Solutions to Problem Set No. 6

UBC Metro Vancouver Physics Circle 2018-2019

March 14, 2019

1 Quantum xerox machines

1. From the definition of U, we have $U|1\rangle|+\rangle = |+\rangle|+\rangle$, whatever $|+\rangle$ actually is!

2. We write $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, use linearity, then use the xeroxing property:

$$U|1\rangle|+\rangle = U|1\rangle\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(U|1\rangle|0\rangle + U|1\rangle|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right).$$

3. Let's revisit our answer from part (1):

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

   Note that we simply multiplied unit vectors $|0\rangle$ and $|1\rangle$, treating them formally like algebraic variables x, y. This is clearly different from our answer to part (2)! The current answer involves classical outcomes $|0\rangle|1\rangle$ and $|1\rangle|0\rangle$ in the mixture, but the answer to part (2) did not. It follows that the cloning operator U cannot be linear, in contravention of the laws of quantum mechanics! Hence, there are no quantum xerox machines.

4. We now suppose that $|\psi_n\rangle$ is a quantum state of n coins, and U acts as

$$U|1^n\rangle|\psi_n\rangle = |\psi_n\rangle|\psi_n\rangle.$$

   Here, $|1^n\rangle = |1\cdots 1\rangle$ is the state of n quantum coins where all of them are in the $|1\rangle$ state. The cloning operator U "overwrites" $|1^n\rangle$ on the first set of coin with the state

$|\psi_n\rangle$ of the second set. Define a particular state on n coins,

$$|+\rangle = \frac{1}{\sqrt{2}}(|01^{n-1}\rangle + |11^{n-1}\rangle) = \frac{1}{\sqrt{2}}(|01\cdots 1\rangle + |11\cdots 1\rangle),$$

analogous to the state $|+\rangle$ on a single coin. Then we can repeat the same argument as above! We apply the cloning operator one way, then use linearity and apply it again, and get different answers. It follows that we cannot clone the state on n coins!

There is an even simpler argument: if we can clone the state on n coins, then in particular, we can clone the state $|\psi_1\rangle|1^{n-1}\rangle$, where $|\psi_1\rangle$ is now the state on a *single* coin. In other words,

$$U|1^n\rangle|\psi_1\rangle|1^{n-1}\rangle = |\psi_1\rangle|1^{n-1}\rangle|\psi_1\rangle|1^{n-1}\rangle.$$

It we forget about all but the first coin in each batch, we have effectively cloned a single coin! We have already argued that this is impossible. This formalises the intuition that being able to clone the state of n coins should let you clone the state on a single coin.

5. We can directly check the teleportation operator is linear. Define $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ for some constants α, β. Then

$$\begin{aligned} T|1\rangle|\psi\rangle = T|1\rangle(\alpha|0\rangle + \beta|1\rangle) &= \alpha T|1\rangle|0\rangle + \beta T|1\rangle|1\rangle \\ &= \alpha|0\rangle|1\rangle + \beta|1\rangle|1\rangle = (\alpha|0\rangle + \beta|1\rangle)|1\rangle = |\psi\rangle|1\rangle. \end{aligned}$$

Thus, the teleportation operator is linear. It turns out that quantum teleportation can in fact be done! Though the operator T is somewhat complicated.

2 Hamming it up

1. The key thing to check in all cases is that, if any single physical bit (corresponding to a single column) is erased, the remaining pair of physical bits are distinct. There are many possible answers, but here are a few you can arrive at by trial and error:

$00 \to 000$	001	111
$01 \to 011$	010	100
$10 \to 101$	100	010

$$11 \to 110 \quad 111 \quad 001$$

2. The rule is to represent logical bits $x_1 x_2$ by physical bits $x_1 x_2 p_1$, where

$$p_1 = x_1 \oplus x_2.$$

Thus, filling out the table gives the first answer listed above:

$$\begin{aligned} x_1 x_2 &\to x_1 x_2 p_1 \\ 00 &\to 000 \\ 01 &\to 011 \\ 10 &\to 101 \\ 11 &\to 110 \end{aligned}$$

3. Mod 2 arithmetic means that $2 = 0$. This means that, for instance,

$$x_1 \oplus p_2 = x_1 \oplus x_1 \oplus x_2 = 2x_1 \oplus x_2 = x_2.$$

So we can recover x_2 by adding x_1 and p_1. Similarly, we can recover x_1 from adding x_2 and p_1. The moral is that with two overlapping circles in our Venn diagram, we can recover any single bit from the remaining two!

