) =$ (3.48)

Besides the controlled-U, there is another two qubit gate that is frequently used in quantum computation, which is SWAP gate. It's defined as

$$\mathsf{SWAP} = \sum_{a,b} |a\rangle\langle b| \otimes |b\rangle\langle a|. \tag{3.49}$$

The swap gate can be decomposed into summation form:

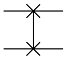
$$\mathsf{SWAP} = \frac{1}{2}(I \otimes I + X \otimes X + Y \otimes Y + Z \otimes Z). \tag{3.50}$$

The SWAP gate exchange the state for two qubits:

$$\mathsf{SWAP}|\psi\rangle \otimes |\varphi\rangle = |\varphi\rangle \otimes |\psi\rangle. \tag{3.51}$$

The quantum circuit notation for SWAP gate is as follows:

$$\mathsf{SWAP} = \tag{3.52}$$



Exercise 47. *Try to write down the matrix form of the above mentioned two qubit gates, including* CNOT, $C(Z)$ *and* SWAP.

3.2.3 Universal quantum gate

Recall that for classical computation, a given set of classical gates is called universal if any Boolean function can be computed using classical circuit built with these gates. For quantum computation, the universal gates can be defined similarly. A set of quantum gates are called are called universal if any n-qubit unitary can be approximated via the quantum circuit built from these gates.

To make it more rigorous, consider a given collection of quantum gates,

$$\mathcal{U} = \{U_1, \cdots, U_n\}, \tag{3.53}$$

using the quantum circuit model, we could compose these gates in many different ways, each composition will result in a unitary operator. The set of unitary operators obtained from this gate set will be denoted as

$$\mathbf{Circ}(\mathcal{U}; n) = \{U = F_{\mathrm{circ}}(U_1, \cdots, U_n) | U_1, \cdots, U_n \in \mathcal{U}\}. \tag{3.54}$$

Definition. 3.5: Universal gate set

A set of quantum gates $\mathcal{U} = \{U_1, \cdots, U_m\}$ (not necessarily finite) is called universal if, for any $n \geq 1$, the unitary transform $V \in U(2^n)$ can be approximated with arbitrary accuracy and up to an overall phase by quantum circuit constructed with this gate set. More precisely, for all n-qubit unitary operator V and $\forall \varepsilon > 0$, there exist a unitary transform $U \in \mathbf{Circ}(\mathcal{U}; n)$ and a phase factor $e^{i\varphi}$ such that

$$\|U - e^{i\varphi} V\| \leq \varepsilon.$$

That is, for all $n \geq 1$, $\mathbf{Circ}(\mathcal{U}; n)$ is dense in $U(2^n)$.

Two-level unitary transformations are universal

When dealing with the problems related to the universal quantum gates, it is useful to introduce the notion of *two-level unitary transformation* in a given basis. Consider a d-dimensional Hilbert space $\mathcal{H}$ with standard basis $|0\rangle, \cdots, |d-1\rangle$, we call a unitary transformation U a two-level unitary transformation[8] if it can be decomposed as a direct sum $U = U^{(2)} \oplus I^{(d-2)}$, where $U^{(2)}$ is a unitary transformation acting on the two-dimensional space spanned by two basis states $|i\rangle$ and $|j\rangle$. Notice that the definition depends on the choice of the basis. In general, a two-level unitary transformation in basis $\mathcal{B}_1$ will not be a two-level unitary transformation in basis $\mathcal{B}_2$.

The set of all unitary transformations on $\mathcal{H}$ is denoted as $U(\mathcal{H})$. There is a useful result about the decomposition of arbitrary unitary transformation in $U(\mathcal{H})$ as a product of two-level unitary transformations.

Theorem. 3.8: Two-level gates are universal

Any unitary transformation U in $U(\mathcal{H})$ can be decomposed as a product of two-level unitary transformations $U_1, \cdots, U_k$ with $k \leq \frac{n(n-1)}{2}$, where $n = \dim \mathcal{H}$.

Proof. Since the basis is fixed, we can prove this in matrix form. Suppose that $U = (u_{ij})$ is a $n \times n$ matrix,what we want to do is make the entries of the first column be $1, 0, \cdots, 0$. To this end, we will construct some two-level unitary matrices $U_1, \cdots, U_n$ such that $U_n \cdots U_2 U_1 U = I$, thus $U = U_1^\dagger U_2^\dagger \cdots U_n^\dagger$. If $u_{2,1} = 0$, set $U_1 = I$; if $u_{2,1} \neq 0$, set

$$U_1 = \begin{pmatrix} \frac{u_{11}^*}{\sqrt{|u_{11}|^2+|u_{21}|^2}} & \frac{u_{21}^*}{\sqrt{|u_{11}|^2+|u_{21}|^2}} \\ \frac{u_{21}}{\sqrt{|u_{11}|^2+|u_{21}|^2}} & \frac{-u_{11}}{\sqrt{|u_{11}|^2+|u_{21}|^2}} \end{pmatrix}^{(1,2)} \oplus I^{(n-2)} = U^{(1,2)} \oplus I^{(n-2)},$$

where $U^{(1,2)}$ is a unitary matrix acting on the space spanned by the 1st and 2nd basis states. Then matrix $U_1 U$ will be of the form

$$U_1 U = \begin{pmatrix} u'_{11} & u'_{12} & \cdots & u'_{1n} \\ 0 & u'_{22} & \cdots & u'_{2n} \\ u'_{31} & u'_{32} & \cdots & u'_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ u'_{n1} & u'_{n2} & \cdots & u'_{nn} \end{pmatrix}$$

[8] This is different from the notion of a unitary transformation acting non-trivially on one qubit (recall the fact and one qubit is a two-level system), which means that U can be decomposed as a tensor product $U = U^{(2)} \otimes I^{(2^{n-1})}$. Thus, only for $\dim \mathcal{H} = 2$, two notions are the same.

