Quantum Mechanics Quick Reference

Ivan Damgård – version 4, December 2010

This note presents a few basic concepts and notation from quantum mechanics in very condensed form. It is not the intention that you should read all of it immediately. The text is meant as a source from where you can quickly recontruct the meaning of some of the basic concepts, without having to trace through too much material.

1 The Postulates

Quantum mechanics is a mathematical model of the physical world, which is built on a small number of postulates, each of which makes a particular claim on how the model reflects physical reality. All the rest of the theory can be derived from these postulates, but the postulates themselves cannot be proved in the mathematical sense. One has to make experiments and see whether the predictions of the theory are confirmed.

To understand what the postulates say about computation, it is useful to think of how we describe classical computation. One way to do this is to say that a computer can be in one of a number of possible states, represented by all internal registers etc. We then do a number of operations, each of which takes the computer from one state to another, where the sequence of operations is determined by the input. And finally we get a result by observing which state we have arrived in.

In this context, the postulates say that a classical computer is a special case of something more general, namely a quantum computer. More concretely they say how one describes the states a quantum computer can be in, which operations we can do, and finally how we can read out the result. In this respect, the claim of quantum mechanics is that this describes the most general form of computation that nature allows.

2 States

The first postulate is that *states of a physical system correspond to unit vectors in a complex vector space.*

A state is usually written $|\phi\rangle$, where ϕ is the name of the vector. A vector of this form should be thought of as a column vector, i.e., if $|\phi\rangle \in \mathbf{C}^M$, then

$$|\phi\rangle = \begin{pmatrix} a_1 \\ a_2 \\ . \\ . \\ a_M \end{pmatrix}$$

If the above are the coordinates of $|\phi\rangle$ with respect to the basis vectors $|e_1\rangle, ..., |e_M\rangle$, we may of course also write $|\phi\rangle$ as a linear combination of basis vectors:

$$|\phi\rangle = \sum_{i=1}^{M} a_i \, |e_i\rangle$$

In general the dagger operator † means "transpose complex conjugate", so $|\phi\rangle^\dagger$, written as $\langle\phi|$, is a row vector containing the complex conjugates of the coordinates of $|\phi\rangle$:

$$\langle\phi| = (a_1^*, ..., a_M^*)$$

Thus, the matrix product $\langle\phi||\psi\rangle$, written $\langle\phi|\psi\rangle$, is a single number, namely the inner product of $\langle\phi|$ and $|\psi\rangle$. The matrix product $|\psi\rangle\langle\phi|$ is an n by n matrix, and is called the *outer product*.

As a first example of this, think of a 1-bit register in a computer. In a classical computer, only two states are possible, namely 0 and 1. In a quantum computer these are only 2 out of an infinite number of possible states. More precisely, a quantum bit, or a *qubit* is a physical object with states in a 2-dimensional space, the standard basis vectors are called $|0\rangle, |1\rangle$, and should be thought of as the classical states 0 and 1. In general the state of a qubit is $\alpha|0\rangle + \beta|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$. This is also known as a *superposition* of 0 and 1. Such a state "contains" both 0 and 1. One may say that it has not decided yet whether it wants to be 0 or 1. The coefficients α and β measure how much "one-ness" or "zero-ness" the state contains.

Given two objects in states $|\phi\rangle, |\psi\rangle$, say in N, resp. M dimensional spaces V, W, the joint system has state corresponding to a vector $|\phi\rangle \otimes |\psi\rangle$. By definition, $|\phi\rangle \otimes |\psi\rangle$ is a vector with NM coordinates, namely all (ordered) pairwise products of coordinates of $|\phi\rangle$ and $|\psi\rangle$. We say that $|\phi\rangle \otimes |\psi\rangle$ is an element in $V \otimes W$, the tensor product of V and W, which has dimension NM.

Usually, when computing with tensor products, the $\otimes$ symbol is omitted, and we write $|0\rangle \otimes |1\rangle = |0\rangle|1\rangle = |01\rangle$. Example: Given two qubits is states $\alpha|0\rangle + \beta|1\rangle$, $\gamma|0\rangle + \delta|1\rangle$, the joint system is in state

$$(\alpha \, |0\rangle + \beta \, |1\rangle) \otimes (\gamma \, |0\rangle + \delta \, |1\rangle) = \alpha\gamma \, |00\rangle + \alpha\delta \, |01\rangle + \beta\gamma \, |10\rangle + \beta\delta \, |11\rangle$$

This also exemplifies the rules for computing with tensor products: the tensor product behaves from an algebraic point of view exactly like multiplication, which is also why the $\otimes$ symbol is often omitted. Hence one is always allowed to "multiply out" the parentheses following normal rules of algebra, and having done this, one is allowed to manipulate single terms as follows: $\alpha|0\rangle\gamma|0\rangle = \alpha\gamma|0\rangle|0\rangle = \alpha\gamma|00\rangle$.

In general, suppose we have a computer containing n qubits. The state of these can be described in a vectorspace of dimension $N = 2^n$. The standard basis vectors are the tensor products of all classical states of each of the n qbits. Put another way: n bits can be in 2^n distinct classical states, and these therefore form the basis of the space in which the state of n quantum bits lives. In standard notation the basis vectors are

$|00..0\rangle, |00...1\rangle, ..., |11...1\rangle$. This is sometimes also written $|0\rangle, |1\rangle, |2\rangle, ..., |N-1\rangle$, where $|i\rangle$ is the basis vector corresponding to the bit string that is i in binary notation.

A state in a tensor product space $V \otimes W$ that can be written as $|\phi\rangle \otimes |\psi\rangle$ is called a *product state*. Product states are in general states where one part of the system is "independent" of the other. For instance, if the system is in state $|\phi\rangle \otimes |\psi\rangle$, then the first part of the system remains in state $|\phi\rangle$ no matter what happens to the pother part. But it is important to understand that $V \otimes W$ contains many vectors that are not product states. If a state is not a product state, it is called *entangled*. For instance, two qubits may be in the state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, and this state is entangled.

The intuitive reason for this name is the following: the state above describes a system that contains an equal amount of 00 and 11. There is nothing referring to 01 or 10. So if we use the interpretation above of a superposition, namely that the system has not yet decided what it wants to be, it is clear that these two qubits can later decide to be both 0 or both 1, but they can never decide to be 01 or 10. This means that if we measure the first qubit (see details on measurements below), and get 0 as result, for instance, the other qubit must also be in state 0. In this way, something we do to the first qubit instantly affects the other, and this is why such states are called entangled.

Two qubits in state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ is called an EPR pair (named after Einstein, Podolsky and Rosen). In fact, Einstein came up with this example state to show that quantum mechanics was wrong: the point is that quantum mechanics predicts that measuring one particle instantly affects the other, *no matter how far apart they are*. Einstein believed that this would violate the principle from relativity theory, that information cannot travel faster than the speed of light. In fact, there is no contradiction, although the reason for this is rather subtle: measuring one qubit will produce a random bit: 0 or 1 with probability 1/2, and there is no way to bias the result in any particular direction. Transferring information means we can get a bit *given as input* and somehow send it to the other side. If we wanted to use EPR pairs to send information, we would have to make the measurement result depend on the input bit, and this is not possible.

3 State Changes

The second postulate of quantum mechanics is that *a system evolves from one state to another by being subjected to a unitary linear mapping.*

A matrix is unitary if it is a square matrix, every column is a unit vector and the columns are pairwise orthogonal. A linear mapping is unitary of its matrix is unitary. A unitary matrix U is invertible and the inverse is $U^\dagger$. The postulate claims that any such matrix correspond to a physical process that could in principle take place, and any physical process corresponds to such a matrix. Concretely, if we have system in state $|\psi\rangle$ and subject it to a physical process corresponding to a linear mapping with matrix U, then the new state is $U|\psi\rangle$.

This means that quantum computation is and must be reversible: if we start our computer in some state $|\psi\rangle$, do some operations and arrive in some new state, $U|\psi\rangle$, it is

always possible to "undo" this and recreate the state $|\psi\rangle$, namely by doing some operations corresponding to $U^\dagger$. We cannot be sure, however, that we can do so *efficiently*.

The standard operator or matrix that computes a classical function $f : \{0,1\}^n \to \{0,1\}^m$ is called U_f and is defined as follows: it operates on a space of dimension $2^n 2^m$. We write basis vectors in this space as $|x, y\rangle$ where $x \in \{0,1\}^n, y \in \{0,1\}^m$. U_f is described by its action on the basis vectors: it sends $|x, y\rangle$ to $|x, f(x) \oplus y\rangle$. This is just a permutation of the basis vectors and so is certainly a unitary operation.

Note that we could not have defined U_f by requiring that it sends $|x\rangle$ to $|f(x)\rangle$: if f is not injective, the corresponding linear mapping will not be unitary.

