ECSE 308 Notes

Me

October 15, 2022

Contents

1 Chapter A

2 Chapter B

3 Chapter C Notes

3.1 C1 Parity Check

Definition. For any k-bit sequence, $d = d_i$, $0 \leq i \leq k-1$. We can append another bit, d_k so that the transmitted sequence is $k+1$ bits, with.

For even parity, $d_k = \oplus_{i=0}^{k-1} d_i$. The transmitted $k+1$ bit sequence has an even number of bits. The receiver can verify by taking $\oplus$ over all its $k+1$ entries, and

$$r \in \mathbb{B}^{k+1} \text{ is valid } \iff \bigoplus_{i=0}^{k} r_i = 0$$

What to Remember (Even/Odd Parity check).

1. For a k-bit sequence, it becomes a $k+1$-bit sequence.
2. Even parity, the parity check bit is
$$d_k = \bigoplus_{i=0}^{k-1} d_i$$
3. To verify even parity, if $r \in \mathbb{B}^{k+1}$ is valid, if and only if
$$\bigoplus_{i=0}^{k} r_i = 0$$
4. In like fashion, for odd parity, the parity check bit is
$$d_k = 1 \oplus \left(\bigoplus_{i=0}^{k-1} d_i \right) = \bigodot_{i=0}^{k-1} d_i = 0$$
5. To verify odd parity, if $r \in \mathbb{B}^{k+1}$ is valid, (recall that $\oplus$ is the odd counting function), if and only if
$$\bigoplus_{i=0}^{k} r_i = 1$$

3.2 C2 Internet Checksum

Definition. Fix a bit sequence of size $M \times l$, we can view this as a matrix with m rows and l columns. If d is the sequence, then

$$d = \left[d_1, d_2, \cdots, d_m\right]$$

Where each d_m is a vector of size l on its own. So d is a vector containing vectors.

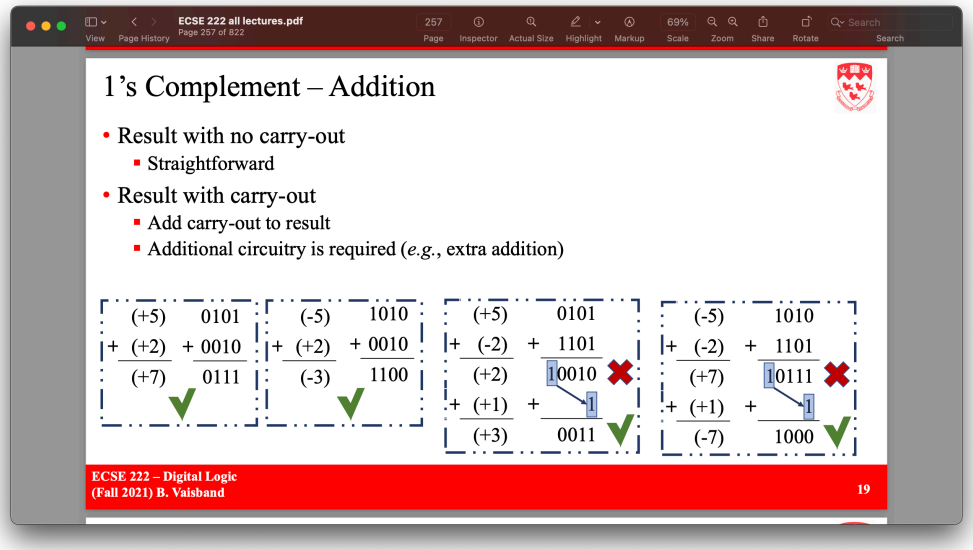
The checksum d_0 is defined inductively. For each $1 \leq m \leq M$,

- We first initialize $d_0 = \vec{0}$, and
- for each $m \in [1, M]$, $d_0 = d_m + d_0$. This is done by one's complement addition.