We now construct U_2 in the same spirit. If $u'_{31} = 0$, set $U_2 = I$; if $u'_{31} \neq 0$, set U_2 as

$$U_2 = \begin{pmatrix} \frac{{u'}^*_{11}}{\sqrt{|u'{}_{11}|^2+|u'{}_{31}|^2}} & \frac{{u'}^*_{31}}{\sqrt{|u'_{11}|^2+|u'{}_{31}|^2}} \\ \frac{u'31}{\sqrt{|u'{}_{11}|^2+|u_{31}|^2}} & \frac{-u'{}_{11}}{\sqrt{|u'{}_{11}|^2+|u'{}_{31}|^2}} \end{pmatrix}^{(1,3)} \oplus I^{(n-2)} = U^{(1,3)} \oplus I^{(n-2)},$$

Then the first column of $U_2 U_1 U$ will be $(u''{}_{11}, 0, 0, u''{}_{41}, \cdots, u''{}_{n1})^T$. Similarly, we can construct $U_3, \cdots, U_{n-1}$ such that

$$V = U_{n-1} \cdots U_1 U = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ 0 & v_{22} & \cdots & v_{2n} \\ 0 & v_{32} & \cdots & v_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & v_{n2} & \cdots & v_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & v_{22} & \cdots & v_{2n} \\ 0 & v_{32} & \cdots & v_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & v_{n2} & \cdots & v_{nn} \end{pmatrix} = I^{(1)} \oplus U',$$

where the second equality is from the unitarity of V. With this dimension reduction process and by induction on the rank of matrix U, we complete the proof. Since in each dimension reduction process $I^{(n-d)} \oplus U^{(d)} \to I^{(n-d+1)} \oplus U^{(d-1)}$ (d to $d-1$ dimensions), at most $n-1$ two-level unitary matrices are needed, thus the total number of two-level unitary matrices appeared in the decomposition is less than or equal to $(n-1)+(n-2)+\cdots+1 = \frac{n(n-1)}{2}$. □

Theorem 3.8 is quite crucial when we try to prove some given gate set is universal. We can first try to prove that any two-level unitary operators can be realized by quantum circuit built from these gates, then from the Theorem 3.8 we conclude that the gate set is universal.

Theorem. 3.9: Universal gate set

The CNOT gate together with all single qubit gates form a universal gate set.

The proof is not difficult but tedious; we won't prove the theorem here. Interested readers can consult Nielsen and Chuang's book.

Having establish the fact that all single qubit gates together with CNOT gate, we now take one step further. Can we find some basic single qubit gates that can approximate all single unitary gates with arbitrary accuracy? The answer is yes! The Hadamard gate and T-gate can achieve this. Proving this fact is a little beyond our scope, let us just recall that T gate is a rotation along z-axis with angle $\pi/4$, HTH will be a rotation along x-axis. Combining them, we obtain a rotation along an axis $\hat{n}$ with an irrational multiple of 2π. Applying it enough number of times, we can approximate all rotation along $\hat{n}$ with arbitrary accuracy. Since we are considering an irrational $\alpha \in [0, 1]$, on an periodic real line $\mathbb{R}/\mathbb{Z}$, $n\alpha$ never be an integer, thus dense on the line. We obtain a rotation

$$R_{\hat{n}}(\theta), \alpha \in \mathbb{R}. \tag{3.55}$$

Applying Hadamard conjugate, we obtain

$$HR_{\hat{n}}(\theta)H = R_{\hat{m}}(\gamma), \gamma \in \mathbb{R} \tag{3.56}$$

Due to the fact that $\hat{m}$ and $\hat{n}$ are non-parallel, from Theorem 3.7, we see that any single qubit unitary operator can be realized by $R_{\hat{m}}$ and $R_{\hat{n}}$. This further implies the H and T can approximate all single unitary gates with arbitrary accuracy.

Theorem. 3.10: Universal gate set

The CNOT gate, Hadamard gate H and T-gate are universal.

3.3 Complexity of quantum circuit model

There are two type of complexity we will encounter in quantum computation, one is query complexity and the other is circuit complexity. Both types of complexity describe how the resources required to solve a problem scale with the size of the input. They are typically expressed as functions of the number of input qubits. This mirrors how classical computational complexity is often described in terms of the size of the input (number of bits). For example, if we consider a polynomial complexity, it means that the complexity grows as some polynomial function of the number of qubits. A typically used notation is called big O notation for measuring complexity for both quantum and classical cases.

Definition. 3.6: Big O notation

If an algorithm has a time complexity of $O(f(n))$, this means that for sufficiently large n, the algorithm's runtime grows no faster than $f(n)$, up to a constant factor. Formally, an algorithm is $O(f(n))$ if:

$$T(n) \leq C \cdot f(n) \quad \text{for all sufficiently large } n,$$

where $T(n)$ is the actual time (or space, or number of queries) taken by the algorithm, and C is a constant.

The following are some examples:

- **Constant Time:** $O(1)$ — The runtime does not depend on the input size.
  - Example: Accessing an element in an array by its index.
- **Logarithmic Time:** $O(\log n)$ — The runtime grows logarithmically with the input size.
  - Example: Binary search in a sorted array.

- **Linear Time:** $O(n)$ — The runtime grows proportionally to the input size.
  - Example: Iterating through all elements in an array.
- **Linearithmic Time:** $O(n \log n)$ — The runtime grows proportionally to n times $\log n$.
  - Example: Efficient sorting algorithms like Merge Sort or Quick Sort (average case).
- **Quadratic Time:** $O(n^2)$ — The runtime grows proportionally to the square of the input size.
  - Example: Nested loops over an array (e.g., bubble sort).
- **Exponential Time:** $O(2^n)$ — The runtime grows exponentially with the input size.
  - Example: Solving the traveling salesman problem using brute force.

In Big O notation, constant factors and smaller terms are ignored because they become insignificant as n grows large. For example:

$$T(n) = 5n^2 + 3n + 10 \quad \text{is} \quad O(n^2),$$

because as n grows, the n^2 term dominates, and constants like 5 and lower-order terms like $3n$ become irrelevant. Big O provides a way to compare the efficiency of algorithms in a general sense and helps predict their behavior with large inputs. If an algorithm has polynomial query complexity like $O(n^2)$, it means the number of oracle queries grows quadratically with n, the number of qubits. If an algorithm has polynomial circuit complexity like $O(n^3)$, the number of quantum gates grows cubically with n. Thus, when we say that the complexity is polynomial, we are referring to how the resources (queries or gates) scale as a function of the number of qubits, similar to how complexity in classical computation is expressed in terms of the number of bits.

Both query and circuit complexity depend on the number of input qubits. Algorithms with polynomial complexity are typically referred to as efficient, whereas those with exponential complexity are considered inefficient. In quantum computation, our goal is to discover efficient quantum algorithms that exhibit exponential speedup compared to their classical counterparts[9]. This means that while the best classical algorithm has exponential complexity, the quantum algorithm achieves polynomial complexity.

The complexities of many quantum algorithms are well understood. For a comprehensive collection of this information, you can visit the Quantum Algorithm Zoo, curated by

[9] Since exponential speedup is a big jump. Also notice that this speedup works only for large n; when n is small, a polynomial function can be greater than an exponential function.

Stephen Jordan (https://quantumalgorithmzoo.org/). It's worth exploring if you're interested. In the next section, we will take a closer look at some typical quantum algorithms like Grover search and Shor's algorithm.

