Quantum Information Processing

Quantum Information Theory

Richard Cleve

Institute for Quantum Computing & Cheriton School of Computer Science

University of Waterloo

October 30, 2023

Abstract

The goal of these notes is to explain the basics of quantum information processing, with intuition and technical definitions, in a manner that is accessible to anyone with a solid understanding of linear algebra and probability theory.

These are lecture notes for the third part of a course entitled "Quantum Information Processing" (with numberings QIC 710, CS 768, PHYS 767, CO 681, AM 871, PM 871 at the University of Waterloo). The other parts of the course are: a primer for beginners, quantum algorithms, and quantum cryptography. The course web site `http://cleve.iqc.uwaterloo.ca/qic710` contains other course materials, including some video lectures.

I welcome feedback about errors or any other comments. This can be sent to `cleve@uwaterloo.ca` (with "Lecture notes" in subject, if at all possible).

Contents

1 Quantum states as density matrices

Let's begin by considering a couple of situations where Alice has an apparatus for creating quantum states for Bob, who has no apparatus. When Bob needs a specific state, he asks Alice to create it and send it to him.

Figure 1: Alice uses her apparatus to prepare states for Bob.

The story of the fake-plus state

Suppose that Bob asks Alice to create a qubit in state $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and send it to him. But suppose that Alice's state preparation device is broken in that it can *only* prepare qubits in state $|0\rangle$ or $|1\rangle$.

What is Alice to do? She cannot create the state $|+\rangle$ literally. Suppose she tries to fake it by flipping a fair coin and then creating either $|0\rangle$ or $|1\rangle$, depending on the coin's outcome—while keeping the coin's outcome secret from Bob. The state that Alice creates can be described as

$$\begin{cases} |0\rangle & \text{with probability } \frac{1}{2} \\ |1\rangle & \text{with probability } \frac{1}{2}. \end{cases} \tag{1}$$

Let's call this the *fake-plus state*.

How good is this fake-plus state as a substitution for the real plus state $|+\rangle$? Is there any way that Bob can tell the difference? If, for any measurement that Bob can perform, the outcome probabilities are exactly the same then the substitution is a good one. Note that if Bob measures the fake plus state in the computational basis then the outcome probabilities are the same as measuring $|+\rangle$ in the computational basis. So far so good.

But there are other measurements for which the outcome probabilities are different. Can you think of one?

Exercise 1.1. *Give a measurement which has different outcome probabilities for the fake-plus state* (1) *than for the true plus state* $|+\rangle$.

So the fake-plus state is *not* a good substitute for the plus state.

The story of the fake-fake-plus state

Now, let's consider a different scenario. Suppose that Bob doesn't want a $|+\rangle$ state; instead, he wants Alice to prepare for him a fake-plus state, the state in Eq. (1). But this time, let's suppose that Alice's apparatus is broken in a different way: it can *only* prepare $|+\rangle$ and $|-\rangle$ states.

What can Alice do in this case? It won't do to send Bob a $|+\rangle$ state, because we already know that the plus state and the fake-plus state do not behave the same for all measurements. What if Alice tries to fake it this time by flipping a fair coin and then sending $|+\rangle$ or $|-\rangle$, depending on the outcome (again keeping the coin outcome secret). Such a state can be described as

$$\begin{cases} |+\rangle & \text{with probability } \frac{1}{2} \\ |-\rangle & \text{with probability } \frac{1}{2}. \end{cases} \tag{2}$$

Let's call this the *fake-fake-plus state*. Is the fake-fake-plus state (2) a good substitute for the fake-plus state (1)?

The two states certainly don't look the same. So your first guess might be that there's a measurement for which the outcome probabilities are different. But it turns out that, for *every* measurement that Bob can make, the outcome probabilities for state (1) are exactly the same as they are for state (2). Even though the two states don't look the same, the fake-fake-plus state is a good substitute for the fake-plus state.

1.1 Probabilistic mixtures of states

We are now in the realm of *probabilistic mixtures* of states. These are states where a random process is used to decide which pure state to prepare. Let $(p_1, p_2, ..., p_m)$ be a probability vector and let $|\psi_1\rangle, |\psi_1\rangle, \ldots, |\psi_m\rangle$ be d-dimensional quantum states (they need not be orthogonal). Imagine that $k \in \{1, 2, \ldots, m\}$ is sampled according the the probability distribution $(p_1, p_2, ..., p_m)$, and then state $|\psi_k\rangle$ is produced (but k is not revealed). Such a state can be described as

$$\begin{cases} |\psi_1\rangle & \text{with probability } p_1 \\ |\psi_2\rangle & \text{with probability } p_2 \\ \vdots & \vdots \qquad\qquad \vdots \\ |\psi_m\rangle & \text{with probability } p_m. \end{cases} \tag{3}$$

These states are called *mixed states*. The "ordinary" states, describable by a single normalized vector $|\psi\rangle$, are called *pure states*. In Eq. (3), if one of the probabilities is 1 and the others are 0 then the state is a pure state.

Let's look at some examples of probabilistic mixtures of states. We have already seen these two mixed states:

$$\begin{cases} |0\rangle & \text{with probability } \frac{1}{2} \\ |1\rangle & \text{with probability } \frac{1}{2} \end{cases} \quad \text{and} \quad \begin{cases} |+\rangle & \text{with probability } \frac{1}{2} \\ |-\rangle & \text{with probability } \frac{1}{2}, \end{cases} \tag{4}$$

and I claimed that they are indistinguishable—but I did not explain why. How do these two states compare with

$$\begin{cases} |0\rangle & \text{with probability } \frac{1}{2} \\ |-\rangle & \text{with probability } \frac{1}{2}? \end{cases} \tag{5}$$

Are they also indistinguishable from this state?

Mixed states for qubits can be probability distributions on any number of vectors, for example

$$\begin{cases} |0\rangle & \text{with prob. } \frac{1}{4} \\ |1\rangle & \text{with prob. } \frac{1}{4} \\ |+\rangle & \text{with prob. } \frac{1}{4} \\ |-\rangle & \text{with prob. } \frac{1}{4} \end{cases} \quad \text{and} \quad \begin{cases} |0\rangle & \text{with prob. } \frac{1}{3} \\ -\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle & \text{with prob. } \frac{1}{3} \\ -\frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle & \text{with prob. } \frac{1}{3}. \end{cases} \tag{6}$$

And the probability distribution need not be uniform, as shown in this example

$$\begin{cases} \cos(\frac{\pi}{8})|0\rangle - \sin(\frac{\pi}{8})|1\rangle & \text{with prob. } \cos^2(\frac{\pi}{8}) \\ \sin(\frac{\pi}{8})|0\rangle + \cos(\frac{\pi}{8})|1\rangle & \text{with prob. } \sin^2(\frac{\pi}{8}). \end{cases} \tag{7}$$

Definition 1.1 (indistinguishable states)**.** *Two probabilistic mixtures of states are* indistinguishable *if, for all possible measurements, the outcome probabilities are the same for the two states.*

Now, consider the six mixed states appearing in in (4)(5)(6)(7) above. Which pairs are indistinguishable? To address such questions, we need to understand these kinds of states better. A very useful approach is to express these states in terms of their *density matrices*.

1.2 Density matrices

Given a probability distribution on a set of state vectors, one might be tempted to consider the *weighted average* of the state vectors $p_1 |\psi_1\rangle + p_2 |\psi_2\rangle + \cdots + p_m |\psi_m\rangle$ as a useful object. However, this kind of average turns out to be of little use. One indicator that it's not worth much is that the weighted average can change dramatically depending on the global phases associated with the vectors—which shouldn't matter. Also, notice that, for the second mixed state in Eq. (6), the weighted average is the zero vector.

Mixed states can be nicely characterized by a different kind of averaging, which occurs in the definition of the density matrix.

Definition 1.2 (density matrix)**.** *For a mixed state of the form of Eq.* (3)*, where* $|\psi_1\rangle, |\psi_1\rangle, \ldots, |\psi_m\rangle$ *are d-dimensional, its* density matrix *is the* $d \times d$ *matrix*

$$\rho = p_1 |\psi_1\rangle \langle\psi_1| + p_2 |\psi_2\rangle \langle\psi_2| + \cdots + p_m |\psi_m\rangle \langle\psi_m| . \tag{8}$$

Note that the density matrix of any pure state $|\psi\rangle$ is $|\psi\rangle \langle\psi|$. For example, the state $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ has density matrix

$$\rho = \big(\alpha_0 |0\rangle + \alpha_1 |1\rangle\big)\big(\bar{\alpha}_0 \langle 0| + \bar{\alpha}_1 \langle 1|\big) \tag{9}$$

$$= \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \begin{bmatrix} \bar{\alpha}_0 & \bar{\alpha}_1 \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} |\alpha_0|^2 & \alpha_0 \bar{\alpha}_1 \\ \alpha_1 \bar{\alpha}_0 & |\alpha_1|^2 \end{bmatrix} . \tag{11}$$

The entries along the diagonal are the absolute values squared of the amplitudes and the off-diagonal entries are cross-terms involving the amplitudes.

Also note from Definition 1.2 that the density matrix of a probabilistic mixture of pure states is the weighted average of the density matrices of the pure states. For example, the density matrices of $|0\rangle$ and $|1\rangle$ are

$$|0\rangle \langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad |1\rangle \langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{12}$$

and the density matrix of the first mixed state in Eq. (4) is

$$\tfrac{1}{2} |0\rangle\langle 0| + \tfrac{1}{2} |1\rangle\langle 1| = \tfrac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \tfrac{1}{2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} . \tag{13}$$

Regarding the second mixed state in Eq. (4), the density matrices of $|+\rangle$ and $|-\rangle$ are

$$|+\rangle\langle+| = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad \text{and} \qquad |-\rangle\langle-| = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{14}$$

and the density matrix of their mixture is

$$\tfrac{1}{2}|+\rangle\langle+| \;+\; \tfrac{1}{2}|-\rangle\langle-| = \tfrac{1}{2}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + \tfrac{1}{2}\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{15}$$

Notice that the density matrices for the two mixed stated in Eq. (4) are the same. This is related to the fact that the states are indistinguishable—which will be explained shortly.

Exercise 1.2 (a straightforward calculation)**.** *Work out the density matrices of the mixed states appearing in* (5)(6)(7)*.*

Let me make a comment about global phases in vector states. When we represent (pure) states as vectors, there's this issue that if multiply the vector by a unit complex number (of the form $e^{i\theta}$, for $\theta \in \mathbb{R}$) then it's essentially the same state; it's indistinguishable from the original state. The density matrix of $e^{i\theta}|\psi\rangle$ is

$$e^{i\theta}|\psi\rangle\langle\psi|e^{-i\theta} = |\psi\rangle\langle\psi|. \tag{16}$$

So global phases don't even show up in density matrices, which is nice. It means that, using density matrices, we don't need to define an equivalence relation to account for global phases.

Now, let's look at some examples of density matrices for higher dimensional systems than qubits. The density matrix of $|00\rangle$ is

$$|00\rangle\langle 00| = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{17}$$

and the density matrices of the other computational basis states are also diagonal matrices with a 1 in one position.

The density matrix of the Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ is

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix} \tag{18}$$

and the density matrix of the state

$$\begin{cases} |00\rangle & \text{with probability } \frac{1}{2} \\ |11\rangle & \text{with probability } \frac{1}{2} \end{cases} \tag{19}$$

is

$$\tfrac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \tfrac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{20}$$

If we were adopt the terminology used for state (1), we might call state (19) a *fake-Bell state*. Notice that the difference between the density matrices of the Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ and the fake-Bell state (19) is in the two off-diagonal entries of their density matrices.

1.2.1 Effect of unitaries on mixed states

Every probabilistic mixture of states has a density matrix associated with it; however, different probabilistic mixtures can result in the same density matrix (for example the two states in Eq. (4)). Suppose that we apply a unitary operation U on a probabilistic mixture of states. How does this affect the density matrix?

If we begin with a mixed state of the form

$$\begin{cases} |\psi_1\rangle & \text{with prob. } p_1 \\ |\psi_2\rangle & \text{with prob. } p_2 \\ \vdots & \vdots & \vdots \\ |\psi_m\rangle & \text{with prob. } p_m \end{cases} \tag{21}$$

then applying U changes the state to

$$\begin{cases} U|\psi_1\rangle & \text{with prob. } p_1 \\ U|\psi_2\rangle & \text{with prob. } p_2 \\ \vdots & \vdots & \vdots \\ U|\psi_m\rangle & \text{with prob. } p_m. \end{cases} \tag{22}$$

This is because, whatever $|\psi_k\rangle$ is randomly selected, it gets converted to $U|\psi_k\rangle$.

Let ρ denote the density matrix of the original state. That is,

$$\rho = \sum_{k=1}^{m} p_k \left|\psi_k\right\rangle \left\langle\psi_k\right| . \tag{23}$$

Then the density matrix after U is applied is

$$\sum_{k=1}^{m} p_k \big(U \left|\psi_k\right\rangle\big) \big(U \left|\psi_k\right\rangle\big)^* = \sum_{k=1}^{m} p_k U \left|\psi_k\right\rangle \left\langle\psi_k\right| U^* \tag{24}$$

$$= U \left(\sum_{k=1}^{m} p_k \left|\psi_k\right\rangle \left\langle\psi_k\right| \right) U^* \tag{25}$$

$$= U \rho U^* . \tag{26}$$

What is remarkable is that the density matrix of the modified state depends *only* on the density matrix of the original state. It does not depend on what specific probabilistic mixture is used to create the original state.

1.2.2 Effect of measurement on mixed states

Now, let's consider the effect of a measurement of a d-dimensional mixed state. As usual, the computational basis is denoted as $\left|0\right\rangle, \left|1\right\rangle, \ldots, \left|d-1\right\rangle$. Let the mixture be

$$\begin{cases} \left|\psi_1\right\rangle & \text{with prob. } p_1 \\ \left|\psi_2\right\rangle & \text{with prob. } p_2 \\ \quad\vdots & \quad\vdots \qquad\quad \vdots \\ \left|\psi_m\right\rangle & \text{with prob. } p_m . \end{cases} \tag{27}$$

There are two different ways that randomness arises in such a measurement: the randomness that was used to select one of the pure states; and, the randomness that arises in the measurement process for the selected state.

If the selected state is $\left|\psi_j\right\rangle$ then the probability of measurement outcome k is

$$\left| \left\langle k|\psi_j\right\rangle \right|^2 = \left\langle k|\psi_j\right\rangle \left\langle\psi_j|k\right\rangle = \left\langle k\right| \Big(\left|\psi_j\right\rangle \left\langle\psi_j\right| \Big) \left|k\right\rangle \tag{28}$$

(where we are using the fact that the expressions are all products of row matrices and column matrices and that matrix multiplication is associative).

If we average this over all possibilities of $|\psi_j\rangle$ then the probability of outcome k is

$$\sum_{j=1}^{m} p_j \langle k| \left(|\psi_j\rangle \langle\psi_j| \right) |k\rangle = \langle k| \left(\sum_{j=1}^{m} p_j |\psi_j\rangle \langle\psi_j| \right) |k\rangle \tag{29}$$

$$= \langle k| \rho |k\rangle , \tag{30}$$

where ρ is the density matrix of the original state. Also, when the measurement outcome is k, the residual state is $|k\rangle$.

Once again, the result of the operation depends *only* on the density matrix of the original state. It does not depend on what specific probabilistic mixture is used to create the original state.

1.2.3 Information processing solely in terms of density matrices

In sections 1.2.1 and 1.2.2, we saw that, for a mixed state with density matrix ρ:

- Applying a unitary operation U to the state changes it to one with density operator $U\rho U^*$.
- Applying a measurement in the computational basis to the state produces classical and quantum outcomes

$$\begin{cases} \big(0,\, |0\rangle\big) & \text{with prob. } \langle 0| \rho |0\rangle \\ \big(1,\, |1\rangle\big) & \text{with prob. } \langle 1| \rho |1\rangle \\ \quad\vdots & \quad\vdots \qquad\qquad \vdots \\ \big(d{-}1,\, |d{-}1\rangle\big) & \text{with prob. } \langle d{-}1| \rho |d{-}1\rangle. \end{cases} \tag{31}$$

In both cases, the result of the operation depends only on the density matrix of the state (not on the specific probabilistic mixture that is used to generate the state).

From this, we can deduce[1] the following theorem.

Theorem 1.1. *Whenever two mixed states have the same density matrix, the states are equivalent.*

Theorem 1.1 implies that the fake-plus state (1) and fake-fake-plus state (2) are indistinguishable.

[1] Actually, we have not yet shown that the outcomes of exotic measurements depend only on the density matrix. Although this is indeed true, it is more convenient to address this later. (Exotic measurements are those where the state being measured is isometrically embedded into a larger space, followed by a unitary and measurement in the larger space.)

1.3 Some properties of matrices

Prior to our use of the density matrix framework, our matrices have mostly been unitary, representing unitary operations on quantum states. Now, we also have density matrices that describe states (and which are not unitary). Henceforth, more types of matrices will arise as we develop our models of quantum information theory. In this section, we review some useful definitions and properties of matrices that will be used.

Definition 1.3 (normal matrix)**.** *A matrix $M \in \mathbb{C}^{d\times d}$ is* normal *if $M^*M = MM^*$.*

An important property of normal matrices is that they are diagonalizable in some orthonormal basis.

Theorem 1.2 (spectral theorem)**.** *A matrix $M \in \mathbb{C}^{d\times d}$ is* normal *if and only if there exists a unitary $U \in \mathbb{C}^{d\times d}$ such that*

$$M = U^* \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix} U. \tag{32}$$

Although Definition 1.3 is the common textbook definition of *normal*, the statement of Theorem 1.2 can be taken as an alternative definition.

To help understand normal matrices, it's useful to see examples of *ab*normal matrices. Consider these two:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad\qquad \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}. \tag{33}$$

The first matrix is not normal because it is not even diagonalizable. The second matrix is diagonalizable but not unitarily (it has eigenvectors $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, which are not orthogonal).

For any normal matrix, by Theorem 1.2, we can imagine it to be diagonal in the coordinate system of some orthonormal basis. Note that a square matrix M is unitary (defined as $M^*M = I$) if and only if all its eigenvalues have absolute value 1 (i.e., they are points on the unit circle in $\mathbb{C}$. And a matrix M is Hermitian (defined as $M = M^*$) if and only if all its eigenvalues are in $\mathbb{R}$.

Definition 1.4 (positive). *A matrix $M \in \mathbb{C}^{d\times d}$ is* positive[2] *if and only if, M is normal and, for all states $|\psi\rangle \in \mathbb{C}^d$, it holds that $\langle\psi| M |\psi\rangle \geq 0$.*

A normal matrix is positive if and only if all of its eigenvalues are in $\mathbb{R}$ and greater than or equal to 0.

Definition 1.5 (trace). *The* trace *of a matrix $M \in \mathbb{C}^{d\times d}$ (denoted as $\mathrm{Tr}(M)$) is defined as the sum of its diagonal entries*

$$\mathrm{Tr}(M) = \sum_{k=1}^{d} M_{k,k}. \tag{34}$$

As simple as the definition of the trace is, it has some interesting properties. An obvious property is that it is linear. That is, for all $A, B \in \mathbb{C}^{d\times d}$ and all $\alpha, \beta \in \mathbb{C}$,

$$\mathrm{Tr}(\alpha A + \beta B) = \alpha \,\mathrm{Tr}(A) + \beta \,\mathrm{Tr}(B). \tag{35}$$

Also, for all $A \in \mathbb{C}^{d_1\times d_2}$ and $B \in \mathbb{C}^{d_2\times d_1}$,

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA). \tag{36}$$

Equation (36) implies that the trace is coordinate system independent, in the sense that, for all $S, A \in \mathbb{C}^{d\times d}$ where S is invertible,

$$\mathrm{Tr}(S^{-1}AS) = \mathrm{Tr}(A). \tag{37}$$

Also, notice that, in Eq. (36), A and B need not be square matrices. For example,

$$\mathrm{Tr}(|\psi\rangle\langle\phi|) = \mathrm{Tr}(\langle\phi|\psi\rangle) = \langle\phi|\psi\rangle\,. \tag{38}$$

A word of caution: here are some properties that, in general, the trace does *not* have:

⚠ In general, the trace is not multiplicative (in the sense that the determinant is). In general, $\mathrm{Tr}(AB) = \mathrm{Tr}(A)\,\mathrm{Tr}(B)$ does *not* hold.

⚠ Moreover, Eq. (36) does not mean that you can arbitrarily reorder any product in the argument of the trace. For example, $\mathrm{Tr}(ABC) = \mathrm{Tr}(BAC)$ does *not* hold in general. But, for the trace of a product, the product can always be *cyclically* permuted as $\mathrm{Tr}(A_1A_2\ldots A_{m-1}A_m) = \mathrm{Tr}(A_mA_1A_2\ldots A_{m-1})$.

[2] In some communities, the terminology *positive semidefinite* is used instead of *positive*.

1.3.1 Characterizing density matrices

Not all matrices arise as the density matrix of some probabilistic mixture of states. The following theorem precisely characterizes which matrices are density matrices.

Theorem 1.3 (characterization of valid density matrix)**.** *A matrix $\rho \in \mathbb{C}^{d\times d}$ is the density matrix of some probabilistic mixture of pure states if and only if ρ is positive and $\mathrm{Tr}(\rho) = 1$.*

Exercise 1.3. *Prove Theorem 1.3.*

Theorem 1.4 (characterization of pure states)**.** *If $\rho \in \mathbb{C}^{d\times d}$ is a density matrix then ρ is a pure state if and only if $\mathrm{Tr}(\rho^2) = 1$.*

Exercise 1.4. *Prove Theorem 1.4.*

1.4 Bloch sphere for qubits

The set of density matrices for qubits has a nice representation as points in the *Bloch sphere*. In this section, I explain this correspondence. Consider the Pauli matrices

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{39}$$

(where in this context I is an honorary Pauli matrix). Every 2×2 matrix can be expressed as a linear combination of I, X, Y, Z. Since $\mathrm{Tr}(I) = 2$ and $\mathrm{Tr}(X) = \mathrm{Tr}(Y) = \mathrm{Tr}(Z) = 0$, we can express any density matrix ρ as

$$\rho = \frac{I + c_x X + c_y Y + c_z Z}{2}. \tag{40}$$

Let's develop a geometric picture for the set of all possible triples (c_x, c_y, c_z) that correspond to valid 2×2 density matrices. Let's start with pure states. Any pure state of a qubit can be written as

$$|\psi\rangle = \cos(\tfrac{\theta}{2})\,|0\rangle + e^{i\phi}\sin(\tfrac{\theta}{2})\,|1\rangle\,, \tag{41}$$

for some $\theta, \phi \in [0, 2\pi]$.