  One's Complement Addition, fix two vectors of size l, $\vec{x}$, $\vec{y}$, and consider their binary representations, and add them together.

  $$\vec{x} + \vec{y} = \left(\sum_{i=0}^{l-1} x_i 2^i\right) + \left(\sum_{i=0}^{l-1} y_i 2^i\right) = \sum_{i=0}^{l} s_i 2^i$$

  The result may be a $l+1$ digit number. If this is so, then the carry-out of the sum is added again to the result. Do this for all $m \leq M$.



- At the end of the process, flip all the bits in d_0

$$d_0 = \Big(1 \oplus d_{l-1}, \cdots 1 \oplus d_0\Big)$$

The result is the following error-checking scheme.

What to Remember (Internet Checksum).

1. If there is no error, then
$$\boxed{\sum_{j\geq 0}^{M} d_j = 0}$$
2. The internet checksum can detect all one-bit error patterns.
3. For a sequence of bits divided in M chunks, the transmitted sequence has $M+1$ chunks.
4. The internet checksum d_0 is obtained by adding all d_j in one's complement, and flipping all the bits (like in one's complement).

3.3 C3 Linear Codes

Definition. Any non-empty, subspace C of F^n with dimension k where $1 \leq k \neq n$ is called a (n, k) linear code. In particular,

If $F + \mathbb{Z}_2$, the n-parity check code is a subspace.

3.3.1 Introduction to Hamming Weights

Fix a code $C \subseteq F^n$, where F^n in this section will always denote the finite cartesian product of $F = GF(p)$. Elements in C are called code words. For any $c \in C$, we define its Hamming Weight

$$\text{wt}(c) = \Big|\{i,\, a_i \neq 0\}\Big|$$

$\text{wt}(c)$ simply counts the number of non-zero digits. Therefore

- For every $x \in C$, $0 \leq \text{wt}(x) \leq n$,
- If $x \in C$, $\text{wt}(x) = 0 \iff x = 0$ (recall that C is a subspace of F^n, so $0 \in C$),
- $\text{wt}(\cdot)$ induces a metric on C, $d(\cdot, \cdot)$

$$d(x, y) = \text{wt}(x - y), \quad \forall x,\, y \in C$$

To verify that the properties of a metric all hold for this $d(x, y)$ on C

1. $d(x, x) = 0 \iff \text{wt}(x - x) = 0 \iff x - x = 0 \iff x = x$,
2. For every pair of elements $x, y \in C$, $d(x, y) \geq 0$. This comes from the fact that $x - y \in C$ and $\text{wt}(x - y) \geq 0$,
3. If $v, u, w \in C$ then

$$d(v, w) \leq d(v, u) + d(u, w)$$

This is an easy consequence of

$$\text{wt}(a + b) \leq \text{wt}(a) + \text{wt}(b) \tag{1}$$

If we write $v - u = a$, and $u - w = b$, then $a + b = v - w$. Indeed, if $a = (a_i)$, $b = (b_i)$, then if for some $1 \leq j \leq n$ such that $a_j + b_j \neq 0$ then either $a_j \neq 0$ or $b_j \neq 0$. Therefore Equation (1) holds.

Furthermore, define a ball $B_r(w)$ about some $w \in F^n$ as follows

$$B_r(w) = \{x \in F^n,\, d(w,x) \leq r\} \tag{2}$$

Notice how this differs from a regular ball in a metric space. We now turn our attention to a quantity which will be of great interest. Fix a (n,k) linear code C, the minimum distance d of C is defined to be the smallest distance between two distinct code words in C, precisely

$$d = \min\{d(v,w),\, v,\, w \in C,\, v \neq w\} \tag{3}$$

This immediately implies that $d \geq 1$. If our code contains only one element (namely if C is the trivial subspace), then $d = +\infty$.

