ENEE 420
FALL 2012
COMMUNICATION SYSTEMS

ANSWER KEY TO TEST # 1:

1. ____

1.a. As shown in class, $H_2(\boldsymbol{u}) = H_2(M) = \log_2 M$.

1.b. The optimal code $C^\star_{2^L} : \{1, \ldots, 2^L\} \to \{0,1\}^\star$ is the one that corresponds to the full tree with 2^L terminal nodes (labelled $1, \ldots, 2^L$). Every codeword having length L, this code is indeed optimal since its average code length coincides with the entropy of the source, namely $H_2(M) = \log_2 M = \log_2 \left(2^L\right) = L$. One way to describe $C^\star_{2^L}$ is as follows: For each $m = 1, \ldots, 2^L$, write $C^\star_{2^L}(m)$ as the L-bit binary expansion of $m-1$.

1.c. Consider now the general case $M = 2^L + K$ with integers L and K satisfying

$$L = 1, 2, \ldots \quad \text{and} \quad K = 0, \ldots, 2^L - 1.$$

With code $C^\star_{2^L} : \{1, \ldots, 2^L\} \to \{0,1\}^\star$ described in Part **1.b**, we first set

$$C^\star_M(m) = C^\star_{2^L}(m), \quad m = 1, \ldots 2^L - K.$$

Next, on the remaining range $m = 2^L - K + 1, \ldots, 2^L + K$, group the symbols in pairs, say

$$2^L - K + 2k + 1 \quad \text{and} \quad 2^L - K + 2(k+1), \quad k = 0, 1, \ldots, K-1.$$

The corresponding codewords are now defined by

$$C^\star_M(2^L - K + 2k + 1) = [C^\star_{2^L}(2^L - K + k), 0]$$

and

$$C^\star_M(2^L - K + 2(k+1)) = [C^\star_{2^L}(2^L - K + k), 1].$$

This corresponds to the following procedure (up to a relabeling of the nodes): Starting with the full binary tree associated with $C^\star_{2^L}$ (for the alphabet $\{1, \ldots, 2^L\}$), keep the terminal nodes corresponding to the symbols $m = 1, \ldots, 2^L - K$, and at each of the remaining nodes (which correspond to the symbols $m = 2^L - K + 1, \ldots, 2^L$ for the alphabet $\{1, \ldots, 2^L\}$), extend the tree by adding its two siblings. This code is optimal because it will be produced by the Huffman algorithm.

It is immediate that $M - K$ codewords have length L, and that $2K$ codewords have length $L+1$, so that

$$\begin{aligned} L(M) &= \frac{1}{M}\left((2^L - K)L + 2K(L+1)\right) \\ &= \frac{1}{M}\left((2^L + K)L + 2K\right) \\ &= L + 2\left(\frac{K}{M}\right). \end{aligned} \tag{1.1}$$

1.d. To have $L(M) = H_2(M)$ means that the optimal code $C_M^\star$ achieves the entropy bound. However, we know that this happens if and only if the underlying pmf is of the form

$$\frac{1}{M} = 2^{-m(x)}, \quad x = 1, \ldots, M$$

with positive integers $m(1), \ldots, m(M)$. Obviously this requires $m(1) = \ldots = m(M) = m^\star$ with $m^\star$ determined by

$$\frac{1}{M} = 2^{-m^\star}.$$

Put another way, the equality $L(M) = H_2(M)$ requires that M be a power of two!

2. ____

2.a. Here

$$\boldsymbol{G} = [1111|1] \quad \text{with} \quad \boldsymbol{P} = [1111].$$

2.b. Since C is a linear $(5,1)$-block code, we get $n = 5$ and $k = 1$, so that there are $2^4 = 16$ distinct cosets.

2.c. To construct the standard array we recall that for this repetition code we have

$$\mathcal{C} = \{00000, 11111\}.$$

See table below.

2.d. The binary vector $\boldsymbol{r} = (11001)$ is received. Going to the standard array, we check that $\boldsymbol{r}$ is in the coset C_8 with leader $\boldsymbol{x}_8 = (00110)$, and that

$$(11001) = \boldsymbol{x}_8 + \mathbf{1}_5$$

so that the decoding will return the codeword $\mathbf{1}_5$, hence the binary $\hat{m} = 1$ was sent.

2.e. When using a repetition code, Nearest Neighbor decoding is equivalent to majority decoding. As a result, the received vector $\boldsymbol{r} = (11001)$ is decoded into $\widehat{\boldsymbol{c}}_{\text{Near}} = (11111)$.

