Coding Theory Homework

Week 4 (Section 1.11 - 1.12)

Exercise 1.11.2b

> Let C = {001, 101, 110}. Determine wether *C* will detect the error pattern 001

001 + 001 = 000
101 + 001 = 100
110 + 001 = 111

None of the three words 000, 100 or 111 is in C so detects the error pattern 001.

Exercise 1.11.4

> Which error patterns will the code *C* = K^n detect?

None, since *C* contains all words there will always be atleast one sum that will be in *C*.

Exercise 1.11.5

> 1. Let *C* be a code which contains the zero word as a codeword. Prove that if the error pattern u is a codeword, then *C* will not detect u.

C = {000, 010, 110}
u = 010

000 + 010 = 010
010 + 010 = 000
110 + 010 = 100

When adding the codeword u to the zero word the resulting word will always be the same as the codeword u itself. Since this codeword is in *C* the error pattern is not detected.

> 2. Prove that no code will detect the zero pattern u = 0

As with the previous exercise. When adding the zero word to a codeword the result will always be the codeword itself. Since atleast one of sums is in *C*, *C* does not detect the error pattern 0.

Exercise 1.11.7b

> Determine the error patterns detected by each code in Exercise 1.9.7 by using the IMLD tables constructed there.

C = {000, 001, 010, 011}

w	000 + w	001 + w	010 + w	011 + w	v
000	000*	001	010	011	000
001	001	000*	011	010	001
010	010	011	000*	001	010
011	011	010	001	000*	011
100	100*	101	110	111	000
101	101	100*	111	110	001
110	110	111	100*	101	010
111	111	110	101	100*	011

The error patterns for the above IMLD are {100,101,110,111}.

Exercise 1.11.10b

> Find the error patterns detected by each of the following codes and compare your answer with those in exercise 1.11.7

C = {000, 001, 010, 011}

000 + 000 = 000
000 + 001 = 001
000 + 010 = 010
000 + 011 = 011
001 + 001 = 000
001 + 010 = 011

001 + 011 = 010
010 + 010 = 000
010 + 011 = 001
011 + 011 = 000

The set of error patterns that cannot be detected in *C* is {000, 001, 010, 011}.
Therefore All error patterns in K^n{000, 001, 010, 011} or {100, 101, 110, 111}.

Exercise 1.11.12b,c,g,h

> Find the distance of each of the following codes
>
> b. *C* = {000, 001, 010, 011}

d(000, 001) = 1
d(000, 010) = 1
d(000, 011) = 2
d(001, 010) = 2
d(001, 011) = 1
d(010, 011) = 1

The distance for *C* is d = 1.

> c. *C* = {0000, 0001, 1110}

d(0000, 0001) = 1
d(0000, 1110) = 3
d(0001, 1110) = 4

The distance for *C* is d = 1.

> g. *C* = {00000, 11110, 01111, 10001}

d(00000, 11110) = 4
d(00000, 01111) = 4
d(00000, 10001) = 2
d(11110, 01111) = 2
d(11110, 10001) = 4
d(01111, 10001) = 4

The distance for *C* is d = 2.

> h. *C* = {000000, 101010, 010101, 111111}

d(000000, 101010) = 3
d(000000, 010101) = 3
d(000000, 111111) = 6
d(101010, 010101) = 6
d(101010, 111111) = 3
d(010101, 111111) = 3

The distance for *C* is d = 3.

Exercise 1.11.13

> Find the distance of the code formed by adding a parity check diget to K^n

Exercise 1.11.19b,c,g,h

> For each code *C* in exercise 1.11.12 find the error patterns which Theorem 1.11.14 guarantees *C* will detect.
>
> b. *C* = {000, 001, 010, 011}

The distance for *C* is d = 1.

Since d - 1 = 0 the theorem does not help in determining which error patterns *C* will detect.

> c. *C* = {0000, 0001, 1110}

The distance for *C* is d = 1.

Same as above

> g. *C* = {00000, 11110, 01111, 10001}

The distance for *C* is d = 2.

We can guarantee that *C* detects all the patterns with a weight of 1 (d - 2 = 1).

00001, 00010, 00100, 01000, 10000

> h. *C* = {000000, 101010, 010101, 111111}

The distance for *C* is d = 3.