Definition. A code C uses the Nearest Neighbour Decoding if w is a received word, and if $w \notin C$, then w is decoded as the code word in C that is the closest to it with respect to $d(\cdot,\cdot)$. If $w \in C$ then it is decoded as w. If there is a tie, then the nearest neighbour is chosen arbitrarily.

WTS. If C is an n-code with minimum distance d, (meaning that it does not have to be a subspace). If C uses Nearest Neighbour Decoding, and if $t = d(w,c)$, there w and c are the received and transmitted words, then

- If $t < d$, then C can detect t errors,
- If $2t < d$, then C can correct t errors.

Proof. Let $c \in C$ and the received code w satisfies $d(w,c) = t < d$, so the received code is t distance away from the original code. Since $w \notin C$ by definition of d (if $w \in C$ then that would mean $d \leq t$). C can detect such an error, and we say that C can correct $t < d$ errors.

If $2t < d$, then w lies only within the t-ball of c. This is because the t-balls about the code words in C are pairwise disjoint. Indeed, if $w \in B_t(c) \cap B_t(c')$ for another code word $c' \in C$, then

$$d \leq d(c',c) \leq d(w,c) + d(w,c') \leq 2t$$

Leading to a contradiction. $\square$

We are ready to prove the main theorems in this Chapter concerning Linear Codes.

WTS. Let C be an (n, k)-code that uses Nearest Neighbour Decoding, and d be the same as above. Then

1. $d = \min\{\text{wt}(w),\, w \in C,\, w \neq 0\}$,
2. C can detect $t \geq 1$ errors if and only if $t < d$,
3. C can correct $t \geq 1$ errors if and only if $2t < d$,
4. If C can correct $t \geq 1$ errors, and denote $|F| = q$ (if $F = \mathbb{Z}_2$, then $|F| = 2$), then

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t \leq q^{n-k}$$

Proof. We skip the proof for now. □

Definition. The coding rate for a (n, k)-linear code is k/n.

Definition. A binary linear code is called MDS (maximum distance-separable) code if and only if d is equal to its upper bound. (Also called Perfect Codes).

$$d = \{d(x, y),\, x, y \in C,\, x \neq y\} = (n - k) + 1$$

3.4 C4 Linear Codes (Generator)

Definition. A systematic linear code refers to the structure in which the k information bits are preserved as the k most significant bits.

This means that they are shifted to the left-most entries in the array.

- Advantages: Ease of extraction without complex mapping

This means that the generator matrix G for code $C \subseteq F^n$, is in the form

$$G = \begin{bmatrix} I_k & A \end{bmatrix}$$

Where A is a $k \times (n-k)$ matrix.

Codeword Generation

Let D be a $1 \times n$ matrix representing a k-bit data stream, with

$$D = \Big(d_{k-1},\, d_{k-2},\, \cdots,\, d_0 \Big)$$

D comes in the form of a row vector, and we right-multiply it with a $k \times n$ matrix G to obtain the transmitted vector $c \in C$, where $C \subseteq F^n$ is a k-dimensional subspace.

$$c = DG,\, C \in \mathbb{B}^{1 \times n}$$

We call this matrix G, the generator matrix. This matrix always exists given any linear code $C \subseteq F^n$ with dimension k. A few observations must be made,

- Let T be the linear transformation that 'encodes' our k-bit codes into row vectors in F^n, in symbols

  $$T : \mathbb{B}^{1 \times k} \to \mathbb{B}^{1 \times n},$$

  G will be the matrix that represents this linear transformation.

- The generator matrix G has k-independent rows, where each row represents a valid code-word $g_i \in C \subseteq F^n$. Such that any $c \in C$ is in the span of the rows of G.

- The parity check matrix, H is a $(n-k) \times n$, and satisfies

$$GH^T = 0$$

If we left-multiply by some $d \in \mathbb{B}^{1\times k}$ then $dG = c$ for some $c \in C$, and $cH^T = 0$ for every $c \in C$.