The density matrix of this state is

$$|\psi\rangle\langle\psi| = \begin{bmatrix} \cos^2(\frac{\theta}{2}) & e^{-i\phi}\cos(\frac{\theta}{2})\sin(\frac{\theta}{2}) \\ e^{i\phi}\cos(\frac{\theta}{2})\sin(\frac{\theta}{2}) & \sin^2(\frac{\theta}{2}) \end{bmatrix} \tag{42}$$

$$= \frac{1}{2}\begin{bmatrix} 1+\cos(\theta) & e^{-i\phi}\sin(\theta) \\ e^{i\phi}\sin(\theta) & 1-\cos(\theta) \end{bmatrix} \tag{43}$$

$$= \frac{1}{2}\begin{bmatrix} 1+\cos(\theta) & \big(\cos(\phi)-i\sin(\phi)\big)\sin(\theta) \\ \big(\cos(\phi)+i\sin(\phi)\big)\sin(\theta) & 1-\cos(\theta) \end{bmatrix} \tag{44}$$

$$= \frac{I+\cos(\phi)\sin(\theta)X+\sin(\phi)\sin(\theta)Y+\cos(\theta)Z}{2}. \tag{45}$$

Therefore, the coefficients in Eq. (40) are

$$(c_x, c_y, c_z) = (\cos(\phi)\sin(\theta),\ \sin(\phi)\sin(\theta),\ \cos(\theta)). \tag{46}$$

These triples are the *polar coordinates* of points on the surface of a sphere, as shown in figure 2.

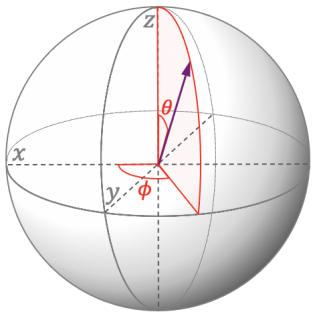

Figure 2: The coordinates (c_x, c_y, c_z) of a pure state are a point on the surface of a sphere.

Think of this sphere as the Earth with the North Pole at the top. Points on the surface can be expressed in terms of their latitude and longitude. The *latitude* θ is the angular distance away from the North Pole. The *longitude* ϕ is an angle representing the East-West distance from some arbitrary[3] starting point. So all the pure states correspond to points[4] on the surface of this sphere, called the *Bloch sphere*.

[3]In geography, the convention is to set 0° at the *Prime Meridian* in Greenwich, UK.

[4]To avoid redundancy, it is natural to restrict the range of θ to $[0, \pi]$.

Where are states $|0\rangle$ and $|1\rangle$ situated on the Bloch sphere? State $|0\rangle$ lies at the North Pole and $|1\rangle$ is at the South Pole, as illustrated in figure 3.

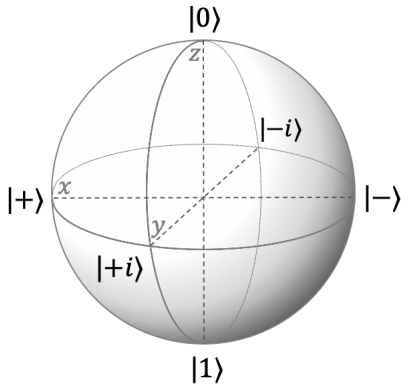

Figure 3: Position of states $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$, $|+i\rangle$, and $|-i\rangle$ on the Bloch sphere.

Notice that $|0\rangle$ and $|1\rangle$ are orthogonal as vectors; however, their positions on the Bloch sphere are 180° apart. Any two orthogonal vectors map to antipodal points on the sphere (180° apart).

Where are $|+\rangle$ and $|-\rangle$? They lie on the equator. State $|+\rangle$ has longitude $\phi = 0$ (it lies at the intersection of the equator and the Prime Meridian) and $|-\rangle$ is at the antipodal point.

Notice the angle-doubling again. The angle between $|0\rangle$ and $|+\rangle$ is 45°, but the angle between their points on the Bloch sphere is 90°. In general, for any two state vectors, if we map them to the sphere, the angular distance between them doubles.

There are two other points on the sphere that are in natural positions relative to the points we have considered so far: those that are 90° from $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$. They correspond to the states

$$|+i\rangle = \tfrac{1}{\sqrt{2}}\,|0\rangle + \tfrac{i}{\sqrt{2}}\,|1\rangle \tag{47}$$

$$|-i\rangle = \tfrac{1}{\sqrt{2}}\,|0\rangle - \tfrac{i}{\sqrt{2}}\,|1\rangle\,. \tag{48}$$

There is some nice symmetry among the six states $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$, $|+i\rangle$, and $|-i\rangle$.

The surface of the Bloch sphere consists of all the pure states, and it turns out the *mixed* states are all the points inside the sphere. For any probabilistic mixture of pure states, its position in the Bloch sphere is the weighted average of the positions of the pure states. For example, an equally weighted mixture of $|0\rangle$ and $|1\rangle$ (the so-called fake-plus state from Eq. (1)) is the point right at the centre of the sphere. And an

equally weighted mixture of $|0\rangle$ and $|+\rangle$ is the midpoint of the line connecting their positions on the sphere.

For qubit systems, it's often very useful to think of states—and the operations acting on them—on the Bloch sphere.

A word of caution:

⚠ Do not conflate state vectors with points on the Bloch sphere!

⚠ The Bloch sphere is for one-qubit states. For higher dimensional systems, there's an analogous geometric shape—but it's not a hypersphere. It's shape is somewhat complicated and it doesn't satisfy all the properties that one might expect to hold based on the case of qubits. For example, not all points on the surface of this shape are pure states (some are mixed states). For qutrits, the shape is 7-dimensional.

2 State transitions in the Kraus form

So far, we have seen various kinds of operations that can be performed on quantum systems. We have seen unitary operations and measurements. There are also operations that are described in words (or annotated quantum circuits), such as "add an ancilla qubit", "measure the second qubit", and "take the first qubit as the output." The Kraus form is a unified framework for describing all of these, as well as some other kinds of quantum operations. We begin with this definition.

Definition 2.1 (Kraus operators)**.** *A sequence of $d_1 \times d_2$ matrices $A_0, A_2, \ldots, A_{m-1}$ is a sequence of* Kraus operators *if*

$$\sum_{k=0}^{m-1} A_k^* A_k = I, \tag{49}$$

where I denotes the $d_2 \times d_2$ identity matrix.

Note that the matrices in the above definition need not be square: if A_k is $d_1 \times d_2$ then $A_k^* A_k$ is $d_2 \times d_2$.

At first glance, Definition 2.1 may look mysterious. In this section, we'll see that several quantum state transformations, including measurements, unitary operations (and other natural transformations) are expressible in terms of Kraus operators.

2.1 Measurements via Kraus operators

For any Kraus operators $A_0, A_2, \ldots, A_{m-1} \in \mathbb{C}^{d_1 \times d_2}$, define the the following measurement operation, whose classical output is $k \in \{0, 1, \ldots, m-1\}$.

Input to the measurement: is a d_2-dimensional quantum system, whose state can be described by a $d_2 \times d_2$ density matrix ρ.

Output of the measurement: can be described as as the probabilistic mixture

$$\begin{cases} \left(\quad 0\,, \quad \dfrac{A_0 \rho A_0^*}{\mathrm{Tr}(A_0 \rho A_0^*)} \quad \right) & \text{with prob. } \mathrm{Tr}(A_0 \rho A_0^*) \\ \left(\quad 1\,, \quad \dfrac{A_1 \rho A_1^*}{\mathrm{Tr}(A_1 \rho A_1^*)} \quad \right) & \text{with prob. } \mathrm{Tr}(A_1 \rho A_1^*) \\ \quad \vdots \qquad\quad \vdots & \qquad\quad \vdots \qquad\quad \vdots \\ \left(m{-}1\,, \quad \dfrac{A_{m-1} \rho A_{m-1}^*}{\mathrm{Tr}(A_{m-1} \rho A_{m-1}^*)} \right) & \text{with prob. } \mathrm{Tr}(A_{m-1} \rho A_{m-1}^*), \end{cases} \tag{50}$$

where the first component is the classical outcome $k \in \{0, 1, \ldots, m-1\}$ and the second component is the residual state, which is a $d_1 \times d_1$ density matrix (if $d_1 \neq d_2$ then the dimensions of the input and output systems are different).

The first question is whether the above makes sense. Are the probabilities non-negative real numbers that sum to 1? Are the residual states valid density matrices? To get an idea why this measurement makes sense, consider the case of pure states. If $\rho = |\psi\rangle\langle\psi|$ then, for all k,

$$\mathrm{Tr}(A_k \rho A_k^*) = \mathrm{Tr}(A_k |\psi\rangle\langle\psi| A_k^*) = \mathrm{Tr}(\langle\psi| A_k^* A_k |\psi\rangle) = \langle\psi| A_k^* A_k |\psi\rangle . \tag{51}$$

Clearly, $\langle\psi| A_k^* A_k |\psi\rangle \geq 0$, since this is the inner product of $A_k |\psi\rangle$ with itself. Also,

$$\sum_{k=0}^{m-1} \mathrm{Tr}(A_k \rho A_k^*) = \sum_{k=0}^{m-1} \langle\psi| A_k^* A_k |\psi\rangle = \langle\psi| \left(\sum_{k=0}^{m-1} A_k^* A_k \right) |\psi\rangle = \langle\psi|\psi\rangle = 1. \tag{52}$$

The more general case where ρ is a mixed state can by analyzed by averaging over pure states.

Exercise 2.1 (straightforward)**.** *Show that, for an arbitrary $d_2 \times d_2$ density matrix ρ, it holds that $\mathrm{Tr}(A_k \rho A_k^*) \geq 0$ (for all k) and $\sum_{k=0}^{m-1} \mathrm{Tr}(A_k \rho A_k^*) = 1$. Also show that $(A_k \rho A_k)/\mathrm{Tr}(A_k \rho A_k^*)$ is a valid density matrix (for all k).*

Next, we'll see some measurements expressed in terms of Kraus operators.

2.1.1 Computational basis measurements

Let us begin with the basic measurement with respect to the computational basis. For a d-dimensional system, the computational basis is $|0\rangle, |1\rangle, \ldots, |d-1\rangle$. To express this measurement in the Kraus form, set

$$A_k = |k\rangle\langle k| \tag{53}$$

for each $k \in \{0, 1, \ldots, d-1\}$. It's easy to check that these are valid Kraus operators, in the sense of Definition 2.1.

Exercise 2.2. *Show that $A_0, A_1, \ldots, A_{d-1}$, as defined as in Eq.* (53)*, are valid Kraus operators.*

For A_k, as defined as in Eq. (53), it holds that

$$\mathrm{Tr}(A_k \rho A_k^*) = \mathrm{Tr}(|k\rangle\langle k| \, \rho \, |k\rangle\langle k|) = \mathrm{Tr}(\langle k|k\rangle \, \langle k| \, \rho \, |k\rangle) = \langle k| \, \rho \, |k\rangle \tag{54}$$

and

$$\frac{A_k \rho A_k^*}{\mathrm{Tr}(A_k \rho A_k^*)} = \frac{|k\rangle \, \langle k| \, \rho \, |k\rangle \, \langle k|}{\langle k| \, \rho \, |k\rangle} = |k\rangle \, \langle k| \tag{55}$$

(where we have used the fact that $\langle k| \, \rho \, |k\rangle$ is a scalar). This is consistent with our definition of the measurement in the computational basis in section 1.2.2.

2.1.2 Projective measurements

A *projective measurement* is a measurement with respect to orthogonal subspaces. In the case of pure states, the effect of such a measurement is for the state to project to one of the subspaces, where the probabilities are the projection lengths squared.

These measurements were discussed in the notes [*Part 1: A Primer for Beginners*, Section 8.1]. What follows is a description of these measurements in the Kraus form using projectors.

Definition 2.2 (projector)**.** *A matrix* Π *is a* projector *if* Π *is normal and* $\Pi^2 = \Pi$*.*

Note that the eigenvalues of a projector are 0 or 1. Geometrically, if a projector is applied to a vector then the result is its component in the 1-eigenspace.

Definition 2.3 (orthogonal and complete projectors)**.** *Let* $\Pi_0, \Pi_1, \ldots, \Pi_{m-1} \in \mathbb{C}^d$ *be a sequence of projectors. The projectors are* orthogonal *if* $\Pi_j \Pi_k = 0$ *(the zero matrix) for all* $j \neq k$*. The projectors are* complete *if* $\Pi_0 + \Pi_1 + \cdots + \Pi_{m-1} = I$*.*

Here's a simple example of orthogonal and complete projectors in $\mathbb{C}^3$.

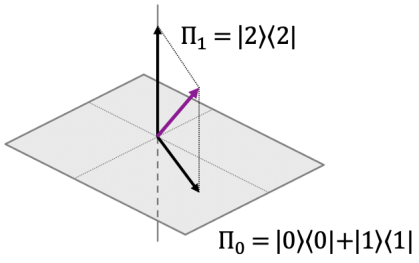

Figure 4: $\Pi_0 = |0\rangle\langle 0| + |1\rangle\langle 1|$ and $\Pi_1 = |2\rangle\langle 2|$ are orthogonal and complete projectors in $\mathbb{C}^3$.

It's easy to see that any sequence of orthogonal and complete projectors are Kraus operators.

Exercise 2.3. *Show that if* $\Pi_0, \Pi_1, \ldots, \Pi_{m-1}$ *are orthogonal and complete projectors then they are Kraus operators, in that they satisfy Eq.* (49).

Therefore, a sequence of orthogonal and complete projectors defines a measurement in the Kraus form. The probability of outcome k is $\mathrm{Tr}(\Pi_k \rho \Pi_k) = \mathrm{Tr}(\rho \Pi_k)$.

Let's look at what these measurements do for pure states. If $\rho = |\psi\rangle\langle\psi|$ then

$$\mathrm{Tr}(\rho \Pi_k) = \mathrm{Tr}\big(|\psi\rangle \langle\psi| \Pi_k\big) \tag{56}$$

$$= \langle\psi| \Pi_k |\psi\rangle \tag{57}$$

$$= \big|\Pi_k |\psi\rangle\big|^2, \tag{58}$$

which is the projection length squared of $|\psi\rangle$ to the 1-eigenspace of Π_k. And the corresponding residual state can be shown to be $\Pi_k |\psi\rangle$ normalized.

2.1.3 Measuring the first of two registers

Suppose that we have a system consisting of two registers, with respective dimensions d_1 and d_2.

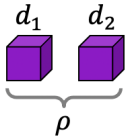

Figure 5: A system consisting of a d_1-dimensional register and a d_2-dimensional register.

All the pure states on this combined system are $d_1 d_2$-dimensional vectors and the density matrices are in $d_1 d_2 \times d_1 d_2$ matrices. A measurement of the *first* register in the computational basis can be defined along the lines of the notes [*Part 1: A Primer for Beginners*, Section 8.2]. In the language of Kraus operators, we can define this measurement as follows. Let $|0\rangle, |1\rangle, \ldots, |d_1 - 1\rangle$ be the computational basis for the first register and set the Kraus operators to be

$$A_k = \big(|k\rangle\langle k|\big) \otimes I, \tag{59}$$

for $k \in \{0, 1, \ldots, d_1 - 1\}$ (where I denotes the $d_2 \times d_2$ identity matrix). Following

Eq. (50), for each $k \in \{0, 1, \ldots, d_1\}$, the output can be shown to be[5]

$$\left(k\,, \quad |k\rangle\langle k| \otimes \frac{\big(\langle k| \otimes I\big)\rho\big(|k\rangle \otimes I\big)}{\mathrm{Tr}\Big(\big(\langle k| \otimes I\big)\rho\big(|k\rangle \otimes I\big)\Big)} \right) \tag{60}$$

with probability $\mathrm{Tr}\Big(\big(\langle k| \otimes I\big)\rho\big(|k\rangle \otimes I\big)\Big)$.

2.1.4 Trine state measurement

So far, all the Kraus measurements that we've seen are projective measurements. However, Kraus measurements need not be projective, as the next example shows.

Let's begin by considering the problem of distinguishing between the trine states

$$|\phi_0\rangle = |0\rangle \tag{61}$$

$$|\phi_1\rangle = -\tfrac{1}{2}\,|0\rangle + \tfrac{\sqrt{3}}{2}\,|1\rangle \tag{62}$$

$$|\phi_2\rangle = -\tfrac{1}{2}\,|0\rangle - \tfrac{\sqrt{3}}{2}\,|1\rangle\,, \tag{63}$$

which are three vectors in $\mathbb{C}^2$, with angle 120° between each pair.

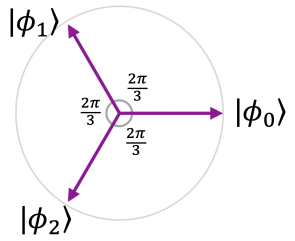

Figure 6: The trine states in $\mathbb{C}^2$.

Suppose that we're given one of these states (we're not told which one) and our goal is to perform a measurement that guesses the state correctly with as high a probability as possible. It turns out the optimal performance (for a worst-case input state) cannot attained by any projective measurement in $\mathbb{C}^2$.

[5]The key step is that $\big(|\psi\rangle\langle\psi| \otimes I\big)\rho\big(|\psi\rangle\langle\psi| \otimes I\big) = \big(|\psi\rangle \otimes I\big)\Big(\big(\langle\psi| \otimes I\big)\rho\big(|\psi\rangle \otimes I\big)\Big)\big(\langle\psi| \otimes I\big)$
$= |\psi\rangle\langle\psi| \otimes \Big(\big(\langle\psi| \otimes I\big)\rho\big(|\psi\rangle \otimes I\big)\Big)$.

Define these three Kraus operators

$$A_0 = \sqrt{\tfrac{2}{3}}\,|\phi_0\rangle\langle\phi_0| \;=\; \begin{bmatrix} \sqrt{2/3} & 0 \\ 0 & 0 \end{bmatrix} \tag{64}$$

$$A_1 = \sqrt{\tfrac{2}{3}}\,|\phi_1\rangle\langle\phi_1| \;=\; \frac{1}{4}\begin{bmatrix} \sqrt{2/3} & -\sqrt{2} \\ -\sqrt{2} & \sqrt{6} \end{bmatrix} \tag{65}$$

$$A_2 = \sqrt{\tfrac{2}{3}}\,|\phi_2\rangle\langle\phi_2| \;=\; \frac{1}{4}\begin{bmatrix} \sqrt{2/3} & \sqrt{2} \\ \sqrt{2} & \sqrt{6} \end{bmatrix}. \tag{66}$$

Notice that these are *not* projectors (because of the factor $\sqrt{2/3}$), and they are not orthogonal. Nevertheless, since $A_0^*A_0 + A_1^*A_1 + A_2^*A_2 = I$, these are valid Kraus operators. Using this measurement for the trine state distinguishing problem results in success probability $\frac{2}{3}$. This is not achievable using projective measurements (however, it is achievable using one of the so-called exotic measurements, where the system is embedded into a larger dimensional space before a projective measurement).

2.2 Quantum channels via Kraus operators

For a sequence of Kraus operators, $A_0, A_2, \ldots, A_{m-1} \in \mathbb{C}^{d_1 \times d_2}$ define the following state transformation, called a *quantum channel*, which maps quantum states to quantum states with no classsical side information.

Input to the channel: is a d_2-dimensional quantum system, whose state can be described by a $d_2 \times d_2$ density matrix ρ.

Output of the channel: is a d_1-dimensional quantum system, whose state is

$$A_0\rho A_0^* + A_1\rho A_1^* + \cdots + A_{m-1}\rho A_{m-1}^*. \tag{67}$$

2.2.1 Unitary operations

Any $d \times d$ unitary operation U corresponds to a quantum channel with one single Kraus operator U. The channel maps each $d \times d$ density matrix ρ maps ρ to $U\rho U^*$. We can think of quantum channels as generalizations of unitary operations.

2.2.2 Decoherence of a qubit

I will first explain what this channel does, and then show you two different ways of "implementing" the channel in terms of Kraus operators. The decoherence channel changes the state of its input qubit from

$$\rho = \begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{bmatrix} \qquad \text{to} \qquad \begin{bmatrix} \rho_{00} & 0 \\ 0 & \rho_{11} \end{bmatrix}. \tag{68}$$

The diagonal density matrix can be viewed as a probabilistic mixture of $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. On the Bloch sphere, the diagonal density matrices are on the axis connecting $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. The effect of the channel is to move the state "horizontally" (i.e., parallel to the equatorial plane) to the vertical axis connecting $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$.

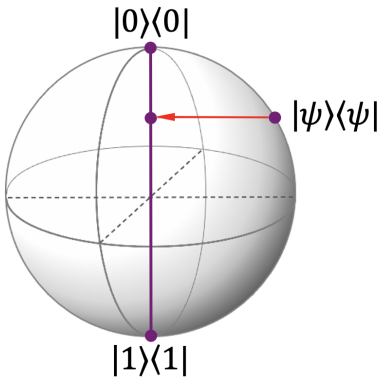

Figure 7: Effect of the decoherence channel on pure state $|\psi\rangle\langle\psi|$.

I will show you two operationally different ways of implementing this channel.

Measuring without looking at the outcome

Our first way of implementing the decoherence channel can be intuitively thought of as measuring the qubit in the computational basis—but without looking at the classical outcome. We might imagine that Bob performs the measurement, but covers his eyes so that he doesn't see the classical outcome. But let's think about it this way: Bob sends the qubit to Alice, who performs the measurement (and sees the outcome) and then Alice sends the qubit back to Bob, but she does not send him the classical output of the measurement.

The quantum part of the outcome of Alice's measurement is

$$\begin{cases} |0\rangle & \text{with prob. } \langle 0|\,\rho\,|0\rangle \\ |1\rangle & \text{with prob. } \langle 1|\,\rho\,|1\rangle. \end{cases} \tag{69}$$

Since Alice obtains the classical outcome, from *her* perspective, the quantum outcome is always either $|0\rangle\langle 0|$ or $|1\rangle\langle 1|$. But Bob does not receive the classical outcome so, from *his* perspective, the quantum outcome is the density matrix

$$\langle 0|\rho|0\rangle\,|0\rangle\langle 0| + \langle 1|\rho|1\rangle\,|1\rangle\langle 1|\,. \tag{70}$$

This can be expressed in the Kraus form by setting the Kraus operators of a quantum channel to $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. Then a density matrix $\rho = \left[\begin{smallmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{smallmatrix}\right]$ maps to

$$\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix} = \begin{bmatrix}\rho_{00} & 0\\ 0 & \rho_{11}\end{bmatrix}. \tag{71}$$

Probabilistic mixture of I and Z

Another way of implementing the decoherence channel is intuitively based on applying a randomly selected unitary to the state. Bob sends the qubit to Alice, who does the following. She flips a fair coin, and then either applies I or Z to the qubit, depending on the outcome of the coin flip. Then she sends the qubit back to Bob, but she does not reveal the coin flip.

Since Alice knows outcome of the coin flip, from *her* perspective, the state is either ρ or $Z\rho Z$. But Bob does not know the coin flip so, from *his* perspective, the state is

$$\begin{cases} \rho & \text{with prob. } \frac{1}{2} \\ Z\rho Z & \text{with prob. } \frac{1}{2}. \end{cases} \tag{72}$$

and the density matrix of this mixture is

$$\tfrac{1}{2}\rho + \tfrac{1}{2}Z\rho Z. \tag{73}$$

This can be expressed in the Kraus form by setting the Kraus operators of a quantum channel to $\frac{1}{\sqrt{2}}I$ and $\frac{1}{\sqrt{2}}Z$. Then a density matrix $\rho = \left[\begin{smallmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{smallmatrix}\right]$ maps to

$$\left(\tfrac{1}{\sqrt{2}}I\right)\rho\left(\tfrac{1}{\sqrt{2}}I\right)^* + \left(\tfrac{1}{\sqrt{2}}Z\right)\rho\left(\tfrac{1}{\sqrt{2}}Z\right)^* = \tfrac{1}{2}\rho + \tfrac{1}{2}Z\rho Z \tag{74}$$

$$= \tfrac{1}{2}\begin{bmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{bmatrix} + \tfrac{1}{2}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{75}$$

$$= \begin{bmatrix}\rho_{00} & 0\\ 0 & \rho_{11}\end{bmatrix}. \tag{76}$$

Comparison of the two implementations of the decoherence channel

From Bob's perspective, who doesn't receive any classical measurement outcomes or coin flip outcomes, the two implementations of the decoherence channel are identical. However, they are not literally the same. In the first implementation, the state is actually measured and it cannot be recovered at a later time, even with the classical information that Alice has. In the second implementation, Alice performs no measurement. If, at some later time, she reveals the coin flip to Bob then he can recover the initial state (by applying either I or Z to the state).