4 Measurements

4.1 Projective Measurements

We first describe so called *projective* measurements which will be sufficient for quantum computing. A measurement on an object in vectorspace V is specified by a set of subspaces $W_1, ..., W_t$ with the property that they are pairwise orthogonal and together, they span the entire space V. In other words, V is the direct sum of the W_i's. A measurement can be thought of as asking a system in state $|\phi\rangle \in V$ which subspace it is in. Although the system may not be in any of the subspaces initially, the measurement forces it to choose one of the subspaces, as follows:

Let P_j be the linear mapping that projects V on the subspace W_j. The measurement specified by $W_1, ..., W_t$ returns a result in $\{1, 2, .., t\}$. To find the probability that result is i is retured, we look at the projection $P_i|\phi\rangle$ on the corresponding subspace, and the probability is the length squared of the projection. In other words, it is the inner product of this projection with itself:

$$\langle\phi|P_i^\dagger P_i\,|\phi\rangle = \langle\phi|P_i\,|\phi\rangle.$$

Here, the last equality follows because P_i is a projection, so $P_i^2 = P_i$ and $P_i^\dagger = P_i$. Since $|\phi\rangle$ is always a unit vector, the probability for all outcomes sum to 1, as they should.

The state after measurement is always decided by the rule: "it's in the state you measured it to be in". Concretely, this means that if the system said it was in subspace W_j, it really is there, in other words the state is basically the projection $P_i|\phi\rangle$. However, a state must have length 1, so the state actually is

$$\frac{1}{\sqrt{\langle\phi|P_i|\phi\rangle}} P_i\,|\phi\rangle,$$

i.e., we normalize by dividing by the length of the projection.

Measurements are the only non-reversible operations in quantum mechanics: after measurement, information about the state before measurement is lost.

The *standard measurement* in quantum computing is to decide on a basis $v_1, ..., v_N$ of the space, and let the subspaces for the measurement be $W_1, ..., W_N$ where W_i is the

1-dimensional space spanned by v_i. Measuring a state $|\psi\rangle$ can be described by writing its coordinates in the basis $v_1, ..., v_N$:

$$|\phi\rangle = \begin{pmatrix} a_1 \\ a_2 \\ . \\ . \\ a_N \end{pmatrix}$$

The *measurement in basis* $v_1, ..., v_N$ returns a result in $\{1, .., N\}$, result i is returned with probability $|a_i|^2$, and in this case, the state after measurement is $\frac{a_i}{|a_i|}v_i$.

With a quantum computer, we usually end the computation by a measurement *in the computational basis*. This just refers to the fact that the state of a quantum computer is the state of, say n qbits, and the standard basis for this space corresponds to the 2^n classical states of n bits, as described above. This is called the computational basis. In this case, the i'th basis vector will often be called $|i\rangle$, corresponding to the bit string that is i in binary notation. The state we have is then in ket-notation

$$|\phi\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$$

and the measurement in the computational basis will return result i with probability $|a_i|^2$.

4.2 When you measure only part of a system

Suppose you have a state of n qubits, and you are interested in measuring only the first one of them. A first observation about this is the principle you might call *At the end, you may as well measure everything*. Namely, if your state is the final result of some quantum computation, you may as well assume that all the qubits are measured: if you look only at the first bit of the classical result, the distribution you will see is the same as if you had measured only the first bit.

We will prove this for $n = 2$ (only for simplicity, the generalization to any n is straightforward). Let the state be $|\Psi\rangle = \sum b_1, b_2 \in \{0, 1\}\alpha_{b_1 b_2}|b_1, b_2\rangle$.

Measuring the first bit can described as a projective measurement asking the system if it is in the subspace V_0 spanned by $|00\rangle, |01\rangle$ or in the subspace V_1 spanned by $|10\rangle, |11\rangle$. This so since states in $V_0(V_1)$ are superpositions of exactly those classical possibilities where the first bit is 0(1). The projection of $|\Psi\rangle$ to V_0 is a vector with coordinates $\alpha_{00}, \alpha_{01}, 0, 0$ so the square of its length is $|\alpha_{00}|^2 + |\alpha_{01}|^2$, so this is the probability that we measure a 0 as result. Similarly, we measure 1 as result with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$.

If instead we measure both bits, we will, by the general rule in the previous subsection, get 00 with probability $|\alpha_{00}|^2$, 01 with probability $|\alpha_{01}|^2$, 10 with probability $|\alpha_{10}|^2$ and 11 with probability $|\alpha_{11}|^2$. From this it is clear that the first bit of the result is 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$, i.e., the same as in the previous experiment.

There is also a different way to describe what happens if you measure only one bit, a way that is more convenient if your state is not the final result, and if you want to describe what happens if you compute some more on the remaining bits.

So let us assume that we have a state $|\Phi\rangle$ for n qubits. We can always write $|\Phi\rangle$ as:

$$|\Phi\rangle = \sum_{(x_1,...,x_n)\in\{0,1\}^n} \alpha_{x_1,...,x_n} |x_1, ..., x_n\rangle$$

Splitting this sum in two according to the value of the first bit, we get

$$|\Phi\rangle = \sum_{(x_2,...,x_n)\in\{0,1\}^{n-1}} \alpha_{0,x_2,...,x_n} |0\rangle|x_2, ..., x_n\rangle + \sum_{(x_2,...,x_n)\in\{0,1\}^{n-1}} \alpha_{1,x_2,...,x_n} |1\rangle|x_2, ..., x_n\rangle$$

From this it follows that we can also write

$$|\Phi\rangle = \alpha |0\rangle|\Phi_0\rangle + \beta |1\rangle|\Phi_1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$ and where $|\Phi_0\rangle, |\Phi_1\rangle$ are both legal states for $n-1$ qubits, i.e., they are both unit vectors. From this, we can directly read off the effect of measuring the first bit: we will get result 0 with probability $|\alpha|^2$ and then the remaining bits will be in state $|\Phi_0\rangle$ – and we get result 1 with probability $|\beta|^2$ and then the remaining bits will be in state $|\Phi_1\rangle$.

This is so, since the measurement splits the space in two subspaces, according to the value of the first bit. And furthermore, the projections of $|\Phi\rangle$ on these two subspaces are $\alpha|0\rangle|\Phi_0\rangle$, respectively $\beta|1\rangle|\Phi_1\rangle$.

This rewriting of $|\Phi\rangle$ can be used to show the *Principle of deferred measurement*: Suppose a qubit in a circuit is measured before the computation is over, and the resulting classical bit is used to control if a subsequent gate U is executed or not. Then an equivalent circuit is obtained by pushing the measurement to the end and replacing the classically controlled U by a quantum Control-U operation. This is essentially what exercise 4.35 in Nielsen and Chuang says. By letting U be the identity, this covers also the simpler case where the measured bit is not used to control anything.

4.3 General Measurements

The most general measurements allowed by quantum mechanics can be seen as a generalization of the projective ones. This is easy to see by rephrasing slightly the way a projective measurement is defined: to specify a projective measurement, we could just specify the projection operators $P_1, ..., P_t$, say by giving their matrices in some basis (this uniquely defines the subspaces we talked about before). The fact that the subspaces are pairwise orthogonal and span the entire space is equivalent to saying that $1 = \sum_i P_i$ which is the same as

$$1 = \sum_i P_i^\dagger P_i$$

- again because $P_i^2 = P_i$ and $P_i^\dagger = P_i$. A general quantum measurement is now simply defined by a set of mappings that are not necessarily projections, but are only required to satisfy the above equation.

To be more precise, the third postulate of quantum mechanics says that *The most general measurement possible is defined as follows*: we need a set of mappings $\mathcal{M} = \{M_1, ..., M_t\}$ satisfying the *completeness condition:*

$$1 = \sum_i M_i^\dagger M_i$$

Measuring state $|\phi\rangle$ with measurement $\mathcal{M}$ produces one of $1, 2, ..., t$ as outcome, the probability of outcome i is

$$p(i) = \langle\phi| M_i^\dagger M_i |\phi\rangle$$

It is easy to see that the completeness condition implies $\sum_i p(i) = 1$. If the outcome was i, then the state after measurement is

$$\frac{1}{\sqrt{\langle\phi| M_i^\dagger M_i |\phi\rangle}} M_i |\phi\rangle.$$

Because the M_i's do not have to be projections, a general measurement cannot be directly interpreted as "asking the system which subspace it is in". However, any measurement can be implemented "almost" as a projective measurement, as follows: to measure state $|\phi\rangle$, we prepare another known and fixed state $|\psi\rangle$, a so called *ancilla*, and then perform a projective measurement on $|\phi\rangle \otimes |\psi\rangle$. By choosing appropriately the state $|\psi\rangle$ and the projections, this can be made to produce exactly the same effect as any given general measurement $\mathcal{M}$.