3.3.1 Query complexity

Query complexity refers to the number of times an algorithm needs to access or ”query” an oracle (a black-box function) to solve a problem. It focuses on how efficiently the algorithm can extract information from the oracle. In quantum computation, query complexity often highlights the advantage quantum algorithms have over classical ones. For example, Grover's algorithm achieves quadratic speedup in query complexity, requiring only $O(\sqrt{N})$ queries to search an unsorted list of N elements, whereas a classical algorithm requires $O(N)$ queries.

3.3.2 Circuit Complexity

Circuit complexity refers to the number of quantum gates (unitary operations) and the depth (number of layers of gates that can be applied in parallel) needed to implement a quantum algorithm. This is a more direct measure of the resources needed to execute a quantum computation. Circuit complexity focuses on the structure of the quantum circuit that performs the computation, considering factors like the number of qubits, gates, and time steps (depth). Algorithms like Shor's algorithm and quantum Fourier transform are analyzed in terms of circuit complexity to evaluate their practical feasibility on quantum computers.

Chapter 4

Quantum Algorithms I: Speed-Up in Oracle Query Complexity

Quantum algorithms represent a revolutionary leap in computational theory, offering capabilities that far exceed the limits of classical computation for certain types of problems. These algorithms leverage key principles of quantum mechanics—such as superposition, entanglement, and interference—to solve complex tasks with unprecedented efficiency.

In this chapter, we focus on a class of quantum algorithms that rely on query oracles, which play a central role in evaluating specific functions or problems. As discussed in the previous section, quantum complexity can be understood in two ways: one is the number of queries made to an oracle, and the other is the circuit depth or number of basic gates in the quantum circuit. Here, we will delve into several foundational algorithms that emphasize the former, namely Deutsch-Jozsa, Simon's algorithm, and Grover's search. Each of these exploits quantum oracles to achieve exponential speedups or optimal search capabilities compared to their classical counterparts.

This section aims to deepen your understanding of these query-based quantum algorithms by exploring their mathematical foundations, quantum circuit designs, and the broader implications they have for computational theory. By examining how quantum queries provide insight into hidden properties of functions or databases, we will see the extraordinary potential of quantum computation in solving problems that are considered intractable for classical computers.

4.1 Deutsch-Jozsa algorithm

The Deutsch–Jozsa algorithm, introduced by David Deutsch and Richard Jozsa in 1992, is a deterministic quantum algorithm. While it has limited practical applications, it is notable for being one of the earliest quantum algorithms to demonstrate an exponential speed advantage over any deterministic classical algorithm. Studying these algorithm will provide us with valuable insights into how quantum algorithms function.

Problem statement of Deutsch-Jozsa algorithm

The function

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

accepts n-bit binary values as inputs and outputs either a 0 or a 1 for each input. It's guaranteed that the function is either constant (producing 0 for all inputs or 1 for all inputs) or balanced (outputting 1 for exactly half of the inputs and 0 for the other half). The objective is to determine whether the function is constant or balanced by querying the oracle. For instance, consider a one-bit input function $f : \{0,1\} \to \{0,1\}$. The function is constant if $f(0) = f(1)$ and balanced if $f(0) = f(1) \oplus 1$ (which simply means it is not constant, where $\oplus$ denotes addition modulo 2). The task is equivalent to checking whether $f(0) \oplus f(1) = 0$ (constant) or 1 (balanced).

This is a black box problem that can be solved efficiently and with perfect accuracy by a quantum computer. 'Efficient' means that we solve the problem with polynomial number of queries to the black box. In contrast, a deterministic classical computer would require an exponential number of queries to the black box to arrive at a solution. It is also mentioning that the classical probabilistic computer can also solve this problem efficiently. So the quantum advantage of this algorithm is over deterministic classical computer.

Classical brute-force solution

In the worst-case scenario, a classical algorithm requires more than half of the inputs, i.e., $2^{n-1} + 1$ evaluations, to confidently distinguish between constant and balanced functions.

4.1.1 Deutsch algorithm: $n = 1$ Deutsch-Jozsa

If you do not use quantum mechanics, you need to calculate f twice to accomplish the task. There is no way around it. What if you can use qubits and unitary operations? In order to do this, you need to "translate" f to a unitary operation U_f. Deutsch did it like this

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle. \tag{4.1}$$

Having U_f we are now ready to run Deutsch algorithm:

1. Apply Hadamard gates to an input state $|0,1\rangle$: $H \otimes H|0,1\rangle$.
2. Apply U_f: $U_f H \otimes H|0,1\rangle$.
3. Apply Hadamard to the first qubit: $H \otimes IU_f H \otimes H|0,1\rangle$.

4. Measure the first qubit in the computational basis.

Let's see, step by step, what happens:

$$\begin{aligned}|0,1\rangle &\rightarrow \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)\\ &\rightarrow \frac{1}{2}(|0,0\oplus f(0)\rangle - |0,1\oplus f(0)\rangle + |1,0\oplus f(1)\rangle - |1,1\oplus f(1)\rangle)\\ &= \frac{1}{2}(|0,f(0)\rangle - |0,1\oplus f(0)\rangle) + \frac{1}{2}(|1,f(1)\rangle - |1,1\oplus f(1)\rangle). \qquad (4.2)\end{aligned}$$

We stopped just before the final measurement. It's easy to calculate that the probability of getting 0 is

$$p(0) = \frac{1}{4}|1 + (-1)^{f(0)\oplus f(1)})|^2.$$

Thus $p(0)$ is 1 if the function is constant and 0 if it's balanced. This means that if we run our U_f only once we get the right answer with 100% certainty. Recall that in the classical world we need to calculate the function f twice. What happened here?

As you can see, in the second step of the algorithm we, sort of, evaluated the function f on all possible inputs and encoded this evaluation in a quantum superposition of qubits - some people call it "quantum parallelism". Of course, this superposition doesn't amount to much unless we can measure it. If we measured it in the computation basis, we would lose these parallel evaluations, so we need to be smarter than that and this is why we transfer it, via the final Hadamard, to the measurable amplitude of probability.

4.1.2 Deutsch-Jozsa algorithm

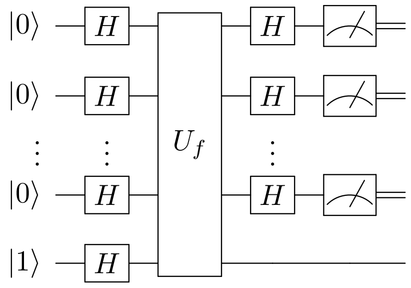
Now let us consider the general case with f have n-bit input. Similar as $n = 1$ case, the classical oracle must be replaced with a quantum oracle U_f which plays the role of evaluating the functions of f. By definition, the quantum oracle U_f is of the form

$$U_f|x\rangle|y\rangle = |x\rangle|y\oplus f(x)\rangle \qquad (4.3)$$

where $x = x_1 \cdots x_n$ is a n-bit sting, and $|x\rangle := |x_1\rangle \cdots |x_n\rangle$.