So there's an advantage to the second implementation. But there is also a disadvantage: if Bob asks Alice later on "what was the measurement outcome?", she cannot answer that question. There is no classical bit $b \in \{0, 1\}$ that Alice can produce and send to Bob such that, if Bob then measures his decohered state in the computational basis, the outcome is guaranteed to be b.

Exercise 2.4 (conceptual)**.** *Suppose that Bob believes that he has figured out a new way of measuring a qubit that is reversible. His idea is to first implement the random unitary method to create the decohered state, which can serve as the quantum outcome (remembering what the coin flip is, so that the can undo the unitary later on). Now all that's lacking is that classical outcome. Bob's idea is to measure the decohered qubit in the computational basis to obtain a bit that can serve as the the classical outcome of the measurement. Will doing all this result in a faithful simulation of the measurement operation? And, after all these operations have been performed, is there a way for Bob to recover the original state?*

2.2.3 General measurement without seeing the outcome

For any sequence of Kraus operators $A_0, A_2, \ldots, A_{m-1}$, we have defined an associated measurement in section 2.1 and an associated channel in section 2.2. The associated channel can *always* be interpreted as performing the associated measurement without looking at the classical outcome.

2.2.4 General mixed unitary channels

For any sequence of unitary operations $U_0, U_1, \ldots, U_{m-1}$ with associated probabilities $p_0, p_1, \ldots, p_{m-1}$, consider the operation where $k \in \{0, 1 \ldots, m-1\}$ is randomly chosen according to probabilities $p_0, p_1, \ldots, p_{m-1}$ and then U_k is applied. If the selected k is

not revealed then this procedure maps any input state ρ to the output state

$$p_0\, U_0\, \rho\, U_0^* \;+\; p_1\, U_1\, \rho\, U_1^* \;+\; \cdots \;+\; p_{m-1}\, U_{m-1}\, \rho\, U_{m-1}^*. \tag{77}$$

This is easy to express in the Kraus form, by setting the Kraus operators to

$$A_k = \sqrt{p_k}\, U_k \tag{78}$$

for $k \in \{0, 1, \ldots, m-1\}$.

Exercise 2.5 (may be challenging)**.** *Can* every *quantum channel, as defined in Eq.* (67)*, be expressed as a probability distribution on a set of unitary operations? Either prove this to be the case or give a counterexample.*

2.2.5 Adding an ancilla

A natural quantum operation is to append an ancilla in state $|\psi\rangle$ after a register. Let $|\psi\rangle$ be d_2-dimensional. The input to this operation is a d_1-dimensional system, whose state is described by a $d_1 \times d_1$ density matrix ρ. The output is a $d_1 d_2$-dimensional system, whose state is $\rho \otimes \big(|\psi\rangle\langle\psi|\big)$.

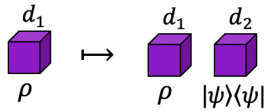

Figure 8: The operation of appending an ancilla in state $|\psi\rangle$ after a register.

This can be expressed as a channel in the Kraus form with one Kraus operator

$$A_0 = I \otimes |\psi\rangle\,. \tag{79}$$

Applying the channel to state $\rho \in \mathbb{C}^{d_1 \times d_1}$ produces the state

$$\begin{aligned}
A_0 \rho A_0^* &= \big(I \otimes |\psi\rangle\big)\rho\big(I \otimes \langle\psi|\big) && (80)\\
&= \big(I \otimes |\psi\rangle\big)\big(\rho \otimes [\,1\,]\big)\big(I \otimes \langle\psi|\big) \qquad \text{where } [\,1\,] \text{ is a } 1\times 1 \text{ matrix} && (81)\\
&= \big(I\rho I\big) \otimes \big(|\psi\rangle\,[\,1\,]\,\langle\psi|\big) && (82)\\
&= \rho \otimes \big(|\psi\rangle\langle\psi|\big). && (83)
\end{aligned}$$

(The insertion of the 1×1 matrix $[\,1\,]$ above is an optional step to make the product easier to parse in a form where the identity $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ can be applied [*Part 1: A Primer for Beginners*, Section 6.6, Lemma 6.1].)

An explicit example is the addition of an ancilla in state $|0\rangle$ after a qubit. This is accomplished by the Kraus operator

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{84}$$

Note that we have addressed the case of adding an ancilla in a pure state. What if we want to add an ancilla in a mixed state? I leave this as an exercise.

Exercise 2.6. *Suppose that we want to add an ancilla in the mixed state $\sigma \in \mathbb{C}^{d_2 \times d_2}$ to the end of a d_1-dimensional system. Show how to express this as a quantum channel in the Kraus form.*

2.2.6 Partial trace

Suppose that Bob is in possession of a system consisting of two registers. Let his first register be d_1-dimensional and his second register be d_2-dimensional. Suppose that Bob wants to discard his first register. What does this mean? Intuitively, we can imagine that Bob sends his first register to a faraway place where he will never access it again. Another way of thinking about this is that the first register doesn't move, but Bob decides to henceforth completely ignore it. He ghosts his first register. What's the state of Bob's remaining register?

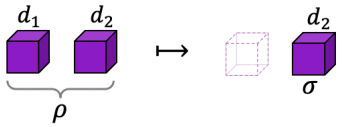

Figure 9: Tracing out the first of two registers.

This question arose the context of pure states in the notes [*Part 1: A Primer for Beginners*, Section 6.3]. If one restricts to pure states (representable as unit vectors) then subsystems might not have states of their own. For example, there is no pure state that captures the state of the second qubit of the Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$.

However, we are now working in a broader context that includes mixed states (representable as density matrices). In this broader context, subsystems always have well-defined states of their own. The states of subsystems are captured by a quantum channel called the *partial trace*.

One way of deriving the definition of the partial trace is to use the fact that what happens to the discarded system is inconsequential to the remaining system. In particular, there is no harm in measuring the discarded system in some orthonormal basis, say $|0\rangle, |1\rangle, \ldots, |d_1 - 1\rangle$ (and not looking at the classical or quantum outcomes of the measurement). Following Eq. (59), the quantum channel corresponding to this measurement (without looking at the classical outcome) has Kraus operators

$$|k\rangle\langle k| \otimes I, \tag{85}$$

for $k \in \{0, 1, \ldots, d_1 - 1\}$. But the output of this channel includes the residual quantum state of the first register (see Eq. (60)). To eradicate this residual state, we modify the Kraus operators to

$$\langle k| \otimes I. \tag{86}$$

It's easy to check that $\langle 0| \otimes I$, $\langle 1| \otimes I$, $\ldots$, $\langle d_1 - 1| \otimes I$ are valid Kraus operators and the quantum channel that they define is the partial trace.

Definition 2.4 (partial trace)**.** *This definition is in the context of a system with a d_1-dimensional register and a d_2-dimensional register. There are two partial traces. The* partial trace $\mathrm{Tr}_1 : \mathbb{C}^{d_1 d_2 \times d_1 d_2} \to \mathbb{C}^{d_2 \times d_2}$ *is defined as, for all $\rho \in \mathbb{C}^{d_1 d_2 \times d_1 d_2}$,*

$$\mathrm{Tr}_1(\rho) = \sum_{k=0}^{d_1 - 1} \big(\langle k| \otimes I\big)\rho\big(|k\rangle \otimes I\big). \tag{87}$$

And the partial trace $\mathrm{Tr}_2 : \mathbb{C}^{d_1 d_2 \times d_1 d_2} \to \mathbb{C}^{d_1 \times d_1}$ *is defined as, for all $\rho \in \mathbb{C}^{d_1 d_2 \times d_1 d_2}$,*

$$\mathrm{Tr}_2(\rho) = \sum_{k=0}^{d_2 - 1} \big(I \otimes \langle k|\big)\rho\big(I \otimes |k\rangle\big). \tag{88}$$

The subscript of Tr denotes which system is being traced out. In the above definition, the measurement is with respect to the computational basis, but the channel is the same if a different orthonormal basis is used.

Recall that the *trace* of a square matrix is the sum of its diagonal entries. We can also call this the *full trace* and its definition can be written as $\mathrm{Tr}(\rho) = \sum_{k=0}^{d-1} \langle k| \rho |k\rangle$. And the entries of the partial trace of ρ are sums of the matrix entries of ρ.

For the case of two 1-qubit registers,

$$\mathrm{Tr}_1 \begin{bmatrix} \rho_{00,00} & \rho_{00,01} & \rho_{00,10} & \rho_{00,11} \\ \rho_{01,00} & \rho_{01,01} & \rho_{01,10} & \rho_{01,11} \\ \rho_{10,00} & \rho_{10,01} & \rho_{10,10} & \rho_{10,11} \\ \rho_{11,00} & \rho_{11,01} & \rho_{11,10} & \rho_{11,11} \end{bmatrix} = \begin{bmatrix} \rho_{00,00} + \rho_{10,10} & \rho_{00,01} + \rho_{10,11} \\ \rho_{01,00} + \rho_{11,10} & \rho_{01,01} + \rho_{11,11} \end{bmatrix} \tag{89}$$

$$\mathrm{Tr}_2 \begin{bmatrix} \rho_{00,00} & \rho_{00,01} & \rho_{00,10} & \rho_{00,11} \\ \rho_{01,00} & \rho_{01,01} & \rho_{01,10} & \rho_{01,11} \\ \rho_{10,00} & \rho_{10,01} & \rho_{10,10} & \rho_{10,11} \\ \rho_{11,00} & \rho_{11,01} & \rho_{11,10} & \rho_{11,11} \end{bmatrix} = \begin{bmatrix} \rho_{00,00} + \rho_{01,01} & \rho_{00,10} + \rho_{01,11} \\ \rho_{10,00} + \rho_{11,01} & \rho_{10,10} + \rho_{11,11} \end{bmatrix}. \tag{90}$$

It should be noted that, although a measurement was introduced to derive[6] the formulas for the partial trace, the measurement does not have to occur. If one register is discarded then the state of the other register is given by the formula for the partial trace whether or not the discarded register is measured.

Now, let's calculate the state of the second qubit of the Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. Applying the formula in Eq. (89) to the density matrix of the state, we obtain

$$\mathrm{Tr}_1\Big(\big(\tfrac{1}{\sqrt{2}}|00\rangle + \tfrac{1}{\sqrt{2}}|11\rangle\big)\big(\tfrac{1}{\sqrt{2}}\langle 00| + \tfrac{1}{\sqrt{2}}\langle 11|\big)\Big) = \mathrm{Tr}_1 \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix} \tag{91}$$

$$= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{92}$$

There's something remarkable about this. Until now, all our mixed states have been expressed as probabilistic mixtures of pure states. However, a mixed state can arise from a process without any explicit occurrence of randomness or measurement. For the pure state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, the state of each of its individual qubits is $\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$.

[6] An alternative way of deriving the formula for $\mathrm{Tr}_1 : \mathbb{C}^{d_1 d_2 \times d_1 d_2} \to \mathbb{C}^{d_2 \times d_2}$ is to define Tr_1 as the unique linear operator with the property that, for all $\rho \in \mathbb{C}^{d_1 \times d_1}$ and $\sigma \in \mathbb{C}^{d_2 \times d_2}$, $\mathrm{Tr}_1(\rho \otimes \sigma) = \mathrm{Tr}(\rho)\sigma$.

3 State transitions in the Stinespring form

In the previous section, I showed you how to express quantum measurements and quantum channels in terms of Kraus operators. In this section, I'm going to show you another form for expressing state transitions, called the *Stinespring form*.

3.1 Measurements in the Stinespring form

Imagine that the input state is a d-dimensional register. First, we append an m-dimensional ancilla register in some computational basis state, say $|0\rangle$. The combined system is md-dimensional. Next, we apply some $md \times md$ unitary operation U to the combined system. Finally, we measure one register in the computational basis, yielding a classical outcome k and a residual quantum state in the other register.

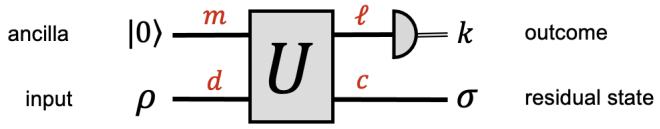

Figure 10: Quantum circuit for a measurement in the Stinespring form.

It's natural for the dimensions of the registers coming out of U to be the same as those of the registers going in ($\ell = m$ and $c = d$). But we allow for the dimensions of the outgoing registers to be different, as long as the total dimension is the same ($md = \ell c$). To get a feeling for this, consider the case where all the dimensions are powers of 2. In that case, we can assume that each register is a bunch of qubits.

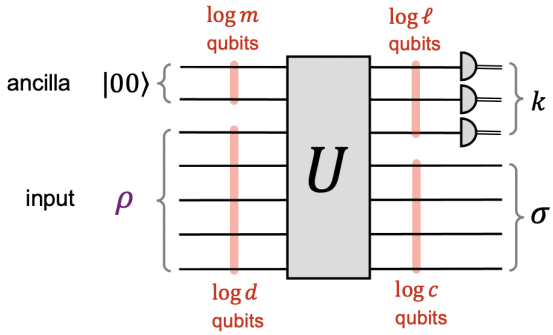

Figure 11: Quantum circuit on qubits for a measurement in the Stinespring form.

In this example, the input state is 5 qubits, whereas the residual outgoing state is 4 qubits. Also, the ancilla is 2 qubits, whereas the number of qubits that are measured

is 3. As long as the total number of qubits going into U and coming out of U is preserved, this makes perfect sense.

Also, note that some of the dimensions can be 1. A 1-dimensional register is essentially the same as no register. A 1×1 density matrix is $\begin{bmatrix} 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \end{bmatrix} \otimes \rho = \rho$. For example, for a measurement in an orthonormal basis specified by U, a very strict translation into the form of figure 11 is obtained by setting $m = c = 1$ and $\ell = d$ (where $U = I$ in the case of the computational basis).

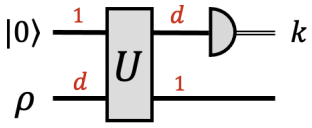

Figure 12: Measurement with respect to an orthonormal basis specified by U.

But figure 12 is pedantic, and we can freely omit the wires of dimension 1 (and omit any I gates). With this relaxation, we can denote a measurement with respect to an orthonormal basis in the Stinespring form as follows.

Figure 13: Measurement with respect to the computational basis and a basis specified by U.

Remember the "exotic measurements" in the notes [*Part 1: A Primer for Beginners*, Section 9]? It should be clear that those measurements are subsumed by these Stinespring measurements.

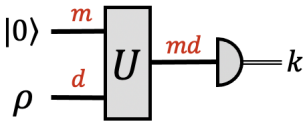

Figure 14: Exotic measurement in the Stinespring form.

3.2 Channels in the Stinespring form

As we noted earlier, one way of thinking about a channel is as a measurement where we don't look at the classical part of the outcome. So we could define Stinespring channels that way. We'll do that, but we'll simplify things by noting that, if we're

not going to see the classical outcome then, instead of performing the measurement, we can trace out that register, like this.

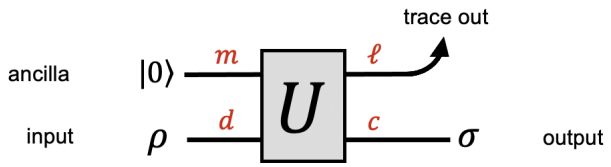

Figure 15: Quantum circuit for a channel in the Stinespring form.

Notice the circuit notation that I'm using here for tracing out a register: the imagery is supposed evoke that the register is tossed away.

Here are some of our most basic channels in the Stinespring form (loosely in the form of figure 15).

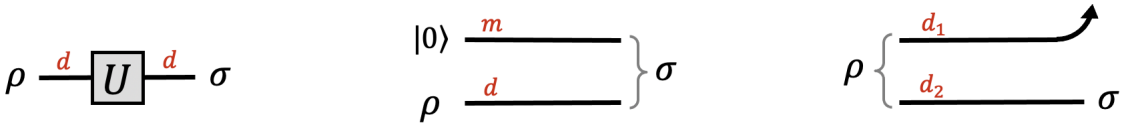

Figure 16: Unitary channel, add ancilla $|0\rangle\langle 0|$ channel, and partial trace Tr_1 channel.

All these channels are rather trivial examples. In the next subsections, we review some more interesting examples.

3.2.1 Decoherence of a qubit

The qubit decoherence channel was defined in the Kraus form in section 2.2.2. The output of this channel corresponds to the residual state when a qubit is measured in the computational basis (but where we don't see the classical part of the outcome). Here's a Stinespring circuit for the decoherence channel.

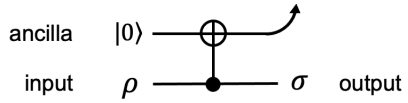

Figure 17: Decoherence of a qubit channel.

How does this work? Consider the case of a pure state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$. The CNOT gate causes the state of the two qubits to become $\alpha_0 |00\rangle + \alpha_1 |11\rangle$, and tracing out

the first qubit yields the mixed state

$$\begin{bmatrix} |\alpha_0|^2 & 0 \\ 0 & |\alpha_1|^2 \end{bmatrix} = |\alpha_0|^2 \, |0\rangle\langle 0| + |\alpha_1|^2 \, |1\rangle\langle 1| \,, \tag{93}$$

which is consistent with the definition of this channel. So at least this works for the special case of pure states.

Exercise 3.1 (easy)**.** *Show that the circuit in figure 17 implements the decoherence channel, as defined in section 2.2.2.*

3.2.2 Reset channel

Here's a very simple channel that I haven't mentioned before, that I'll call the *reset channel*. The input is a qubit and the output is a qubit in state $|0\rangle$ (regardless of what the input state is). Here's a very simple Stinespring circuit for this.

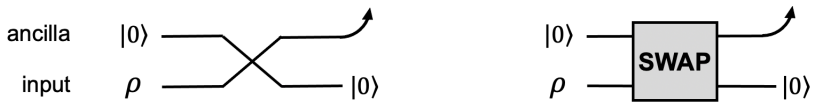

Figure 18: Circuit for the reset channel (with two different notations for the SWAP gate).

It's obvious that this circuit works: it traces out the input qubit and produces a qubit in state $|0\rangle$ as output. The following question about the reset channel is non-trivial.

Exercise 3.2. *Express the reset circuit in the Kraus form (in terms of Kraus operators). (Hint: two Kraus operators suffice.)*

Later in this section, we will see recipes for converting between the Stinespring form and the Kraus form, but there is a simple solution to the above which you might try to discover directly.

3.2.3 Depolarizing channel

The *depolarizing channel* is fundamental, and used as a natural model of noise. We'll be seeing more of this channel when we get to the subject of quantum error-correcting codes. The channel is parameterized by $p \in [0, 1]$, and it maps an input qubit in state $\rho \in \mathbb{C}^{2\times 2}$ to an output qubit in state

$$p\rho + (1-p) \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{94}$$

In other words, with probability p, the state is left alone and with probability $1-p$ the state is changed to the maximally mixed state $\left[\begin{smallmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{smallmatrix}\right]$. Here's what the effect of this channel looks like on the Bloch sphere.

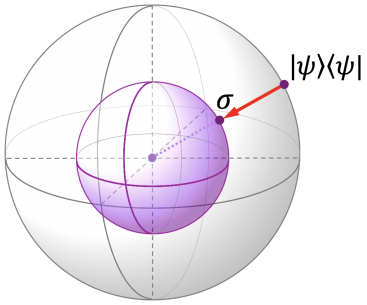

Figure 19: Effect of depolarizing channel on pure state $|\psi\rangle\langle\psi|$.

The maximally mixed state is at the centre of the Bloch sphere. The channel moves states towards the centre. In fact, the channel shrinks the entire Bloch sphere by a factor of p towards the centre.

Can we represent this channel in Stinespring form? Here's one Stinespring circuit for this channel, where $R = \left[\begin{smallmatrix} \sqrt{1-p} & -\sqrt{p} \\ \sqrt{p} & \sqrt{1-p} \end{smallmatrix}\right]$.

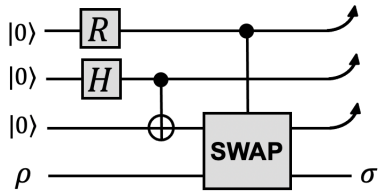

Figure 20: The depolarizing channel in the Stinespring form.

At first glance, this circuit may look complicated. But we can understand it by first looking at the two middle qubits. The H and CNOT gate are manufacturing a Bell state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. Note that each qubit of the Bell state is in state $\left[\begin{smallmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{smallmatrix}\right]$. Then the rest of the circuit applies

$$\begin{cases} \mathsf{SWAP} & \text{with prob. } p \\ I & \text{with prob. } 1-p. \end{cases} \tag{95}$$

This can be seen by noting that the circuit is equivalent to this.

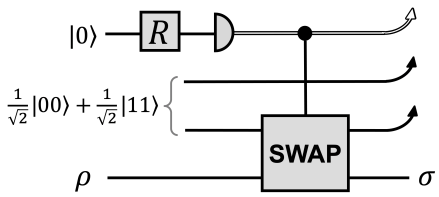

Figure 21: An equivalent circuit for the depolarizing channel.

Notice that this Stinespring form uses three ancilla qubits. Can this channel be constructed with fewer ancilla qubits?

A very easy way to reduce the ancilla to two qubits is to skip the Bell state and just initialize one ancilla qubit to the mixed state $\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$. Technically, this isn't in the Stinespring form of figure 15, since that requires all ancilla qubits to be initialized to a pure state. But a construction allowing an ancilla to be initialized to a mixed state might nevertheless be useful in some contexts.

However, it turns out that there is a different Stinespring circuit for the depolarizing channel that uses only two ancilla qubits initialized to state $|00\rangle$. The construction is rather elegant, and I leave it as an exercise.

Exercise 3.3. *Give a Stinespring form for the depolarizing channel that uses only two ancilla qubits in initial state* $|00\rangle$*.*

Is two qubits the optimal size of the ancilla? It turns out that one ancilla qubit is not enough for the depolarizing channel.

Exercise 3.4. *Prove that there is no Stinespring form for the depolarizing channel that uses only one ancilla qubit.*

And we can make more a fine-grained distinction regarding the size of the ancilla: what if the ancilla is allowed to be a qutrit?

Exercise 3.5. *Is there a Stinespring form for the depolarizing channel that uses one qutrit as ancilla? Justify your answer.*

3.3 Equivalence of Kraus and Stinespring channels

Recall from Definition 2.1 that $A_0, A_1, \ldots, A_{\ell-1}$ is a sequence of Kraus operators if

$$\sum_{k=0}^{\ell-1} A_k^* A_k = I. \tag{96}$$

Associated with any sequence of Kraus operators, we have two transformations: a Kraus measurement and a Kraus channel. We're now going to prove two theorems.