If H is a $(n-k) \times n$ matrix, with rank (dimension of column-span) of $(n-k)$, this forces the row-space of H^T to have dimension $(n-k)$, since

$$C = \operatorname{im} T = \{T(x),\, x \in \mathbb{B}^k\}$$

We usually say that H is a $(n-k)\times n$ matrix, with rank (column-span) $n-k$. And this forces the row-space of H^T to have dimension $(n-k)$.

The following equation is of utmost importance, if the rank of H is $n-k$, then

$$\boxed{C = \{w \in \mathbb{B}^n,\, wH^T = 0\}} \tag{4}$$

Every valid codeword $c \in C \subseteq \mathbb{B}^n$ must satisfy this property.

The proof is quite straight-forward, we have already shown that $C \subseteq \{w \in \mathbb{B}^n, wH^T = 0\}$, the second part of the proof just consists of showing that $\dim \ker H^T = (n-k)$. The reader should consult Chapter 8.8 of Nicholson's Linear Algebra for more details.

- Assume that $c \in C$ is transmitted, and the receiver receives some $v \in F^n$, where possibly $v \notin C$, then we define
  - $z = v - c$ as the error,
  - If $v \notin C$, then by Equation (4), $s = vH^T \neq 0$. Such a word $s \in \mathbb{B}^{(n-k)\times 1}$ is called the syndrome.

Note a couple of things, the receiver has the vector v, and wishes to recover $c \in C$. For this task, it suffices to recover z, the error, since we can always subtract $c = v - z$. Furthermore, $cH^T = 0 \iff Hc^T = 0$, and by linearity,

$$c = v - z \implies Hc^T = Hv^T - Hz^T \implies s = Hv^T = Hv^T$$

If we can solve for the error vector v, then we can solve for the original codeword by subtraction, what remains is to solve the homogeneous system of equations $s = Hv^T$.

- The set of all possible errors that can be applied to c is $\mathbb{B}^{1\times n}$, but the syndrome is only a $\mathbb{B}^{(n-k)\times 1}$ vector. So there can be no one-to-one correspondence. The take-away is this, given a syndrome vector $s = Hv^T$, where v is the received vector, the search for the error vector z, has to be reduced from $\mathbb{B}^n$ to $\mathbb{B}^{n-k}$ for a one-to-one correspondence.
- Intuitively speaking, the larger $n - k$, the larger the syndrome vector; and the more likely we can recover the correct error vector, and thus recover c.

3.5 Arithmetic in GF(2)

3.5.1 Multiplication of Polynomials in GF(2)

If we want to multiply $(x^5 + x^2 + 1)(x^3 + 1)$, we write the coefficients of the two polynomials in the form

$$(x^5 + x^2 + 1)(x^3 + 1) \iff (5,\ 2,\ 0)(3,\ 0)$$

Find the set of sums across the two tuples, given by

$$(8,\ 5,\ 5,\ 2,\ 3,\ 0)$$

Then retain only the ones that have an odd number of terms (here the two 5 cancel out), and we get

$$(8,\ 3,\ 2,\ 0) \iff x^8 + x^3 + x^2 + x^0$$

Let us try with another example with $(1 + x)(x + x^2)$

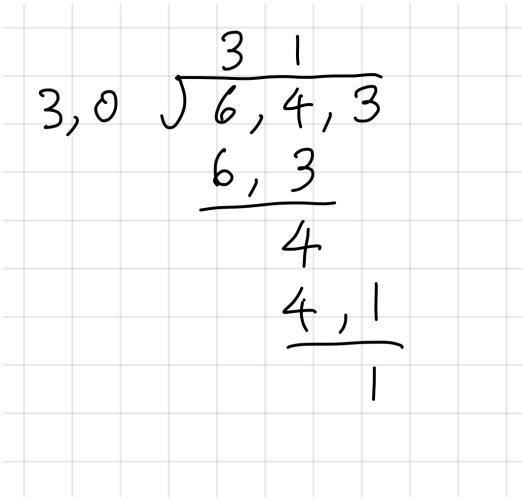
$$(1 + x)(x + x^2) \iff (1,\ 0)(1,\ 2)$$