If we are only interested in the probabilities of the different outcomes of a measurement, it sufficient to specify only the mappings $E_i = M_i^\dagger M_i$ and forget about the M_i's. The set of mappings $E_1, ..., E_t$ is called a POVM (Positive Operator Value Measurement). The reason for the name is that the E_i's by construction are guaranteed to be positive operators. Positive operators are linear operators with real, non-negative eigenvalues, see the section below for details. It turns out that *any* set of positive operators $E_1, ..., E_t$ with $\sum_i E_i = 1$ corresponds to some measurement in the way specified here.

5 Normal Operators, the Trace Function and Density Matrices

A vector $|v\rangle$is an *eigenvector* for the linear operator A if $A|v\rangle = \lambda|v\rangle$ for some scalar λ, which is called an *eigenvalue*.

An operator A is *normal* if $AA^\dagger = A^\dagger A$. The Spectral decomposition theorem says that A is normal if and only if you can choose an orthonormal basis such that the matrix of A written in this basis is diagonal. Put another way, there exists a basis consisting of eigenvectors of A, and if you write A in this basis, the eigenvalues of A will appear on the diagonal. For a proof see N&C, page 72.

Some special types of normal matrices: A is *Hermitian* if $A = A^\dagger$, in this case all eigenvalues of of A are real. A is *positive* if all eigenvalues of A are non-negative. Note that unitary operators are also normal, and some are even both Hermitian and Unitary, such as the X, H and Z gates.

The trace function tr applies to a matrix and is simply the sum of all elements on the diagonal. It holds in general for matrices A, B and scalars a, b that $tr(aA + bB) = a \cdot tr(A) + b \cdot tr(B)$ and $tr(AB) = tr(BA)$. From this also follows that the trace is basis invariant, namely for a unitary matrix U, we have $tr(U^\dagger AU) = tr(UU^\dagger A) = tr(A)$, so if we think of U as transforming to coordinates in a different basis, this shows that if we write A in another basis, the trace is the same. Therefore the trace can be thought of as a property of a *linear operator*, not just a property of a matrix.

Thinking of states as being unit vectors is some space is sufficient for quantum computing, but is not sufficient is all scenarios, in particular cases that include communication or cryptography. Suppose A is about to send a state to B. Suppose she does this by choosing which state to send among a set of states $|\psi_1\rangle, ..., |\psi_t\rangle$, such that with probability p_i the state $|\psi_i\rangle$ is chosen. How can we describe the object A sends? certainly not by single unit vector. We could of course simply specify the set of pairs $\{p_i, |\psi_i\rangle\}$, this is called an *ensemble*. But it turns out that there is a more compact and convenient way to do it, namely by giving the so called *density matrix*. For the ensemble $\{p_i, |\psi_i\rangle\}$, the density matrix is

$$\rho = \sum_i p_i \, |\psi_i\rangle\langle\psi_i|$$

The motivation for the density matrix approach is as follows: the only way B can infer any information on the state he has been sent is to preform some measurement on it. It turns out that the behavior of any measurement B could possibly do, can be predicted given only the density matrix. Namely, given measurement $\mathcal{M} = \{M_1, ..., M_t\}$, and density matrix ρ, the probability of outcome i is

$$p(i) = tr(M_i^\dagger M_i \rho)$$

(where tr is the trace operator), and after the measurement, if the outcome was i, we will have an ensemble with density matrix

$$\rho_i = \frac{1}{tr(M_i^\dagger M_i \rho)} M_i \rho M_i^\dagger$$

This is proved in Nielsen and Chuang p.99-100. It is very important to understand that different ensembles may have the same density matrix. This means that even though A may prepare a state to send in several different ways (resulting in different ensembles), if these different methods result in the same density matrix, B will *never* be able to tell the difference, since any measurement he can perform will behave the same way, as long the density matrix is the same. It therefore makes good sense to say that the state A sends *is the density matrix*. In other words, we may think of density matrices as a generalization of the state concept we have seen earlier.

In general, any matrix that is positive and has trace 1 is a valid state in this sense, i.e., there is an ensemble that would result in this matrix being produced.

An example: if A sends the ensemble $\{(\frac{1}{2}, |0\rangle), (\frac{1}{2}, |1\rangle)\}$, i.e. he flips a fair coin to decide whether to send $|0\rangle$ or $|1\rangle$, the density matrix can directly be computed using the definition to be $\frac{1}{2} \cdot I$, i.e., one half times the identity matrix. Suppose instead A decides at random to send either $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, i.e., the ensemble is $\{(\frac{1}{2}, |+\rangle), (\frac{1}{2}, |-\rangle)\}$. Again by direct computation one can verify that the same density matrix results.

In general, two ensembles $\{p_i, |\psi_i\rangle\}$, $\{q_i, |\phi_i\rangle\}$ are equivalent, i.e., they have the same density matrix if and only if there exists a *unitary* matrix with entries u_{ij} such that for every i

$$\sqrt{p_i}\,|\psi_i\rangle = \sum_j u_{ij}\sqrt{q_j}\,|\phi_j\rangle$$

If the underlying ensemble is such that some $p_i = 1$, i.e., A sends a particular $|\psi_i\rangle$ all the time, we say we have a *pure* state, and everything reduces essentially to the unit vector formalism we have seen earlier, namely the density matrix will then be $|\psi_i\rangle\langle\psi_i|$, but this matrix uniquely defines the vector $|\psi_i\rangle$ and vice versa. In the following, we will talk about a state as being a density matrix or a vector, choosing whatever is more convenient (although of course we will have to use density matrices for non-pure states). This correspondence can also be used to see what happens if a receiver B decides to do some computation on the state ρ he has received, that is, to apply a unitary transform U to it. Here, the rule is that:

After applying U to a system in state ρ, the new state is $U\rho U^\dagger$.

The reason why this is the only reasonable way to define how U acts on a density matrix is that if the density matrix was actually a pure state $|\psi\rangle$, that is, $\rho = |\psi\rangle\langle\psi|$, then the state after operating with U should of course be $U|\psi\rangle$ which when written as a density matrix becomes $U|\psi\rangle\langle\psi|U^\dagger = U\rho U^\dagger$.

It may be helpful to think of density matrices as the quantum analogue of a classical phenomenon: a classical probabilistic algorithm is basically a computer program that makes random choices underway. As a result, even if the input is fixed, the output is not any particular value. Instead, we can say that the output is a probability distribution over possible output values. Clearly, there may be many different algorithms that will (on this particular input) produce the same output distribution. If some receiver is given the output value, he has no chance whatsoever of determining which of these algorithms were actually used.

One may now think of the classical output distribution as corresponding to the density matrix in the quantum case, and the different classical algorithms correspond to different ways of preparing the same density matrix.

There is an even more compelling reason why density matrices correspond to probability distributions: since a density matrix is normal, it can be diagonalized, i.e., there is an orthonormal basis in which it is diagonal. Since it is also positive and has trace 1,

we see that the diagonal written in this basis defines a probability distribution. This also shows that there is a standard, so called canonical way to prepare an ensemble resulting in a given density matrix ρ, namely use the eigenvectors forming the basis in which ρ is diagonal and use the probabilities appearing on the diagonal.

6 When you only have access to a part of the system

6.1 The partial trace

Consider the familiar EPR state $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, which is a state for two qbits. Suppose A prepares such a state and sends one of the qbits to B. Which state is B's qbit in? Seen, of course, from B's point of view, who only has access to the one particle A sends him.

To answer this, we will look at a more general setting: suppose Alice holds a part of the system living in state space A, Bob holds a part living in state space B, and that the two systems together is in (possibly mixed) state ρ^{AB}. So this means that ρ^{AB} is a linear operator acting $A \otimes B$. We will now define a linear mapping Tr_A which maps operators on $A \otimes B$ to operators on B, constructed in such a way that if we compute $\rho^B = Tr_A(\rho^{AB})$, we can meaningfully claim that ρ^B is the state held by Bob.

To define this mapping consider first arbitrary $|a_1\rangle, |a_2\rangle \in A$ and $|b_1\rangle, |b_2\rangle \in B$. Then $|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$ is an operator on $A \otimes B$. Furthermore, any operator on $A \otimes B$ can be written as a linear combination of terms of form $|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$. For instance, let $|a_1\rangle, |a_2\rangle$, respectively $|b_1\rangle, |b_2\rangle$ vary over all standard basis vectors of A, respectively B. We then define

$$Tr_A(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = tr(|a_1\rangle\langle a_2|)\, |b_1\rangle\langle b_2|$$

and extend linearly to any operator on $A \otimes B$. Here tr is the ordinary trace. Tr_A is called the *partial trace*, which should be self-explanatory from the definition, and when we compute Tr_A, we say that we "trace out A".