Theorem 3.1 (Kraus to Stinespring)**.** *Any transformation in the Kraus form can be simulated in the Stinespring form.*

Theorem 3.2 (Stinespring to Kraus)**.** *Any transformation in the Stinespring form can be simulated in the Kraus form.*

3.3.1 Kraus to Stinespring

In this section we prove Theorem 3.1. For Kraus operators $A_0, A_1, \ldots, A_{\ell-1} \in \mathbb{C}^{c\times d}$, consider the block matrix

$$\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{\ell-1} \end{bmatrix}, \tag{97}$$

which is an $\ell c \times d$ matrix. The columns of this matrix are orthonormal because

$$\begin{bmatrix} A_0^* & A_1^* & \cdots & A_{\ell-1}^* \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{\ell-1} \end{bmatrix} = \sum_{k=0}^{\ell-1} A_k^* A_k = I. \tag{98}$$

One consequence of this is that $d \leq \ell c$. Otherwise, the number of orthonormal vectors would have to exceed the dimension of the space in which they exist, which is impossible.

So we have ℓ Kraus operators that are $c \times d$ matrices and $d \leq \ell c$. To make the dimensions work out nicely, I'd like to assume that d divides ℓc. For this to hold, we might have to increase ℓ. It's straightforward to show that this can be done, where the new value of ℓ is less than double the original value of ℓ. If ℓ is increased we can add more Kraus operators that are zero matrices. Note that the larger set of matrices are still Kraus operators. So we can assume that d divides ℓc and set $m = \frac{\ell c}{d}$. Then we have $\ell c = md$.

Now, consider to the block matrix in Eq. (97) of Kraus operators again. Since its columns are orthonormal, we can extend this set of ℓc-dimensional column vectors to be an orthonormal basis of size ℓc. If we add these column vectors to the block

matrix, we end up with a square unitary matrix. Call this $\ell c \times \ell c$ matrix U, which is of the form

$$U = \left[\begin{array}{c|c} A_0 & \\ A_1 & \\ \vdots & W \\ A_{\ell-1} & \end{array}\right]. \tag{99}$$

Now consider this circuit.

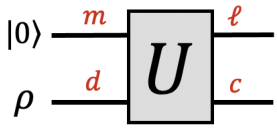

Figure 22: Stinespring circuit (omitting the final measurement/trace-out stage).

The input to the circuit consists of two registers: an m-dimensional ancilla and our d-dimensional input state. The circuit applies U to this. We can calculate the density matrix of the output state as

$$U\big(|0\rangle\langle 0| \otimes \rho\big)U^* = \left[\begin{array}{c|c} A_0 & \\ A_1 & \\ \vdots & W \\ A_{\ell-1} & \end{array}\right] \begin{bmatrix} \rho & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \left[\begin{array}{c|c} A_0^* & A_1^* \quad \cdots \quad A_{\ell-1}^* \\ \hline & \\ W^* & \\ & \end{array}\right] \tag{100}$$

$$= \left[\begin{array}{c|c} A_0\rho & \\ A_1\rho & \\ \vdots & 0 \\ A_{\ell-1}\rho & \end{array}\right] \left[\begin{array}{c|c} A_0^* & A_1^* \quad \cdots \quad A_{\ell-1}^* \\ \hline & \\ W^* & \\ & \end{array}\right] \tag{101}$$

$$= \begin{bmatrix} A_0^*\rho A_0 & A_0^*\rho A_1 & \cdots & A_0^*\rho A_{\ell-1} \\ A_1^*\rho A_0 & A_1^*\rho A_1 & \cdots & A_1^*\rho A_{\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\ell-1}^*\rho A_0 & A_{\ell-1}^*\rho A_1 & \cdots & A_{\ell-1}^*\rho A_{\ell-1} \end{bmatrix}. \tag{102}$$

For the Stinespring channel, the final step is to trace out the first register. This partial trace is the sum of the $c \times c$ blocks along the diagonal, which is

$$\sum_{k=0}^{\ell-1} A_k \rho A_k^*. \tag{103}$$

For the Stinesrping measurement, the final step is to measure the first register in the computational basis. This measurement is defined in the Kraus form in section 2.1.3, and it is straightforward to deduce that the probability that the outcome of this measurement is k is $\mathrm{Tr}(A_k \rho A_k^*)$.

3.3.2 Stinepring to Kraus

In this section I give a brief overview of the proof of Theorem 3.2. The Stinespring form consists of three stages. The first two are: adding an ancilla in state $|0\rangle$; and then applying a unitary operation U. The third stage is the partial trace for the Stinespring channel, and the measurement of the first register for the Stinespring measurement. Note that, in section 2, we have Kraus forms for each of these individual operations. We can compose these Kraus forms to obtain a Kraus channel from a Stinespring channel. And we can compose them to obtain a Kraus measurement from a Stinespring measurement.

3.4 Unifying measurements and channels

I have been describing state transitions as if there's a clear dichotomy between measurements and channels. You either measure and get a classical outcome and a residual state or you apply a channel and get just a quantum state as outcome. In fact, there's a general notion that unifies these.

Let $f : \{0, 1, \ldots, \ell - 1\} \to T$ be some function. Suppose that we apply the Kraus measurement and then apply f to the classical outcome. So the classical outcome is $f(k)$, rather than k.

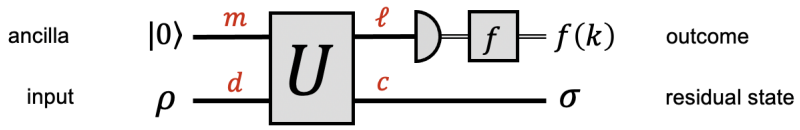

Figure 23: Generalized quantum transformation.

If f is a constant function, then seeing $f(k)$ provides us with no information about k. So that corresponds to the case of a channel. The other extreme case is where f is a bijection, for which knowing $f(k)$ provides full information about k. And there are in-between cases where f is not constant nor a bijection. In those cases, we receive *partial* information about k. The classical outcome $f(k)$ might narrow down the possible values of k, but without uniquely determining k.

3.5 POVM measurements

A final topic concerns *POVM measurements* (POVM stands for *positive operator valued measure*[7]). This is a simplified way of describing a Kraus measurement, that works if we *only* care about the classical outcome (so we do *not* care about the residual quantum state).

Recall that, for Kraus operators $A_0, A_1, \ldots, A_{\ell-1}$, the associated measurement of a state ρ produces outcome k with probability

$$\mathrm{Tr}(A_k \rho A_k^*) = \mathrm{Tr}(\rho\, A_k^* A_k). \tag{104}$$

For each Kraus operator A_k, define $E_k = A_k^* A_k$. All we need to know is the sequence $E_0, E_1, \ldots, E_{\ell-1}$ to define the classical part of the measurement outcome. And we can characterize such sequences $E_0, E_1, \ldots, E_{\ell-1}$ in a simple way.

Definition 3.1 (POVM elements)**.** *A sequence $E_0, E_1, \ldots, E_{\ell-1}$ is a* sequence of POVM elements *if, for all k, it holds that E_k is positive,*[8] *and*

$$E_0 + E_1 + \cdots + E_{\ell-1} = I. \tag{105}$$

For a sequence of POVM elements $E_0, E_1, \ldots, E_{\ell-1}$ and a quantum state ρ, applying the associated *POVM measurement* produces outcome $k \in \{0, 1, \ldots, \ell - 1\}$ with probability $\mathrm{Tr}(\rho E_k)$.

⚠ A word of caution: for a POVM measurement, there is no way to define a residual quantum state. This is because we cannot uniquely deduce a set of underlying Kraus operators from POVM elements. We can find an A_k such that $E_k = A_k^* A_k$, but this A_k is not unique, and for a different choices of A_k the residual state is different. So we should use Kraus operators if we want to be able to refer to the quantum state after the measurement.

[7] Regarding terminology, a *positive operator valued measure* is a generalization of the notion of a *measure*, that you may have seen in probability theory or functional analysis. The word "measure" is distinct from "measurement". So it makes sense to say "POVM measurement".

[8] Meaning that E_k is normal and all its eigenvalues are nonnegative reals.

4 Distance measures between states

This section is about distance measures between quantum states. We'll see various ways of quantifying how different two quantum states are, including the *fidelity* and the *trace distance*. I'll also show you the Holevo-Helstrom Theorem, which relates trace distance to the operational problem of distinguishing between a pair of states by a measurement.

Recall that we consider two quantum states to be *indistinguishable* if, for any measurement procedure, the probability distribution of outcomes is identical between the two states. For example, $|0\rangle$ and $-|0\rangle$ are indistinguishable. In [*Part 2: Quantum algorithms*, section 1] we saw different probabilistic mixtures that resulted in the same density matrix. In all such cases, the two states are indistinguishable; we don't even consider them to be different states.

Definition 4.1 (distinguishable states)**.** *We say two states are* distinguishable *if they're not indistinguishable. In other words, if for some measurement procedure, the outcome probabilities are different for the two states.*

For example, the $|0\rangle$ and the $|+\rangle$ state are distinguishable. Note that our definition of distinguishable does not require us to be able to perfectly tell the two states apart. Another example is $\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$ and $|+\rangle\langle +|$.

Definition 4.2 (perfectly distinguishable states)**.** *Define two states to be* perfectly distinguishable *if there is a measurement procedure that perfectly tells them apart.*

For example, $|+\rangle$ and $|-\rangle$ are perfectly distinguishable.

We have three qualitative categories: indistinguishable, distinguishable, and perfectly distinguishable. Can we quantify how different two states are?

4.1 Operational distance measure

An operational way of quantifying the difference between two states is based on the guess-the-state game that we've seen several times.

For any two states, which in general can be mixed states, imagine the game where Alice flips a fair coin to decide which of the two states to set a quantum system to, and then she sends the quantum system to Bob. Bob knows what the two possible states are, but Alice does not tell him which one she chose. Bob's goal is to apply a measurement procedure measurement to the state that he received and to use the

classical outcome to guess which state it was. We can write the success probability of Bob's optimal measurement procedure as $(1+\delta)/2$, where $\delta \in [0,1]$.

The trivial strategy for Bob is to just guess a random bit (ignoring the system that he receives from Alice). That strategy succeeds with probability $\frac{1}{2}$. We can think of δ as how much better one can do than that baseline. Another way of viewing δ is as the success probability minus the failure probability. It's sometimes useful to look at δ that way.

For a given pair of quantum states, what's the best δ attainable? If the two states are indistinguishable then $\delta = 0$ is the best possible. At the other extreme, if the two states are perfectly distinguishable then the success probability can be 1, so $\delta = 1$.

An in-between case is when the two states are $|0\rangle$ and $|+\rangle$. These are not perfectly distinguishable, but the distinguishing probability can be as high as

$$\cos^2(\tfrac{\pi}{8}) = \frac{1+\cos(\frac{\pi}{4})}{2} = \frac{1+\frac{1}{\sqrt{2}}}{2}, \tag{106}$$

so $\delta = \frac{1}{\sqrt{2}} \approx 0.707$. Another in-between case is $\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$ vs. $|+\rangle\langle +|$, for which the best possible distinguishing probability is $\frac{3}{4} = \left(1+\frac{1}{2}\right)/2$, so in that case $\delta = \frac{1}{2}$.

This is one natural way of quantifying the distance between two states, in terms of the highest distinguishing probability possible. A natural question is: given two density matrices ρ_0 and ρ_1, what is the highest distinguishing probability possible? In section 4.5, we'll see a systematic way of addressing such questions.

4.2 Geometric distance measures

Now, let's look at the distance between two quantum states form a different perspective: a geometric perspective.

4.2.1 Euclidean distance

If we have two d-dimensional pure states, $|\psi_0\rangle$ and $|\psi_1\rangle$ then it seems natural to take the Euclidean distance between their state vectors $\big||\psi_0\rangle - |\psi_1\rangle\big|_2$.

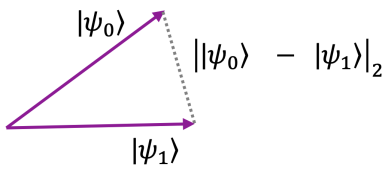

Figure 24: Euclidean distance between pure states $|\psi_0\rangle$ and $|\psi_1\rangle$.

If their Euclidean distance is small then it's intuitively natural to think of the states as "close". But, if their Euclidean distance is large, should we think of the states as "far apart"? Not necessarily, because of global phases. Note that $-|\psi\rangle$ is the unit vector farthest away from $|\psi\rangle$ (the Euclidean distance being 2), even though these are the same state.

Also, it is not so clear how this notion of Euclidean distance can be extended to mixed-states.

4.2.2 Fidelity

Another notion of distance is called the *fidelity*, which for pure states is the absolute value of the inner product between the state vectors $\big|\langle\psi_0|\psi_1\rangle\big|$.

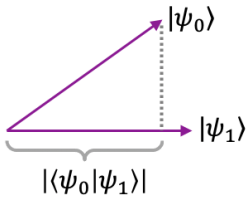

Figure 25: Fidelity between pure states $|\psi_0\rangle$ and $|\psi_1\rangle$.

This distance measure is calibrated in reverse in the sense that *large* fidelity means "close" and *small* fidelity means "far". Clearly, fidelity 1 means indistinguishable and fidelity 0 means perfectly distinguishable (since orthogonal states are perfectly distinguishable). Notice how the absolute value takes care of any distinctions between vectors due to global phases.

For the in-between values of fidelity, if the fidelity is close to 1 means that the states are "close".

What about the fidelity between mixed states? It turns out that there is a definition of fidelity for mixed states. If ρ_0 and ρ_1 are the density matrices of the two mixed states, then the fidelity is given by the formula

$$F(\rho_0, \rho_1) = \mathrm{Tr}\Big(\sqrt{\sqrt{\rho_0}\,\rho_1\sqrt{\rho_0}}\Big). \tag{107}$$

I'm not going to explain this formula, but I want you to see it. You cannot simplify this formula by cyclically permuting one of the $\sqrt{\rho_0}$ factors to the other side because of the square root within the trace. One reasonable property that this has is that, it agrees with the definition $\big|\langle\psi_0|\psi_1\rangle\big|$ for the very special case of pure states. This is easy to verify, which I'll leave as an exercise.

Exercise 4.1. *Prove that, if* $\rho_0 = |\psi_0\rangle\langle\psi_0|$ *and* $\rho_1 = |\psi_1\rangle\langle\psi_1|$ *then it holds that* $F(\rho_0, \rho_1) = \big|\langle\psi_0|\psi_1\rangle\big|$.

After looking at that expression for fidelity involving all those square roots of matrices, let's think about what it means to take the square root of a matrix. This is part of a more general *functional calculus* on square matrices.

4.3 Functional calculus for linear operators

Suppose that M is a normal matrix and $f : \mathbb{C} \to \mathbb{C}$. Then we can define f applied to the matrix M as follows. Since M is normal, we can diagonalize M in some orthonormal basis (the columns of some unitary U) as

$$M = U^* \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix} U. \tag{108}$$

Define $f(M)$ as the matrix where f is applied to each eigenvalue, namely,

$$f(M) = U^* \begin{bmatrix} f(\lambda_1) & 0 & \cdots & 0 \\ 0 & f(\lambda_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f(\lambda_d) \end{bmatrix} U. \tag{109}$$

Square root of a positive matrix

Every $x \in \mathbb{C}$ has at least one square root, and if $x \neq 0$ then x has two square roots. However, if $x \in \mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$ then x has a unique square root in $\mathbb{R}_+$. Whenever $M \geq 0$, there is a natural definition of $\sqrt{M}$ since then the eigenvalues of M are in $\mathbb{R}_+$ and have unique square roots in $\mathbb{R}_+$.

⚠ A word of caution: in general, for positive matrices L and M, it does *not* hold that $\sqrt{LM} = \sqrt{L}\sqrt{M}$. This is because L and M may not be simultaneously diagonalizable. They are each diagonalizable, but not necessarily with respect to the same orthonormal basis.

Von Neumann entropy of a positive matrix

Whenever $M \geq 0$, it makes sense to define $M \log M$ (which is related to the von Neumann Entropy of a quantum state, defined as $\mathrm{Tr}\big(M \log M\big)$), that will come up in a later part of the course.

Absolute value of any matrix

We can also define the absolute value of a normal matrix M using the functional calculus, as the absolute values of all the eigenvalues

$$|M| = U^* \begin{bmatrix} |\lambda_1| & 0 & \cdots & 0 \\ 0 & |\lambda_2| & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & |\lambda_d| \end{bmatrix} U. \tag{110}$$

Notice that

$$|M| = \sqrt{M^* M}, \tag{111}$$

and this interesting because for *any* (not necessarily normal) matrix M, it holds that $M^*M \geq 0$. Therefore, we have a definition of the absolute value that extends to any matrix (not necessarily diagonalizable).

4.4 Trace norm and trace distance

Now, I'd like to show you a very interesting distance measure between quantum states, called the trace distance. First, I'll show you the definition of the trace norm of a matrix, which is the trace of the absolute value of the matrix.

Definition 4.3 (trace norm)**.** *For any $M \in \mathbb{C}^{d \times d}$, the* trace norm *of M is defined as*

$$\|M\|_1 = \mathrm{Tr}\big(|M|\big) = \mathrm{Tr}\big(\sqrt{M^* M}\,\big). \tag{112}$$

The notation is as a norm with subscript 1 (sometimes this alternative notation is used, with the subscript "tr").

It's not too hard to show that, if M is normal then the trace norm of M is the 1-norm of the vector of eigenvalues of M. In other words, if the eigenvalues of M are $\lambda_1, \lambda_2, \ldots, \lambda_d$ then

$$\|M\|_1 = |\lambda_1| + |\lambda_2| + \cdots + |\lambda_d|. \tag{113}$$

Now we are ready to define the trace distance between two states. It's the trace norm of the difference between their density matrices.

Definition 4.4 (trace distance)**.** *For any two d-dimensional states, whose density matrices are ρ_0 and ρ_1, the* trace distance *between them is*

$$\|\rho_0 - \rho_1\|_1. \tag{114}$$

Of all the different matrix norms on which we could base a distance measure, what's so special about the trace norm? Why is this a meaningful measure of distance between states? The answer is given by the amazing Holevo-Helstrom Theorem.

4.5 The Holevo-Helstrom Theorem

Remember the δ that arose in our discussion of state distinguishability? We defined δ to be the advantage over random guessing in the guess-the-state game. In fact, that δ is exactly the trace norm multiplied by $\frac{1}{2}$. So the trace norm coincides with how well the states can be distinguished in the guess-the-state game!

Theorem 4.1 (Holevo-Helstrom Theorem)**.** *Let ρ_0 and ρ_1 be the density matrices of two d-dimensional states. If one of these two states is prepared by the flip of a fair coin and then the best distinguishing procedure succeeds with probability*

$$\frac{1 + \frac{1}{2}\|\rho_0 - \rho_1\|_1}{2}. \tag{115}$$

(If the trace distance had been defined as $\frac{1}{2}\|\rho_0 - \rho_1\|_1$ instead of $\|\rho_0 - \rho_1\|_1$ then the factor of $\frac{1}{2}$ would not apear in Eq. (115). But we're stuck with the standard definition.)

To prove the Holevo-Helstrom Theorem, we need to show:

- There is a measurement whose success probability is $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$.
- No measurement can perform better than $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$.

4.5.1 Attainability of success probability $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$

In this section, we prove the attainability part of Theorem 4.1. Namely, that there exists a measurement that attains success probability $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$, where ρ_0 and ρ_1 are the density matrices of the states that we wish to distinguish between.

Note that, since ρ_0 and ρ_1 are Hermitian, $\rho_0 - \rho_1$ is also Hermitian. But notice that, because of the minus sign, $\rho_0 - \rho_1$ need not be positive. In general, $\rho_0 - \rho_1$ has some negative eigenvalues.

Consider the two projectors, Π_0 and Π_1, defined as follows. Let Π_0 be the projector onto the space of all eigenvectors of $\rho_0 - \rho_1$ whose eigenvalues are ≥ 0. Let Π_1 be the projector onto the space of all eigenvectors of $\rho_0 - \rho_1$ whose eigenvalues are < 0. Then Π_0 and Π_1 are orthogonal projectors and $\Pi_0 + \Pi_1 = I$. Therefore Π_0 and Π_1 are the elements of a POVM measurement. We'll show that this measurement succeeds with probability $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$.

First note that

$$\big(\Pi_0 - \Pi_1\big)\big(\rho_0 - \rho_1\big) = \big|\rho_0 - \rho_1\big|. \tag{116}$$

To see why this is so, think of $\Pi_0 - \Pi_1$ and $\rho_0 - \rho_1$ in diagonal form (they are simultaneously diagonalizable). $\Pi_0 - \Pi_1$ has a $+1$ eigenvalue in all the positions where the corresponding eigenvalue of $\rho_0 - \rho_1$ is non-negative. $\Pi_0 - \Pi_1$ has a -1 eigenvalue in all the positions where the corresponding eigenvalue of $\rho_0 - \rho_1$ is negative. Therefore, $\Pi_0 - \Pi_1$ flips the sign of all the negative eigenvalues of $\rho_0 - \rho_1$, resulting in $\big|\rho_0 - \rho_1\big|$.

Note that Eq. (116) implies that

$$\mathrm{Tr}\big((\Pi_0 - \Pi_1)(\rho_0 - \rho_1)\big) = \|\rho_0 - \rho_1\|_1. \tag{117}$$

We can also expand

$$\begin{aligned}\mathrm{Tr}\big((\Pi_0 - \Pi_1)(\rho_0 - \rho_1)\big) &= \mathrm{Tr}(\Pi_0\rho_0) - \mathrm{Tr}\,\Pi_0\rho_1) - \mathrm{Tr}(\Pi_1\rho_0) + \mathrm{Tr}(\Pi_1\rho_1) &(118)\\ &= \big(\mathrm{Tr}(\Pi_0\rho_0) + \mathrm{Tr}(\Pi_1\rho_1)\big) - \big(\mathrm{Tr}(\Pi_0\rho_1) + \mathrm{Tr}(\Pi_1\rho_0)\big), &(119)\end{aligned}$$

where $\frac{1}{2}\big(\mathrm{Tr}(\Pi_0\rho_0) + \mathrm{Tr}(\Pi_1\rho_1)\big)$ is the success probability,[9] and $\frac{1}{2}\big(\mathrm{Tr}(\Pi_0\rho_1) + \mathrm{Tr}(\Pi_1\rho_0)\big)$ is the failure probability.[1] So the success probability minus the failure probability is equal to $\frac{1}{2}$ times the trace distance.

Note that our proof of this part is constructive. It actually gives us a recipe to determine the optimal measurement for distinguishing between two mixed states: consider the matrix that is one density matrix subtracted from the other density matrix; take the projector to the positive eigenspace of this matrix and the projector to the negative eigenspace of this matrix; that's the POVM measurement that solves the distinguishing problem with success probability $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$.

[9] Averaged over the random choice of the state.

4.5.2 Optimality of success probability $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$

So far, we've proved that a particular measurement attains the proposed success probability. But how do we know that there isn't an even better measurement? In this section, we prove the optimality part of Theorem 4.1. Namely, that there does not exist a measurement whose success probability exceeds $\frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1$.