The set of sums is $(2,\ 3,\ 1,\ 2)$, and flipping the required coefficients we get

$$(3, 1) \implies (1 + x)(x + x^2) = x^3 + x$$

3.5.2 Long Division of Polynomials in GF(2)

Assume that $G(X) = (3, 0) = x^3 + x^0$, and $D(X) = (3, 1, 0) = x^3 + x^1 + x^0$, if we wish to find $R(X)$ then we wish to find the remainder of $\frac{X^3 D(X)}{G(X)}$



3.6 Cyclic Redundancy Check

We begin with some size considerations. Suppose that

- We are given a data vector, D of k-bits. We can represent D using a binary polynomial of $k-1$ degree, denoted by $D(X)$
- In CRC, the transmitter then generates a $n-k$ bit sequence called a Frame-Check Sequence (FCS) or $R(X)$. ($n-k-1$ polynomial),
- The transmitted frame, $T(X)$ is n bits in size. ($n-1$ polynomial).
- For any CRC scheme, where n, and k are fixed numbers. There has to be a generator polynomial, $G(X)$ of degree $n-k$.
- $R(X)$ is the remainder when $X^{n-k}D(X)$ is divided by $G(X)$.
- $X^{n-k}D(X)$ simply shifts the bits in $D(X)$ leftwards by $n-k$. $R(X)$ has to have degree $n-k$.
- The transmitted frame $T(X) = X^{n-k}D(X) + R(X)$.

The following example should enlighten things a bit. Let $G(X) = x^3+1$, and the corresponding information bits be $D = [1101] = x^3 + x^2 + x^0$. Degree on $G(X)$ is $3 = n-k$, and Degree on $D(X)$ is $3 = k-1$, therefore $k = 4$ and $n-7$. Shifting $D(X)$ by 3 bits to the left yields

$$X^{n-k}D(X) = X^3D(X) = x^6 + x^5 + x^3$$

Dividing $X^{n-k}/G(X)$ give us the remainder $R(X) = x^1$. Since $R(X)$ must have degree $n-k-1 = 2$, we can write $R(X) = [010]$. And the transmitted frame is $T(X) = [1101010]$.

What to Remember. Cyclic Redundancy Check Computations

Vector sizes

1. D: Data Vector, k bits. Degree: $k-1$,
2. G: Generator Polynomial, Degree: $n-k$,
3. R: Remainder/Frame Check Sequence, Degree: $n-k-1$.

Computations

1. Given D and G. Determine n, and k by

$$\deg G = n - k \tag{5}$$
$$\deg D = k - 1 \tag{6}$$

2. Shift D by $n - k$ bits.
3. Compute $R(X)$ by remainder of $X^{n-k}D(X)/G(X)$,
4. Concatenate $T(X) = X^{n-k}D(X) + R(X)$.
5. Extra: Verify that $T(X)$ has n-bits in total. (A polynomial of $n - 1$ degree).

Factoids

1. Given a transmitted codeword $T(X)$ from a generator polynomial $G(X)$. It is always possible to fool the scheme by using an error $E(X) = G(X)$. This is because $T(X) + E(X) = T(X) + G(X)$ divides $G(X)$.
2. Simple to implement,
3. Well suited to detect burst errors (useful because common transmission errors are burst errors),
4. Irreducible polynomial of degree m, not divisible by any polynomial with degree $0 < k < m$.
5. Primitive polynomial of degree m, not divisible by any polynomial in the form $x^k + 1$, $0 < k < 2^m$
6. If G is an irreducible polynomial of degree L, then CRC can detect all BURST errors of $0 < k < L$.
7. If G is a primitive polynomial of degree L, then CRC can detect all two-bit errors separated by $N < 2^L$ bits.
8. A factor of $(x + 1)$ detects any odd number of bit errors.
9. Single error patterns are detected by G having two or more terms.

