

# Exploring Abstract Algebra with Computer Software

## PREP Workshop 2004

### Section 13 Appendix: Error Correcting Codes

(Based on a project written by Loren Larson, originally using Maple)

Let  $\mathbf{a} = (a_0 a_1 \dots a_{m-1})$  denote a string of  $m$  0's and 1's. We associate to this "message" a "message polynomial" over  $Z_2$ , namely the polynomial  $a(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{m-1} x^{m-1}$ . For example, the message 01101 corresponds to the polynomial  $x + x^2 + x^4$  (here  $m = 5$ ). Let  $g(x)$  be a polynomial in  $Z_2[x]$  which has degree  $k$  and constant coefficient 1. The *polynomial code* with *encoding polynomial*  $g(x)$  encodes the message  $\mathbf{a}$  as the code word  $\mathbf{b}$  of length  $n = m + k$  corresponding to the polynomial  $b(x)$ , where  $b(x) = a(x)g(x)$ .

For example, let  $g(x) = 1 + x^2 + x^3$ , and  $a(x) = x + x^2 + x^4$ , then

```
gap> R:=Integers mod 2;
GF(2)
gap> x:=X(R,"x");
x
gap> g := 1+x^2+x^3;
Z(2)^0+x^2+x^3
gap> a := x+x^2+x^4;
x+x^2+x^4
gap> b:=a*g;
x+x^2+x^3+x^4+x^5+x^6+x^7
```

We call this an  $(m, n)$  polynomial code, since it encodes  $m$ -digit messages into  $n$ -digit messages by polynomial multiplication.

#### Section 13 Appendix, Project

13A.1 Use the coding polynomial  $g(x)$  as above to encode all possible four digit messages (for example, 0000, 0001, ...).

The *weight*  $w(\mathbf{a})$  of a word  $\mathbf{a}$  is the number of 1's it contains. The *distance* between two words is the weight of their difference (or sum, since the

underlying ring is  $Z_2$ ). The distance between two words is the number of places (coordinates) where the words differ. For example, the distance between  $\mathbf{a} = 11010$  and  $\mathbf{b} = 01011$  is 2 (their (coordinatewise) “sum” is 10001).

13A.2 What is the minimum nonzero weight of the encoded words in 13A.1?

13A.3 Prove that the minimum distance obtained over all pairs of two distinct code words (in an  $(m,n)$  polynomial code) is the minimum nonzero weight of the code words.

We consider another approach to the  $(4,7)$  code considered above.

```
gap> Factors(g);
[ Z(2)^0+x^2+x^3 ]
gap> polyring:=PolynomialRing(GF(8));
<algebra-with-one over GF(2^3), with 1 generators>
gap> factors:=Factors(polyring,g);
[ Z(2^3)^3+x, Z(2^3)^5+x, Z(2^3)^6+x ]
```

Let  $\alpha$  denote a root of  $g$  in  $\text{GF}(8)$ . The elements of the extension field  $Z_2(\alpha)$  are linear combinations of  $1, \alpha$  and  $\alpha^2$ . We may choose  $\alpha$  to be

$-\text{Z}(2^3)^3$

.

```
gap> alpha:=-Z(2^3)^3;
Z(2^3)^3
gap> alpha^2;
Z(2^3)^6
gap> alpha^3;
Z(2^3)^2
gap> alpha^4;
Z(2^3)^5
gap> alpha^5;
Z(2^3)
gap> alpha^6;
Z(2^3)^4
```

```

gap> alpha^7;
Z(2)^0
gap> 1+alpha;
Z(2^3)
gap> 1+alpha^2;
Z(2^3)^2
gap> alpha+alpha^2;
Z(2^3)^4
gap> 1+alpha+alpha^2;
Z(2^3)^5

```

This gives us the following table for powers of  $\alpha$  as linear combinations of  $1, \alpha$  and  $\alpha^2$ , as follows:

$$\alpha = \alpha$$

$$\alpha^2 = \alpha^2$$

$$\alpha^3 = 1 + \alpha^2$$

$$\alpha^4 = 1 + \alpha + \alpha^2$$

$$\alpha^5 = 1 + \alpha$$

$$\alpha^6 = \alpha + \alpha^2$$

$$\alpha^7 = 1$$

Note also that the elements of the ring  $Z_2[x]/\langle x^7 - 1 \rangle$  can be identified by polynomials in  $\alpha$  over  $Z_2$  of degree less than 7. For example, the code word 1110100 corresponds to the polynomial  $f(\alpha) = 1 + \alpha + \alpha^2 + \alpha^4$ .

