Chapter 5

Perfect Codes

Having established how linear codes work in general, we now begin to to look more closely at particular codes and types of codes. These will still be linear, but first we will return briefly to ideas from Section 1.4 about more general codes.

We know (Proposition 1.15) that a q-ary (n, M, d) code satisfies:

$$M \cdot \sum_{k=0}^{t} \binom{n}{k} (q-1)^k \leq q^n, \quad \text{where } t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

This is because the spheres $S(c, t)$ round all the codewords have to fit into $\mathbb{F}_q^n$ without overlapping. If these spheres also fill $\mathbb{F}_q^n$ then we have equality and the code is perfect. For a given q, n and d, we have made M as large as possible. Thus for a code to be perfect we need $|S(x,t)| = \sum_{k=0}^{t} \binom{n}{k}(q-1)^k$ to divide q^n. (If q is a prime power, this means that $|S(x,t)|$ is a power of q.)

There are two very trivial solutions to this, with $t = 0$, or $t = n$ (see Q60) More useful, though still regarded as trivial, are the odd binary repetition codes, for example $\{(0,0,0),(1,1,1)\} \subseteq \mathbb{F}_2^3$ and $\{(0,0,0,0,0),(1,1,1,1,1)\} \subseteq \mathbb{F}_2^5$ (see Q63). Apart from these, computer searches have found that relatively few values for q, n and d have $|S(x,t)|$ dividing q^n. Of non-trivial parameters which meet this criterion, most are parameters for a Hamming code; we shall consider these in section 5.1. Remarkably, for $n \leq 1000$, $d \leq 1000$ and $q \leq 100$ there are only three other possibilities. For $q = 2$ a [23,12,7] code or a [90,78,5] code would be perfect, if it exists; for $q = 3$ a [11,6,5] code would be perfect. In 1949, Golay showed that a [90,78,5] binary code does not exist (see Q74), but he specified perfect codes for the other two sets of parameters. We shall study these in Section 5.3. Later work has gone on to show that all non-trivial perfect codes either have the same parameters as the Hamming Codes, or the Golay codes, though we won't prove this in this course.

5.1 Hamming codes

If a code is to be able to correct one symbol-error, we need $d \geq 3$. For a linear code, by Theorem 4.11, we can achieve this by making sure that no two columns of its check

matrix are multiples of each other. This is simplest in the binary case.

Definition 5.1. For $r \geq 2$, $n = 2^r - 1$, let the matrix $H \in M_{r,n}(\mathbb{F}_2)$ have as columns all non-zero vectors in $\mathbb{F}_2^r$. The **binary Hamming code of redundancy** r, is $\mathrm{Ham}_2(r) = \{\mathbf{x} \in \mathbb{F}_2^n \mid \mathbf{x}H^t = 0\}$.

Example 37. For $r = 2$ and $r = 3$ respectively, we can take

$$H_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad H_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \qquad \triangle$$

Note that, according to the definition, we could have had the columns in any order. Thus strictly $\mathrm{Ham}_2(r)$ is not just one code, but an equivalence class of codes.

Proposition 5.2. *For $r \geq 2$, $Ham_2(r)$ is a perfect $[2^r - 1, 2^r - r - 1, 3]$ code, with check matrix H.*

Proof. There are five things to check here:

- $n = 2^r - 1$: There are $|\mathbb{F}_2^r - \{\mathbf{0}\}|$ columns in H.
- H is a check-matrix for $\mathrm{Ham}_2(r)$: Since H is certainly acting-check, we need only show its rows are linearly independent. This is clear since its columns include standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_r$, so each row has (at least) one entry 1 where all other rows have 0.
- $k = 2^r - r - 1$: As H is a check-matrix, it has $n - k$ rows, so $n - r = k$.
- $d = 3$: No column is a multiple of another, but columns $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_1 + \mathbf{e}_2$ are linearly dependent.
- $\mathrm{Ham}_2(r)$ is perfect: We know $|\mathrm{Ham}_2(r)| = 2^k = 2^{2^r - r - 1}$, and that by Lemma 1.7 $|S(c, 1)| = 1 + n = 2^r$. But then since the $S(c, 1)$ are disjoint, $|\bigcup_{c \in C} S(c, 1)| = 2^{2^r - r - 1} \cdot 2^r = 2^n$. So the spheres fill the space.

