Solution Set #4

Quantum Error Correction
Instructor: Daniel Gottesman

Problem #1. Using Bounds on QECCs

a) $[[15, 5, 3]]$: Clearly the quantum Singleton bound is satisfied ($n - k = 10 \geq 2(d-1) = 4$), so the code can exist. The quantum Hamming bound is also satisfied, $3n + 1 = 46 \leq 2^{n-k} = 1024$, so the code can be non-degenerate. In fact, the quantum Gilbert-Varshamov bound is satisfied too, $9n(n-1)/2 + 3n + 1 = 991 \leq 2^{n-k} = 1024$, so we know the code must exist. (Actually, even a $[[15, 9, 3]]$ code exists.)

$[[21, 16, 3]]$: The quantum Singleton bound is satisfied, $n - k = 5 \geq 2(d-1) = 4$, but the quantum Hamming bound is not: $3n + 1 = 64 \not\leq 2^{n-k} = 32$. Thus, just referring to these bounds, this code could exist but would have to be degenerate. (In fact, using linear programming bounds, it has been shown that no code with these parameters exists.)

$[[9, 2, 5]]$: The quantum Singleton bound is not satisfied, $n-k = 7 \not\geq 2(d-1) = 8$. Therefore, the code cannot exist. (A $[[9, 1, 5]]$ code would satisfy the quantum Singleton bound, but still does not exist; indeed, there is not even a $[[9, 1, 4]]$ QECC. The smallest QECC correcting two errors uses 11 qubits.)

b) The relevant bound to use in this problem is the No-Cloning bound: We cannot split the system into two pieces, such that each can recover the encoded state. Note that this is *not* the same as insisting that we cannot recover given only half of the original qubits.

In fact, the answer to the first question is "yes": it is possible to find an encoding which corrects for the erasure of any single person's qubits (even though Alice has 3 qubits out of 6 total). For instance, one way to do this is to encode the qubit in the 5-qubit code, give one of the five qubits to Charlie and two qubits to each Alice and Bob, and then to give Alice a sixth qubit in the state $|0\rangle$. The 5-qubit code has distance 3 and can thus correct two erasure errors, so it can certainly correct for the absence of Bob or Charlie. It can also correct for the absence of Alice, since two qubits of the code plus one irrelevant qubit have been erased.

Similarly, we can easily find an encoding that corrects for the absence of just Bob or just Charlie. We can use the above strategy or even the simpler strategy of giving the "encoded" qubit directly to Alice and then use $|0\rangle$ states for the remaining five qubits.

On the other hand, we certainly cannot correct for the absence of any pair of people, or even Alice vs. Bob and Charlie, as either situation would violate the no-cloning bound.

Problem #2. Applying the Linear Programming Bounds

a) The generators of the $[[5, 1, 3]]$ code are $X \otimes Z \otimes Z \otimes X \otimes I$, plus 3 cyclic shifts of it. A little multiplication shows us that the other elements are the remaining cyclic shift of $X \otimes Z \otimes Z \otimes X \otimes I$, plus all cyclic shifts of $Y \otimes X \otimes X \otimes Y \otimes I$ and $Z \otimes Y \otimes Y \otimes Z \otimes I$. All of these 15 elements have weight 4, and the remaining element of the stabilizer is the identity, weight 0, so the weight enumerator is

$$A(x) = 1 + 15x^4. \tag{1}$$

Then, applying the quantum MacWilliams identity, we find

$$
\begin{aligned}
B(x) =& \frac{1}{16}\left[(1+3x)^5 + 15(1-x)^4(1+3x)\right] && (2)\\
=& \frac{1}{16}\big[16 + (5\cdot 3 - 15\cdot 4 + 15\cdot 3)x + (10\cdot 9 + 15\cdot 6 - 15\cdot 4\cdot 3)x^2 && (3)\\
&+ (10\cdot 27 - 15\cdot 4 + 15\cdot 6\cdot 3)x^3 + (5\cdot 81 + 15 - 15\cdot 4\cdot 3)x^4 + (243 + 15\cdot 3)x^5\big] && (4)\\
=& 1 + 30x^3 + 15x^4 + 18x^5. && (5)
\end{aligned}
$$

b) The quantum MacWilliams identity tells us (using $A_0 = B_0 = 1$)

$$
\begin{aligned}
4B(x) =& 4 + 4B_1x + 4B_2x^2 + 4B_3x^3 + 4B_4x^4 && (6)\\
=& (1+3x)^4 + A_1(1+3x)^3(1-x) + A_2(1+3x)^2(1-x)^2 + A_3(1+3x)(1-x)^3 + A_4(1-x)^4. && (7)
\end{aligned}
$$

This gives us five linear equations:

$$
\begin{aligned}
4 =& \ 1 + A_1 + A_2 + A_3 + A_4 && (8)\\
4B_1 =& \ 12 + 8A_1 + 4A_2 + 0A_3 - 4A_4 && (9)\\
4B_2 =& \ 54 + 18A_1 - 2A_2 - 6A_3 + 6A_4 && (10)\\
4B_3 =& \ 108 + 0A_1 - 12A_2 + 8A_3 - 4A_4 && (11)\\
4B_4 =& \ 81 - 27A_1 + 9A_2 - 3A_3 + A_4. && (12)
\end{aligned}
$$

Applying now $A_1 = B_1$ and $B_j \geq A_j$, we find

$$
\begin{aligned}
A_1 + A_2 + A_3 + A_4 =& 3 && (13)\\
A_1 + A_2 + 0A_3 - A_4 =& -3 && (14)\\
& && (15)
\end{aligned}
$$

for the first two equations. Since $A_j \geq 0$, these two equations already tell us that $A_4 = 3$, and $A_1 = A_2 = A_3 = 0$, giving us a weight enumerator of $A(x) = 1 + 3x^4$. We already know a [[4,2,2]] code with this weight enumerator (from problem 1a on problem set #2), so there is no need to calculate the remaining inequalities or the quantum shadow enumerator, as we know they will be satisfied. We find the code's dual enumerator to be

$$
B(x) = 1 + 18x^2 + 24x^3 + 21x^4. \qquad (16)
$$