Let E_0 and E_1 be the POVM elements of any 2-outcome POVM measurement for distinguishing between ρ_0 and ρ_1. We'll prove an upper bound on it's performance.

First, notice that E_0 and E_1 are simultaneously diagonalizable.[10] Why? Because $E_1 = I - E_0$. So if E_0 is diagonal is some coordinate system then so is E_1. Therefore, we can write

$$E_0 = U^* \begin{bmatrix} p_0 & 0 & \cdots & 0 \\ 0 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{d-1} \end{bmatrix} U \quad \text{and} \quad E_1 = U^* \begin{bmatrix} q_0 & 0 & \cdots & 0 \\ 0 & q_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{d-1} \end{bmatrix} U. \tag{120}$$

The eigenvalues of E_0 and E_1 are between 0 and 1 and corresponding eigenvalues sum to 1. It follows that the largest eigenvalue of $E_0 - E_1 \leq 1$.

Definition 4.5 (infinity norm)**.** *For a normal matrix M, the* infinity norm[11] $\|M\|_\infty$ *is defined as the absolute value of the largest eigenvalue of M.*

In this language, we have that $\|E_0 - E_1\|_\infty \leq 1$. We will make use of the following lemma, which is an instance of Hölder's inequality for matrices.

Lemma 4.1. *For any Hermitian L and M,* $\mathrm{Tr}(LM) \leq \|L\|_\infty \|M\|_1$.

Now, let's continue proving that any POVM measurement for distinguishing between ρ_0 and ρ_1 does not outperform the measurement that we constructed. Define

$$A = \frac{\rho_0 + \rho_1}{2} \qquad \text{and} \qquad B = \frac{\rho_0 - \rho_1}{2}. \tag{121}$$

Note that $\rho_0 = A + B$, $\rho_1 = A - B$, and $\mathrm{Tr}(A) = 1$.

[10] Please note that this only holds because it's a 2-outcome POVM measurement. For POVM measurements with 3 or more outcomes, the matrices might not be simultaneously diagonalizable.

[11] This is equivalent to the *spectral norm*, which is defined for all matrices.

The success probability (averaged over inputs) is

$$\begin{aligned}
\tfrac{1}{2}\operatorname{Tr}(E_0\rho_0) + \tfrac{1}{2}\operatorname{Tr}(E_1\rho_1) &= \tfrac{1}{2}\operatorname{Tr}(E_0(A+B)) + \tfrac{1}{2}\operatorname{Tr}(E_1(A-B)) && (122)\\
&= \tfrac{1}{2}\operatorname{Tr}((E_0+E_1)A) + \tfrac{1}{2}\operatorname{Tr}((E_0-E_1)B) && (123)\\
&\leq \tfrac{1}{2}\operatorname{Tr}(A) + \tfrac{1}{2}\|E_0-E_1\|_\infty \|B\|_1 && (124)\\
&= \tfrac{1}{2} + \tfrac{1}{4}\|\rho_0-\rho_1\|_1. && (125)
\end{aligned}$$

This proves optimality and we have now completed the proof of the Holevo-Helstrom Theorem.

4.6 Purifications and Ulhmann's Theorem

Next, I'd like to show you what a *purification* of a quantum state is. Any mixed state can be viewed as a pure state on some larger system with part of that larger system traced out.

Let ρ be any density matrix of a d-dimensional system. Suppose ρ can be written as a probabilistic mixture of m pure states, as

$$\rho = \sum_{k=0}^{m-1} p_k \left|\psi_k\right\rangle\!\left\langle\psi_k\right|. \tag{126}$$

Now, consider the pure state on a d-dimensional register and an m-dimensional register

$$\left|\phi\right\rangle = \sum_{k=0}^{m-1} \sqrt{p_k}\left|\psi_k\right\rangle \otimes \left|k\right\rangle. \tag{127}$$

It's easy to see that $\operatorname{Tr}_2 \left|\phi\right\rangle\!\left\langle\phi\right| = \rho$. The pure state $\left|\phi\right\rangle$ is called a *purification* of ρ. This is one way to purify ρ, but the purification of a state is not unique.

Now, let's recall the previous strange-looking definition of fidelity between general mixed states that I showed you earlier—but which I didn't explain. Using our language of the trace norm, we can rewrite the expression for fidelity as

$$F(\rho_0,\rho_1) = \left\|\sqrt{\rho_0}\sqrt{\rho_1}\right\|_1. \tag{128}$$

For any two density matrices, we can take purifications of them and then take the inner product of these purifications. The result will depend on which purification we choose. The following theorem relates the fidelity between mixed states with inner products of their purifications.

Theorem 4.2 (Ulhmann's Theorem)**.** *For any two mixed states ρ_0 and ρ_1, the fidelity between them is the maximum $\langle\phi_0|\phi_1\rangle$ taken over all purifications $|\phi_0\rangle$ and $|\phi_1\rangle$.*

For further details, please see [Nielsen and Chuang, *Quantum Computation and Quantum Information*, pp. 410–411].

4.7 Fidelity vs. trace distance

We've discussed fidelity and trace distance, mostly the latter. Known relationships between them are

$$1 - F(\rho_0, \rho_1) \leq \|\rho_0 - \rho_1\|_1 \leq \sqrt{1 - F(\rho_0, \rho_1)^2}. \tag{129}$$

In particular, suppose that we have two mixed states with density matrices ρ_0 and ρ_1, and we want to show that their trace distance is small. One way to do this is to construct purifications of them whose inner products are close to 1. By Ulhmann's Theorem and the second inequality above, for any purifications $|\phi_0\rangle$ and $|\phi_1\rangle$ we have

$$\|\rho_0 - \rho_1\|_1 \leq \sqrt{1 - \langle\phi_0|\phi_1\rangle^2}. \tag{130}$$

5 Simple quantum error-correcting codes

In this section, we begin the subject of quantum error-correcting codes, which can protect quantum states from noise. We'll first briefly review some results about error-correcting codes for *classical* information. Then we'll consider *quantum* error-correcting codes and I'll explain Shor's nine-qubit quantum error-correcting code.

Let's start by discussing what noise is. Broadly speaking, noise is when information gets disturbed. Imagine that Alice wants to send some bits to Bob, but their communication channel is flawed.

Figure 26: Alice sending Bob a bit over a noisy communication channel.

Suppose that, for each bit $b \in \{0,1\}$ that Alice sends via the channel, what Bob receives is

$$\begin{cases} b & \text{with prob. } 1-\epsilon \\ \neg b & \text{with prob. } \epsilon, \end{cases} \tag{131}$$

for some parameter $\epsilon \in [0, \frac{1}{2}]$. So each bit gets flipped with probability ϵ. This mapping from states of bits to states of bits is called a *binary symmetric channel*, and we refer to it as BSC (or as BSC_ϵ, to specify the parameter ϵ).

This channel can be viewed as a classical analogue of the depolarizing channel. Recall that the depolarizing channel takes a qubit as input and produces as output a probabilistic mixture of that state and the maximally mixed state. The binary symmetric channel outputs a probabilistic mixture of the input bit and a classical maximally mixed state (which is a uniformly distributed random bit). If we identify bit 0 with the probability vector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and bit 1 with $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ then the binary symmetric channel BSC_ϵ is a linear mapping such that

$$\mathsf{BSC}_\epsilon \begin{bmatrix} 1 \\ 0 \end{bmatrix} = (1-2\epsilon) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2\epsilon \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \tag{132}$$

$$\mathsf{BSC}_\epsilon \begin{bmatrix} 0 \\ 1 \end{bmatrix} = (1-2\epsilon) \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 2\epsilon \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}. \tag{133}$$

Now suppose that Alice wants to communicate a bit $b \in \{0, 1\}$ to Bob and their communication channel is a binary symmetric channel BSC_ϵ. Can Alice and Bob reduce the noise level to a smaller ϵ? The obvious way is for Alice and Bob to get a better communication hardware, with a smaller parameter ϵ. But Alice and Bob have an alternative to investing in better hardware: they can use an error-correcting code.

5.1 Classical 3-bit repetition code

Perhaps the simplest error-correcting code is the 3-bit repetition code, which works as follows. To transmit bit $b \in \{0, 1\}$, Alice encodes b into three copies of b. Then Alice sends each of these three bits through the channel. Then Bob takes the majority value of the three bits that he receives (which might not all be the same, because some bits might get flipped by the channel).

How well does this perform? The system succeeds if no more than one bit is flipped (because that doesn't change the majority) and it fails if two or more bits are flipped. Assume that the channel behaves independently for each bit that passes through it.

Then the failure probability can be calculated as follows. There are three ways that one bit can be flipped, each occurring with probability $\epsilon^2(1 - \epsilon)$. And there is one way that all three bits can flip, occurring with probability ϵ^3. So the failure probability is

$$3\epsilon^2(1 - \epsilon) + \epsilon^3 = 3\epsilon^2 - 2\epsilon^3. \tag{134}$$

Here's a plot of the failure probability resulting from this scheme as a function of the original failure probability of the channel.

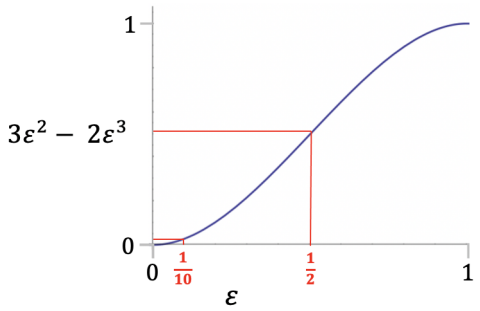

Figure 27: Failure probability of the 3-bit repetition code as a function of ϵ.

If $\epsilon = \frac{1}{2}$ then there is no improvement: the success probability remains at $\frac{1}{2}$. But this should not be surprising, because, if $\epsilon = \frac{1}{2}$ then the channel sends no information: it just outputs a random bit uncorrelated with the bit that Alice is sending. But when ϵ is smaller there is an advantage. For example, when $\epsilon = \frac{1}{10}$ the failure probability from using the code is around $\frac{1}{35}$. And the smaller ϵ is the more pronounced the error reduction is. If $\epsilon = \frac{1}{1000}$ then the failure probability from using the codes is around $\frac{1}{300000}$ (a three-hundred-fold decrease).

What price are we paying for this improvement? The main cost is that that three bits have to be sent instead of one.

Definition 5.1 (rate of a code)**.** *The* rate *of a code is the inverse of the expansion in message length due to the encoding.*

The rate of this code is $\frac{1}{3}$. Each bit of the encoding conveys $\frac{1}{3}$ of a bit of the data to be transmitted.

If the data is a long string of bits then there will be errors, but fewer errors using the code. With no code, the expected fraction of errors is ϵ. With the code, the expected fraction of errors is $3\epsilon^2 - 2\epsilon^3$. Suppose that Alice wants to send a Gigabyte to Bob (that's around 8.5 billion bits) and their communication channel has noise parameter $\epsilon = \frac{1}{100}$. Without the code, the fraction of errors will be around 85 million. With the code, the fraction of errors will be about 24 thousand. That's significantly fewer errors. Achieved at the cost of sending three Gigabytes instead of one.

But suppose the we don't want *any* errors. Can this be achieved? One approach is to use a larger repetition code than 3-bits. That reduces the error probability for each bit. But this also reduces the rate—so, in the above example, many more Gigabytes would have to be sent. But there are much better error-correcting codes than repetition codes.

5.2 Brief remarks about the existence of good classical codes

An error-correcting code need not separately encode each bit. Rather, each block of n bits (a message) can be encoded into a block of m bits (a codeword). The rate of such a code is $\frac{n}{m}$. Note that a small rate means a large message expansion (an inefficiency); whereas, a rate close to 1 means a small message expansion.

A fundamental result about the existence of good classical error-correcting codes can be informally stated as:

> For a binary symmetric channel with any error parameter $\epsilon < \frac{1}{2}$, the success probability for encodings of long strings can be made arbitrarily close to 1, *while maintaining a constant rate.*

So, in fact, Alice doesn't have to send many Gigabytes in order to get every single bit through to Bob correctly.

I'm going to state the result about good error-correcting codes more precisely. Please note that I'm not going to give the details of the construction or the analysis. The theory of error-correcting codes is a large field of study, that could easily take a course its own course to explain.

In order to state the result, we need some basic definitions for block codes. Such a code consists of an *encoding* function $E : \{0,1\}^n \to \{0,1\}^m$ and a *decoding* function $D : \{0,1\}^m \to \{0,1\}^n$. The rate of such a code is $\frac{n}{m}$.

Assuming that the communication channel is a binary symmetric channel with error parameter ϵ, the error probability of a code of the above form is defined as the maximum, for all $a_1 a_2 \dots a_n \in \{0,1\}^n$, of

$$\Pr\big[D(\mathsf{BSC}_\epsilon(E(a_1 a_2 \dots a_n))) \big] \neq a_1 a_2 \dots a_n. \tag{135}$$

Just to be clear: success means all the bits are successfully received at the other end; that there are no errors.

Definition 5.2 (Shannon entropy)**.** *The* Shannon entropy *of a probability vector* $(p_1, p_2, \dots, p_d)$ *is defined as*

$$H(p_1, p_2, \dots, p_d) = -\sum_{k=1}^{d} p_k \log p_k \,. \tag{136}$$

Define $R : [0, \frac{1}{2}] \to [0,1]$ as $R(\epsilon) = 1 - H(\epsilon, 1-\epsilon)$. Here is a plot of this function.

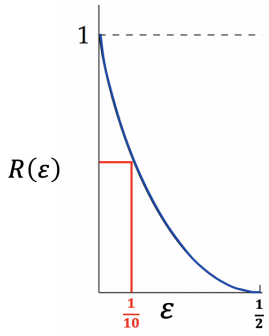

Figure 28: The rate function $R(\epsilon)$.

I'm now going to state the result about good multi-bit error-correcting codes. Let the noise level be any $\epsilon < 1/2$. Think of that as a property of the communication channel that you're stuck with using.

Then you can select any rate r, as long as $r < R(\epsilon)$. And you can select an arbitrarily small $\delta > 0$, which is your desired error probability bound. Then there exists an error-correcting code $E : \{0,1\}^n \to \{0,1\}^m$, $D : \{0,1\}^m \to \{0,1\}^n$ with rate $\frac{n}{m} > r$ and whose failure probability is less than δ.

There are additional considerations, that I'd like to mention, even though I won't go in the details:

- One is the block-length n. The smaller δ is, the larger n has to be.
- Another is the computational cost of computing E and D. The bottom line is that there are codes for which this can be done efficiently.

OK, so that was a very brief overview of error-correcting codes for classical information. The question is what happens with quantum error-correcting codes.

5.3 Shor's 9-qubit quantum error-correcting code

We started with a simple classical error-correcting code, the 3-bit repetition code. So we might be tempted to start with a quantum repetition code, where a qubit is encoded as three copies of itself.

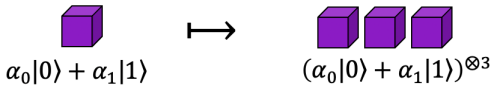

Figure 29: A naïve first attempt at a 3-qubit quantum repetition code.

Of course, this fails for multiple reasons, starting with the fact that a general quantum state cannot be copied (the no-cloning theorem).

It's easy to copy classical information, and all error-correcting codes for classical information are based on some sort of redundancy. But the no-cloning theorem kind of suggests that redundancy for quantum information might not be possible. That was the thinking shortly after Shor's algorithms for factoring and discrete log came out. But the underlying intuition that no-cloning implies no-redundancy was wrong.

I'm going to show you the first error-correcting code, that was discovered by Peter Shor in the mid-1990s. It's a 9-qubit code that is constructed by combining two 3-qubit codes that protect against very limited error types.

5.3.1 3-qubit code that protects against one X error

Let's start with a simple 3-qubit code that protects against a very restricted error-set. Suppose the only error possible is a Pauli X, a bit flip. Then these encoding and decoding circuits protect against up to one X-error.

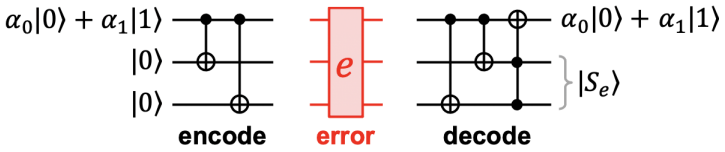

Figure 30: 3-qubit code that protects against one X error.

The *encoding* circuit takes a qubit as input and produces three qubits as output. Then the encoded data is affected by an error e, where e can be any one of these four unitary operations:

$I \otimes I \otimes I$	$X \otimes I \otimes I$	$I \otimes X \otimes I$	$I \otimes I \otimes X$
(no error)	(flip 1st qubit)	(flip 2nd qubit)	(flip 3rd qubit)

(137)

Then the three qubits are input to the *decoding* circuit.

It turns out that, in all four cases of e, the data is correctly recovered. The final state of the first qubit is the same as its initial state. It's a straightforward exercise to verify these, but I recommend that you work through some of these cases to convince yourself.

If you work out the final states, you will see that the second and third qubits end up in a state that depends on what the error operation e is. We'll refer to these two qubits as the *syndrome of the error*, denoted as $|s_e\rangle$. So, not only does the encoding/decoding protect the data against the errors, but the decoding process also reveals what the error e is.

What about other errors? Specifically, what happens if there is a Z-error on one of the three encoded qubits? A Z-error is not corrected; instead it's passed through. By this, I mean that if the data is in state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ then applying a Z-error to one of the three encoded qubits causes the output of the decoding circuit to be $\alpha_0 |0\rangle - \alpha_1 |1\rangle$, which is the same as applying Z directly to the original data. So this encoding/decoding does *not* protect against an Z-error, but passes it through.

5.3.2 3-qubit code that protects against one Z error

Now, here's another 3-qubit code which protects against up to one Z-error.

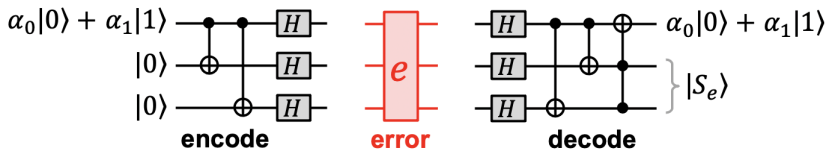

Figure 31: 3-qubit code that protects against one Z error.

The encoding/decoding is like the code in figure 30 for protecting against X-errors, but with a layer of Hadamard gates added to the end of the encoding and the beginning of the decoding. This is essentially a re-purposing of our code for X-errors into a code for Z-errors. Think of a Z-error and the H gates. Since $HZH = X$ and $HH = I$, any Z-errors effectively become X-errors and then are handled as the previous encoding/decoding in figure 30.

5.3.3 9-qubit code that protects against one Pauli error

By combining the 3-qubit codes from figures 30 and 31, we can obtain a 9-qubit code that protects against any Pauli error (I, X, Y, or Z) in any one of the nine qubit positions. The encoding/decoding circuits are the following.

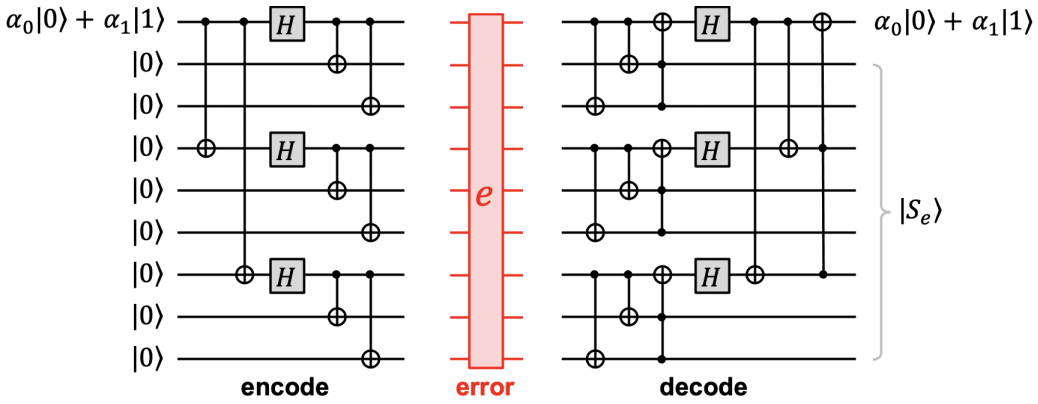

Figure 32: Shor's 9-qubit code.

A nice way of understanding how this code works is in terms of it's inner part (the last two layers of gates in the encoding part and first three layers of gates in the decoding part) and an outer part (the other gates). The inner part consists of three blocks, where each block is a copy of our code for X-errors from figure 30. And the outer part is our code for Z-errors from figure 31.

What happens if there's an X-error? It's corrected by the inner part. What happens if there's a Z-error? A Z-error is passed through by the inner part and then corrected by the outer part. What happens if there's a Y-error? Since $Y = iXZ$ (and the phase i makes no difference in this context), this can be viewed as a Z-error and an X-error. The inner part corrects the X-error and the outer part corrects the Z-error.

So if the error e is any Pauli error in one qubit position then it is corrected by this code; the data is recovered in the final state of the first qubit. There are 28 different one-qubit Pauli errors e (including $I^{\otimes 9}$) that this code protects against. The final state of eight qubits after the first qubit is the *error syndrome* $|s_e\rangle$, and contains information about which error was corrected.[12]

In fact, this codes corrects against an *arbitrary* error in any one qubit position and it is also effective for communicating through a channel with depolarizing noise.

5.4 Quantum error models

Let's step back and consider some of the error models that the Shor code protects against. There are several error models, but let's focus on two that are the most closely related to our discussion.

Worst-case unitary noise

By *worst-case unitary noise* noise, we mean that there is a set of possible unitary errors, and the error can be any one of them. For example, the Shor code is resilient against an arbitrary one-qubit Pauli error on a 9-qubit encoding.

Figure 33: Arbitrary unitary error on one single qubit.

In fact, if a code is resilient against any 1-qubit Pauli error then it is resilient against *any* 1-qubit unitary operation U. To see why this is so, note that any 2×2 unitary U can be expressed as

$$U = \eta_0 I + \eta_x X + \eta_y Y + \eta_z Z, \tag{138}$$

[12] A curiosity is that some of the 28 possible one-qubit Pauli errors have the error syndrome.

for some $\eta_0, \eta_x, \eta_y, \eta_z \in \mathbb{C}$. Then it's a straightforeward exercise to deduce from figure 32 that, if the error is set to $e = I^{\otimes j-1} \otimes U \otimes I^{\otimes 8-j}$ (that is, applying U to the j-th qubit), then the output will state is

$$\big(\alpha_0 |0\rangle + \alpha_1 |1\rangle\big) \otimes \big(\eta_0 |s_0\rangle + \eta_x |s_{x,j}\rangle + \eta_y |s_{y,j}\rangle + \eta_z |s_{z,j}\rangle\big), \tag{139}$$

where $|s_0\rangle$, $|s_{x,j}\rangle$, $|s_{y,j}\rangle$, $|s_{z,j}\rangle$ are the respective error syndromes of I, X, Y, Z in position j.

More generally, there exist other codes, with m-qubit encodings which have the property there, for some threshold k, the error can be an arbitrary unitray operation acting on any subset of the qubits of size k.