□

If we choose the right order for the columns of H, we can decode very neatly, without even making a syndrome table. Notice that in the matrices above, the columns actually count in binary, from left to right. For example, in H_3 the sixth column is 110, which is the binary representation of six.

Algorithm: Decoding for binary Hamming codes.

1. Use the $\mathrm{Ham}_2(r)$ with check-matrix H having column j equal to the binary representation of j, for $1 \leq j \leq n = 2^r - 1$.
2. Having received $\mathbf{y} \in \mathbb{F}_2^n$, compute its syndrome $S(\mathbf{y}) = \mathbf{y}H^t$
3. If $S(\mathbf{y}) = 0$, then $\mathbf{y}$ is a codeword, so decode as $\mathbf{y}$.

4. Otherwise, read $S(\mathbf{y})$ as a number j in binary, and decode $\mathbf{y}$ by changing its jth entry.

Example 38. Suppose we are using $\mathrm{Ham}_2(3)$, with check-matrix H_3 as above, and we receive $\mathbf{y} = (0,1,1,0,1,0,1)$. Then $S(\mathbf{y}) = \mathbf{y}(H_3)^t = (0,1,1)$. Since 011 in binary is three, we change the third entry of y, and decode it as $(0,1,0,0,1,0,1)$. △

Why does this algorithm work? Since the $S(\mathbf{c},1)$ partition $\mathbb{F}_2^n$, any received $\mathbf{y}$ equals $\mathbf{c}+\mathbf{x}$, where $\mathbf{c} \in \mathrm{Ham}_2(r)$, and $w(\mathbf{x}) \leq 1$. If $w(\mathbf{x}) = 1$, then $\mathbf{x} = \mathbf{e}_j$ for some j. So $S(\mathbf{y}) = (\mathbf{c}+\mathbf{e}_j)H^t = \mathbf{e}_j H^t = j^{th}$ row of H^t = binary representation of j. By changing the jth entry of $\mathbf{y}$, we find $\mathbf{y} - \mathbf{e}_j = \mathbf{c}$.

We can also make Hamming codes with $q > 2$, but we must take more care to avoid two columns of H being multiples of each other. Let $\mathbf{v}$ be one of the $q^r - 1$ non-zero vectors in $\mathbb{F}_q^r$, and consider the set $L_{\mathbf{v}} = \{\lambda\mathbf{v} \mid \lambda \in \mathbb{F}_q, \lambda \neq 0\}$. Then $|L_{\mathbf{v}}| = q-1$, and if $\mathbf{w} \in L_{\mathbf{v}}$, we have $L_{\mathbf{w}} = L_{\mathbf{v}}$. So the $L_{\mathbf{v}}$ partition $\mathbb{F}_q^n - \{\mathbf{0}\}$, and there are $(q^r-1)/(q-1)$ of them.

Example 39. For $q = 3,\ r = 2$ we have $8/2 = 4$ sets:

$$L_{(0,1)} = \{(0,1),(0,2)\},\ L_{(1,0)} = \{(1,0),(2,0)\},$$
$$L_{(1,1)} = \{(1,1),(2,2)\},\ L_{(1,2)} = \{(1,2),(2,1)\}.$$

△

Definition 5.3. For $r \geq 2,\ n = (q^r-1)/(q-1)$, let the sets $L_{\mathbf{v}}$ as above partition $\mathbb{F}_q^n - \{\mathbf{0}\}$. Construct $H \in M_{r,n}(\mathbb{F}_q)$ by taking one column from each $L_{\mathbf{v}}$. The **Hamming code of redundancy** r, is $\mathrm{Ham}_q(r) = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x}H^t = 0\}$.