It is useful to notice two facts:

- $(|a_1\rangle \otimes |b_1\rangle)(\langle a_2| \otimes \langle b_2|) = |a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$
- $tr(|a_1\rangle\langle a_2|) = \langle a_2|a_1\rangle$

Plugging this into the definition, we get the "rule of thumb" for computing the partial trace:

$$Tr_A(|a_1b_1\rangle\langle a_2b_2|) = Tr_A((|a_1\rangle \otimes |b_1\rangle)(\langle a_2| \otimes \langle b_2|)) = \langle a_2|a_1\rangle \cdot |b_1\rangle\langle b_2|$$

As an example, let us apply this to the EPR state $|\phi\rangle$. To apply the rule we must write the state in the form of a density matrix, so we think of it as the matrix $|\phi\rangle\langle\phi|$ (but it's the same state anyway!). Plugging in what ϕ actually is, we get that

$$|\phi\rangle\langle\phi| = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$$

The rule of thumb now tells us that

$$\begin{aligned} Tr_A(|\phi\rangle\langle\phi|) &= \frac{1}{2}(Tr_A(|00\rangle\langle 00|) + Tr_A(|00\rangle\langle 11|) + Tr_A(|11\rangle\langle 00|) + Tr_A(|11\rangle\langle 11|)) \\ &= \frac{1}{2}(\langle 0|0\rangle|0\rangle\langle 0| + \langle 1|0\rangle|0\rangle\langle 1| + \langle 0|1\rangle|1\rangle\langle 0| + \langle 1|1\rangle|1\rangle\langle 1|) \\ &= \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \\ &= \frac{1}{2}I \end{aligned}$$

Note that this is the *same density matrix* or state, that B would see, if A had instead flipped a fair coin and had sent to B either $|0\rangle$ or $|1\rangle$ according to the outcome. By what we said earlier on density matrices, there is no way B can distinguish how A really prepared the state to send.

Why the Partial Trace? The way in which the partial trace is defined is by no means arbitrary. It is, in fact, the only way that makes sense. This is shown on page 107 of N&C. The idea is as follows: suppose the state ρ^{AB} in question lives in a space $A \otimes B$ and Bob does a measurement on his part of the state. We can of course think of this as a measurement that is performed on the entire state ρ^{AB}, although it must be of special form, namely the measurement operators M_j must be of form $I \otimes M'_j$ where M'_j acts on B. From the postulates, one can compute the distribution $\mathcal{D}$ that the measurement result must have.

However, on the other hand, if we want a way to compute from ρ^{AB} a matrix ρ_B describing what Bob's state is then ρ_B should at least satisfy that doing the measurement defined by the M'_j's on ρ^B produces the correct result distribution $\mathcal{D}$. It turns out that that this demand already implies that $\rho_B = Tr_A(\rho^{AB})$[1].

The Partial Inner Product There is yet another way to express what the partial trace does. This is based on something called the partial inner product: If we have $|a_1\rangle, |a_2\rangle \in A$ and $|b_1\rangle, |b_2\rangle, |e\rangle, |e'\rangle \in B$, then we define

$$\langle e|(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|)\,|e'\rangle = |a_1\rangle\langle a_2|\langle e|b_1\rangle\langle b_2|e'\rangle$$

This has a bit the same flavor as the partial trace, where this time we trace out B: we take something that lives in $A \otimes B$ and produce something that lives only in A. This is no coincidence - if we let $|e_1\rangle, ..., |e_N\rangle$ be a basis of B, then

$$\sum_i \langle e_i|(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|)|e_i\rangle = \sum_i |a_1\rangle\langle a_2|\langle e_i|b_1\rangle\langle b_2|e_i\rangle \quad (1)$$

$$= |a_1\rangle\langle a_2| \sum_i \langle e_i|b_1\rangle\langle b_2|e_i\rangle \quad (2)$$

$$= |a_1\rangle\langle a_2| \cdot tr(|b_1\rangle\langle b_2|) \quad (3)$$

[1] The proof in N&C is done using the concept of an observable, which we do not need to consider explicitly at this point. Observables are defined in N&C on p. 87

In other words, we have in fact computed $Tr_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|)$.

We need to note for later that the partial inner product is well defined for any operator (matrix) M that acts on $A \otimes B$, i.e., we claim that writing something like $\langle e|M|e'\rangle$ makes good sense. This is simply because $A \otimes B$ has a basis where all basis vectors are of form $|a\rangle \otimes |b\rangle$. In such a basis, as also discussed above, it is not hard to see that M can always be written as a linear combination of matrices of form $|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$, so then, since the partial inner product is of course linear, it is clear what $\langle e|M|e'\rangle$ should be.

6.2 Quantum operators

Usually, we say that a quantum state evolves by applying a unitary transform to it. This is true if we have control over and access to an entire system. But if we communicate a state and something happens to it underway, say some noise on the channel, we need a more general formalism. The point is that our state interacts with the environment underway, but the receiver has no access to the entire environment, only to the state that is received.

So it would be nice, if we could describe what can happen in this situation, referring only to the state we are interested in and the space in which it lives.

This can be done with the *Operator Sum* formalism. In this formalism, an quantum operation $\mathcal{E}$ is described as a set of linear mappings $\mathcal{E} = \{E_1, ..., E_t\}$, all defined over the same vectorspace V with the property that

$$\sum_i E_i E_i^\dagger = I$$

In the most general case, the operation $\mathcal{E}$ can act on a state (or density matrix) ρ, provided ρ is also defined over the space V. The state after applying operator $\mathcal{E}$ is

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$$

The basic claim here is that if we send a quantum state $|\psi\rangle$ over some channel, then no matter what happens to it underway, the received state can be described as $\mathcal{E}(|\psi\rangle\langle\psi|)$ for some quantum operation $\mathcal{E}$.

The same operation can sometimes be expressed in different ways. In general, two quantum operations $\mathcal{E}, \mathcal{F}$ are equivalent, if and only if we have, for $\mathcal{E} = \{E_1, ..., E_t\}$, $\mathcal{F} = \{F_1, ..., F_t\}$ that $E_i = \sum_j u_{ij} F_j$, where the matrix formed by the u_{ij}'s is unitary. This is Thm. 8.2 in N&C.

A particularly simple and nice case of quantum operations is the case where $E_i = \sqrt{p_i} U_i$, where U_i is unitary and the p_i's are non-negative real numbers with $\sum_i p_i = 1$. In this case what the quantum operation is doing is simple to describe: with probability p_i, it applies the unitary transform U_i to the input state.

Otherwise, the way in which a quantum operation could happen physically is more complicated. But based on the postulates, the most general thing that could happen is as follows: the environment takes the input state ρ of, say n qbits and an auxiliary register

of m qbits, which without loss of generality can be assumed to be in state $|0^m\rangle$. The environment applies a unitary transform U to $\rho \otimes |0^m\rangle\langle 0^m|$. As a result you get the state $U(\rho \otimes |0^m\rangle\langle 0^m|)U^\dagger$. Then the environment passes on as output the n qbits that contained ρ before.

The reason why this corresponds to the operator sum formalism above is that, by what we have seen before, the state of the n first qubits is

$$Tr_m(U(\rho \otimes |0^m\rangle\langle 0^m|)U^\dagger)$$

where Tr_m is the partial trace where we trace out the state of the last m qubits. It is now possible to compute the operator sum representation of what has happened to the state. We can do this using the expression from (1) for the partial trace. Let $\{|f_i\rangle\}$ be a basis for the space where the m first qubits live. Then we get

$$Tr_m(U(\rho \otimes |0^m\rangle\langle 0^m|)U^\dagger) = \sum_i \langle f_i|U(\rho \otimes |0^m\rangle\langle 0^m|)U^\dagger\,|f_i\rangle = \sum_i \langle f_i|U\,|0^m\rangle\rho\,\langle 0^m|U^\dagger\,|f_i\rangle$$

The last equality can be shown by noting that ρ is always a linear combination of pure states $|\Psi_j\rangle\langle\Psi_j|$, and likewise U is a linear combination of operators of form $|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$. Since furthermore both sides of the equality are expressions that are linear in ρ and U, it is enough to prove it in the case where we replace ρ by $|\Psi_j\rangle\langle\Psi_j|$ and U by $|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$. This in turn follows from the definition of partial inner product.

The above expression for $Tr_m(U(\rho \otimes |0^m\rangle\langle 0^m|)U^\dagger)$ is exactly of the form we expect from the operator sum formalism, where we set $E_i = \langle f_i|U|0^m\rangle$.