Depolarizing noise

Our discussion of classical error-correcting codes was based on the binary symmetric channel, which flips any bit passing through it with probability ϵ. A quantum analogue of the binary symmetric channel is the depolarizing channel, which can be defined as the mixed unitary channel of the form

$$\begin{cases} I & \text{with prob. } 1-\epsilon & \text{(no error)} \\ X & \text{with prob. } \epsilon/3 & \text{(bit flip)} \\ Y & \text{with prob. } \epsilon/3 & \text{(bit+phase flip)} \\ Z & \text{with prob. } \epsilon/3 & \text{(phase flip)} \end{cases} \tag{140}$$

for a parameter $\epsilon \leq \frac{3}{4}$. This is like the binary symmetric channel, but allowing for the fact that a qubit can be flipped in more than one way (bit-flip, phase-flip, or both).

Exercise 5.1. *Show that the definition of the depolarizing channel in Eq.* (140) *is equivalent to our earlier definition of the depolarizing channel, as mapping each state ρ to a convex combination of that state and the maximally mixed state, namely*

$$p\rho + (1-p)\big(\tfrac{1}{2} |0\rangle\langle 0| + \tfrac{1}{2} |1\rangle\langle 1|\big) \qquad \textit{(for some } p\textit{).} \tag{141}$$

In the depolarizing noise model, we assume that every qubit of the encoded state is independently affected by a depolarizing channel. So all of the qubits incur an error; however, there is a bound ϵ on the severity of that error.

Figure 34: Depolarizing error on every qubit.

How does the Shor code perform in this model? For the depolarizing channel with parameter ϵ, let's think of ϵ as the *effective error probability*, since the qubit passes through the channel is undisturbed with probability $1-\epsilon$. If a qubit is encoded by the Shor code and each of the nine qubits of the encoding incurs a depolarizing error with parameter ϵ then we can analyze the effective error probability resulting from the encoding/decoding process. It turns out that this effective error probability is upper bounded $c\epsilon^2$, for some constant[13] c. So, if ϵ is small enough to begin with, then the reduction due to squaring ϵ is more than the effect of multiplying by c, so the effective error is decreased. The cost is that Alice has to send nine qubits to convey just one qubit. So the rate of this code is $\frac{1}{9}$.

Are there better codes than this? And are there good multi-qubit quantum error-correcting codes? This will be discussed in section 6.

5.5 Redundancy vs. cloning

The Shor code encodes one qubit in state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ into nine qubits, in state $\alpha_0 |0\rangle_L + \alpha_1 |1\rangle_L$ where

$$|0\rangle_L = \left(\tfrac{1}{\sqrt{2}} |000\rangle + \tfrac{1}{\sqrt{2}} |111\rangle\right)^{\otimes 3} \tag{142}$$

$$|1\rangle_L = \left(\tfrac{1}{\sqrt{2}} |000\rangle - \tfrac{1}{\sqrt{2}} |111\rangle\right)^{\otimes 3}. \tag{143}$$

Sometimes, $|0\rangle_L$ and $|1\rangle_L$ are referred to as the *logical zero* and *logical one* of the code.

The encoded state has the property that, if any error is inflicted on any one of its qubits then the data can still be recovered. To be clear, note that the recovery procedure is not provided with any information about *which* qubit has been affected by an error.

Which of the nine qubits contains the data? The answer is that no individual qubit contains any information about the qubit. The information is somehow "spread out" among the nine qubits.

A natural question is whether there is a more efficient code that requires fewer than nine qubits and protects against any one-qubit error. This question will be answered in section 6.

I'd like to briefly mention a different type of error, called an *erasure error*. For this type of error, the positions of the affected qubits are known. One way of viewing this

[13] I don't know the exact constant, but it is less than 36.

is that some of the qubits are "lost," but we know the positions of the lost qubits—and the remaining qubits are undisturbed. For example, suppose that a qubit is encoded via the Shor code into nine qubits and then qubit 2 and qubit 6 go missing.

Figure 35: Erasure error on 2 qubits.

It turns out that the Shor code can handle any two erasure errors. From any seven of the nine encoded qubits, the data can be recovered.

Finally, here's a theorem that's very easy to prove but it helps clarify the distinction between redundancy and copying.

Theorem 5.1 (non-existence of a 4-qubit code protecting against two erasure errors)**.** *There does not exist a 4-qubit code that protects against two erasure errors.*

Proof. Suppose that we had such a code. Then, take the first two qubits and think of them as an encoding with two missing qubits. Same with the last two qubits.

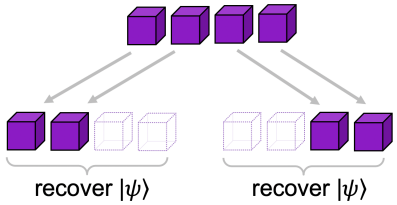

Figure 36: 4-qubit code protecting against two erasure errors violates the no-cloning theorem.

If the code was resilient against two erasure errors then we could recover the data from each of these, thereby producing two copies of the data. This would contradict the no-cloning theorem. □

6 Calderbank-Shor-Steane codes

In this section, we will begin with a quick overview of classical linear error-correcting codes, and then I'll explain how to construct good *quantum* error-correcting codes from certain classical linear codes using a method due to Calderbank, Shor and Steane. These are commonly called CSS codes.

In section 5.2, I stated results about multi-bit classical error-correcting codes, without saying anything about how these codes work. We're going to begin by looking at some of the structure of classical linear codes.

6.1 Classical linear codes

Here we consider certain block codes that encode n-bit data as m-bit codewords.

Definition 6.1 (linear code)**.** *An error-correcting code is* linear *if the the mapping from data to codewords is a linear mappping from* $\{0,1\}^n$ *to* $\{0,1\}^m$*, with respect to the field* $\mathbb{Z}_2$*. In particular, the set of codewords is a linear space.*

The 3-bit repetition code from section 5.1 is a linear code. In particular, the set of its codewords is $\{000, 111\}$, which is a 1-dimensional subspace of $\{0,1\}^3$.

Another example of a linear code is the code given by the table in figure 37.

data	codeword
000	0000000
001	0001111
010	0110011
011	0111100
100	1010101
101	1011010
110	1100110
111	1101001

Figure 37: A 7-bit classical error-correcting code.

This code linearly maps 3-bit strings of data into 7-bit codewords. The codewords are a 3-dimensional subspace of $\{0,1\}^7$. In fact, the three strings 1010101, 0110011, 0001111 (codewords for 001, 010, 100) are a basis for the 3-dimensional subspace. In section 6.3, I'll show you how codes that have this linear structure (and additional properties) can be converted into quantum error-correcting codes in a systematic

way. And the 7-bit code in figure 37 will serve as a running example to illustrate the construction.

Although an error-correcting code is a mapping from $\{0,1\}^n$ to $\{0,1\}^m$, its error-correcting properties can be deduced from properties of it set of codewords, and we frequently refer to a code by its set of codewords.

Definition 6.2 (Hamming distance)**.** *For any two binary m-bit strings, their* Hamming distance *is defined as the number of bit positions in which they are different. That's how many bits you need to flip to convert between the two strings.*

Definition 6.3 (distance of a code)**.** *The* distance of a code *is the minimum Hamming distance between any two codewords. For linear codes, this is equivalent to the minimum distance from any non-zero codeword to the zero codeword.*

What's the minimum distance of the 7-bit code in figure 37? Since it's a linear code, it suffices to check the minimum Hamming weight of all the non-zero codewords, which is 4. So the minimum distance is 4.

The minimum distance of a classical code is closely related to its error-correcting properties.

Theorem 6.1. *If the minimum distance of a code is d then $\lfloor \frac{d-1}{2} \rfloor$ is the number of errors that can be corrected. In other words, as long as the number of bits flipped is strictly less than $\frac{d}{2}$, they can be corrected.*

Proof. First think of the subset of the m-bit strings that are the codewords, and associate with each codeword a neighbourhood that consists of all m-bit strings whose distance from that codeword is strictly less than $\frac{d}{2}$. Figure 38 is a schematic illustration of the codewords (in blue) and their associated neighborhoods (in grey).

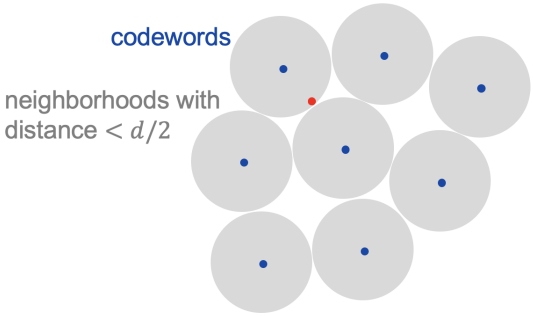

Figure 38: Neighborhoods with associated with associated with codewords.

Note that none of the neighborhoods intersect (because if they did then there would be an m-bit string whose distance from two different codewords is strictly less than $\frac{d}{2}$, violating the fact that the minimum distance is d). If fewer than $\frac{d}{2}$ bits of any codeword are flipped then the perturbed codeword stays within the same neighborhood (e.g., the red point in figure 38). Therefore, there is a unique codeword whose distance is less than $\frac{d}{2}$ from the perturbed codeword. So there is an unambiguous decoding. $\square$

As an example, suppose that, for the code in figure 37, one bit of a codeword is flipped, resulting in the string 10001010. Can you find the unique codeword whose distance is 1 from this string?

6.1.1 Dual of a linear code

You may recall the *dot product* between binary strings $a, b \in \{0,1\}^m$ that we previously saw in the context of Simon's algorithm, which is

$$a \cdot b = a_1 b_1 + a_2 b_2 + \cdots + a_m b_m \bmod 2. \tag{144}$$

Remember that this this is not an inner product because the dot product of a nonzero vector with itself can be zero. But it has some nice properties and we can very loosely think of two strings as being "orthogonal" if their dot product is zero.

Definition 6.4 (dual code)**.** *For any linear code whose codewords are $C \subseteq \{0,1\}^m$, define the dual code as*

$$C^\perp = \big\{ a \in \{0,1\}^m \;\big|\; \text{for all } b \in C,\, a \cdot b = 0 \big\}. \tag{145}$$

The set $C^\perp$ can be loosely thought as all codewords that are orthogonal to C.

The codewords of the 7-bit code in figure 37 are

$$\begin{aligned} C = \big\{ & 0000000, 1010101, 0110011, 1100110, \\ & 0001111, 1011010, 0111100, 1101001 \big\}, \end{aligned} \tag{146}$$

and the dual of that code is

$$\begin{aligned} C^\perp = \big\{ & 0000000, 1010101, 0110011, 1100110, \\ & 0001111, 1011010, 0111100, 1101001, \\ & 1111111, 0101010, 1001100, 0011001, \\ & 1110000, 0100101, 1000011, 0010110 \big\}. \end{aligned} \tag{147}$$

Every vector in C has dot product zero with every vector in $C^\perp$.

Note that $C^\perp$ is a superset of C (it contains all of C, plus additional strings). This can happen because the dot product is not an inner product. What's the minimum distance of $C^\perp$ in this example? The minimum distance of $C^\perp$ is 3. Note that this means that $C^\perp$ can correct against a 1-bit error.

6.1.2 Generator matrix and parity check matrix

For every linear code with n-bit data and m-bit codewords, we associate two matrices.

One matrix is the $n \times m$ *generator matrix*, G, which expresses the encoding process as a linear operator. Namely, for all $a \in \{0,1\}^n$,

$$E(a) = aG. \tag{148}$$

The convention in coding theory is that binary strings are row vectors and the matrix multiplication is on the right side.

For our running 7-bit code example, the generator matrix is

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{149}$$

and to encode a three-bit string $a \in \{0,1\}^3$, we right multiply by the generator matrix

$$E(a) = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix}. \tag{150}$$

It's not hard to see that the rows of G are a basis for the set of codewords.

Another interesting matrix associated with a linear code is called the $m \times (m-n)$ *parity check matrix*, H, which can be used to check whether a given string is a codeword or not. For any string $b \in \{0,1\}^m$, $b \in C$ if and only if $bH = 0^{m-n}$, the zero vector. The columns of H are a basis for $C^\perp$, the dual of C.

For our 7-bit code example, a parity check matrix is

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{151}$$

Notice that the space generated by the rows of G is orthogonal to the space generated by the columns of H. This is expressed succinctly by the fact that $GH = 0^{n\times(m-n)}$, the $n \times (m-n)$ zero matrix.

6.1.3 Error-correcting via parity-check matrix

Let $C \subset \{0,1\}^m$ be a linear code with with distance d and let H be a parity check matrix of C. Here's how to correct errors using H.

For any codeword $b \in \{0,1\}^m$, let b' be the perturbed codeword after an error has been applied. It's useful to think of an m-bit *error vector*, e, that has a 1 in each position where a bit is flipped, and a 0 in the other positions. Then we can write $b' = b + e$, where the vectors are added bitwise (mod 2). If fewer than $\frac{d}{2}$ bits of b are flipped then the Hamming weight of e is less than $\frac{d}{2}$.

Now, consider what happens if we multiply b' by the parity check matrix H

$$b'H = (b+e)H \tag{152}$$
$$= bH + eH \tag{153}$$
$$= eH \tag{154}$$

(where we are using the fact that $bH = 0$ because b is a codeword).

We call bH the *syndrome of the error* e. For linear codes, the syndrome depends only on the error, and not on the codeword on which the error occurred. This property will be especially valuable when we construct quantum error-correcting codes based on classical linear codes.

Referring back to our running example 7-bit code, here's a table of the syndromes of all the error under consideration. That is, where at most one bit is flipped.

e	eH
0000000	0000
1000000	1001
0100000	1010
0010000	1011
0001000	1100
0000100	1101
0000010	1110
0000001	1111

Figure 39: Syndromes associated with errors.

In general, all errors e that are of Hamming weight less than $\frac{d}{2}$ have unique syndromes.

Let's go through am example of an error-correction procedure using the syndrome table in figure 39. Suppose that we receive the string 1001010 from the channel (and we're not told what the error is). Multiplying by the parity check matrix, we obtain $\begin{bmatrix}1\,0\,0\,1\,0\,1\,0\end{bmatrix}H = \begin{bmatrix}1\,0\,1\,1\end{bmatrix}$, so the syndrome of the error is 1011. Looking at the syndrome table, we can see that this syndrome corresponds to the error 0010000 (i.e., the third bit is flipped). So we can flip the third bit again to correct the error, obtaining the original codeword $b = 1011010$. To get the data $a \in \{0,1\}^3$ from the codeword b, we can use the fact that $\begin{bmatrix}a_0\,a_1\,a_2\end{bmatrix}G = b$ (where G is a generator matrix for the code) and solve the system of linear equations to get a.

As an aside, for large m and where d grows as a function of m, the syndrome table can be of size exponential in m. So it is not efficient to explicitly construct the entire table of errors and syndromes. Good error-correcting codes with large block sizes are designed with special additional structural properties which enable the error as a function of the syndrome to be computed efficiently.

All this information about classical linear codes is useful for understanding the methodology of CSS codes.

6.2 $H \otimes H \otimes \cdots \otimes H$ revisited

Before discussing the CSS code construction, I'd like to show you some more nice properties of the m-fold tensor product of Hadamard gates. We've already seen in the notes [*Part 2: Quantum algorithms*, section 6.3.1] that

$$H^{\otimes m} \left|0^m\right\rangle = \tfrac{1}{\sqrt{2^m}} \sum_{b \in \{0,1\}^m} \left|b\right\rangle, \tag{155}$$

and, more generally, for any $w \in \{0,1\}^m$,

$$H^{\otimes m} |w\rangle = \tfrac{1}{\sqrt{2^m}} \sum_{b \in \{0,1\}^m} (-1)^{b \cdot w} |b\rangle \,, \tag{156}$$

where $b \cdot w$ denotes the dot-product $b_0 w_0 + b_1 w_1 + \cdots + b_{m-1} w_{m-1}$.

Now, here's a generalization of the above two equations related to states that are uniform superpositions over the elements of a linear subspace $C \subseteq \{0,1\}^m$. Applying $H^{\otimes m}$ to such a state results in an equally weighted superposition of the elements of $C^\perp$. Namely,

$$H^{\otimes m}\Big(\tfrac{1}{\sqrt{|C|}} \sum_{a \in C} |a\rangle\Big) = \tfrac{1}{\sqrt{|C^\perp|}} \sum_{b \in C^\perp} |b\rangle \,. \tag{157}$$

Also, for a uniform superposition over the elements of a linear subspace $C \subseteq \{0,1\}^m$ offset by some $w \in \{0,1\}^m$, there is a similar expression with phases,

$$H^{\otimes m}\Big(\tfrac{1}{\sqrt{|C|}} \sum_{a \in C} |a + w\rangle\Big) = \tfrac{1}{\sqrt{|C^\perp|}} \sum_{b \in C^\perp} (-1)^{b \cdot w} |b\rangle \,. \tag{158}$$

Notice that Eq. (157) generalizes Eq. (155), since $\{0^m\}$ is the zero-dimensional linear subspace of $\{0,1\}^m$ and $\{0^m\}^\perp = \{0,1\}^m$. Similarly, Eq. (158) generalizes Eq. (156).

Exercise 6.1. *Prove Eqns.* (157) *and* (158).

Hint: if G is the $n \times m$ generator matrix of C then

$$\tfrac{1}{\sqrt{|C|}} \sum_{a \in C} |a\rangle = \tfrac{1}{\sqrt{2^n}} \sum_{b \in \{0,1\}^n} |bG\rangle \,. \tag{159}$$

6.3 CSS codes

A CSS code is based on two classical linear codes, $C_0, C_1 \subseteq \{0,1\}^m$, that are related by these properties:

- $C_0 \subsetneq C_1$
- $C_0^\perp \subseteq C_1$.

Two relevant parameters are $k = \dim(C_1) - \dim(C_0)$ and d, the distance of code C_1. From this, we will construct a quantum error-correcting code that encodes k qubits as m qubits, and protects against any errors in fewer than $\frac{d}{2}$ qubits.

From our running example, we can take

$$\begin{aligned} C_0 = \big\{&0000000,\ 1010101,\ 0110011,\ 1100110,\\ &0001111,\ 1011010,\ 0111100,\ 1101001\big\}, \end{aligned} \tag{160}$$

$$\begin{aligned} C_1 = \big\{&0000000,\ 1010101,\ 0110011,\ 1100110,\\ &0001111,\ 1011010,\ 0111100,\ 1101001,\\ &1111111,\ 0101010,\ 1001100,\ 0011001,\\ &1110000,\ 0100101,\ 1000011,\ 0010110\big\}. \end{aligned} \tag{161}$$

Clearly $C_0 \subsetneq C_1$ and $C_0^\perp = C_1$. For this example, $k = \dim(C_1) - \dim(C_0) = 4 - 3 = 1$, and $d = 3$ (the distance of code C_1).

In general, let G_0 and G_1 be generator matrices for C_0 and C_1, respectively. Suppose that G_0 is an $n \times m$ matrix and G_1 is an $(n+k) \times m$ matrix. We can express

$$G_1 = \left[\frac{G_0}{W}\right], \tag{162}$$

where W is a $k \times m$ matrix representing the additional rows to add to G_0 in order to extend the span from C_0 to C_1.

In our running example, we have

$$G_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{163}$$

$$G_1 = \left[\begin{array}{ccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]. \tag{164}$$

6.3.1 CSS encoding

For linear codes $C_0 \subset C_1 \subseteq \{0,1\}^m$ such that $C_0^\perp \subseteq C_1$, let $k = \dim(C_1) - \dim(C_0)$. The related CSS code encodes a k-qubit state as an m-qubit state as follows.

For all $v \in \{0,1\}^k$, define the *logical basis state* $|v\rangle_L$ as the m-qubit state

$$|v\rangle_L = \tfrac{1}{\sqrt{|C_0|}} \sum_{a \in C_0} |a + vW\rangle\,. \tag{165}$$

Then an arbitrary k-qubit (pure) state

$$\sum_{v\in\{0,1\}^k} \alpha_v |v\rangle \tag{166}$$

is encoded as the m-qubit state

$$\sum_{v\in\{0,1\}^k} \alpha_v |v\rangle_L = \sum_{v\in\{0,1\}^k} \alpha_v \Big(\tfrac{1}{\sqrt{|C_0|}} \sum_{a\in C_0} |a + vW\rangle \Big). \tag{167}$$

Returning to our running example, we have logical qubits

$$\begin{aligned} |0\rangle_L = {} & |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ & |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle \end{aligned} \tag{168}$$

$$\begin{aligned} |1\rangle_L = {} & |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ & |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle . \end{aligned} \tag{169}$$

The state $|0\rangle_L$ is a uniform superposition of the elements of C_0. The state $|1\rangle_L$ is a uniform superposition of the elements of C_0+1111111. Any 1-qubit state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ is encoded as the 7-qubit state $\alpha_0 |0\rangle_L + \alpha_1 |1\rangle_L$. This is called the *Steane code* and we'll show that it protects against any 1-qubit error.

6.3.2 CSS error-correcting

Let $C_0 \subset C_1 \subseteq \{0,1\}^m$ be linear such that $C_0^\perp \subseteq C_1$ and let $k = \dim(C_1) - \dim(C_0)$. We will show how to perform error-correction for the related CSS code, correcting any Pauli error acting on fewer than $\frac{d}{2}$ qubits, where d is the distance of code C_1.

We begin by showing how to correct X-errors acting on fewer than $\frac{d}{2}$ qubits. The encoding of a k-qubit state is of the form of Eq. (167). This state is a superposition of basis states from the larger code C_1, whose minimum distance is d.

We can write the X-error operator as $E_e = X^{e_0} \otimes X^{e_1} \otimes \cdots \otimes X^{e_{m-1}}$, where $e \in \{0,1\}^m$ has Hamming weight less than $\frac{d}{2}$. For encoded data $|\psi\rangle$, the state $E_e |\psi\rangle$ is the encoding subjected to the error.

Based on the parity check matrix H_1 of C_1, we can create a circuit than produces the error syndrome $|S_e\rangle$ in an ancillary register.

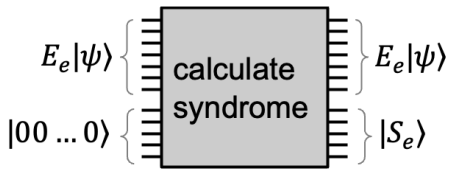

Figure 40: Computing the X-error syndrome.

Since the syndrome depends only on the error vector e and not any computational basis state from C_1, the output of the circuit is the product state $\left(E_e\,|\psi\rangle\right) \otimes |S_e\rangle$.

In our running example, this syndrome is computed using the parity check matrix

$$H_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{170}$$

and, based the entries of H_1, a circuit for producing the X-error syndrome is this.

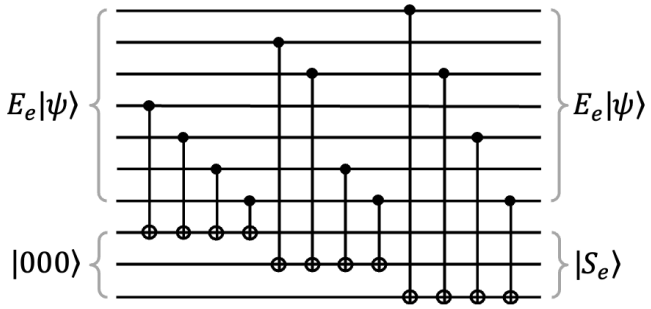

Figure 41: Explicit circuit computing the X-error syndrome for the Steane code.