4. We now have logical bits x_1, x_2, x_3. For the physical bits p_1, p_2, p_3 in the overlap of two circles, we calculate them by adding the two logical xs in the overlapping circles. This means that

$$\begin{aligned} p_1 &= x_1 \oplus x_2 \\ p_2 &= x_1 \oplus x_3 \\ p_3 &= x_2 \oplus x_3. \end{aligned}$$

The natural guess for p_4 is the sum of *all three* xs, corresponding to the three overlapping circles:

$$p_4 = x_1 \oplus x_2 \oplus x_3.$$

Let's first check that we can erase x_1 and recover it from the xs and ps. Once again

using the fact that $2 = 0$ in mod 2 arithmetic,

$$p_1 \oplus x_2 = x_1 \oplus x_2 \oplus x_2 = x_1 \oplus 2x_2 = x_1.$$

That is, we recover the logical bit x_1 by adding the physical bits p_1 and x_2. From similar calculations, we find that

$$\begin{aligned} x_1 &= p_1 \oplus x_2 = p_2 \oplus x_3 \\ x_2 &= p_1 \oplus x_1 = p_3 \oplus x_3 \\ x_3 &= p_2 \oplus x_1 = p_3 \oplus x_2. \end{aligned}$$

5. You may have noticed that p_4 was not really necessary for encoding the three logical bits x_1, x_2, x_3. But in the last column above, we wrote the xs solely in terms of ps. Now comes the punchline. We can swap the role of the ps and xs, treating p_1, p_2, p_3, p_4 as logical bits, and writing the xs in terms of them:

$$\begin{aligned} x_1 &= p_4 \oplus p_1 \oplus p_2 \\ x_2 &= p_4 \oplus p_1 \oplus x_3 \\ x_3 &= p_4 \oplus p_2 \oplus p_3. \end{aligned}$$

Our original definitions

$$\begin{aligned} p_1 &= x_1 \oplus x_2 \\ p_2 &= x_1 \oplus x_3 \\ p_3 &= x_2 \oplus x_3 \\ p_4 &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \end{aligned}$$

now allow us to recover any of the p_i if they are corrupted. This is the Hamming code!

3 Summoning: possibilities and impossibilities

1. At r_1, Alice knows b_1 but not b_2.

   This is because b_1 comes from c_1, and it's possible to travel from c_1 to r_1. However, b_2 comes from c_2, and it is not possible to travel from c_2 to r_1.

   Similarly, at r_2 Alice knows b_2 but *not* b_1.

2. Let's imagine our perfect protocol is being implemented. Since $b_1 = 1, b_2 = 0$ results in the state being handed in at r_1, Alice at r_1 must always hand the state in when she sees that $b_1 = 1$. Since she doesn't know about b_2, she hands the state in at r_1 even when $b_2 = 1$.

   By the same argument applied to Alice at r_2, she must hand in the state at r_2 when $b_1 = 1$ and $b_2 = 1$.

   This means $b_1 = 1, b_2 = 1$ results in two copies of the state being produced, if we assume the protocol is perfect.

3. Since making two copies of a quantum state is impossible, there can be no perfect protocol!

4. Notice that to an Alice sitting at point s, all three call point c_1, c_2, c_3 look *identical*.

   It is natural then that Alice do the same thing with regard to each call point.

   Since she holds three shares of her error-correcting code, she should send one share to each call point.

5. Consider the share at $c_i, i = 1, 2, 3$. If you see $b_i = 1$, you know you want to construct the state at r_i, so it's natural that you should send your share there.

   If instead $b_i = 0$, you know you do *not* need the state at r_i, so it is natural to send your share the other way (to r_{i-1}) since it *might* be needed there.

   If (for example) we have $b_1 = 0, b_2 = 1, b_3 = 0$, r_2 will receive the shares from c_2 and c_3. Since you only need two out of of three shares to get the state, you can recover the state at r_2.

6. Suppose $b_1 = 1, b_2 = 1, b_3 = 0$. Then:

   - r_1 gets one share (from c_1);

- r_2 gets two shares (from c_2, c_3);
- r_3 gets zero shares.

Thus, the same protocol works if one or two of the bs equal 1.

However, if three of the bs have $b = 1$ then every r_i gets one share, and no-one can reconstruct the state.

In fact, if any of one, two or three bs may have $b = 1$, it is impossible to complete the summoning task (in the sense of returning the state in full to one of the called-to diamonds). See "A Quantum Paradox of Choice", Adlam and Kent, for details.