The fact to take away from this is that we now have a formula for how one computes the operator sum representation for a process where some input state ρ together with an environment in pure state $|e_0\rangle$ is subjected to unitary operation U. After this, considering only the space where ρ lives, the state has now changed to $\mathcal{E}(\rho)$, where $\mathcal{E} = \{E_i\}$ is a quantum operation with $E_i = \langle f_i|U|e_0\rangle$, and where $\{|f_i\rangle\}$ is a basis for the space where the environment lives,

Finally, note that the operator sum formalism is a natural generalization of unitary operations: a unitary operation U can be described as a quantum operation $\mathcal{F}_U = \{U\}$. Indeed, we have by definition that $\mathcal{F}_U(\rho) = U\rho U^\dagger$ which is identical to the way we have seen before that unitary operations act on general states.

7 Quantum Circuits are as powerful as Classical Circuits

We first consider the Toffoli gate, this can thought of as a classical gate with 3 input and output bits, defined as $\textsc{Toffoli}(x, y, z) = (x, y, z \oplus xy)$. The Toffoli gate is universal, i.e., any classical computable function can be computed using a circuit of Toffoli gates, and furthermore, since one can simulate a Nand gate, fan-out and other standard elementary gates using a single call to Toffoli, the number of Toffoli gates needed to compute a

function coincides, up to a constant factor, with standard measures of circuit complexity of a function. In particular, for Nand,

$$\textsc{Toffoli}(a, b, 1) = (a, b, \neg ab),$$

and for fan-out

$$\textsc{Toffoli}(1, a, 0) = (1, a, a).$$

One detail that will be important is that these simulations of standard gates using Toffoli requires that we specially prepare some of the inputs in fixed states, and we may only be interested in one or two of the outputs. This means that a circuit C_f of Toffoli gates computing function $f : \{0,1\}^n \to \{0,1\}^m$ will have the following behavior:

$$C(x, x_0) = (f(x), g(x))$$

where x is the actual input, x_0 is a fixed string of bits that contain all those specially prepared inputs we need to make the Toffoli gates do the right thing, $f(x)$ is the actual output and $g(x)$ represents the set of values that happens to be on all the other wires. We are not interested in computing g, but the value sitting there will clearly be some function of x (which we call g).

We would now like to use the existence of such a circuit C_f for any function f to argue that there exists a quantum circuit of similar size as C_f that computes the operator U_f. The good news is that since the Toffoli gate is reversible, we can think of it as a quantum operator, whose behavior on classical basis states is "the same" as it's classical definition. Concretely, we define the quantum Toffoli gate by

$$\textsc{Toffoli}(|x\rangle|y\rangle|z\rangle) = |x\rangle|y\rangle|z \oplus xy\rangle$$

It is easy to see that this just defines a permutation of the basis vectors, so this is indeed a unitary operation. Therefore, we can think of C_f as a quantum circuit, whose behavior on classical input is as defined above.

The bad news is that this behavior is not the one we defined for U_f. One might think that this does not matter, after all we wanted to compute f, and this value is indeed computed as part of the output. Nevertheless, direct usage of C_f in the quantum case would not work. If we would only use the circuit on classical input, there would be no problem, but in general, when we are done, the qubits holding $f(x)$ are *entangled* with those holding $g(x)$. For instance, if we place the input register in an equally weighted superposition of all classical inputs, then we will end in the state

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

With such a state, it will *not* work to just forget about the qubits holding $g(x)$. If, for instance, that register is measured, it will collapse to one classical state $g(y)$, for a random y. But then the first register will also collapse, to $f(y)$, and our nice superposition holding

information about all f-values is gone. So to get anything useful out of this, we would have to keep the state in the last qubits alive indefinitely, and if we wanted compute f many times, we would need more and more auxiliary qubits. This is not reasonable: when a subroutine computes a function using some auxiliary memory, it should be possible to reuse that memory many times.

To solve this, we add an extra step where we first compute exactly what is expected from U_f and second, the auxiliary qubits are returned to the state they were in at the start of the computation. This means in particular that they cannot be entangled with the qubits holding the result.

To describe this in a simple way, we the observation that the Toffoli gate is its own inverse. This means that the inverse operation of what the quantum circuit C_f does can be implemented easily, we simply execute the gates in C_f in the reversed order. The resulting circuit is called C_f^{-1}. We now have the following algorithm for computing U_f, where we describe what it does on any classical input strings $x \in \{0,1,\}^n, y \in \{0,1\}^m$. As usual, this determines what happens to any input state:

1. We start with input state $|y\rangle|x\rangle|x_0\rangle$ (where x_0 is fixed as above), and apply C_f to the last two registers. We get the state $|y\rangle|f(x)\rangle|g(x)\rangle$.
2. We apply m C-NOT gates, where each bit in $|f(x)\rangle$ gets to act as control bit and a corresponding bit in $|y\rangle$ is target bit. This x-or's $f(x)$ to y, so we have the state $|y \oplus f(x)\rangle|f(x)\rangle|g(x)\rangle$.
3. Finally, we apply C_f^{-1} to the last two registers, resulting in the state $|y \oplus f(x)\rangle|x\rangle|x_0\rangle$.

Since the last register is returned to its original state, we are allowed to ignore it, as far as the input/output behavior is concerned, so we really have an implementation of U_f. If we ignore degenerate cases where C_f has size sublinear in m, the circuit we have built has size linear in the size of C_f. We therefore have proved:

Theorem 1. *For any classical function $f : \{0,1\}^n \to \{0,1\}^m$ with classical circuit complexity $CC(f)$, there exists a quantum circuit of size $O(CC(f))$ computing U_f.*

8 On variants of CSS codes

Codewords in a CSS code are defined from two classical codes C_1, C_2 where $C_2 \subset C_1$ and $C_1, C_2^{\perp}$ both correct t errrors. If C_1 is a (n, k_1)-code and C_2 is a (n, k_2)-code, then the $CSS(C_1, C_2)$ code can encode $k_1 - k_2$ qubits into n and can correct for arbitrary quantum operations applied to at most t of the n qubits.

Consider the cosets of C_2 in C_1, there are $2^{k_1-k_2}$ such cosets. Choose one element from each coset, to get a set $C_{1/2} = \{v_1, ..., v_{2^{k_1-k_2}}\}$ of codewords from C_1.

The CSS codewords are defined as

$$|v_k + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |v_k + y\rangle, \quad v_k \in C_{1/2}$$

Suppose we also choose one element in each coset of C_1 in the entire set of 2^n binary vectors. Since there are 2^{n-k_1} cosets, we get a set $C_{all/1} = \{x_1, ..., x_{2^{n-k_1}}\}$. And finally, we

choose one element from the cosets of $C_2^{\perp}$ in the set of all vectors. There are $2^{n-(n-k_2)} = 2^{k_2}$ of those, so we get a set $C_{all/2^{\perp}} = \{z_1, ..., z_{2^{k_2}}\}$. A useful fact to note is that there is a 1-1 correspondence between pairs (v_k, x_i) and cosets of C_2 in the entire space, there are $2^{k_1-k_2}2^{n-k_1} = 2^{n-k_2}$ such pairs, and this is indeed the number of cosets of C_2 in the entire space. More concretely, v_k, x_i designates the coset $x_i + v_k + C_2$ of C_2.

Now, we can define a set of codes that are equivalent to the $CSS(C_1, C_2)$ code in terms of error correction capabilities, namely for any $x \in C_{all/1}, z \in C_{all/2^{\perp}}$, we have a code $CSS(C_1, C_2)_{z,x}$ where codewords are defined by

$$|\xi_{v_k,z,x}\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{y \cdot z} |v_k + x + y\rangle, \quad v_k \in C_{1/2}$$

Note that the codeword $|v_k + C_2\rangle$ defined above is a special case: we have $|v_k + C_2\rangle = |\xi_{v_k,(0,..,0),(0,...,0)}\rangle$.

The encoding operation for one of the codes $CSS(C_1, C_2)_{z,x}$ can be described as a unitary mapping $U_{z,x}$ that takes as input a basis state of form $|k\rangle|0^{n-(k_1-k_2)}\rangle$, where k is a string of length $k_2 - k_1$ bits, and outputs

$$U_{z,x} |k\rangle \big|0^{n-(k_1-k_2)}\big\rangle = |\xi_{v_k,z,x}\rangle$$

To use CSS codes for quantum cryptography, one needs two important facts about them, that are proved in the exercises:

- The set of states $\{|\xi_{v_k,z,x}\rangle\}$ form an orthonormal basis for the space of n qubits.
- For the state $\sum_{j \in \{0,1\}^n} |j\rangle|j\rangle$ of n EPR pairs, we have

$$2^{-n/2} \sum_{j \in \{0,1\}^n} |j\rangle|j\rangle = 2^{-n/2} \sum_{v_k \in C_{1/2}, x \in C_{all/1}, z \in C_{all/2^{\perp}}} |\xi_{v_k,z,x}\rangle|\xi_{v_k,z,x}\rangle$$

9 The Lo-Chau key exchange protocol.

In this section, we give some material on quantum exchange that replaces the treatment in N&C of Lo and Chau and in particular state a better definition of security for quantum key exchange than the one given in N&C.