Exercise 48. *Prove that quantum oracle U_f is a unitary operation.*

The quantum circuit for Deutsch-Jozsa algorithm is as follows



The Deutsch-Jozsa algorithm works as follows:

1. **Initial Setup**: Start with an n-qubit register initialized to the state $|0\rangle^{\otimes n}$ and an auxiliary qubit initialized to $|1\rangle$:

$$|0\rangle^{\otimes n} \otimes |1\rangle$$

2. **Apply Hadamard Gate**: Apply the Hadamard gate to each of the n qubits of the input register and the auxiliary qubit. The Hadamard transformation takes the state to a superposition:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

3. **Oracle (Black-box) Query**: The oracle implements a unitary transformation U_f, defined as:

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$$

where $y \oplus f(x)$ denotes addition modulo 2 (XOR). After applying the oracle, the state becomes:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

Here we have used $|x\rangle \otimes \frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle) - |1 \oplus f(x)\rangle) = (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Since the auxiliary qubit is no longer needed, we can discard it, leaving:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$$

4. **Apply Hadamard Gate Again**: Apply the Hadamard gate to each of the n qubits of the input register. This transforms the basis states as:

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

After applying this transformation, the state becomes:

$$\frac{1}{2^n} \sum_{z=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{f(x)+x \cdot z} \right) |z\rangle$$

5. **Measurement**: Measure the n qubits. The result depends on whether the function $f(x)$ is constant or balanced:

   - **If $f(x)$ is constant**: The sum collapses to the state $|0\rangle^{\otimes n}$, and we always measure $z = 0$.
   - **If $f(x)$ is balanced**: The sum leads to destructive interference for the $z = 0$ term, and we never measure $z = 0$.

   If the measurement yields $|0\rangle^{\otimes n}$, the function is *constant.* If the measurement yields anything else, the function is *balanced*

Exercise 49. *Verify the states at each step align with that provided above.*

Exercise 50. *The formula*

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle \tag{4.4}$$

is a crucial trick that we will use repeatedly hereinafter, prove it and remember it. Notice that we have use the convention $x \cdot z = x_1 z_1 \oplus \cdots \oplus x_n z_n$.

The algorithm solves the problem with only one evaluation of the oracle, compared to the exponentially growing number of evaluations required by a classical algorithm. The quantum algorithm achieves this exponential speedup by leveraging quantum superposition and interference.

4.2 Grover's Search Algorithm

Grover's search algorithm is one of the cornerstone algorithms in quantum computation, providing a quadratic speedup for unstructured search problems. The search algorithm for an unstructured database is crucial in various applications. For example, it can be used to find the lowest-priced product, determine the shortest path to a destination via transfers, and more.

This Grover's algorithm is proposed by Lov Grover in 1996. Classical algorithms require $O(N)$ queries to search an unsorted database of N elements, but Grover's algorithm reduces this to $O(\sqrt{N})$ using quantum parallelism and amplitude amplification.

4.2.1 Problem Statement

Problem statement of Grover search algorithm

The goal of Grover's algorithm is to search for a unique item (or marked element) in an unsorted database. Given an oracle function $f : \{0,1\}^n \to \{0,1\}$, where $f(x) = 1$ for the marked item x_{marked} and $f(x) = 0$ for all other items, the task is to find x_{marked} using fewer queries to the oracle than classical algorithms would require.

We will assume that the search space contains $N = 2^n$ possible items. Classical search requires $O(N)$ oracle queries, since at the worst case, we need to query the oracle N times to find the marked item. The Grover's algorithm reduces this to $O(\sqrt{N})$.

4.2.2 Quantum Oracle

Recall that the classical oracle is a function $f : \{0,1\}^n \to \{0,1\}$, where $f(x) = 1$ for the marked item x_{marked} and $f(x) = 0$ for all other items. We can generalized this to a quantum oracle U_f.

The oracle is a quantum operation U_f that flips the sign of the amplitude of the marked state. It acts as follows:

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

For the marked state x_{marked}, $f(x_{\text{marked}}) = 1$, so the phase of this state is flipped:

$$U_f |x_{\text{marked}}\rangle = - |x_{\text{marked}}\rangle$$

For all other states $x \neq x_{\text{marked}}$, $f(x) = 0$, so their amplitudes remain unchanged.

Exercise 51. *This quantum oracle is related the the one defined in Eq. (4.3) by setting $|y\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Check it by yourself.*

4.2.3 Grover Iteration

The algorithm operates by repeatedly applying a process known as *Grover iteration* or *Grover operator* to amplify the amplitude of the marked state. Notice that if there are M marked items, we can introduce

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:\text{unmarked}} |x\rangle \tag{4.5}$$

which is the equal superposition state of all unmarked items; and

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x:\text{marked}} |x\rangle. \tag{4.6}$$

The crucial fact is that the equal superposition state of all items

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x} |x\rangle \tag{4.7}$$

can be rewritten as a superposition of $|\alpha\rangle$ and $|\beta\rangle$:

$$|\psi\rangle = \frac{\sqrt{N-M}}{\sqrt{N}} |\alpha\rangle + \frac{\sqrt{M}}{\sqrt{N}} |\beta\rangle. \tag{4.8}$$

$|\beta\rangle$ is our target state and we need to amplify the amplitude for $|\beta\rangle$.

$$\cos\frac{\theta}{2} = \frac{\sqrt{N-M}}{\sqrt{N}}, \quad \sin\frac{\theta}{2} = \frac{\sqrt{M}}{\sqrt{N}} \tag{4.9}$$

which is useful for us to understand Grover search geometrically. Hereinafter, for simplicity, we will assume $M = 1$, viz., there is only one marked item. We can set

Each Grover iteration consists of two main steps:

1. **Oracle Query (Phase Flip)**: Apply the oracle U_f, which flips the phase of the marked state.

2. **Amplitude Amplification (Grover diffusion operator)**: Apply the Grover operator D, which amplifies the probability amplitude of the marked state by inverting all amplitudes about their average. The diffusion operator can be expressed as:

   $$D = 2\,|\psi\rangle\,\langle\psi| - I$$

   where $|\psi\rangle$ is the equal superposition state $\frac{1}{\sqrt{N}} \sum_x |x\rangle$ and I is the identity operator.