ℓ	Leader $\boldsymbol{x}_\ell$		
1	00000	00000	11111
2	00001	00001	11110
3	00010	00010	11101
4	00100	00100	11011
5	01000	01000	10111
6	10000	10000	01111
7	00011	00011	11100
8	00110	00110	11001
9	01100	01100	10011
10	11000	11000	00111
11	00101	00101	11010
12	01010	01010	10101
13	10100	10100	01011
14	01001	01001	10110
15	10010	10010	01101
16	10001	10001	01110

3. ____

3.a.

$$0001001100001110011110001100011011000111011011100\ldots \tag{1.2}$$

3.b. The resulting bit stream will be

00000 0	00001 0	00000 1	00010 1	00011 0	00010 0	00011 1	00101 0	00111 1

01000 0	00111 0	00100 1	00001 1	01010 1	01011 1	00101 1	01011 0

N	N binary	Phrase	Codeword
0	00000	(empty)	
1	00001	0	00000-0
2	00010	00	00001-0
3	00011	1	00000-1
4	00100	001	00010-1
5	00101	10	00011-0
6	00110	000	00010-0
7	00111	11	00011-1
8	01000	100	00101-0
9	01001	111	00111-1
10	01010	1000	01000-0
11	01011	110	00111-0
12	01100	0011	00100-1
13	01101	01	00001-1
14	01110	10001	01010-1
15	01111	1101	01011-1
16	10000	101	00101-1
17	10001	1100	01011-0

3.c.

00000 0	00001 0	00000 1	00010 1	00011 0	00010 0	00011 1	00101 0	00111 1

01000 0	00111 0	01000 1	00001 1	01010 1	01011 1	00100 1	01011 0

There is no reason for the receiver to conclude that a transmission error has occurred: Indeed, although $\boxed{\text{00101 1}}$ has been received incorrectly as $\boxed{\text{00100 1}}$, this error does not prevent from the decoding operation to proceed, resulting in 0011 being decoded instead of 101.

N	Received codeword	N binary	Decoded phrase
0		00000	(empty)
1	00000-0	00001	0
2	00001-0	00010	00
3	00000-1	00011	1
4	00010-1	00100	001
5	00011-0	00101	10
6	00010-0	00110	000
7	00011-1	00111	11
8	00101-0	01000	100
9	00111-1	01001	111
10	01000-0	01010	1000
11	00111-0	01011	110
12	01000-1	01100	1001
13	00001-1	01101	01
14	01010-1	01110	10001
15	01011-1	01111	1101
16	00100-1	10000	0011
17	01011-0	10001	1100

3.d.

00000 0	00001 0	00000 1	00010 1	00011 0	00010 0	00011 1	00101 0	00111 1

00001 1	11000 0	00111 0	01000 1	01010 1	01011 1	00101 1	01011 0

There is reason for the receiver to conclude that a transmission error has occurred: The tenth received codeword $\boxed{11000\ 0}$ cannot be decoded because it calls for adding bit value 0 to the pattern to be found at $N = 24 = 11000$ which is larger that 11, and so has not beeen constructed yet!

It should be pointed out that the previous codeword $\boxed{01000\ 0}$ is already transmitted in error as codeword $\boxed{00001\ 1}$. However, as in Part **3.c** the receiver will not be able to conclude that a transmission error has occurred.

N	Received codeword	N binary	Decoded phrase
0		00000	(empty)
1	00000-0	00001	0
2	00001-0	00010	00
3	00000-1	00011	1
4	00010-1	00100	001
5	00011-0	00101	10
6	00010-0	00110	000
7	00011-1	00111	11
8	00101-0	01000	100
9	00111-1	01001	111
10	00001-1	01010	01
11	11000-0	01011	Cannot decode
12	01111-0	01100	
13	01000-1	01101	
14	01010-1	01110	
15	01011-1	01111	
16	00101-1	10000	
17	01011-0	10001	

4.

4.a. The encoding

$$C(\boldsymbol{m}) = (\mathtt{Par}(\boldsymbol{m}), \boldsymbol{m}, \boldsymbol{m}), \quad \boldsymbol{m} \in \mathcal{H}_k \tag{1.3}$$

provided by the code $C : \mathcal{H}_k \to \mathcal{H}_n$ can be expressed as

$$C(\boldsymbol{m}) = \boldsymbol{m}\boldsymbol{G}$$

where the generating $\boldsymbol{G}$ is the $k \times (2k+1)$ matrix given by

$$\boldsymbol{G} = [\boldsymbol{P}|\boldsymbol{I}_k] \quad \text{with} \quad \boldsymbol{P} = \left[\mathbf{1}_k^t|\boldsymbol{I}_k\right].$$

4.b. Here it is plain that the $(k+1) \times n$ matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \left[\boldsymbol{I}_{k+1} \middle| \begin{array}{c} \mathbf{1}_k \\ \boldsymbol{I}_k \end{array} \right].$$

