Exercise Sheet 7

(Solutions)

Exercise 7.1. First, we factorize $x^8 - 1$, which can be writen as

$$x^8 - 1 = (x^4 - 1)(x^4 + 1) = (x - 1)(x + 1)(x^2 + 1)(x^4 + 1).$$

Observe that $x^2 + 1$ and $x^4 + 1$ have no linear factors. However, $x^4 + 1$ is divisible by the irreducible polynomial $x^2 - x - 1$, which gives

$$x^8 - 1 = (x - 1)(x + 1)(x^2 + 1)(x^2 - x - 1)(x^2 + x - 1).$$

One can also see that $x^8 - 1$ has two degree 1 factors and three degree 2 factors by taking α as a primitive element of $\mathbb{F}_9^\times$ and observing that the minimal polynomials of the elements in each of the following tuples are the same: (α^0), (α^1, α^3), (α^2, α^6), (α^4), (α^5, α^7). Thus the number of possible generator polynomials for a length 8 cyclic code (which is the number of linear cyclic codes) is $2^5 = 32$.

Exercise 7.2.

1. Let ω denote a primitive 31st root of unity in $\mathbb{F}_{32}$. First, we write a complete list of minimal polynomials for various powers of ω. Denote the minimal polynomial of ω^i by g_i. Then g_i is also the minimal polynomial of $\omega^{2i}, \omega^{4i}, \ldots$ (e.g., $g_1 = g_2 = g_4 = g_8 = g_{16}$). According to this, the powers of ω for which each g_i is the minimal polynomial are listed below:

$$\begin{array}{rl} g_0: & 0 \\ g_1: & 1, 2, 4, 8, 16 \\ g_3: & 3, 6, 12, 24, 17 \\ g_5: & 5, 10, 20, 9, 18 \\ g_7: & 7, 14, 28, 25, 19 \\ g_{11}: & 11, 22, 13, 26, 21 \\ g_{15}: & 15, 30, 29, 27, 23 \end{array}$$

   Thus the degree of g_0 is 1 and the rest of the g_i's have degree 5. Now in order to design the code, we need to take three degree 5 polynomials that are factors of the generating polynomial $g(x)$ for the code (because the dimension of the code must be 16, we need the degree of the generating polynomial to be $31-16 = 15$), and wee need the generator polynomial to contain 6 consecutive powers of ω as its roots (as the distance of the code must be at least 7). We see that a suitable choice is $g(x) = g_1(x)g_3(x)g_5(x)$, which has $\omega^1, \ldots, \omega^6$ as its roots.

2. Let $H(z) = 1 - \sigma_1 z + \sigma_2 z^2 - \sigma_3 z^3$ be the error-locator polynomial. Thus $H'(z) = -\sigma_1 + 2\sigma_2 z - 3\sigma_3 z^2$. As we will be working with the coefficients of these polynomials in characteristic two, we can simplify the polynomials as $H(z) = 1 + \sigma_1 z + \sigma_2 z^2 + \sigma_3 z^3$ and $H'(z) = \sigma_1 + \sigma_3 z^2$. Let $S(z) := S_1 + S_2 z + S_3 z^2 + \cdots$ be a power series defined by the S_i. According to the Newton relations, we must have

$$H(z) \cdot S(z) = -H'(z),$$

thus,

$$(1+\sigma_1 z+\sigma_2 z^2+\sigma_3 z^3)\cdot(S_1+S_2 z+S_3 z^2+\cdots)=\sigma_1+\sigma_3 z^2.$$

We already know that $\sigma_1 = S_1$. Now comparing the coefficients of various powers of z on both sides (namely, z^2 and z^3), we obtain

$$S_3+\sigma_1 S_2+\sigma_2 S_1=\sigma_3 \Rightarrow S_3+S_1 S_2+\sigma_2 S_1=\sigma_3, \tag{1}$$

and,

$$S_4+\sigma_1 S_3+\sigma_2 S_2+\sigma_3 S_1=0 \Rightarrow S_4+S_1 S_3+\sigma_3 S_1=\sigma_2 S_2. \tag{2}$$

Substituting (1) in (2) gives

$$\sigma_2 S_2 = S_4+S_1 S_3+S_3 S_1+S_1^3 S_2+\sigma_2 S_1^2,$$

thus,

$$\sigma_2=(S_4+S_1^2 S_2)/(S_2+S_1^2).$$

Using this in (1) finally gives

$$\sigma_3=S_3+S_1 S_2+S_1(S_4+S_1^2 S_2)/(S_2+S_1^2).$$

3. As seen in the lecture, a simple decoding algorithm will first compute the syndromes $S_1,\dots,S_6$ from the received word y as $S_i := y(\omega^i)$, and then uses the identities obtained in the previous parts to compute $\sigma_1,\sigma_2,\sigma_3$, and thus, the error locator polynomial $H(z)$ (if all the S_i are zero, no error has occurred and decoding stops right away). The roots of $H(z)$ include the powers of ω at which the errors have occurred. By erasing y at the obtained positions, we can apply an erasure decoding algorithm (which amounts to solving a system of linear equations) to find the exact set of errors.

Exercise 7.3. The vector $(y_0, y_1, \dots, y_{p-1})$ which encodes the evaluations of the line at various points of F_p (except for a bounded number of errors) can be seen as a "received word" in a Reed-Solomon code of dimension $k=2$ and length $n=p$. The minimum distance of this code is $d=n-k+1=p-1$, and thus, the code is able to uniquely correct any set of up to $\lfloor (d-1)/2\rfloor = \lfloor (p-2)/2\rfloor = (p-3)/2$ errors, and it is guaranteed that the number of erroneous evaluations do not exceed this amount. There are various algorithms that can be applied in this problem. But perhaps the simplest solution is to enumerate all lines (there are p^2 of them) and find the one with the closest evaluation vector (in Hamming weight) to $(y_0, y_1, \dots, y_{p-1})$. By the argument above, this must be the line we are looking for.