Due to the fact that $|\psi\rangle$ in the plane spanned by $|\alpha\rangle$ and $|\beta\rangle$, the Grover iteration can be illustrated as a geometric transformation, see Fig. 4.1 We can introduce the Grover iteration operator as

$$G = DU_f = (2\,|\psi\rangle\,\langle\psi| - I)U_f. \tag{4.10}$$

Thus from the equal superposition state $|\psi\rangle$, one Grover iteration rotation $3\theta/2$ towards our target state $|\beta\rangle$.

Exercise 52. *Check that $G|\psi\rangle$ works as that illustrated in Fig. 4.1. Also try to show that the Grover iteration can be regarded as a rotation matrix in the basis of $|\alpha\rangle$ and $|\beta\rangle$.*

From Fig. 4.1, it's clear that θ in Eq. (4.9) must be small to reach the high precision. This means that M/N must be small. The Grover search can reach a high precision for large database.

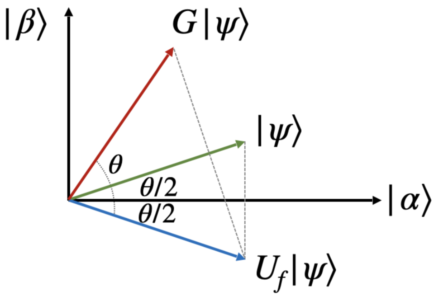


Figure 4.1: Illustration of the Grover iteration.

4.2.4 Steps of Grover's Algorithm

The following is a detailed procedure for implementing Grover search:

1. **Initialization**:
   - Start with n qubits in the state $|0\rangle^{\otimes n}$.
   - Apply Hadamard gates $H^{\otimes n}$ to create an equal superposition of all possible states:

   $$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

2. **Grover Iterations**:
   - Apply the Grover iteration $O(\sqrt{N})$ times. Each iteration consists of:

     (a) **Oracle Application**: Apply the oracle U_f, flipping the sign of the marked state.

     (b) **Amplitude Amplification**: Apply the diffusion operator D, amplifying the marked state's amplitude.

3. **Measurement**: After $O(\sqrt{N})$ iterations, the probability amplitude of the marked state will be close to 1. Measure the state to obtain x_{marked} with high probability.

4.2.5 Mathematics of Amplitude Amplification

Suppose the number of marked item is $M = 1$, let us analyze the iteration process more carefully. The key to Grover's algorithm lies in how the amplitudes of the states evolve under each Grover iteration. Initially, all states have equal amplitude $\frac{1}{\sqrt{N}}$. After applying the oracle and the diffusion operator, the amplitude of the marked state increases while the amplitude of the unmarked states decreases.

Each Grover iteration performs a rotation in a two-dimensional subspace spanned by the equal superposition of all states and the marked state. The angle of rotation is $\theta \approx 2\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$. After $k = O(\sqrt{N})$ iterations, the state vector is rotated close to the marked state, maximizing the probability of measuring x_{marked}.

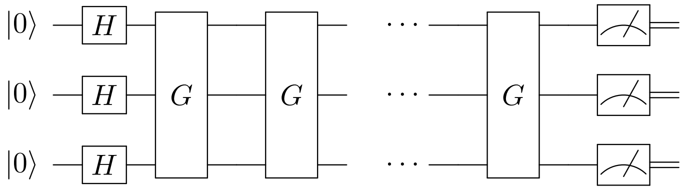
The optimal number of Grover iterations is approximately $\frac{\pi}{4}\sqrt{N}$. If the number of iterations exceeds this, the amplitudes will begin to rotate away from the marked state, reducing the success probability.

4.2.6 Quantum Circuit of Grover's Algorithm

The quantum circuit for Grover's algorithm consists of:

- **Hadamard Gates**: Apply $H^{\otimes n}$ to initialize the superposition state.
- **Grover iteration** G: Implement the Grover iteration many times, each Grover iteration consists of
  - **Oracle** U_f: Implement the oracle that flips the phase of the marked state.
  - **Diffusion Operator**: Apply the Grover diffusion operator, typically constructed using Hadamard gates, Z gates, and controlled operations.
- **Measurement**: Measure the qubits to obtain the marked state.

The following is an example of three-qubit circuit for Grover search:



Each iteration consists of applying the oracle U_f followed by the diffusion operator D.

4.2.7 Advantages and Limitations

- **Quadratic Speedup**: Grover's algorithm provides a quadratic speedup over classical search algorithms, requiring only $O(\sqrt{N})$ queries to the oracle.
- **Generality**: It works for any unstructured search problem where the oracle function can be defined.

- **Limitation**: Grover's algorithm only provides a quadratic speedup, so for very large databases, the improvement may not be sufficient to surpass classical methods.
- **Multiple Solutions**: If there are multiple marked elements, the algorithm still works but the number of iterations must be adjusted accordingly. The optimal number of iterations decreases as the number of marked elements increases.

Grover's Search Algorithm is a powerful tool in the quantum computing repertoire, illustrating how quantum mechanics can provide computational speedups for problems like unstructured search. Its quadratic speedup, while modest compared to the exponential gains offered by algorithms like Shor's factoring algorithm, demonstrates the potential of quantum algorithms for solving problems more efficiently than classical algorithms.

4.3 Simon's algorithm

Simon's algorithm is a quantum algorithm designed to solve the problem of finding a hidden string s in a specific class of functions more efficiently than classical algorithms. The s in the bit-wise addition, play the similar role as period of the function, thus the problem is also known as period finding problem in some cases. It is a notable example of quantum speedup, demonstrating how quantum computing can outperform classical methods in specific scenarios. The algorithm is proposed by Daniel R. Simon in 1994.

To be more precise, the Simon's algorithm is to solve the following problem:

Problem statement of Simon's algorithm

We want to find a hidden string s of length n given a function $f : \{0,1\}^n \to \{0,1\}^n$ that satisfies:

- **Periodic Condition**: $f(x) = f(x \oplus s)$ for all x in $\{0,1\}^n$, where $\oplus$ denotes bitwise XOR.
- **Distinct Values**: The function produces distinct outputs for any x such that x and $x \oplus s$ differ.

The goal is to determine the hidden string s with as few evaluations of f as possible.

Since f is a periodic function, note that in bitwise addition, any string satisfies $s + s = 0$. This differs slightly from our intuition for real-valued periodic functions, where if T is a period, then $2T$ is also a period. In bitwise addition, however, $2T$ always equals zero.

Classically, the best approach involves evaluating f multiple times, requiring up to $O(2^{n/2})$ evaluations [1]. This is due to the need to find pairs of inputs that yield the same output, which can be computationally intensive.