4.c. Note that

$$\boldsymbol{H}^t = \left[\begin{array}{c} \boldsymbol{I}_{k+1} \\ \mathbf{1}_k^t|\boldsymbol{I}_k \end{array} \right],$$

so that

$$\begin{aligned}
\mathcal{C} &= \left\{ \boldsymbol{x} \in \mathcal{H}_n : \boldsymbol{x}\boldsymbol{H}^t = \boldsymbol{0}_{k+1} \right\} \\
&= \left\{ (x, \boldsymbol{y}, \boldsymbol{z}) \in \mathcal{H}_n : \begin{array}{c} x \in \{0,1\} \\ \boldsymbol{y}, \boldsymbol{z} \in \mathcal{H}_k \end{array} \quad \text{and} \quad \begin{array}{c} x + \boldsymbol{z}\boldsymbol{1}_k^t = 0 \\ \boldsymbol{y} + \boldsymbol{z} = \boldsymbol{0}_k \end{array} \right\} \\
&= \left\{ (x, \boldsymbol{y}, \boldsymbol{z}) \in \mathcal{H}_n : \begin{array}{c} x \in \{0,1\} \\ \boldsymbol{y}, \boldsymbol{z} \in \mathcal{H}_k \end{array} \quad \text{and} \quad \begin{array}{c} x + \texttt{Par}(\boldsymbol{z}) = 0 \\ \boldsymbol{y} = \boldsymbol{z} \end{array} \right\} \\
&= \left\{ (\texttt{Par}(\boldsymbol{z}), \boldsymbol{z}, \boldsymbol{z}) : \ \boldsymbol{z} \in \mathcal{H}_k \right\},
\end{aligned} \tag{1.4}$$

as expected.

4.d. With $\boldsymbol{c} = (\texttt{Par}(\boldsymbol{z}), \boldsymbol{z}, \boldsymbol{z})$ for some $\boldsymbol{z}$ in $\mathcal{H}_k$, we have $\boldsymbol{c} = \boldsymbol{0}_n$ if and only if $\boldsymbol{z} = \boldsymbol{0}_k$. With this in mind,

$$\begin{aligned}
d_H(\mathcal{C}) &= \min \left(w_H(\boldsymbol{c}) : \begin{array}{c} \boldsymbol{c} \neq \boldsymbol{0}_n \\ \boldsymbol{c} \in \mathcal{C} \end{array} \right) \\
&= \min \left(w_H(\texttt{Par}(\boldsymbol{z}), \boldsymbol{z}, \boldsymbol{z}) : \begin{array}{c} \boldsymbol{z} \neq \boldsymbol{0}_k \\ \boldsymbol{z} \in \mathcal{H}_k \end{array} \right) \\
&= \min \left(\texttt{Par}(\boldsymbol{z}) + 2w_H(\boldsymbol{z}) : \begin{array}{c} \boldsymbol{z} \neq \boldsymbol{0}_k \\ \boldsymbol{z} \in \mathcal{H}_k \end{array} \right) \\
&= 3
\end{aligned} \tag{1.5}$$

as easily seen through the following argument: First it is plain that $d_H(\mathcal{C}) \leq 3$ since

$$d_H(1, \boldsymbol{z}, \boldsymbol{z}) = 3$$

if we select $\boldsymbol{z}$ in $\mathcal{H}_k$ with exactly one non-zero component. We now show that it is not possible to find $\boldsymbol{z} \neq \boldsymbol{0}_k$ in $\mathcal{H}_k$ such that

$$d_H(1, \boldsymbol{z}, \boldsymbol{z}) = 2.$$

This would amount to

$$\texttt{Par}(\boldsymbol{z}) + 2w_H(\boldsymbol{z}) = 2.$$

If $\texttt{Par}(\boldsymbol{z}) = 0$, then $2w_H(\boldsymbol{z}) = 2$, i.e., $w_H(\boldsymbol{z}) = 1$ and so $\texttt{Par}(\boldsymbol{z}) = 1$, a contradiction. On the other hand, if $\texttt{Par}(\boldsymbol{z}) = 1$, then $2w_H(\boldsymbol{z}) = 1$ and a contradiction again arises.

It is also not possible to find $\boldsymbol{z} \neq \boldsymbol{0}_k$ in $\mathcal{H}_k$ such that

$$d_H(1, \boldsymbol{z}, \boldsymbol{z}) = \texttt{Par}(\boldsymbol{z}) + 2w_H(\boldsymbol{z}) = 1.$$

Indeed, this requirement leads necessarily to $w_H(\boldsymbol{z}) = 0$ and $\texttt{Par}(\boldsymbol{z}) = 1$, with the latter contradicting the former!

4.e. Because $d_H(C) = 3 = 2 + 1$, all error patterns with exactly two bit reversals will be detected. Some error patterns with three bit reversals will not be detected, e.g. with $k = 3$, $\boldsymbol{m} = (1, 1, 1)$, $\boldsymbol{c} = (1, 1, 1, 1, 1, 1, 1)$ and $\boldsymbol{r} = (0, 0, 1, 1, 0, 1, 1)$ – Contrast this with the fact that all odd-numbered bit reversals would be detected if bit parity check codes are used. Some error patterns with four bit reversals will be detected, e.g., with $k = 3$, $\boldsymbol{m} = (1,1,1)$, $\boldsymbol{c} = (1,1,1,1,1,1,1)$ and $\boldsymbol{r} = (0,0,0,1,1,0,1)$ – Again contrast this with the situation for bit parity check codes.

4.f. Because $d_H(C) = 3 = 2.1 + 1$, a single error (or bit reversal) will always be corrected under Nearest Neighbor decoding.
