29.8 Theorem For a linear code C, the minimum weight among the nonzero code words of C is the same as the minimum distance between two different code words.

Proof For any two code words $w, u \in \mathbb{Z}_2^n$, the distance between w and u is the number of bits where the words differ. That is, the weight of $w - u$ is the distance between w and u. Since C is a subgroup of $\mathbb{Z}_2^n$, $w - u \in C$. Thus the minimum weight of nonzero code words is less than or equal to the minimum distance between two different code words. We also notice that $0 \in \mathbb{Z}_2^n$ is a code word, so the weight of a code word w is the distance between 0 and w, which implies that the minimum distance between two different code words is less than or equal to the minimum weight among the nonzero code words. ◆

If the Hamming distance between any two different code words in a code C is at least d, then we say that C **detects** $d - 1$ **bit errors** since any change to a code word in at most $d - 1$ bits is not a code word. If C is a code in $\mathbb{Z}_2^n$ and for any string $v \in \mathbb{Z}_2^n$, there is at most one code word whose Hamming distance from v is d or less, then we say that C **corrects** d **bit errors**. The idea is that if a string is received that is not a code word, then the best guess for the sent code word is the code word that is closest to the received string. For a code that corrects d bit errors, by taking the closest code to a received string we reconstruct the sent code word as long as the number of errors is at most d.

29.9 Example Let $C = \{(0,0,0,0), (1,0,1,0), (1,1,0,1), (0,1,1,1)\} \subseteq \mathbb{Z}_2^4$. It is not difficult to check that C is a subgroup of $\mathbb{Z}_2^4$, so C is a linear code. The code word $(1,0,1,0)$ has weight 2 and the other two nonzero code words have weight 3. By Theorem 29.8, the minimum distance between any two code words is 2. Thus C detects one-bit errors, but it cannot correct a one-bit error. A received message of $m = (1,0,0,0)$ differs from both $(0,0,0,0)$ and $(1,0,1,0)$ by only one bit, so even if we know m is only incorrect in one position, we would not know if the sent code word was $(0,0,0,0)$ or $(1,0,1,0)$. ▲

There are many schemes to generate codes having various properties, but we will focus on just one method. We can think of an element $(a_0, a_1, a_2, \ldots, a_{n-1}) \in \mathbb{Z}_2^n$ as corresponding to the coefficients of the polynomial $a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} \in \mathbb{Z}_2[x]$. In this way, instead of thinking of code words as strings of zeros and ones of length n, we can think of them as polynomials in $\mathbb{Z}_2[x]$ of degree at most $n - 1$. We note that this correspondence is a group isomorphism ϕ mapping $\mathbb{Z}_2^n$ onto the additive group of polynomials in $\mathbb{Z}_2[x]$ of degree at most $n - 1$.

29.10 Example Let $n = 5$ and $g(x) = x^2 + x + 1$. We define C to be all the multiples of $x^2 + x + 1$, including 0, whose degree is less than 5.

$$\begin{aligned} C &= \{f(x)g(x) \,|\, f(x) \in \mathbb{Z}_2[x] \text{ and either } \deg(f(x)) \leq 2 \text{ or } f(x) = 0\} \\ &= \{0 \cdot g(x),\ 1 \cdot g(x),\ x \cdot g(x),\ (x+1) \cdot g(x), \\ &\qquad x^2 \cdot g(x),\ (x^2+1) \cdot g(x),\ (x^2+x) \cdot g(x),\ (x^2+x+1) \cdot g(x)\} \\ &= \{0,\ x^2+x+1,\ x^3+x^2+x,\ x^3+1, \\ &\qquad x^4+x^3+x^2,\ x^4+x^3+x+1,\ x^4+x,\ x^4+x^2+1\}. \end{aligned}$$

By reading off the coefficients of these polynomials we determine the code words to be

$$\begin{gathered} (0,0,0,0,0)\ (0,0,1,1,1)\ (0,1,1,1,0)\ (0,1,0,0,1) \\ (1,1,1,0,0)\ (1,1,0,1,1)\ (1,0,0,1,0)\ (1,0,1,0,1). \end{gathered}$$

It is not difficult to check that this collection of elements in $\mathbb{Z}_2^5$ is a subgroup of $\mathbb{Z}_2^5$ and therefore gives a linear code. The code is not cyclic since $(0,1,0,0,1)$ is a code word,

but $(1, 0, 1, 0, 0)$ is not a code word. We see that the minimum weight among all the nonzero code words is 2. By Theorem 29.8, the minimum Hamming distance between any two code words is also 2, which implies that the code detects a one-bit error, but it does not correct a one-bit error. ▲

In Example 29.10, we simply read off the coefficients of the polynomials in C to construct a linear code. For the rest of this section we will abuse notation slightly by referring to a set C of polynomials in $\mathbb{Z}_2[x]$ as a linear code if C is a subgroup of $\mathbb{Z}_2[x]$ containing no polynomial of degree n or larger. The fact that ϕ mapping $\mathbb{Z}_2^n$ to the polynomials of degree at most $n-1$ is a group isomorphism assures us that any subgroup $C \leq \mathbb{Z}_2[x]$ having no polynomial of degree n or larger provides a linear code by simply reading off the coefficients of the polynomials in C.

29.11 Theorem Let $g(x)$ be a polynomial in $\mathbb{Z}_2[x]$ of degree less than n. Then $C = \{f(x)g(x) \,|\, f(x) \in \mathbb{Z}_2[x]$ and either $f(x) = 0$ or $\deg(f(x)) < n - \deg(g(x))\}$ is a linear code. Furthermore, if the polynomial $g(x)$ is a factor of $x^n + 1$ in $\mathbb{Z}_2[x]$, then C is a cyclic code.

Proof We first show that C is closed under addition. Let $f(x), h(x) \in \mathbb{Z}_2[x]$ so that each either has degree less than $n - \deg(g(x))$ or is the 0 polynomial. Then $f(x) + h(x)$ is either the zero polynomial or else its degree is less than $n - \deg(g(x))$. Therefore

$$f(x)g(x) + h(x)g(x) = (f(x) + h(x))g(x),$$

which implies that C is closed under addition. Also C contains the 0 polynomial and if $f(x)g(x) \in C$, then $-(f(x)g(x)) = f(x)g(c) \in C$. Thus C is a subgroup of the additive group $G = \{w(x) \in \mathbb{Z}_2[x] \mid w(x) = 0$ or $\deg(w(x)) < n\}$, which means that C is a linear code.

Now we assume that $g(x)$ is a factor of $x^n + 1$ in $\mathbb{Z}_2[x]$, that is, there is a polynomial $h(x) \in \mathbb{Z}_2[x]$ with

$$h(x)g(x) = x^n + 1.$$

Apparently,

$$\deg(h(x)) = n - \deg(g(x)).$$

Let $f(x)g(x) \in C$. If

$$\deg(f(x)g(x)) < n - 1,$$

then

$$(xf(x))g(x) \in C$$

and $xf(x)g(x)$ simply increases by one the degree of each term in the polynomial $f(x)g(x)$. This implies that $xf(x)g(x) \in C$ is a cyclic shift of $f(x)g(x)$. On the other hand, if

$$\deg(f(x)g(x)) = n - 1,$$

then a cyclic shift of the code word $f(x)g(x)$ is

$$p(x) = xf(x)g(x) + (x^n + 1).$$

We have

$$\begin{aligned} xf(x)g(x) + (x^n + 1) &= xf(x)g(x) + h(x)g(x) \\ &= (xf(x) + h(x))g(x) \end{aligned}$$