**🖊 MTH2002** Coding Theory Coursework

School of Engineering and Physical Sciences, University of Lincoln

Tuesday, 8th October 2024

*There may be some errors in the rendering of this document, in these rare cases then please reference the original document:* [*https://bilbois.me/Unorganised/Coding-Theory-Coursework*](https://bilbois.me/Unorganised/Coding-Theory-Coursework)*.*

### Problem CW1 (*25 total marks*)

Consider the code consisting of the following codewords….

#### Part A (10 marks)

First, we can calculate the Hamming distance between each codeword, as follows:

We can then find the minimal distance of the code and use it to determine how many errors it can detect/correct with the *Definition of Distance Theorem*:

Given that has a **minimal distance** ,

1. , where , then **detects errors**; hence .
2. , where , then **corrects errors**; hence .

#### Part B (15 marks)

To find all the words in which follow the two constraints:

1. Their first symbol is equal to ,
2. They have a Hamming distance of exactly from every codeword of simultaneously, i.e. their second symbol isn’t , , or , and their third symbol isn’t , , or .

We can first list all the words which meet the first constraint:

And then **filter them** by the second constraint:

### Problem CW2 (*10 marks*)

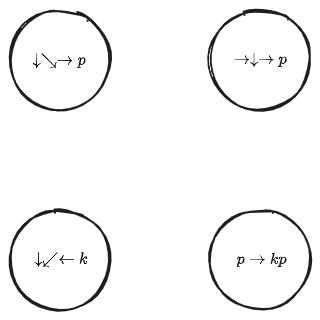
Consider the code in a symmetric channel with symbol error probability Suppose the codeword is sent and is received with exactly one erroneous symbol.

We can find the **probability** of this occurring by determining the ways that it can happen, , and then multiplying this by the probability of each symbol being as we want: one symbol being erroneous, , and three symbols being correct, …

### Problem CW3 (*35 total marks*)

The following code is built to encode certain special moves of a prominent videogame as a combination of inputs from a controller…

#### Part A (5 marks)

We can **draw a graph** where the elements of are represented by vertices which are only connected by an edge if (and only if) their Hamming distance is exactly one:

With this graph, we can count how many edges there are and hence how many elements have a Hamming distance of exactly one. However, there are **no pairs of elements with a Hamming distance of exactly one**, the minimum is two between and .

#### Part B (10 marks)

The **four parameters** of are:

* ,
* ,
* ,
* .

#### Part C (10 marks)

To determine how many errors can detect and correct we can use the *Definition of Distance Theorem*:

Given that has a **minimal distance** (found in the previous part),

1. , where , then **detects errors**; hence .
2. , where , then **corrects errors**; hence .

#### Part D (10 marks)

Given a space of words of length four, over the alphabet used to build , we can calculate the **volume of a solid sphere of radius and centre** , i.e. , by using the *Definition of the Volume of a Solid Hamming Distance Sphere*:

Practically, this represents the number of words with Hamming distance 2 from the word .

### Problem CW4 (*30 total marks*)

The following table lists all codewords of the original code proposed by R. Hamming:

#### Part A (15 marks)

We can prove that **Hamming’s original code is perfect** by using the *Definition of a Perfect Code in Coding Theory*:

If this holds…

Then the code is perfect. Hence, using the *Definition of the Volume of a Solid Hamming Distance Sphere* and given that the parameters of Hamming’s original code are ,

#### Part B (15 marks)

Suppose that some codeword of Hamming’s original code was sent, but the word was received.

The code can reliable correct one erroneous symbol using the nearest neighbour decoding strategy, but if we assume there are two errors in transmission then issues arise since multiple codewords can be Hamming equidistant.

For the word , the following all have a Hamming distance of two:

* ,
* ,
* .

**Hence, there is a chance that the code corrects to the original codeword, and a chance that it doesn’t**.