9.1 The Ideal state for generating a secure key

For the Lo and Chau protocol, the first major point to understand is that Alice and Bob could generate a completely secure key if they could generate between them exactly the state of m EPR pairs, i.e.

$$|\beta_{00}\rangle^{\otimes m} = \frac{1}{2^{m/2}} \sum_{j \in \{0,1\}^m} |j\rangle|j\rangle$$

Namely, if Alice and Bob have exactly this state, they just both measure in the standard basis, get the same bit string, and since Eve cannot be entangled with Alice and Bob (if

she was, they would have a mixed and not a pure state), there is nothing she can do to learn any information on the key, now or later.

Once Alice and Bob have measured to get a key K, this key is uniformly distributed over all m-bit strings, in other words, what they have from Eve's point of view is a mixed state containing all possible keys with equal probability, and this state is not entangled with whatever Eve has. This can expressed by simply saying that the overall state held by Alice, Bob and Eve at the end of the protocol is of form

$$\frac{1}{2^m} \sum_{K \in \{0,1\}^m} |K\rangle\langle K| \otimes \rho^E$$

strictly speaking, this should be written as $\frac{1}{2^m} \sum_{K \in \{0,1\}^m} |K\rangle|K\rangle\langle K|\langle K| \otimes \rho^E$ since Alice and Bob each have a copy of K, but in the following we choose the abbreviation above for convenience.

9.2 Being close enough suffices

The next point is that it is actually enough for Alice and Bob to have a (possibly mixed) state ρ that is *close to* the ideal state. However, we have to be careful with what we mean by "close". In N&C, the so called fidelity is used as a distance measure, and it is shown that if the state Alice and Bob have is in this sense close to the ideal EPR state, then Eve can learn only a negligible amount of information by measuring whatever state she has. This sounds like just the result you want, but is actually not sufficient!

The point is that Eve does not have to measure her state immediately, she could instead wait until the key is used for something, for instance to encrypt a message. She could then make a measurement that depends on the ciphertext. Although it is not clear what she could get out of this, it clearly not a measurement she could have done based only on the state she had just after the key exchange. Therefore we need a security condition that is stronger. This can be done using a different measure of distance, known as the *trace norm distance*.

The trace norm distance is treated in detail in 9.2.1 of N&C, here we just give the definition and the most important result about it. First, we define, for an operator A on C^N, that $|A| = \sqrt{A^\dagger A}$, that is, the positive square root of $A^\dagger A$. By the square root, we mean the following: note that $A^\dagger A$ is a positive operator, it can therefore be diagonalized with non-negative eigenvalues on the diagonal. We then take the (positive) square root of all these values and define the resulting matrix to be the square root of $A^\dagger A$. Now, for two mixed states ρ, σ, we define the trace norm distance to be

$$D(\rho, \sigma) = \frac{1}{2} tr(|\rho - \sigma|).$$

This may look very mysterious at first sight, but actually makes good sense: as an example, suppose ρ and σ can be simultaneously diagonalized, that is, there is an orthonormal basis B in which they are both diagonal. Physically, this means that both states can be prepared by selecting with various probabilities the basis states in B, where the probabilities are

found on the diagonals of ρ and σ. Let $p = \{p_1, .., p_N\}, q = \{q_1, ..., q_N\}$ be the probability distributions defined in this way by ρ and σ, respectively. And define $D(p, q)$ to be the classical statistical distance between p and q, that is

$$D(p, q) = \frac{1}{2} \sum_i |p_i - q_i|$$

It is clear that $(\rho-\sigma)^\dagger = (\rho-\sigma)$, so $|\rho-\sigma|$ is a matrix containing $|p_i - q_i|$ on the diagonal, and hence $D(\rho, \sigma) = D(p, q)$.

A second fact is that operating on the states does not change the distance, i.e., we have for any unitary U that

$$D(U\rho U^\dagger, U\sigma U^\dagger) = D(\rho, \sigma)$$

The final important fact is that if the states are close then any measurement allowed by quantum mechanics will return almost the same distribution of results. We have

Theorem 2. *For any measurement $\mathcal{M}$, let $p(\mathcal{M}, \rho)$ be the distribution of results returned from applying $\mathcal{M}$ to ρ, and let $p(\mathcal{M}, \sigma)$ be the distribution resulting from applying $\mathcal{M}$ to σ. Then*

$$D(\rho, \sigma) = max_{\mathcal{M}} D(p(\mathcal{M}, \rho), p(\mathcal{M}, \sigma))$$

In other words, if $D(\rho, \sigma)$ is very small, then no matter how we operate on the state and no matter how we measure it, the results we obtain will be essentially the same, regardless of whether we start from ρ or from σ. In a nutshell, ρ behaves as σ, except with probability $D(\rho, \sigma)$.

We can now define what a key exchange protocol needs to satisfy to be secure, namely if both parties Alice and Bob agree that the protocol was successful, then state held by Alice, Bob and the eavesdropper Eve is close to a state where the key is completely independent of the state held by Eve. To give a more concrete definition of this, we have to take into account the issue that we cannot guarantee that Alice and Bob always succeed in generating a key. Eve could block all quantum communication, thus making it impossible to generate an information theoretically secure key. To allow for this possibility, we assume that Alice and Bob will publicly agree on the length of the key they can generate, and this length can be 0 in case they conclude that they must give up.

First we define, for $\ell = 1, 2, ..$ that $\sigma_\ell = \frac{1}{2^\ell} \sum_{K \in \{0,1\}^\ell} |K\rangle\langle K|$, i.e., σ_ℓ is the mixed state of a uniformly chosen key K of length ℓ bits. For $\ell = 0$, we define σ_ℓ to be some arbitrary but fixed classical state.

Then we define a *good* state to be one where Alice, Bob and Eve together hold something of the form

$$\sum_{\ell=0}^{m} p_\ell \cdot \sigma_\ell \otimes \rho_\ell^E$$

where the p_ℓ's form a probability distribution and where the ρ_ℓ^E's are arbitrary states held by Eve. In such a good state, Alice and Bob have shared a key of length ℓ with probability

p_ℓ. Eve may be able to find out what ℓ is by measuring her state because the ρ_ℓ^E may be different for different ℓ, but this all she can get because she is not entangled with Alice and Bob once ℓ is fixed. As an example, if Eve blocks all quantum communication, we will have $p_0 = 1$.

We then define a quantum key exchange protocol to be secure if it ends in joint state ρ^{ABE} for Alice, Bob and Eve for which

$$D(\rho^{ABE}, \rho_{good}^{ABE}) \leq 2^{-cn},$$

where c is a constant, n is a security parameter (where $\theta(n)$ qubits are sent in the protocol), and ρ_{good}^{ABE} is a good state.

Next, we observe that if Alice and Bob could generate a state which with large probability is good, and else is something arbitrary, then we are in business. As we shall see, we will be able to ensure the following: either Alice and Bob agree that they cannot generate a secure key, or they go ahead. The event that Alice and Bob go ahead and generate a key, but in fact the state they have is not σ_m has probability $2^{-\alpha n}$, where $2n$ is the total number of qubits we send in the protocol. This event occurring is the only way we can fail to be in a good state, since there is no security requirement if Alice and Bob give up.

Therefore, the global state is of form

$$\rho^{ABE} = (1 - 2^{-\alpha n}) \cdot \rho_{good}^{ABE} + 2^{-\alpha n} \cdot \rho_{error}$$

where ρ_{error} is some arbitrary state. We then have that

$$D(\rho^{ABE}, \rho_{good}^{ABE}) = \frac{1}{2} tr(|2^{-\alpha n}(-\rho_{good}^{ABE} + \rho_{error})|) \leq 2^{-\alpha n}$$

where the first equality is by definition of D, and the inequality follows, first because $tr(|\cdot|)$ is a norm and so satisfies $tr(|A + B|) \leq tr(|A|) + tr(|B|)$ and second because all density matrices have norm 1.

9.3 Key Exchange: how to get a good state from one with not too many errors

The communication model for quantum key exchange is that Alice can send qubits to Bob but the adversary Eve can do anything she wants to the qubits sent. Furthermore, we assume an authentic open classical channel between Alice and Bob: Eve gets everything sent on this channel, but cannot modify what is sent. Using only classical communication, it is impossible to establish a secret key if the adversary has access to all communication, but it is indeed possible with quantum communication. On the other hand, if we did not assume the classical channel, key exchange would be impossible even with quantum communication: there must something in the model that, from Alice's point of view, distinguishes Bob from anyone else. If not, there is no way Alice could know that she is exchanging a key with Bob and not Eve.