[1] Why $O(2^{n/2})$? Try to figure this out for small number of bits, this will help you to understand the problem of Simon's algorithm

4.3.1 Quantum Solution: Simon's Algorithm

The Simon's algorithm also need to using quantum oracles U_f which has a similar form as that for Deutsch-Jozsa algorithm:

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle .$$

Notice that $f(x)$ and y are now n-qubit states. The bitwise addition means that we do not treat them as binary numbers.

Exercise 53. *Prove the that quantum oracle defined above is a unitary operation.*

Simon's algorithm leverages quantum principles, particularly superposition and interference, to solve the problem in $O(n)$ evaluations of f. Unlike the Deutsch-Jozsa and Grover algorithms, Simon's algorithm requires some classical data processing after measurement.

Here's a detailed breakdown of the algorithm's steps:

1. **Initial Setup**
   Prepare two quantum registers:
   - An n-qubit input register initialized to $\left|0\right\rangle^{\otimes n}$.
   - An n-qubit output register also initialized to $\left|0\right\rangle^{\otimes n}$.

   The initial state of the system is:
   $$\left|0\right\rangle^{\otimes n} \left|0\right\rangle^{\otimes n}$$

2. **Apply Hadamard Gates**
   Apply the Hadamard gate H to all qubits in the input register to create a superposition of all possible input states:
   $$H^{\otimes n} \left|0\right\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \left|x\right\rangle$$
   The state now is:
   $$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \left|x\right\rangle \left|0\right\rangle^{\otimes n}$$

3. **Query the Oracle**
   Use the oracle U_f to compute $f(x)$ for each x:
   $$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle$$

After applying the oracle, the state becomes:

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

4. **Measure the Output Register**
   Measure the output register. This measurement yields a value $y = f(x_0)$ for some x_0. The state collapses to:

   $$|x_0\rangle \, |f(x_0)\rangle$$

   Due to the periodic nature of f, any x such that $f(x) = y$ satisfies:

   $$x \oplus s = x' \quad \text{(for some } x' \text{ such that } f(x') = y)$$

5. **Generate Linear Equations**
   The relationships derived from the measurements create linear equations involving s. For example:

   - If we find $f(x_1) = f(x_2)$, we have:

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

   where we have use the fact that any bitwise addition for any x with itself gives zero, $x \oplus x = 0$. Each measurement provides new information about s.

6. **Repeat the Process**
   Repeat steps 2 to 5 approximately n times. Each iteration provides a new equation involving s.

7. **Solve the System of Equations**
   Once enough equations are gathered, use classical linear algebra techniques (e.g., Gaussian elimination) to solve the system for the hidden string s.

Exponential Speedup: Simon's algorithm solves the problem using $O(n)$ evaluations compared to the classical $O(2^{n/2})$.

Chapter 5

Quantum Algorithms II: Speed-Up in Circuit Complexity

In the previous chapter, we discussed quantum algorithms that rely on query oracles. In this section, we shift our focus to algorithms that do not primarily use query oracles but instead transform the problem into a quantum circuit. Key examples include the Quantum Fourier Transform, Quantum Phase Estimation, and Shor's algorithm.

These circuit-based algorithms are not only vital for their computational power but also for their wide-reaching applications in fields like cryptography, optimization, and machine learning. By challenging long-held assumptions about computational complexity, they provide new frameworks for addressing real-world problems. As we examine the inner workings of these algorithms, we encourage you to experiment with practical implementations using Qiskit or actual quantum hardware, such as IBM's quantum computers.

5.1 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is the quantum analogue of the classical discrete Fourier transform (DFT) and plays a crucial role in many quantum algorithms, including quantum phase estimation, which is used to estimate the phase of a given unitary matrix, and Shor's algorithm for integer factorization. The QFT maps a quantum state to its frequency components and operates exponentially faster than the classical Fourier transform, making it a powerful tool in quantum computation.

5.1.1 Definition and Mathematical Overview

The classical discrete Fourier transform on a vector $(x_0, x_1, \ldots, x_{N-1})$ produces a new vector $(y_0, y_1, \ldots, y_{N-1})$, where each y_k is a linear combination of all the elements of the input vector.

Problem statement of QFT

Mathematically, the DFT problem is defined as:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \cdot e^{2\pi ijk/N}$$

where x_j and y_k are complex numbers. In quantum computing, we consider the label of x_j and y_k, given a quantum state $|j\rangle$, where $j \in \{0, 1, \ldots, N-1\}$, the QFT is defined by the transformation:

$$\mathrm{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N}\,|k\rangle$$

This means that for input state

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_j x_j |j\rangle, \tag{5.1}$$

the QFT gives

$$\mathrm{QFT}\,|\psi\rangle = \frac{1}{\sqrt{N}} \sum_k y_k |k\rangle. \tag{5.2}$$

Exercise 54. *Prove the QFT operation defined above is unitary.*

5.1.2 QFT Algorithm

We will assume $N = 2^n$ to be the number of basis states, where n is the number of qubits. One crucial point worth mentioning is that, to better understand the QFT, a solid understanding of binary number operations is essential. Suppose we have $j = j_1 j_2 \cdots j_n$ in binary number, this means

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0. \tag{5.3}$$

This implies

$$j/N = 0.j_1 j_2 \cdots j_n = j_1/2^1 + j_2/2^2 + \cdots + j_n/2^n. \tag{5.4}$$

Essentially, the QFT employs techniques involving binary numbers in the phase $e^{2\pi ijk/N}$, which contains

$$k \times 0.j_1 j_2 \cdots j_n. \tag{5.5}$$

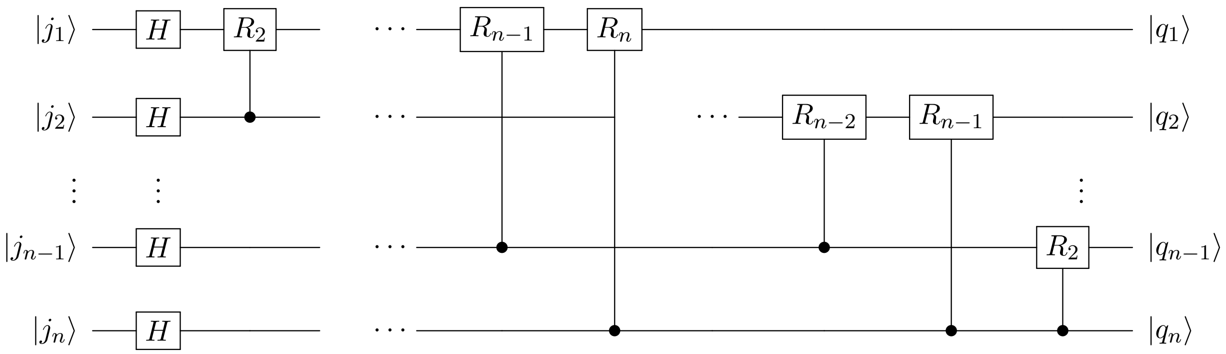
We only need to consider its decimal part, because the integer part contributes 1 to $e^{2\pi ijk/N}$. It can be proved via a little algebra of binary numbers that

$$\text{QFT} = \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)\cdots(|0\rangle + e^{2\pi i 0,j_1 j_2 \cdots j_n}) \tag{5.6}$$