It's easy to confirm that, for any computational basis state in the code word, this computes the syndrome $|S_e\rangle$ associated with error e. Once the error syndrome S_e has been computed, the error e can be deduced and undone. So this procedure corrects X-errors, as long as there are fewer than $\frac{d}{2}$ of them.

What about Z-errors? To correct against Z-errors, we apply an H operation to each qubit of the encoded data. This converts the encoding to the Hadamard basis, where Z-errors are X-errors.

What does the encoding look like in the Hadamard basis? Using the results from section 6.2, this is

$$H^{\otimes m}\left(\sum_{v\in\{0,1\}^k}\alpha_v\left|v\right\rangle_L\right)=\sum_{v\in\{0,1\}^k}\alpha_v H^{\otimes}\left(\frac{1}{\sqrt{|C_0|}}\sum_{a\in C_0}\left|a+vW\right\rangle\right) \tag{171}$$

$$=\sum_{v\in\{0,1\}^k}\alpha_v\left(\frac{1}{\sqrt{|C_0^\perp|}}\sum_{b\in C_0^\perp}(-1)^{b\cdot(vW)}\left|b\right\rangle\right). \tag{172}$$

Since $C_0^\perp\subseteq C_1$, this state is also a superposition of computational basis states from C_1. Therefore, we can apply the procedure in figure 40 again in the Hadamard basis.

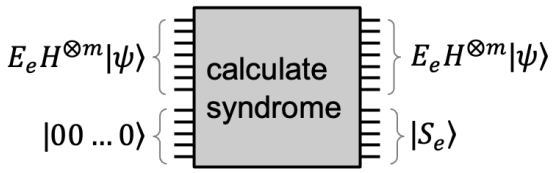

Figure 42: Computing the Z-error syndrome.

From the syndrome, the Z-errors (which are X-errors in the Hadamard basis) can be undone. Then $H^{\otimes m}$ is applied again to return to the computational basis.

So far, we can correct X-errors and Z-errors. Since $Y=iXZ$, each Y-error is like an X-error and a Z-error, and each of those is corrected by the two aforementioned procedures.

6.3.3 CSS code summary

The Steane 7-qubit CSS code and the Shor 9-qubit code both protect against a 1-qubit error; however, note that the Steane code has better rate.

In general, the performance of a CSS code depends on the performance of the classical linear codes $C_0\subset C_1\subseteq\{0,1\}^m$ (with $C_0^\perp\subseteq C_1$) on which it is based. Since good classical codes of the above form exist, there exist qualitatively good quantum error-correctng codes. It turns out that there exists a threshold $\epsilon_0=0.055...$ such that, for the depolarizing channel with error parameter $\epsilon<\epsilon_0$, there exist CSS codes of constant rate and arbitrarily small failure probability. This is qualitatively equivalent to what's achievable with classical error-correcting codes, although the specific constants are smaller.

This gives an idea of what quantum error-correcting codes can accomplish. But this is just the beginning. There are many other quantum error-correcting codes, some of which are better in certain respects than CSS codes. Also, the depolarizing channel is a kind of standard noise model, but it's not the only one. Quantum error-correcting codes that perform well against this model tend to perform well against variations of this error model. And there is some fine-tuning possible if one knows the exact error model.

7 Very brief remarks about fault-tolerance

The error-correcting codes that I've described assume that the noise is restricted to the communication channel. They assume that the encoding and decoding processes are not subject to any noise. What about noise during the execution of a quantum circuit?

Suppose that we want to execute a quantum circuit, but there is noise *during the computation*. One simple way of modeling this is to assume that there is a depolarizing channel at each qubit applied at each time step.

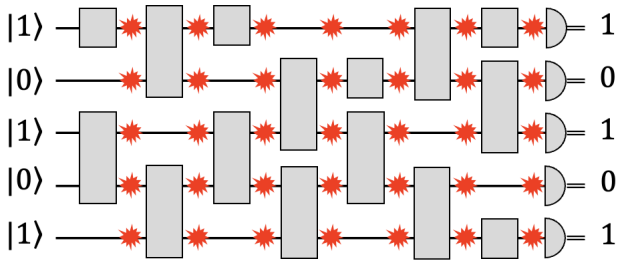

Figure 43: Noisy gates modeled by a depolorizing channel at each qubit at each time step.

How can we cope with this kind of noise? If the error parameter of the depolorizing channel ϵ is very small then this is OK. Suppose that the size of the computation is less than $\frac{1}{10\epsilon}$. Then, with good probability, none of the qubits incurs a flip (by which I mean a bit-flip, phase-flip, or both), so the computation succeeds.

But, if ϵ is a constant (dictated by the precision of our hardware), then size of the largest circuit possible will also be bounded by a constant. If we want to execute a larger circuit then we need a smaller ϵ, which means better precision in our quantum gates. For very large circuits we would need very high precision components.

But we can do much better than this, due to the celebrated *Threshold Theorem*.

Theorem 7.1 (Threshold Theorem, rough statement)**.** *There exists a constant $\epsilon_0 > 0$ such that if the precision per gate per time step is below ϵ_0 then we can perform arbitrarily large computations without having to further increase the precision.*

The rough idea is to convert the circuit that we want to perform into a another circuit that is fault-tolerant. The fault-tolerant circuit uses error-correcting codes in place of each qubit, and performs additional operations that correct errors at regular time intervals, so they don't accumulate. The known fault-tolerant constructions can be quite elaborate, and use several clever ideas. But what's nice is that the fault-tolerant

circuits are not that much larger asymptotically than the original circuits. In some formulations, the size increase is by a logarithmic factor; and in some formulations, by a constant factor.[14]

This result is quite impressive, if you consider that there is noise during any encoding and decoding operation within the fault-tolerant circuit.

[14]The fault tolerant circuit will not be exactly of the form of a larger version of the kind of circuit that appears in Fig. 43. For the details to work out, fresh ancilla qubits must be injected into the circuit at intermediate times and also qubits must also be measured at intermediate times to perform the corrections.

8 Nonlocality

In this section, I will explain a phenomenon called *nonlocality*, which is a strange kind of behavior that quantum systems can exhibit. One way of explaining this behavior is in terms of games played by a team of cooperating players. They players must individually answer certain questions, and they must do this without communicating with each other. The lack of communication appears to restrict what the players can achieve. However, with quantum information, strange behaviors can occur, that defy what one might intuitively think is possible.

8.1 Entanglement and signalling

First of all, let's note that entangled states cannot be used to communicate instantaneously. What I mean by this is the following. Suppose that Alice has a quantum system in her lab and Bob has a quantum system in his lab, and the two labs are in physically separate locations. Alice and Bob's systems might be in an entangled state—for example one of the Bell states.

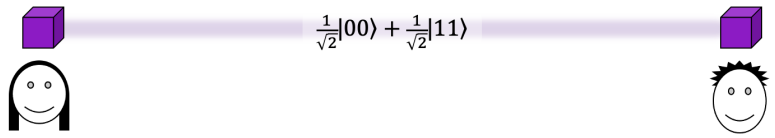

Figure 44: Is there anything Alice can do to *her* qubit that is detectable in Bob's qubit?

Then there is no quantum operation that Alice can perform on her system alone, whose effect is detectable by Bob. If Alice wants to communicate with Bob, she has to send something to him

To see why this is so, consider the density matrix of Bob's system (that is, the density matrix that results when Alice's system is traced out). If Alice performs a unitary operation or measurement on her system then this has no effect on the density matrix of Bob's system. So any kind of measurement that Bob performs on his system will have exactly the same outcome whether or not Alice performs the operation.

And this is not specific to two systems. If there are three or more parties then the same thing holds. Operations on one system have no detectable effect on the other systems.

8.2 GHZ game

Now let's consider a three-player game, commonly referred to as the GHZ game (named after it's inventors, Greenberger, Horne, and Zeillinger).

Let's call the three players Alice, Bob, and Carol.

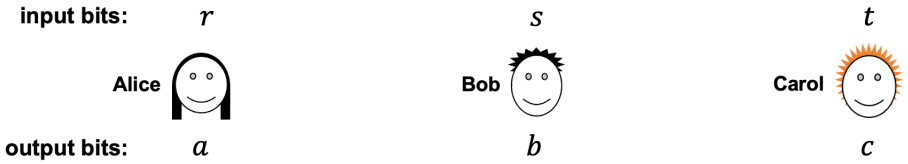

Figure 45: Alice, Bob, and Carol playing the GHZ game.

This game works as follows. Each player receives a 1-bit input. Call the respective input bits r, s, and t. And each player is required to produce a 1-bit output. Call the respective input bits a, b, and c.

The rules of this game are the following.

1. It is promised that the input bits, r, s, and t have an even number of 1s among them. In other words, $r \oplus s \oplus t = 0$. So there are actually three cases of inputs: 000, 011, 101, 110.

2. There is no communication allowed between Alice, Bob, and Carol once the game starts and their input bits are received. So, for example, although Alice will know what her input r is, she does not know what s and t are.

3. This rule defines the *winning conditions* of the output bits. The three players *win* the game if and if $a \oplus b \oplus c = r \vee s \vee t$.

The winning condition $a \oplus b \oplus c = r \vee s \vee t$ is just a condensed way of describing this table.

rst	$a \oplus b \oplus c$
000	0
011	1
101	1
110	1

Figure 46: For each input rst, the required value of $a \oplus b \oplus c$ to win.

For the first case, the XOR of the outputs should be 0 for the players to win. For the other three cases, the XOR of the outputs should be 1 for the players to win.

To get a feeling for this game, here is an example of a strategy that Alice, Bob and Carol could use:

Example of a strategy

Alice receives r as input and produces r as output.

Bob receives s as input and produces $\neg s$ as output.

Carol receives t as input and produces 1 as output.

So how well does this strategy perform? Here I've added the output bits arising from this strategy to the table.

rst	$a \oplus b \oplus c$	abc
000	0	011
011	1	001
101	1	111
110	1	101

Figure 47: Output bits produced by the example strategy (in red).

You can see that the first output bit a is r, the second output bit b is $\neg s$, and the third output bit c is always 1. Consider the first case of inputs 000. There the XOR of the output bits should be 0. And it is 0. So they win in that case. For the second case, the XOR of the output bits should be 1, and it is. So they win in that case too. They also win in the third case. So far, so good. But what happens in the fourth case? In that case, the XOR should be 1. But the output bits have XOR 0. So they lose in that case.

So this is an example of a strategy that wins in three out of the four cases. Is there a better strategy, that wins in all four cases?

8.2.1 Is there a perfect strategy for GHZ?

Call a strategy that wins in every case a *perfect* strategy. Let's see if we can find a perfect strategy for this game.

Alice's output bit is a function of her input bit. Let a_0 denote her output bit if her input bit is 0. And let a_1 denote her output bit if her input bit is 1. Similarly,

define b_0, b_1 as Bob's output bits in the two cases, and c_0, c_1 as Carol's output bits in the two cases. So we have six bits specifying a strategy.

We can express the winning conditions in terms of the four equations

$$a_0 \oplus b_0 \oplus c_0 = 0 \tag{173}$$

$$a_0 \oplus b_1 \oplus c_1 = 1 \tag{174}$$

$$a_1 \oplus b_0 \oplus c_1 = 1 \tag{175}$$

$$a_1 \oplus b_1 \oplus c_0 = 1. \tag{176}$$

The first equation says that, when all three inputs are 0, the XOR of the output bits should be 0. The second equation says that, when the input bits are 011, the XOR of the output bits should be 1. And so on.

So, to find a perfect strategy, we just have to solve this system of equations. Is there a solution?

In fact, there is no solution to this system of equations. These are linear equations in mod 2 arithmetic. Suppose we add the four equations. On the left side we get 0, because each variable appears exactly twice. On the right side, we get the XOR of three 1s, which is 1. So, summing the equations yields $0 = 1$, a contradiction.

It's possible to satisfy any three of the four equations, but not all four. Therefore, there does not exist a perfect deterministic strategy.

I say *deterministic*, because this analysis doesn't consider the case of probabilistic strategies. Could there exist a probabilistic perfect strategy?

There cannot exist a probabilistic perfect strategy either. This is because a probabilistic strategy is essentially a probability distribution over all the deterministic strategies, and the success probability is a weighted average of all the success probabilities of the deterministic strategies (weighted by the probabilities).

If the questions r, s, and t were selected randomly (with probability $\frac{1}{4}$ for each possible input) then the success probability for every deterministic strategy would be at most $\frac{3}{4}$. So the weighted average for any probability distribution on deterministic strategies cannot be higher than that.

Now imagine that you actually carried out this game with Alice, Bob, and Carol. You generate a random triple of questions and check whether their answers win or not. If you just play this once then they might win by luck. In fact, they can win with probability $\frac{3}{4}$. But suppose you play this several rounds in succession and they win every single round?

What if you play four rounds, once for each of the four input possibilities? Will the players necessarily fail in at least one of those rounds? No, not necessarily. Note

that the players might use a different deterministic strategy at each round. Their strategy can satisfy any three of the four equations, and they can arrange to have a different equation violated for each round.

In fact, the player can ensure that they win each round with probability $\frac{3}{4}$. So they would win any four rounds with probability $(\frac{3}{4})^4$, which is slightly more than 30%.

On the other hand, it's highly unlikely that the players would be lucky enough to win, say, 500 rounds. That success probability is $(\frac{3}{4})^{500}$, which less than 1 in a trillion trillion trillion trillion trillion.

So if you did play the game for a large number of rounds—with randomly selected questions at each round—and the players won *every single time*, what would you make of that?

8.2.2 Cheating by communicating

What makes the game non-trivial is that the players cannot communicate with each other. If they could communicate then there would be an easy perfect strategy. For example, suppose that Bob sent his input bit s to Alice. Then Alice knows r and s. She can also deduce t because of the condition that the parity of all three bits is promised to be 0. So Alice knows which of the four cases they're in. A winning strategy is for Alice to output the required parity to win, and Bob and Carol to output 0.

rst	$a \oplus b \oplus c$	abc
000	0	000
011	1	100
101	1	100
110	1	100

Figure 48: Output bits of the cheating-by-communicating strategy (in red).

This wins in all four cases. And there are ways of scrambling up the outputs that obscures this pattern—so that Bob and Carol don't always output 0, and yet the three players always win.

8.2.3 Enforcing no communication

So how can we ensure that they do not communicate? Maybe they each have a very well-concealed transmitter.

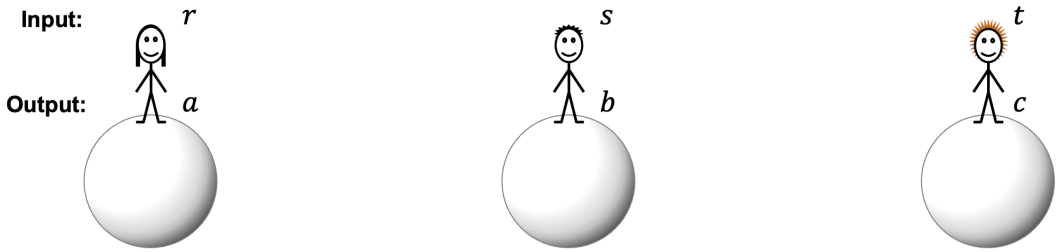

Figure 49: Alice, Bob, and Carol physically far apart.

You could ensure that there's a significant physical distance between the players (here I'm depicting them on separate planets), and also time their inputs and outputs tightly so that they cannot communicate fast enough for messages to reach each other before their deadlines for producing their outputs. For this, we assume that they cannot send signals faster than the speed of light. In physics-terminology we are making the input/output events *space-like separated*.

Then, assuming that the theory of relativity is correct, we can be sure they are not communicating their inputs to each other. But what if this is done and they *still* keep on winning, for every round?

If they players had quantum systems that were entangled, could that possibly help? Recall (as explained in section 8.1) that entanglement does not enable communication. There's no way that Bob can perform an operation on his system that can be detected by Alice. So is there any possible way that Alice, Bob, and Carol could keep on winning this game? If you're not already familiar with scenarios like this then I recommend that pause and think this over. The answer will come at the next page.

8.2.4 The "mystery" explained

The answer is yes, there is a way that they can always win, and I will show you how. They do use entanglement. Let them share the 3-qubit state

$$\tfrac{1}{2}|000\rangle - \tfrac{1}{2}|011\rangle - \tfrac{1}{2}|101\rangle - \tfrac{1}{2}|110\rangle \tag{177}$$

Alice possesses the first qubit, Bob possesses the second qubit and Carol possesses the third qubit.

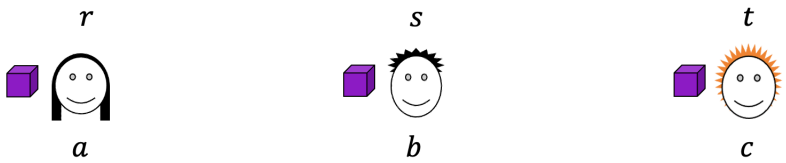

Figure 50: Alice, Bob, and Carol each possess a qubit of a tripartite entangled state.

First, I'll describe the strategy of the players.

Entangled strategy

Alice: if $r = 1$ then apply H; measure and output the result.

Bob: if $s = 1$ then apply H; measure and output the result.

Carol: if $t = 1$ then apply H; measure and output the result.

Now let's see how this strategy performs. There are four cases of inputs.

Let's begin by considering the first case, where $rst = 000$. In that case, neither player applies a Hadamard transform. Therefore, they measure the state in Eq. (177) with respect to the computational basis. The result is

$$\begin{cases} 000 & \text{with prob. } \frac{1}{4} \\ 011 & \text{with prob. } \frac{1}{4} \\ 101 & \text{with prob. } \frac{1}{4} \\ 110 & \text{with prob. } \frac{1}{4}. \end{cases} \tag{178}$$

For all four possibilities, the XOR of the three output bits is 0, which is what it's supposed to be for the 000 case.

Now let's consider the case where $rst = 011$. In that case, Bob and Carol apply Hadamard operations. It's straightforward to check that

$$\begin{aligned}\big(I \otimes H \otimes H\big)\big(\tfrac{1}{2}\,|000\rangle - \tfrac{1}{2}\,|011\rangle - \tfrac{1}{2}\,|101\rangle - \tfrac{1}{2}\,|110\rangle\big) \\ = \tfrac{1}{2}\,|001\rangle + \tfrac{1}{2}\,|010\rangle - \tfrac{1}{2}\,|100\rangle - \tfrac{1}{2}\,|111\rangle\end{aligned} \tag{179}$$

and when this state is measured in the computational basis the result is

$$\begin{cases} 001 & \text{with prob. } \frac{1}{4} \\ 010 & \text{with prob. } \frac{1}{4} \\ 100 & \text{with prob. } \frac{1}{4} \\ 111 & \text{with prob. } \frac{1}{4}. \end{cases} \tag{180}$$

So the XOR of the three output bits is 1, as required for that case.

The cases where $rst = 101$ and 110 are similar to the previous case, due to the symmetry of the state and the strategies. That's how Alice, Bob, and Carol can win with probability 1.

If they play several rounds of the game then they need to possess several copies of the entangled state in Eq. (177), and they consume one copy during each round.

8.2.5 Is the entangled strategy communicating?

So how is this entangled strategy working? Is it somehow communicating? If you look at the outcome distributions for the different cases, you can see that each individual output bit is an unbiased random bit. So the result of Alice's measurement contains absolutely no information about Bob and Carol's inputs. And similarly for the other players. In fact it can be shown that any perfect strategy using entanglement must have the property that each output bit by itself is a random unbiased bit.

Even if we consider pairs of output bits, they are uncorrelated random bits. It's only the *tripartite* correlations among all three output bits that contain information about the inputs.

8.2.6 GHZ conclusions

Let's summarize this GHZ game. It's a game played by a team of three cooperating players who cannot communicate with each other once the game starts. They each receive a bit as their input and are required to produce a bit as their output. There is

a well-defined winning condition for the output bits that depends on what the input bits are.

The following conditions hold:

- Any classical team can succeed with probability at most $\frac{3}{4}$.
- Allowing the players to communicate would enable them to boost their success probability to 1.
- Entanglement cannot be used to communicate.
- Nevertheless, entanglement is another way that the players can boost their success probability to 1. But not by using entanglement to communicate.
- Instead, entanglement enables the measurement outcomes to be correlated in ways that are impossible with classical information.

You might wonder why I showed you a three-player game. Are there two-player games for which the same phenomena occurs? I showed you a three-player game because it's the simplest game that that I'm aware of that illustrates the point.

8.3 Magic square game

Here's an example of a two-player game with a property similar to that of the GHZ game: that there is no perfect classical strategy, whereas there is a perfect strategy using entanglement. It's commonly called the *Magic Square Game* and I will give just a broad overview (without explaining how the entangled strategy works).

A good way of understanding how the Magic Square Game is defined is to first consider the following puzzle. Imagine a 3-by-3 array whose entries are bits.

b_1	b_2	b_3
b_4	b_5	b_6
b_7	b_8	b_9

Figure 51: Are there bits with even parity for each row and odd parity for each column?

Can you find values for the bits such that:

(a) the number of 1s in every row is even; and

(b) the number of 1s in every column is odd?

Please pause to think about how to do this.

You may have come to the realization that it is impossible to do this. Why? Consider the number of 1s among all the nine bits. The row condition implies that there are an odd number of 1s in total. The column condition implies that there are an even number of 1s in in total. This is a contradiction.

Now, keep this in mind, while I describe a two-player game. As with the three-player GHZ game, the players are collaborating and cannot communicate with each other once the game starts. Alice and Bob each receive a trit as input and are each required to return three bits. Think of Alice's input as specifying a row of the array, and think of Bob's input as specifying a column of the array.

Figure 52: Framework for playing the Magic Square Game.

The winning conditions are:

1. Alice's 3-bit output has even parity (think of these as the bits of one of the rows of the array).

2. Bob's 3-bit output has odd parity (think of these as the bits of one of the columns of the array).

3. Alice and Bob's outputs are consistent in the sense that, where Alice's row intersects Bob's column, the bits are the same. (For example, if Alice is queried the second row and Bob is queried the third column then Alice's third bit must be the same as Bob's second bit.)

What can we say about this game?

It turns out that there is no perfect classical strategy for this game. Of the nine possible question pairs, Alice and Bob's strategy must fail for at least one of them. The maximum success probability attainable is $\frac{8}{9}$. The proof of this is based on the fact that there is no way to set the bits of the 3-by-3 array that satisfy the parity conditions.

But there is a perfect entangled strategy that uses two Bell states as entanglement. It's a very interesting strategy, but I won't go into the details of it here.

8.4 Are nonlocal games useful?

So far, you might think that these games are weird curiosities, that have no conceivable application. But these games can be useful for enforcing certain kinds of behavior in cryptographic protocols. A simple example of this involves devices for generating random bits.

Suppose that you purchased such a device that purportedly generates a stream of random bits.

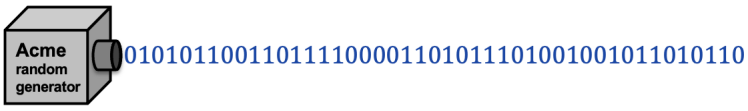

Figure 53: A supposed random number generator. Can you trust it?

How can you know that this is a good device? You could open the box and see if the internal mechanism looks legitimate. But, even if superficially it looks like some process that you think is generating randomness, it's possible that it's a fake random generator, and that the manufacturer is trying to trick you.