The (modified) Lo-Chau protocol in the book is described as if one can use an arbitrary quantum error correcting code. It is actually not entirely clear that this would work, and

one should read it as if CSS codes are used, encoding $m = k_1 - k_2$ qubits into n bits. It is also necessary to modify the condition specifying when Alice and Bob must abort the protocol.

We therefore describe the entire protocol below as it looks after the changes. It uses an (n, m) $CSS(C_1, C_2)$-code that can correct t bit and phase flips. We assume that $t = \mu n$ for a constant μ, which is indeed possible for CSS codes.

Lo-Chau Quantum Key-Exchange Protocol

1. Alice creates $2n$ EPR pairs, each in state $\beta_{00} = (|00\rangle + |11\rangle)/\sqrt{2}$, and picks a random $2n$-bit string b.
2. Alice randomly selects a subset T consisting of n of the $2n$ EPR pairs.
3. For each $i = 1...2n$, if $b_i = 1$, Alice executes H on the second particle in the i'th EPR pair.
4. Alice sends the second particle in each EPR pair to Bob, who announces on the classical channel that he has received n particles. All communication after this point uses the classical channel.
5. Alice sends b and T to Bob.
6. For $i = 1.., 2n$, if $b_i = 1$, Bob applies H to the i'th received particle.
7. For each EPR pair in T, Alice and Bob both measure their particles in the Z basis, and exchange results.
8. If more than $t - \epsilon n$ error occurred, Alice and Bob abort. Here ϵ can be any constant less than μ.
9. Consider the state of the n EPR pairs not in T as the superposition

$$2^{-n/2} \sum_{v_k \in C_{1/2}, x \in C_{all/1}, z \in C_{all/2^\perp}} |\xi_{v_k,z,x}\rangle \left| \tilde{\xi}_{v_k,z,x} \right\rangle$$

   where the $\tilde{\xi}$ indicates that some errors may have been introduced by Eve on Bob's side. Alice then measures for z and x and sends the results to Bob. This makes the state collapse to

$$2^{-m/2} \sum_{v_k \in C_{1/2}} |\xi_{v_k,z,x}\rangle \left| \tilde{\xi}_{v_k,z,x} \right\rangle$$

10. Assuming that less than t bit and phase flips have occurred, Bob corrects the errors using the CSS decoding method, which if successful will produce the state

$$2^{-m/2} \sum_{v_k \in C_{1/2}} |\xi_{v_k,z,x}\rangle |\xi_{v_k,z,x}\rangle$$

11. Alice and Bob both apply $U^\dagger_{z,x}$ on their part of the state (see previous section) and this results in

$$2^{-m/2} \sum_{k \in \{0,1\}^m} |k\rangle |k\rangle$$

    and then they both measure each particle in the Z basis to obtain the shared key.

9.4 Proving Security

We now want to show that the protocol is secure. To this end, note that the only assumption needed to ensure that Alice and Bob have a perfect EPR state before measuring the key is is that no more than t bits and phase flips occur. So given what we said above, the only event that could cause us to finish in a state that is not good, is if Alice and Bob do not abort, but still more than t bit or phase flips occur when Bob measures the error syndromes in step 10. We will argue that the first part of the protocol ensures that this event occurs with probability at most $2^{-\alpha n}$ for some constant α

The intuition of the first part of the protocol is clear: Alice sends one half of $2n$ EPR pairs to Bob. She selects randomly n of them to serve as the test set, and we check how many errors are in the test set. If we do not see many errors in the test set, we believe that there were also not many errors in the set we did not check - the point being, of course, that Eve cannot not know ahead of time which qubits are in the testset. We now make this intuition more precise..

How to detect errors As discussed under error correction, although Eve can introduce arbitrary errors to the qubits sent to Bob, we can always measure the bits in such a way that we force the error to decide if it wants to be a bit flip, a phase flip, both, or nothing. Now, a single EPR pair subjected to one of the these 4 possible errors will be changed to one of the 4 so called Bell states:

$$\text{No error: } \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),\ X : \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),\ Z : \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),\ XZ : \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Since these form an orthogonal basis of the 4 dimensional space for 2 qubits, it is in principle possible to measure EPR pairs in the test set, and make them collapse to one of the 4 possibilities. This measurement can be "separated" into two, namely one that only tests for bit flips and one that only tests for phase flips. To test for a bit flip, we do a projective measurement defined by splitting the 4-dimensional space in the two subspaces spanned by $\{\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)\}$ respectively by $\{\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)\}$. In other words we ask the system to decide if it is in the subspace where a bit flip did, resp. did not occur. As usual, the subspaces could also be defined by projectors projecting to those subspaces, and this is exactly what the text in N&C does, defining the measurement by projectors $\Pi_{bf}, I - \Pi_{bf}$. In the same way, we can define a measurement that determines whether a phase flip occurred, by projectors $\Pi_{pf}, I - \Pi_{pf}$.

It is easy to see that you can do these two measurements in any order you like, and the joint distribution of the outcomes is the same no matter the order. In fact, at the end, you will have projected the input to one of the 4 Bell states, in other words you have done the same as a complete projective measurement. This is what is meant in the textbook by "these measurements commute with the Bell basis". This also means that once we fix the state Alice and Bob hold after the quantum transmission, there is one fixed (classical) probability distribution P_{bf} for the results you get if you would measure all the $2n$ pairs for bit flips, and another P_{pf} for the distribution of phase flips.

Unfortunately, the measurement determining the errors cannot be done as they are described because they require that you operate in a coherent way on both qubits in a pair. Alice and Bob are physically separated and cannot do this. However, Alice and Bob can do something almost as good, namely they can either both measure in the computational basis $|0\rangle, |1\rangle$ (the Z basis) or in the diagonal basis $|+\rangle, |-\rangle$ (the X basis). It is clear that if Alice and Bob both measure in the computational basis, and compare the results, they get exactly the same statistics out as if they had done the measurement defined by $\Pi_{bf}, I - \Pi_{bf}$, after all, the natural way to detect a bit flip is if both measure their bit and compare. In the same way if both measure in the X basis, they detect phase flips with the same output distribution as if $\Pi_{pf}, I - \Pi_{pf}$ had been used. The price is that they cannot test a pair for both bit and phase flips, because the measurements are now destructive.

Looking at the test set is enough Let us now for a while concentrate on bit flips. If Alice and Bob test for bit flips as described above in the test subset, we want to argue that the number of bit flips we would see if we looked for bit flips in the other n positions is not much larger. More precisely, the event we want to avoid is that, for some constant $\epsilon > 0$, when we look for for bit flips inside the test set we see at most $t - \epsilon n$ errors, but if we measure for bit flips outside the test set, we have more than t errors. Call this event Bad.

Remember that there is a fixed distribution P_{bf} for the bit flips. We can think of this as a distribution that produces a string of $2n$ bits, where a 1 in some position means that when measuring the corresponding pair of qubits, the system decided to have a bit flip here, whereas a 0 means no bitflip. P_{bf} is essentially created by Eve since she decides what to do to the qubits sent to Bob. If she does nothing, for instance, P_{bf} will output the all-zero string with probability 1. Since Eve has no idea which bits are in the test set, we can assume that P_{bf} is independent of the choice of test set. This allows us to bound the probability of Bad. Consider the following two experiments:

- Choose at random n among the $2n$ positions as test set. Choose bits in the test set according to P_{bf} "restricted" to the test set. Then choose bits outside the test set such that the entire bit string is distributed according to P_{bf}.
- Choose the entire $2n$ bit string according to P_{bf}, then choose a random test set.

The first experiment corresponds exactly to what is done in the protocol: first we look at the test set, then Bob tries to correct the errors and thereby fixes the string outside the test set. Note, moreover, that since P_{bf} is independent of the choice of test set, the two experiments output identical results, so Bad occurs with the same probability in both experiments.

We can now apply the result of Exercise 12.27 in N&C to the second experiment [2]. It says that, for any $\delta > 0$, the probability of seeing less than δn errors in the test set, but more than $(\delta + \epsilon)n$ errors in the untested bits, is less than $e^{-c\epsilon^2 n}$ for some constant c and all large enough n.

[2] This is actually a variant of the more well known Hoeffding or Chernoff bound.

We can use this result in our case, recalling that $t = \mu n$ for some constant $\mu > \epsilon$. Then we can set δ from the above result to be $\mu - \epsilon$. This means that $t - \epsilon n = \delta n$ and $t = (\delta + \epsilon)n$, so the probability that Alice and Bob try to complete the protocol with too many bit flips is at most $e^{-c\epsilon^2 n}$. Of course, an exactly similar argument can be done for phase flips.