Thus we need to design a circuit to realize this transformation. From Eq. (5.6), we see (we can guess) that we need to use Hadamard gate and the rotation gates of the form

$$R_l = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^l} \end{pmatrix} \tag{5.7}$$

For general n-qubit case, the quantum circuit of QFT algorithm is as follows:



The QFT algorithm works as follows:

1. **Apply Hadamard Gate to the First Qubit**
The first Hadamard gate creates an equal superposition of $|0\rangle$ and $|1\rangle$ on the most significant qubit:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

2. **Apply Controlled Phase Shift Gates**
After applying the Hadamard, the remaining qubits are adjusted with controlled phase shift gates. For the first qubit, the rotations are defined by the phase factor $e^{2\pi i/2^k}$, where k is the position of the qubit.

3. **Repeat for Remaining Qubits**
   The process is repeated for the second qubit, and so forth, until all qubits have been processed.

4. **Reverse the Qubit Order (Optional)**
   After applying the QFT, the qubits are typically in reverse order, so a swap operation is required to reorder the qubits if needed. This is because that the order for output qubit state is reversed.

Exercise 55. *Show that the QFT circuit we give above out the product state the same as Eq. (5.6) with the order reversed, i.e.,*

$$\begin{aligned} |q_1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle) \\ |q_2\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_2 j_3 \cdots j_n}|1\rangle) \\ &\vdots \\ |q_n\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \end{aligned} \tag{5.8}$$

The quantum Fourier transform can be implemented efficiently using a quantum circuit that consists of Hadamard gates and controlled phase shift gates. The circuit has $O(n^2)$ gates, which is exponentially faster compared to the classical Fourier transform, which takes $O(N \log N)$ time.

The Quantum Fourier Transform is a key building block in many quantum algorithms, including:

- **Shor's Algorithm**: QFT is used for order finding, which is the core part of factoring large integers efficiently.
- **Quantum Phase Estimation**: QFT is essential in determining the eigenvalue of a unitary operator.
- **Hidden Subgroup Problems**: Many problems such as the Deutsch-Jozsa and Simon's algorithm can be expressed in terms of finding hidden symmetries, and QFT is a useful tool in solving these efficiently.

5.1.3 Inversing QFT (IQFT)

The inverse of QFT (IQFT) is used to revert the state back from the Fourier domain to the computational basis. The inverse QFT follows the same structure as the QFT but with inverse phase shifts and a reverse ordering of operations. This is crucial for algorithms like quantum phase estimation algorithm and Shor's algorithm, where IQFT is applied after the phase estimation step.

5.2 Quantum Phase Estimation Algorithm

Consider a unitary matrix U, since $U^\dagger U = I$, we see that all eigenvalues of U is of the form $e^{i2\pi\theta}$ with $0 \leq \theta < 1$. The Quantum Phase Estimation (QPE) algorithm is a fundamental quantum algorithm used to estimate the phase (or eigenvalue) of a unitary operator. It is widely applied in quantum algorithms, including Shor's factoring algorithm and various quantum simulation techniques. The goal of QPE is to estimate the phase θ in the eigenvalue equation:

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$

where U is a unitary operator, $|\psi\rangle$ is an eigenstate of U, and θ is the phase that we wish to estimate. The phase θ is a real number in the interval $[0, 1)$.

5.2.1 Problem Statement

The problem of phase estimation can be summarized as follows:

Phase estimation problem

Given a unitary operator U and an eigenstate $|\psi\rangle$ such that:

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle .$$

If we represent θ as binary number, we have

$$\theta = 0.\theta_1\theta_2\theta_3\cdots , \tag{5.9}$$

the goal is to estimate the value of θ to n bits of precision. Notice that the eigenstate $|\psi\rangle$ is assumed to be the input datum.

5.2.2 Quantum phase estimation

The quantum phase estimation algorithm uses two quantum registers:

- **First register**: Contains n qubits, which will store the estimated phase.
- **Second register**: Contains the eigenstate $|\psi\rangle$, which is an eigenstate of the unitary operator U.

The output will be an n-bit approximation of the phase θ, such that:

$$\theta \approx 0.\theta_1\theta_2\ldots\theta_n$$

where $\theta_1, \theta_2, \ldots, \theta_n$ are the binary digits of θ.

The QPE algorithm consists of the following steps:

1. **Prepare Initial State**

   The system starts with two registers:

   - The first register is initialized in the state $|0\rangle^{\otimes n}$.
   - The second register is initialized in the eigenstate $|\psi\rangle$ of U.

   The initial state is:

   $$|0\rangle^{\otimes n} \otimes |\psi\rangle$$

2. **Apply Hadamard Gates**

   Apply Hadamard gates to each qubit in the first register, putting it into a superposition of all possible states:

   $$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |\psi\rangle$$

3. **Apply Controlled-U^{2^j} Operations**

   For each qubit j in the first register, apply the controlled-U^{2^j} operation. This step creates an entanglement between the first and second registers:

   $$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U^k |\psi\rangle$$

   Since $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, the state becomes:

   $$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ik\theta} |k\rangle \otimes |\psi\rangle$$

   At this point, the second register is no longer needed.

4. **Apply the Inverse Quantum Fourier Transform (QFT)**

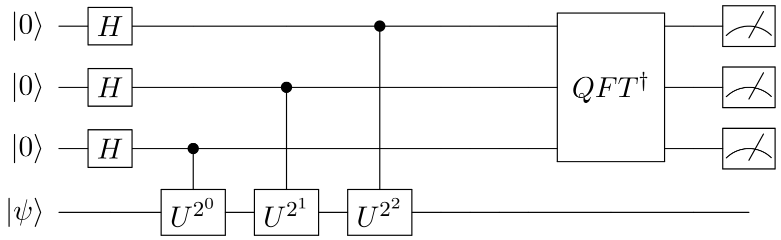
   Apply the inverse quantum Fourier transform (QFT†) on the first register to extract the phase information:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k\theta} |k\rangle$$

5. **Measure the First Register**

   Finally, measure the qubits in the first register. The measurement will yield the binary representation of the phase θ with an n-bit approximation.