4 Chapter C Exercises

4.1 Problem C1

WTS.

Proof. □

4.2 Problem C2

WTS. Assume that transmitting data is divided into chunks of 3-bit long, and then odd-parity is applied to form a parity codeword. In the new 4-bit parity codeword space, list all the valid and invalid codewords.

Proof. We are using odd parity, so for any $c \in \mathbb{B}^4$,

$$c \in C \iff \bigoplus_{i=0}^{3} c_i = 1$$

Equivalently, $c \notin C \iff \bigoplus_{i=0}^{c} c_i = 0$. Let us list C^c, the complement of C.

$$C^c = \{[0000], [1111], [0011], [0101], [1001], [0110], [1010], [1001]\}$$

And therefore

$$C = \{[0001], [0010], [0100], [1000], [1110], [1101], [1011], [0111]\}$$

□

4.3 Problem C3

WTS. Using the 16-bit Internet Checksum:

(a) Calculate the Internet checksum for the four 16-bit words of data as follows: E308 E309 00FF 0F0F.

(b) Suppose that the received data and checksum are contaminated with error as follows: E308 E309 00FF 0F0E 29DF. Verify that the internet checksum can detect the error.

(c) Find an example of a 2-bit error that can fool Internet checksum for the above data and checksum.

Proof. Part a. Procedure for calculating the internet checksum:

1. Using HEX mode on Calculator, add all the data bits together.

$$E308 + E309 + 00FF + 0F0F = 0001D61F$$

2. Since there is a carry-out of '1', we subtract 10000 and add 1, equivalently:

$$0001D61F - 10000 + 1 = D620$$

3. Taking the one's complement, (bitwise XOR) of the sum. This is equivalent to subtracting with FFFF, therefore

$$FFFF - D620 = 29DF$$

Part b. Verify that the internet checksum detects the error. The procedure is as follows

1. Using HEX mode on Calculator, add all the shit together again,

$$E308 + E309 + 00FF + 0F0E + 29DF = 0001FFFD$$

2. Carry-out of '1', we need to subtract 10000 and add 1, hence

$$0001FFFD - 10000 + 1 = FFFE$$

3. Take one's complement. Which equates to

$$FFFF - FFFE = 0001 \neq 0000$$

4. Therefore the internet checksum detects the error.

Part c. We note that $\boldsymbol{E309}$ and $\boldsymbol{E308}$ are one bit different from each other, therefore we can apply the error which swaps $\boldsymbol{9}$ with $\boldsymbol{8}$. Resulting in E309 E308 00FF 0F0F 29DF.

□

4.4 Problem C4

WTS.

Proof. □

4.5 Problem C5

WTS.

Proof. □

4.6 Problem C6

WTS. Find the Hamming distance between the following codewords: A = [000000], B = [101010], C = [000111]

Proof. This is trivial, $d(A, B) = \mathbf{wt}(B) = 3$, $d(A, C) = \mathbf{wt}(C) = 3$, and $d(B, C) = 4$. The minimum distance is 3. □

4.7 Problem C7

WTS. Consider the coding scheme in problem 6, what is the maximum number of detectable bit errors and the maximum number of bit errors that can be corrected? If the receiver receives a codeword of R = [111010], which valid codeword should it correct to? Explain why.

Proof. The maximum number of detectable bit errors is 2. The maximum number of bit errors that can be corrected is 1.

If the receiver receives $\boldsymbol{R} = [111010]$, then it assumes there is 1 error (since it can only correct 1 error). Flipping the second bit corrects $\boldsymbol{R}$ to $\boldsymbol{B} = [101010]$. □

4.8 Problem C8

WTS.

Proof. □