CSC 404 - Activity/Project 14 - Name: Chris Glanzer

Remark 1. The generating matrix G and transpose of the parity check matrix, H, in a [7,4] Hamming Code (modulo 2) is given by

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad H^T = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$i = 2$

Problem 1. For the following, try to do a couple of them by hand. So, you can see the magician tricks in action.

a. For a message $x = 1001 \sim (1, 0, 0, 1)$.

i. What is the resulting codeword $c = xG$? → 1001001

11 + 00 + 01 + 11 = 2 = 0

11 + 01 + 00 + 11 = 2 = 0

11 + 01 + 01 + 10 = 1 = 1

ii. Show that the syndrome, $s = cH^T$, results in $(0, 0, 0)$.

1001001

no errors

b. Suppose you receive the codeword $c = 0111101 \sim (0, 1, 1, 1, 1, 0, 1)$

i. Compute the syndrome, $s = c_1H^T$.

01 + 10 + 11 + 11 + 11 + 00 + 10 = 3 = 1

0 0 1 1 1 0 0

01 + 11 + 10 + 11 + 10 + 01 + 10 = 2 = 0

0 1 0 1 0 0 0

01 + 11 + 11 + 10 + 10 + 00 + 11 = 3 = 1

0 1 1 0 0 0 1

s = 101, index 2

0101101

ii. Determine the position, j, of the syndrome within H^T.

iii. Change the jth bit in the received codeword. What was the original, $k = 4$, bit message?

0101

c. Suppose you receive the codeword $c = 0001100 \sim (0, 0, 0, 1, 1, 0, 0)$

i. Compute the syndrome, $s = c_1H^T$. new way of writing

0001100

1011100 = 2 = 0

1101010 = 1 = 1

1110001 = 0 = 0

010

ii. Determine the position, j, of the syndrome within H^T.

j = 5 0001110

iii. Change the jth bit in the received codeword. What was the original, $k = 4$, bit message?

0001

Remark 2. You may have noticed this already, but there is a specific structure the matrix H^T takes. For $r = 4$, each row/entry is a unique non-zero 4-bit number – i.e., it contains 0001 through 1111 – Neat! For H^T we place the entries with a single 1 on the bottom forming a 4×4 identity matrix and everything else lives on top. For a general r we construct H^T as $2^r - 1$ rows where each row/entry is a unique non-zero r bit number and place the identity matrix on the bottom!

$$H^T = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ 10 \rightarrow \\ \\ \\ \\ 1100 \end{array}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{array} \qquad G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}\begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{array}$$

5 3 4 4

5 4 3 4
1 0 1 0

We are actually free to rearrange the non identity matrix rows anyway we want (as long as we keep the same organization in the matrix G. An easy way to generate H^T is to cycle through numbers 1 to $2^r - 1$ and place their binary representation (with possible padded zeros) in to H^T making sure to place (append!) the single 1 terms in an 'identity matrix' shape on the bottom of H^T. See Replit Link for a version of this – I highly recommend trying to reconstruct a version of this.

Problem 2. (Bigger Hamming $[n, k]$ Codes) Alright, let's work with some bigger Hamming Codes! How about we use $r = 4$ to allow $k = (2^4 - 1) - 5 = 11$ bit messages. (In general, $n = 2^r - 1$ and $k = n - r$.) Use technology to compute the following – you can do these by hand, but it is a pain :-).

a. For the message $x = 10101010101$ compute the corresponding codeword $c = xG$. Then show the syndrome $s = cH^T$ results in 0000 (i.e., with no errors in the codeword we should have all zeros in the syndrome.).

10101010101|1010

b. Suppose you receive the codeword $c_1 = 101001110001010$. Compute the syndrome $s = c_1 H^T$.

S= 1100

c. Determine the position, j, of the syndrome within H^T. (See Above/Replit for H^T)

J= 10

10 100111 001 1010

d. Change the jth bit in the received codeword and output the resulting k-bit message. (Bonus Fun - turn the resulting message back into a decimal number)

10100 111 001 1010

Magic Man...