13A.4 We obtained a factorization for  $g(x)$  above. Write the factorization in terms of  $\alpha$ , that is, regarding the factorization as occurring in  $Z_2(\alpha)$ . Then factor  $x^7 - 1$  in  $Z_2(\alpha)$ .

We now consider the problem of decoding a received message. Let  $\mathbf{v}$  denote a

received word, and let  $v(x)$  be its corresponding polynomial. If  $g(x)$  divides  $v(x)$  then  $v(x)$  is a code word and if no errors were introduced in transmission, then the original message corresponds to the polynomial  $v(x)/g(x)$ . If, however,  $g(x)$  does not divide  $v(x)$  and if there is exactly one error in  $v(x)$  (one bit is incorrect), then there is exactly one code word  $\mathbf{w}$  that has distance 1 from  $\mathbf{v}$ . This follows since any pair of distinct code words has distance 3 between the two words of the pair. Let  $v(x) = w(x) + e(x)$ , where  $e(x)$  is the “error polynomial” (it has weight exactly 1). We now show how to find  $e(x)$ , and hence  $w(x)$ .

Consider an example. The received word  $\mathbf{v} = 0110011$  contains an error (we assume no more than one error), since  $x + x^2 + x^5 + x^6$  is not divisible by  $g(x)$ . We use the division algorithm to find  $e(x)$ .

```
gap> v:=x+x^2+x^5+x^6;
x+x^2+x^5+x^6
gap> v/g;
(x+x^2+x^5+x^6)/(Z(2)+x^2+x^3)
gap> PolynomialReduction(v,[g],MonomialTotalDegreeLess);
[ Z(2)+x, [ Z(2)+x^3 ] ]
```

Thus  $v(x) = w(x) + (1 + x)$ . The error polynomial  $e(x) = 1 + x$  is not a monomial as we had hoped. However, when we substitute  $\alpha$  into this equation, and use the fact that  $1 + \alpha = \alpha^5$ , we see that  $v(\alpha) = w(\alpha) + \alpha^5$ . This is the form we wanted:  $e(x) = x^5$ , and  $w(x) = v(x) - x^5 = x + x^2 + x^6$ . Thus the message word is  $w(x)/g(x)$ :

```
gap> w:=x+x^2+x^6;
x+x^2+x^6
gap> w/g;
x+x^2+x^3
```

The corrected message is 0111.

This method does not work if the received word has more than one error.

13A.5 Consider the  $(4, 7)$  code above. Suppose the received word is 0100111. What is the message assuming that no errors were transmitted? Suppose the received word is 1110111. Assuming exactly one error was transmitted, find the message.

13A.6 The polynomial  $g(x) = 1 + x + x^2 + x^4 + x^8$  generates a  $(7, 15)$  polynomial code. Suppose the letters of the alphabet correspond to the messages given in the table below:

A 1000000	B 0100000	C 0010000	D 0001000
E 1100000	F 0110000	G 0011000	H 0001100
I 1010000	J 0101000	K 0010100	L 0001010
M 1001000	N 0100100	O 0010010	P 0001001
Q 1110000	R 0101100	S 0010110	T 0001011
U 1100001	V 1100010	W 1010100	X 1010010
Y 1111000	Z 0111100	Blank 0011110	! 0001111

Correct any errors and decode the following message, assuming no more than one error in any one received word.

```
100111001100000
011110101100100
011010010101000
011111010110010
101110111111000
001011101111110
111010001000100
100111110110001
001001110011000
100111010110001
001100111110110
000110011111011
000101110111111
```