Below is a quantum circuit that implements the Quantum Phase Estimation algorithm:



Here:

- Apply Hadamard gates to all qubits in the first register.
- Apply controlled-U^{2^j} gates where j is the index of the qubit in the first register.
- Apply the inverse Quantum Fourier Transform (QFT†) to the first register.
- Measure the first register to obtain an n-bit approximation of θ.

Exercise 56. *For the above three qubit QPE, calculate the explicit output state and show that they give the estimated phase.*

The QPE algorithm estimates the phase θ with a high probability of success. If θ can be exactly represented with n bits, the algorithm will output the correct result with probability 1. If θ cannot be exactly represented, the algorithm will output the closest n-bit approximation with a high probability.

Quantum Phase Estimation (QPE) has many applications and, like the Quantum Fourier Transform (QFT), is often used as a subroutine. The Quantum Phase Estimation algorithm is a key quantum algorithm that estimates the phase corresponding to an eigenvalue of a unitary operator. It is widely used in algorithms for factoring, quantum simulations, and quantum chemistry.

- **Shor's Algorithm**: QPE is used to find the period of modular exponentiation.
- **Quantum Simulations**: QPE is used to estimate eigenvalues of Hamiltonians, a critical task in quantum simulations.
- **Quantum Chemistry**: QPE is employed to compute the energy levels of molecular systems.

5.3 Shor's Factoring Algorithm

Shor's algorithm is a groundbreaking quantum algorithm for integer factorization, , it was proposed by Peter Shor in 1994. It finds the prime factors of a large composite integer exponentially faster than the best-known classical algorithms, posing a significant threat to classical cryptographic systems (like RSA), which rely on the difficulty of factorization.

5.3.1 Outline of Shor's Algorithm

Shor's algorithm combines quantum computing with classical number theory, particularly by leveraging the properties of periodic functions to solve the factorization problem. It consists of two main parts:

1. **Classical Part**: Reduce the factoring problem to finding the period of a periodic function.
2. **Quantum Part**: Use a quantum algorithm (Quantum Fourier Transform) to efficiently determine the period.

Problem Statement of Shor's factoring

Given a composite integer N, the goal is to find its prime factors, i.e., find integers p and q such that $N = p \cdot q$.

5.3.2 Shor's Algorithm

The factorization problem is reduced to a problem of finding the period of a periodic function. The key insight is based on the following steps (They will be explained in more details during the lecture):

Step 1: Choose a Random Number a

Pick a random number a such that $1 < a < N$. Then, compute the greatest common divisor (GCD) of a and N, i.e., $\gcd(a, N)$.

- If $\gcd(a, N) \neq 1$, then you've already found a non-trivial factor of N, so the algorithm terminates.
- If $\gcd(a, N) = 1$, proceed to the next step. This ensures that a and N are coprime.

Step 2: Find the Period r

The next goal is to find the period r of the function $f(x) = a^x \mod N$, where r is the smallest integer such that:

$$a^r \equiv 1 \mod N$$

This means that the sequence $a^x \mod N$ is periodic with period r. Finding r is the core of the quantum part of Shor's algorithm.

Step 3: Quantum Fourier Transform to Find r

To find r, the algorithm leverages the Quantum Fourier Transform (QFT). This is where the quantum speedup comes in. The QFT is used to efficiently find the period r by performing the following quantum steps:

1. **Initialize two quantum registers**:
   - The first register is initialized to an equal superposition of all possible states $|x\rangle$ where $x \in \{0, 1, \dots, 2^n - 1\}$.
   - The second register is initialized to the value $|1\rangle$, representing the current value of the modular exponentiation.
2. **Apply modular exponentiation**: Apply the function $f(x) = a^x \mod N$ to the second register, entangling it with the first. This results in the state:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |a^x \mod N\rangle$$

3. **Perform the Quantum Fourier Transform**: Perform the QFT on the first register. This efficiently extracts the period r from the periodic function.
4. **Measure the first register**: Measuring the first register yields information that, with high probability, allows you to deduce the period r.

Step 4: Use the Period r to Find the Factors

Once you have the period r, the next task is to use it to find factors of N. If r is even, then the following holds:

$$a^r \equiv 1 \mod N \implies (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \mod N$$

Thus, N divides the product $(a^{r/2} - 1)(a^{r/2} + 1)$. If neither $a^{r/2} - 1$ nor $a^{r/2} + 1$ is divisible by N, then compute:

$$\gcd(a^{r/2} - 1, N) \quad \text{and} \quad \gcd(a^{r/2} + 1, N)$$

At least one of these GCD computations will yield a non-trivial factor of N.

Step 5: Repeat If Necessary

If the result from Step 4 does not yield factors, the algorithm can be repeated with a different randomly chosen a.

5.3.3 Example of Shor's Algorithm

Let's work through an example where $N = 15$ and we want to factor it using Shor's algorithm.

1. **Choose a random number**: Let $a = 7$.

   $$\text{Compute } \gcd(7, 15) = 1$$

   Since $\gcd(7, 15) = 1$, proceed to the next step.

2. **Find the period r**: Using a quantum computer, we apply modular exponentiation to find the period of $f(x) = 7^x \mod 15$. The result is $r = 4$, as $7^4 \equiv 1 \mod 15$.

3. **Use the period to find the factors**:

   - Since $r = 4$ is even, compute $7^{r/2} - 1 = 7^2 - 1 = 48$.
   - Compute the GCD: $\gcd(48, 15) = 3$, which is a factor of 15.
   - Similarly, $7^{r/2} + 1 = 7^2 + 1 = 50$, and $\gcd(50, 15) = 5$, which is the other factor.

Thus, we have factored $15 = 3 \times 5$.

5.3.4 Quantum Speedup

Shor's algorithm provides an exponential speedup over classical algorithms. Classical factoring algorithms, like the general number field sieve, run in sub-exponential time. In contrast, Shor's algorithm runs in polynomial time, specifically:

$$O((\log N)^2 \log\log N \log\log\log N)$$

where N is the integer to be factored. This makes it feasible to factor large integers that are used in cryptographic protocols (e.g., 2048-bit RSA) on a quantum computer.

Shor's algorithm is one of the most important quantum algorithms because of its potential to break widely used cryptographic systems. By exploiting quantum parallelism and the quantum Fourier transform, it efficiently solves the integer factorization problem, which is intractable for classical computers. The development of practical, large-scale quantum computers could revolutionize cryptography, making Shor's algorithm a key focus of both theoretical and applied quantum computing research.

Bibliography