We can guarantee that *C* detects all the patterns with a weight of 1 or 2.

Exercise 1.11.20

> Let C be the code consisting of all words of length 4 which have even weight. Find the error pattern C detects.

C = {0011, 0101, 0110, 1001, 1010, 1100, 1111}

d(0011, 0101) = 2
d(0011, 0110) = 2
d(0011, 1001) = 2
d(0011, 1010) = 2
d(0011, 1100) = 4
d(0011, 1111) = 2
d(0101, 0110) = 2
d(0101, 1001) = 2
d(0101, 1010) = 4
d(0101, 1100) = 2
d(0101, 1111) = 2
d(0110, 1001) = 4
d(0110, 1010) = 2
d(0110, 1100) = 2
d(0110, 1111) = 2
d(1001, 1010) = 2
d(1001, 1100) = 2
d(1001, 1111) = 2
d(1010, 1100) = 2
d(1010, 1111) = 2
d(1100, 1111) = 2

The distance for C is d = 2

We can guarantee that C detects all the patterns with a weight of 1 (d - 2 = 1).

0001, 0010, 0100, 1000

Exercise 1.12.5 (use u = 010 in place of u = 100)

> Let C = {001, 101, 110}, Does C correct the error pattern u = 010? What about u = 000

w	001 + w	101 + w	110 + w	v

000	001*	101	110	001
001	000*	001	111	001
010	011	111	100*	110
011	010*	110	101	011
100	101	001	010	--
101	100	000*	011	101
110	111	011	000*	110
111	110	010	001	--

The rows of the IMLD where 010 appear are

w	**001 + w**	**101 + w**	**110 + w**	**v**
011	010*	110	101	011
100	101	001	010	--
111	110	010	001	--

010 does not receive an asterisk on every row and thus does not correct *C*.

The rows of the IMLD where 000 appear are

w	**001 + w**	**101 + w**	**110 + w**	**v**
001	000*	001	111	001
101	100	000*	011	101
110	111	011	000*	110

Because 000 received an asterisk on every row we can conclude that 000 does correct *C*.

Exercise 1.12.7

> Prove that the zero pattern is always corrected.

The zero pattern only appears when w is equal to v. This means that no errors have occured and thus it will always correct C.

Exercise 1.12.8

> Which error pattern will the code $C = K^n$ corrent?

Only the zero pattern. Any other pattern will collide with another word in C.

Exercise 1.12.12 (i:b) & (ii:b,c,g,h)

> For each of the following codes C
>
> i. determine the error patterns that C will correct (the IMLD tables for these codes were constructed in Exercise 1.9.7)
>
> ii. find the error patterns that Theorem 1.12.9 guarantees that C corrects

> b. C = {000, 001, 010, 011}

> i.

w	000 + w	001 + w	010 + w	011 + w	v
000	000*	001	010	011	000
001	001	000*	011	010	001
010	010	011	000*	001	010
011	011	010	001	000*	011
100	100*	101	110	111	000
101	101	100*	111	110	001
110	110	111	100*	101	010
111	111	110	101	100*	011

Only 000 and 100 receive an asterisk every time and thus the error patterns that C will correct are {000, 100}.

> ii.

The distance of C is d = 1.

[(d - 1) / 2] = 0

Theorem. 1.12.9 cannot guarantee any error patterns.

> c. *C* = {0000, 0001, 1110}

> ii.

The distance for *C* is d = 1.

[(d - 1) / 2] = 0

Theorem. 1.12.9 cannot guarantee any error patterns.

> g. *C* = {00000, 11110, 01111, 10001}

> ii.

The distance for *C* is d = 2.

[(d - 1) / 2] = 0

Theorem. 1.12.9 cannot guarantee any error patterns.

> h. *C* = {000000, 101010, 010101, 111111}

> ii.

The distance for *C* is d = 3.

[(d - 1) / 2] = 1

Theorem. 1.12.9 can guarantee all words with a weight of 1. These are {000001, 000010, 000100, 001000, 010000, 100000}.

Exercise 1.12.14b

> For each code in Exercise 1.12.12, find an error pattern of weight [(d - 1) / 2] + 1 that *C* does not correct.

> b. *C* = {000, 001, 010, 011}

010