What the manufacturer could do is produce a long sequence of random bits in the factory and store those random bits in two memory devices, and hide one of those memory devices somewhere in the box and keep the other memory. And what the device could actually be doing is just outputting the bits stored in the memory device.

Note that the output would look like random bits to you. But the manufacturer would have a copy of those bits. They would know exactly what the next output bit is going to be. If you use such a fake random generator in a cryptographic context, for example to generate a random secret key, that that could be trouble. The manufacturer would know the key.

Unfortunately, in the context of classical information, there is no remedy for this, even in principle. If you don't trust the manufacturer of your devices, then there's no way that you can be sure that it's really generating random bits, that are unknown to the manufacturer.

But, with *quantum* devices it's possible to certify the randomness of untrusted devices. The "device" would actually consists of two components, that contain entangled quantum systems.

Figure 54: Generating random strings using untrusted devices.

You physically separate the two components and input a short random seed into each component. From this, each component outputs a long string of bits. You check if the inputs and outputs satisfy a certain function, called a test. If they pass the test, and if the input/output events are space-like separated, then you know for sure that the outputs are really random bits (within some precision ϵ, for some small $\epsilon > 0$).

I'm not claiming that such a system is practical to implement for wide usage. But it shows that, in principle, it's possible to actually certify randomness using quantum information. Something that's impossible, even in principle, with classical information.

Nonlocal games also have profound implications in the foundations of physics, which will be the topic of the next section.

9 Bell/CHSH inequality

This section is about the Bell inequality in physics, and its violation. The version that we'll consider was discovered shortly after John Bell's ground-breaking paper, and is called the CHSH inequality, after its authors, Clauser, Horne, Shimoney, and Holt.

9.1 Fresh randomness vs. stale randomness

I'd like to begin by making a distinction between "fresh" randomness and "stale" randomness. By *stale randomness*, I mean something like this. Suppose that I flipped a coin yesterday and I know what the outcome was, but I'm not telling you what it was. Then, from *your* perspective, the outcome is a probability distribution. From your perspective, outcome is "heads" with probability $\frac{1}{2}$ and "tails" with probability $\frac{1}{2}$. But the outcome is already determined. Your probabilities just reflect a lack of information on your part.

Contrast this situation with the case where the coin is spinning in the air right at this very moment. In that case, neither of us know the outcome. The outcome has not been determined yet. Let's call that *fresh randomness*.

But is a coin flip really a random process? Isn't the outcome determined by the present conditions? If we knew the exact shape of the coin, it's exact motion, and the positions of all the air molecules *and* we had an extremely powerful computer then maybe we could determine the value of the coin flip while it's spinning in the air.

Moreover, we should not conflate *forecasting* (being able to predict a future event) with *determinism* (a future event being determined by present conditions).

Consider the weather. Predicting, say, the outside temperature where I live one year for now is for all practical purposes impossible due to the chaotic nature of weather (the so-called *butterfly effect*). In terms of forecasting, this temperature is at best a probability distribution (a different distribution in the summer than in the winter). Nevertheless, it seems that the *precise* future weather is determined by the *precise* present conditions, even if we cannot how exactly what these are.

Whether the intuitive notion of fresh randomness actually exists or is just an illusion is an interesting question. I don't claim to know the answer. All of this discussion is a lead-in to the question:

> *If a qubit in state* $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ *is measured in the computational basis, will the outcome be fresh randomness or stale randomness?*

In the quantum information framework that has been the subject of these notes, the outcome is regarded as fresh randomness, that's spontaneously generated during the measurement process. It doesn't really make sense in our model for Alice to produce a qubit in state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and at the same time to know in advance the outcome of a future measurement (in the computational basis) of that state. Or does it?

9.2 Predetermined measurement outcomes of a qubit?

Let's explore the possibility that the outcomes for measuring a qubit in a state like $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ are predetermined. Think of a qubit as a physical entity, a particle (technically, a spin-$\frac{1}{2}$ particle) that was created at the big bang. Imagine that, *at the time of creation*, a predetermined outcome for each possible measurement outcome was embedded into the particle. So lurking within a qubit is some sort of table of predetermined outcomes. Let's visualize it as a literal table of outcome values.

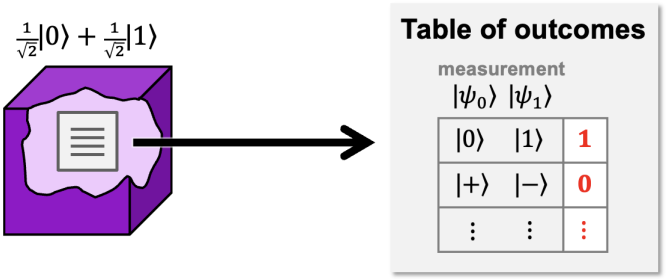

Figure 55: Are predetermined values of all measurement outcomes contained in a qubit?

Imagine that every entry of the table was created randomly so as to conform with the outcome probabilities of quantum measurements. For example, for a measurement in the computational basis, the outcome is an unbiased random bit. So, in that entry of the table, a random bit was inserted (in the figure, it was set to 1). On the other hand, for a measurement in the $|+\rangle/|-\rangle$ basis, the outcome should always be the first state $|+\rangle$. So that entry of the table was set the bit 0. And so on. For every other potential measurement, there is an entry in the table containing a bit that is sampled with the appropriate probability distribution for that measurement of the state. That's an infinitely large table. Maybe there's a compressed way of containing this information, but let's not concern ourselves with that issue. My point is that it's conceivable that the particle contains this table of predetermined measurement outcomes stored within it.

In physics, these are called *hidden variables*. The idea is that these hidden variables represent additional physical properties of systems that are yet to be discovered. When they are discovered, quantum mechanics will be tamed of its randomness. The randomness that arises in quantum theory as it currently exists could merely be a consequence of the fact that we don't know what these hidden variables are. In this way of thinking, a measurement merely extracts a predetermined value from the table of outcomes.

Let's continue developing this model. What happens if we apply a unitary operation to a qubit? This would rearrange the table of outcomes in some systematic way. For example, suppose that we apply a Hadamard transform to the qubit. That would swap the first two bits of the table. This is because, after applying a Hadamard to this $|+\rangle$ state, the state becomes $|0\rangle$, so now a measurement in the computational basis produces 0 for sure. And measuring in the $|+\rangle/|-\rangle$ basis is what produces a random bit. Without going into the details, the effect of any unitary operation can be captured by moving around the entries of the table of outcomes. Unitary operations merely rearrange the stale randomness of the table of outcomes.

And, in this picture, every spin-$\frac{1}{2}$ particle has it's own separate table. If multiple particles are in the $|+\rangle$ state then each one contains an independent random bit for the first entry in its table. This is consistent with what happens when we measure several qubits that are each in the $|+\rangle$ state.

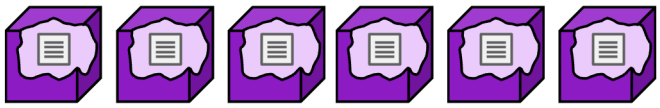

Figure 56: Multiple qubits, each containing a "table" of hidden variables.

What's interesting about this local hidden variables picture is that, so far, everything is consistent with quantum behavior.

We might imagine that this model can be extended to capture all of quantum information theory. For example, when a 2-qubit unitary gate entangles two particles, what might actually be happening is a rearrangement of both tables of outcomes, so that the entries are appropriately correlated for all possible measurements. If the unitary operation creates the state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ then the table entries for measuring each qubit in the computational basis should be: 0 for both particles with probability $\frac{1}{2}$; and 1 for both particles with probability $\frac{1}{2}$.

Let's continue exploring how a local hidden variable model would work.

9.3 CHSH inequality

Imagine a system consisting of two qubits (or two particles), and that there are two measurements, that we'll refer to as M_0 and M_1. We're supposing that each particle contains a full tables of outcomes, but here we'll only care about the parts of the tables that are associated with the measurements M_0 and M_1.

Figure 57: Two particles, with their hidden variables for two measurements, M_0 and M_1.

Call the two predetermined values for the first particle a_0 and a_1. And call the two predetermined values for the second particle b_0 and b_1.

Note that we making an assumption that the hidden variables are *local* hidden variables in the sense that each particle's predetermined outcomes depend only on the measurement performed on that particle, and not the measurements performed on the other particles. What's the justification for this?

The justification is that the particles might be far apart from each other and the timing of the measurements might be such that the two measurement events are space-like separated. This means that there isn't sufficient time for a signal to go from the second particle to the first particle with the information about which measurement is performed. For space-like separated measurement events, the first particle has no way of "knowing" what measurement is being performed on the second particle, even in principle. Therefore, for space-like separated measurements, it's impossible for one particle's outcome to depend on what measurement is performed on the other particle.

I will now describe a property that the bits a_0, a_1, b_0, b_1 must satisfy. It is convenient to describe this property in terms of bits that are expressed as $+1$ and -1

instead of 0 and 1. So we define such a conversion, into "uppercase" bits as

$$A_0 = (-1)^{a_0} \tag{181}$$
$$A_1 = (-1)^{a_1} \tag{182}$$
$$B_0 = (-1)^{b_0} \tag{183}$$
$$B_1 = (-1)^{b_1}. \tag{184}$$

I claim that inequality (185) holds, which is called the CHSH inequality.[15]

Theorem 9.1 (CHSH inequality)**.** *For any $A_0, A_1, B_0, B_1 \in \{+1, -1\}$, it holds that*

$$A_0B_0 + A_0B_1 + A_1B_0 - A_1B_1 \leq 2. \tag{185}$$

Note that each of the four terms on the left side can be $+1$ or -1. So we might imagine that the left side can be as large as 4. But the upper bound is 2.

Proof of Theorem 9.1. Let's see how to prove the CHSH inequality. We can write the left side of Eq. (185) as

$$A_0(B_0 + B_1) + A_1(B_0 - B_1). \tag{186}$$

Now consider the expressions in the parentheses, $B_0 + B_1$ and $B_0 - B_1$. Either B_0 and B_1 have the same sign or they have different signs. If B_0 and B_1 have the same sign then $B_0 + B_1$ can be as large as 2, but then $B_0 - B_1 = 0$. So in that case, the upper bound is 2. If B_0 and B_1 have the different signs then $B_0 - B_1$ can be as large as 2, but then $B_0 + B_1 = 0$. So in that case, the upper bound is also 2. This completes the proof. □

Why should we care about this CHSH inequality? The reason why is that the inequality can be experimentally tested, and if an experiment shows that it's violated then the possibility of a local hidden variable model is refuted.

First of all, let's consider how one could in principle design an experiment to verify that systems satisfy the CHSH inequality. There's some subtlety with this. The problem is that, for any single measurement of the two particles, only one of the four A_sB_t-terms in the inequality can be measured. Here again are the particles

[15] The original inequality along these lines is due to John Bell in 1964. All subsequent variations of it are loosely called *Bell inequalities*. This particularly nice version is due to Clauser, Horne, Shimony, and Holt and is also called the CHSH inequality.

with their outcome tables, where I've taken the liberty of writing the outcomes with uppercase bits (in the ± 1 language).

Figure 58: Two particles, with their hidden variables for M_0 and M_1 specified as $\pm$ bits.

You can choose to perform any single measurement on the first system and get either A_0 or A_1 and then the state is disturbed, so you cannot measure again to get the original value of the other bit. Similarly, you can choose any single measurement on the second system and get B_0 or B_1 (but not both). So you can acquire only one of the four terms A_0B_0, A_0B_1, A_1B_0, A_1B_1 (whose value is $+1$ or -1). To verify the inequality, you would need to see all four terms.

However, the Bell inequality *can* be verified using statistical methods, by making several independent runs, using a separate pair of particles for each run. In each run, $st \in \{00, 01, 10, 11\}$ is chosen randomly and the $\pm$ bit $(-1)^{st}A_sB_t$ is calculated. The factor $(-1)^{st}$ means multiply by -1 in the $st = 11$ case.

If $st \in \{00, 01, 10, 11\}$ is sampled randomly according to the uniform distribution then the *expected value* of $(-1)^{st}A_sB_t$ is

$$\mathrm{E}_{s,t}\big[(-1)^{st}A_sB_t\big] = \tfrac{1}{4}A_0B_0 + \tfrac{1}{4}A_0B_1 + \tfrac{1}{4}A_1B_0 - \tfrac{1}{4}A_1B_1. \tag{187}$$

Does this expression look familiar? It's the left side of the CHSH inequality divided by 4. So we can deduce from the CHSH inequality that

$$\mathrm{E}_{s,t}\big[(-1)^{st}A_sB_t\big] \leq \tfrac{1}{2}. \tag{188}$$

The experiment to statistically verify the CHSH inequality is to make many separate runs, each on a separate pair of particles, of the procedure where you pick a random $st \in \{00, 01, 10, 11\}$ and then measure M_s and M_t to create a sample $(-1)^{st}A_sB_t \in \{+1, -1\}$. If local variables exist then the average over many runs should converge to $\frac{1}{2}$ or less.

Note that, in order to eliminate the possibility of a hidden variable model that is *not local*, the experiment should be implemented so that each pair of measurement

events is space-like separated. If they are not space-like separated then the experiment does not eliminate the possibility that nature is behaving in a conspiratorial way, with signaling between pairs of particles, that permits the outcomes for each particle to depend on both measurements.

The fact that the CHSH inequality can be experimentally verified is remarkable because ...

9.4 Violating the CHSH inequality

... quantum systems can violate the CHSH inequality!

To see how, suppose that the two physically separated qubits are entangled in the Bell state $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$.

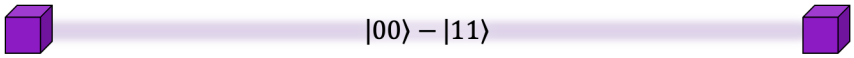

Figure 59: Two physically separated particles in the Bell state $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$.

Consider what happens if a rotation is performed on each qubit, by angle θ_s for the first qubit and θ_t for the second qubit.

Exercise 9.1 (straightforward)**.** *Check that, if $R(\theta_s) \otimes R(\theta_t)$ is applied to state $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$ then the result is*

$$\cos(\theta_s+\theta_t)\big(\tfrac{1}{\sqrt{2}}|00\rangle - \tfrac{1}{\sqrt{2}}|11\rangle\big) + \sin(\theta_s+\theta_t)\big(\tfrac{1}{\sqrt{2}}|01\rangle + \tfrac{1}{\sqrt{2}}|10\rangle\big). \tag{189}$$

If the state in Eq. (189) is measured in the computational basis then the outcome bits ($a_s, b_t \in \{0,1\}$) satisfy

$$\Pr[a_s \oplus b_t = 0] = \cos^2(\theta_s+\theta_t) \tag{190}$$

$$\Pr[a_s \oplus b_t = 1] = \sin^2(\theta_s+\theta_t). \tag{191}$$

It follows that, for the $\pm$ bits $A_s = (-1)^{a_s}$ and $B_t = (-1)^{b_t}$,

$$\begin{aligned}\mathrm{E}[A_s B_t] &= \cos^2(\theta_s+\theta_t) - \sin^2(\theta_s+\theta_t) \\ &= \frac{1+\cos\big(2(\theta_s+\theta_t)\big)}{2} - \frac{1-\cos\big(2(\theta_s+\theta_t)\big)}{2} \\ &= \cos\big(2(\theta_s+\theta_t)\big). \end{aligned} \tag{192}$$

Now define the measurements M_0 and M_1 as follows.

- M_0: rotate by $\theta_0 = -\frac{\pi}{16}$ and then measure in the computational basis.
- M_1: rotate by $\theta_1 = +\frac{3\pi}{16}$ and then measure in the computational basis.

Let's look at the various angles $\theta_s + \theta_t$ that arise for $st \in \{00, 01, 10, 11\}$.

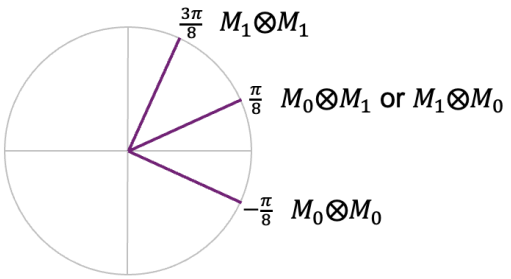

Figure 60: Angles $\theta_s + \theta_t$ that arise for $M_s \otimes M_t$, for the cases $st \in \{00, 01, 10, 11\}$.

When M_0 is performed on both sides, $\theta_0 + \theta_0 = -\frac{\pi}{8}$. When M_0 is performed on one side and M_1 on the other side, $\theta_0 + \theta_1 = \theta_1 + \theta_0 = +\frac{\pi}{8}$. And when M_1 is performed on both sides, $\theta_1 + \theta_1 = \frac{3\pi}{8}$.

Applying Eq. (192), this means that, for measurements M_s and M_t, the $\pm$ outcomes A_s and B_t have the property that

$$\mathrm{E}[A_s B_t] = \begin{cases} \cos(\pm\frac{\pi}{4}) & \text{if } st \in \{00, 01, 10\} \\ \cos(\frac{3\pi}{4}) & \text{if } st = 11 \end{cases} \tag{193}$$

$$= \begin{cases} \frac{1}{\sqrt{2}} & \text{if } st \in \{00, 01, 10\} \\ -\frac{1}{\sqrt{2}} & \text{if } st = 11. \end{cases} \tag{194}$$

It follows that

$$\mathrm{E}[(-1)^{st} A_s B_t] = \tfrac{1}{2}\sqrt{2}, \tag{195}$$

which clearly violates the CHSH inequality—explicitly the upper bound in Eq. (188).

So if we performed the aforementioned experiment of repeatedly picking a random $st \in \{00, 01, 10, 11\}$ and applying the measurements M_s and M_t to a pair of qubits in state $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$ to sample $(-1)^{st} A_s B_t$ then the average would exceed the bound of $\frac{1}{2}$, that was derived under the assumption of local hidden variables. Therefore, in the quantum information framework, local hidden variables cannot exist.

Summary and experimental implementations

Let's summarize the Bell inequality and its violation. Assuming that the measurement outcomes of quantum systems are predetermined by local hidden variables leads to the Bell inequality. But actual quantum quantum systems violate this inequality, by a factor of $\sqrt{2}$. Therefore, quantum systems cannot be based on local hidden variables.

And this behavior of quantum systems has been experimentally verified. The rough idea is to generate two particles in a Bell state and send them out in opposite directions to reach detectors, which are set to measure the particles in various ways.

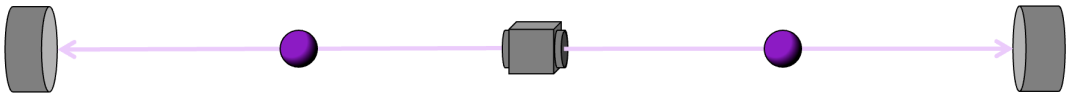

Figure 61: Form of test of the CHSH inequality violation.

In order for such an experiment to be *loophole-free*, the measurement events must be space-like separated; the precision in the state preparation and the measurements must exceed certain thresholds, and the random choices of $st \in \{00, 01, 10, 11\}$ must actually be random. This is non-trivial, but such experiments that are widely regarded as loophole-free have been performed, refuting the existence of local hidden variables.

9.5 Bell/CHSH inequality as a nonlocal game

Now let's look at the Bell/CHSH inequality and its violation in a different way, as a nonlocal game, similar to the ones that we saw in sections 8.2 (the GHZ game) and 8.3 (the Magic Square game).

Define the *CHSH game* as follows. Alice and Bob receive input bits, s and t, and they must produce output bits, a and b.

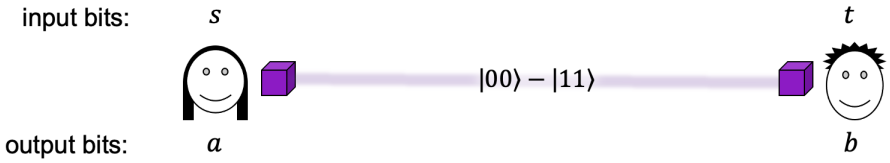

Figure 62: Alice and Bob playing the CHSH game.

The rules of the game are as follows. First, a question pair $st \in \{00, 01, 10, 11\}$ is randomly selected according to the uniform distribution. As usual for nonlocal games,

there is no communication allowed between the players once the game starts. And the players *win* if and only if $a \oplus b = s \wedge t$, which is just a condensed way of specifying the following table.

st	$a \oplus b$
00	0
01	0
10	0
11	1

Figure 63: For each input st, the required value of $a \oplus b$ to win.

What's interesting about the CHSH game is that:

- The maximum winning probability for any *classical* strategy is $\frac{3}{4}$.
- There exists an *entangled* strategy that wins with probability

$$\cos^2\left(\tfrac{\pi}{8}\right) = \tfrac{1}{2}\left(1 + \tfrac{1}{\sqrt{2}}\right) = 0.853... \tag{196}$$

This is essentially the CHSH inequality and its violation, but where the outputs are $\{0, 1\}$-bits instead of $\{+1, -1\}$-bits. The upper bound on the winning probability corresponds to the CHSH inequality (in Eq. (188)), and the quantum strategy that attains success probability $\cos^2\left(\frac{\pi}{8}\right)$ corresponds to the CHSH inequality violation (in section 9.4). However, I will provide a separate analysis of this game.

We can analyze classical strategies for the CHSH game in a manner similar to the way we analyzed the GHZ game in section 8.2.1. First note that any deterministic strategy can be described by four bits, a_0, a_1 (Alice's output bits for the two input possibilities), b_0, b_1 (Bob's output bits for the two input possibilities). And the winning condition can be expressed as the four equations

$$a_0 \oplus b_0 = 0 \tag{197}$$
$$a_0 \oplus b_1 = 0 \tag{198}$$
$$a_1 \oplus b_0 = 0 \tag{199}$$
$$a_1 \oplus b_1 = 1. \tag{200}$$

It's easy to show that at most three of these four equations can be satisfied, so the maximum success probability of any deterministic strategy is $\frac{3}{4}$. Since any classical

probabilistic strategy is essentially a probability distribution on the set of all deterministic strategies, its winning probability cannot be higher than $\frac{3}{4}$.

The entangled strategy for the CHSH game whose probability of winning is $\cos^2(\frac{\pi}{8})$ is very similar to the CHSH inequality violation in section 9.4. Alice and Bob use the entangled state $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$ and they each perform the following on their qubit.

```
1: if input bit is 0 then
2:     apply R(−π/16) to qubit
3: else if input bit is 1 then
4:     apply R(+3π/16) to qubit
5: end if
6: measure qubit and output result
```

Figure 64: Alice and Bob's local behavior based on their input bit.

From Eqns. (189)(190)(191), we can deduce that, for input bits s and t, Alice and Bob's output bits a and b satisfy

$$\Pr[a \otimes b = s \wedge t] = \begin{cases} \cos^2\left(\pm\frac{\pi}{8}\right) & \text{if } st \in \{00, 01, 10\} \\ \sin^2\left(\frac{3\pi}{8}\right) & \text{if } st = 11 \end{cases} \tag{201}$$

$$= \cos^2\left(\tfrac{\pi}{8}\right). \tag{202}$$

Bell inequalities and nonlocal games can be thought of as different ways of expressing the same ideas about nonlocality. One perspective is to consider (and refute) the existence of local hidden variables. Another perspective is to consider communication protocols between separated parties, and what they can accomplish with and without the resource of entanglement.