Testing for both bit and phaseflips We need, of course, to test for both bit and phase flips. The tests we are able to do cannot test the same position for both error types, so we cannot choose two independent test sets for bit and phase flips, as they would overlap with large probability. Instead, the Lo-Chau protocol uses the method of applying Hadamard gates randomly to the qubits sent, and test only for bit flips. Since a Hadamard transform converts a bit flip to a phase flip and vice versa, what happens is that we in fact test for both bit and phase flips at the same time.

More precisely, the experiment can be modeled as follows: choose a $2n$ bit string b_{bf} according to P_{bf} and a string b_{pf} according to P_{pf}. Now, for each bit position, choose at random whether to leave the bits of b_{pf} and b_{bf} in this position alone or interchange them. Call the bits strings obtained b'_{bf} and b'_{pf}. Finally, choose n positions for a test set at random and look at how many 1-bits b'_{bf} has in the test set.

Note that we assume that the decisions to switch bits or not is independent of the bit strings. This is justified since in real life Eve has no information on whether Alice applied an H to a given qubit or not. For every such qubit, Eve will see the same state in any case, namely the mixed state $I/2$.

Now, using the same result as above, one can choose a constant $\epsilon' < \epsilon$ and argue that with exponentially small probability, we will see at most $t - \epsilon n$ 1's in the test set but more than $t - \epsilon' n$ 1's in b'_{bf} outside the test set.

Note that b'_{bf}, b'_{pf} describe the set of bit and phase flips Bob will actually be trying to correct, so we now already know that he will almost certainly be able to handle the bit flips.

The final observation we need is that, since the decision to switch bits is independent of the actual bit strings, we can expect that the number of 1's in b'_{bf} and b'_{pf} (outside the test set) to be almost the same, in fact the probability that they will be different by more than $\epsilon' n$ is exponentially small, and hence the number of 1's in b'_{pf} will also be less than t with overwhelming probability.

This last result follows from the Hoeffding bound (listed below). Intuitively, the bound says that if you make m independent experiments, each outputting 0 or 1, and 1 occurs with probability p each time, then the number of observed 1's will be close to pm except with exponentially small probability. We can now apply this to, e.g., the set of 1's in b_{bf} where the experiment is to decide at random to move the 1 to the other string or leave it alone – we can think of moving the 1 as being equivalent to the experiment outputting 1. The bound then says we can expect about half the 1's to be moved. Doing the same with the 1's in b_{pf} and adding up the error probabilities gives the desired result. We leave the details to the reader.

This therefore finally shows that $Pr(Bad)$ is exponentially small as a function of n.

Theorem 3 (Hoeffding bound). *Let $X_1, ..., X_m$ be independent random variables in the range $[a, b]$, and let $X = X_1 + ... + X_m$. Then*

$$Pr(|X - E(X)| > t) \leq 2e^{-(\frac{2t^2}{n(a-b)^2})}$$

10 A More General Version of the BB84 protocol

BB84 is the general name for QKD protocols where in the first step, one player (Alice) sends qubits to Bob, where each qubit is in a state that is randomly chosen among $|0\rangle, |1\rangle, |+\rangle, |-\rangle$. One can think of $|0\rangle, |1\rangle$ as encodings of classical bits 0/1 in the standard basis, whereas $|+\rangle, |-\rangle$ are encodings of bits 0 and 1 in the "diagonal" basis consisting of $|+\rangle, |-\rangle$. To reflect this in the following, we use a slightly different naming convention that is more convenient here, namely for $\theta = +, \times$, we set

$$|0\rangle_+ = |0\rangle, |1\rangle_+ = |1\rangle, |0\rangle_\times = |+\rangle, |1\rangle_\times = |-\rangle$$

Note that measuring $|b\rangle_+$ in the standard $+$-basis returns b always, whereas a measurement in the $\times$-basis returns a random result. Of course, something similar holds for measurement of $|b\rangle_\times$.

In the textbook, a special version of the BB84 protocol is described, that is derived from the Lo-Chau protocol, such that security of Lo-Chau implies security of this particular variant of BB84. However, a much more general version of the protocol can also be proved secure, albeit using a completely different proof technique. We present a more general protocol, but first describe some parameters it uses: n is total number of qubits sent, and an additional parameter $k \leq n/2$ chosen such that k is $\theta(n)$. Additional parameters that are determined during the protocol are: β, an observed error rate and ℓ, the number of key bits that can be extracted ($\ell = 0$ is possible).

We also use a so called universal hash function $g : \{0,1\}^{n-k} \to \{0,1\}^\ell$. g is chosen at random from a class of functions $\mathcal{G}$, and has to satisfy that for any two distinct inputs x, x', the probability that $g(x) = g(x')$ is at most $2^{-\ell}$. This probability is taken over the choice of g. One possibility is to let $\mathcal{G}$ be the class of functions parametrized by $\alpha \in GF(2^{n-k})$ where $g_\alpha(x)$ is defined to be the least significant ℓ bits of αx. We return to the purpose of g below.

Finally an error correcting code C is used. This is a binary code of length $n - k$ that corrects a fraction $\beta' > \beta$ of errors (except perhaps with negligible probability). The choice of β' is a trade-off between keeping the probability low that Alice and Bob disagree on the final key and keeping the length of the final key as large as possible. It is fine in the following to think of β' as being approximately β.

As we increase the error rate β we want to correct for, the dimension of C needs to be smaller. This means that the length of a syndrome computed w.r.t. C (called m below) will increase. We will see that we need to have $\ell \leq (1 - h(\beta))n - k - m$, where $h()$ is the binary entropy function. Therefore, as β gets closer to the maximum value 1/2, this may go negative, which means that the error rate is too large for us to be able to choose a suitable code. In this case, the protocol simply aborts.

The general BB84 protocol works as follows:

1. Alice sends $(2+\delta)n$ qubits to Bob, where the i'th qubit is in state $|b_i\rangle_{\theta_i}$ where b_i, θ_i are randomly chosen. Let $b = (b_1, ..., b_n)$.
2. Bob measures the i'th qubit received in randomly chosen basis $\hat{\theta}_i$ to get result $\hat{b}_i$, and tells Alice on the classical channel that the qubits were received. Let $\hat{b} = (\hat{b}_1, ..., \hat{b}_n)$.
3. Alice and Bob exchange $\theta_i, \hat{\theta}_i$ for all i on the classical channel. For use in the following, they select the first n indices where $\theta_i = \hat{\theta}_i$ (there will be at least n except with exponentially small probability). For convenience, we renumber the positions, such that the n selected indices are $1, ..., n$.
4. Alice selects a random subset size k subset T of the n indices and sends it to Bob. They then exchange b_T and $\hat{b}_T$ (b, $\hat{b}$ restricted to T) and compute β, the number of indices where b_T and $\hat{b}_T$ differ divided by k.
5. From k and β, Alice chooses a suitable linear code C as described above and sends it to Bob (concretely, this could mean sending the generator matrix). If that is not possible Alice aborts and informs Bob. If C is sent, Alice also computes and sends the syndrome syn of $b_{\bar{T}}$ w.r.t. C. Here, $b_{\bar{T}}$ is b restricted to the complement of T. Let m be the bit length of syn.
6. Bob uses syn to correct the errors in $\hat{b}_{\bar{T}}$, to get $b'_{\bar{T}}$ (we expect, of course, that $b_{\bar{T}} = b'_{\bar{T}}$).
7. Alice chooses a random universal hash function g with output size ℓ, where $\ell < (1 - h(\beta))n - k - m$, and sends g to Bob. The parties compute $K = g(b_{\bar{T}})$ and $K' = g(b'_{\bar{T}})$, respectively.

Several proofs of security of (variants of) this protocol have been published, but the nicest and most intuitive proof is found in Bouman and Fehr: *Sampling in a quantum population, and Applications*, Proc. of Crypto 2010. The intuitive reason why the length of the final key is approximately $(1-h(\beta))n-k-m$ is as follows: to learn information on b, the adversary Eve must measure the qubits sent, but this must introduce errors, since she does not know in advance the bases θ_i used for the encoding. The analysis in B&F shows that $h(\beta)n$ is a good estimate of the number of bits of information Eve can know about b, if the observed error rate is β. However, during the protocol, we tell Eve extra information, namely the k bits in b_T and the m bits in syn, so we can expect to have $n - h(\beta)n - k - m$ secret bits left.

It turns out that, to correct for error rate β, the code needs to be such that $m = h(\beta)n$, so we get that ℓ can be about $(1 - 2h(\beta))(n - k)$ which is positive if β is less than about 11%.

The universal hash function g is needed since it is of course not enough for Alice and Bob to agree on $b_{\bar{T}}$. Eve may have a lot of information on this string so we need to distill out those $(1 - 2h(\beta))(n - k)$ secrets bits that we know it contains, and this is exactly what g can do, provided it has the property we listed above.