Example 40. For $\mathrm{Ham}_3(2)$, H could be $\begin{pmatrix} 0 & 2 & 2 & 2 \\ 1 & 2 & 0 & 1 \end{pmatrix}$. △

Proposition 5.4. *For $r \geq 2$, $Ham_q(r)$ is a perfect $[n, n-r, 3\,]$ code, with check-matrix H.*

Notice that this definition, and Proposition 5.2, actually include the binary case discussed above. The proof of Proposition 5.4 is almost the same as that of Proposition 5.2; proving $d = 3$, and that the code is perfect, need a little more detail (see Q67). Again, a different choice and/or ordering of vectors for the columns of H will give an equivalent code. For the following algorithm, it may be convenient to choose each column so that its last non-zero entry is 1. Again we exploit the fact that the $\mathrm{Ham}_q(r)$ is perfect.

Algorithm: Decoding for Hamming codes.

1. Having received $\mathbf{y} \in \mathbb{F}_q^n$, compute its syndrome $S(\mathbf{y}) = \mathbf{y}H^t$
2. If $S(\mathbf{y}) = 0$, then $\mathbf{y}$ is a codeword, so decode as $\mathbf{y}$.
3. Otherwise, find j and λ such that $S(\mathbf{y})$ is λ times the j^{th} column of H, and decode $\mathbf{y}$ as $\mathbf{y} - \lambda\mathbf{e}_j$.

Example 41. Let us use $\mathrm{Ham}_3(2)$, with $H = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$. If $\mathbf{y} = (2,1,0,1)$ is received, then $S(\mathbf{y}) = (1,2)$, which is twice the last column of H. So we decode to $\mathbf{y} - 2\mathbf{e}_4 = (2,1,0,2)$. △

5.2 Extending and Puncturing

To improve error-correction or -detection we may wish to increase the minimum distance of a code, even at the expense of longer codewords. One way to attempt this is to extend the code. This puts the codewords into a bigger space, so they may be further apart.

Definition 5.5. Let the $[n, k, d]$ code C have check-matrix H. Then the **extended** code $\widehat{C}$ has check-matrix $\widehat{H} = \begin{pmatrix} & & & 0 \\ & H & & \vdots \\ & & & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix}$.

The last row perform a "parity check": it requires that the entries of a codeword add to zero.

The definition implies that the rows of $\widehat{H}$ are linearly independent, but this is true, because the rows of H were linearly independent, and only the last row has a non-zero entry in the last column.

Proposition 5.6. *If C is an $[n, k, d]$ code then $\widehat{C}$ is an $[n', k', d']$ code, with $n' = n + 1$, $k' = k$, and $d \leq d' \leq d + 1$.*

Proof. Clearly $n' = n + 1$. Since $\widehat{H}$ is a check matrix for $\widehat{C}$, it has $n' - k'$ rows, but it also has one more row than H, so $(n+1) - k' = (n-k) + 1$. Using Thm. 4.11, it is clear that $d \leq d'$, because if some set of columns of $\widehat{H}$ is linearly dependent, then certainly those columns of H must also be linearly dependent [1]. On the other hand, if some set of columns of H are linearly dependent, those columns in $\widehat{H}$ will not usually satisfy this relationship; we may need to use the last column to make zero in the last co-ordinate. □

Example 42. Let $C_2 \subseteq F_2^3$ and $C_3 \subseteq F_3^4$ have check-matrices

$$H_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \text{ and } H_3 = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Then $\widehat{C}_2$ and $\widehat{C}_3$ have check-matrices $\widehat{H}_2 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$, and $\widehat{H}_3 = \begin{pmatrix} 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$.
Looking at the columns of H_2 and $\widehat{H}_2$, it is clear that $d(C_2) = 3$ but $d(\widehat{C}_2) = 4$. On the other hand, in both H_3 and $\widehat{H}_3$ columns 2 to 4 add up to zero, so $d(C_3) = d(\widehat{C}_3) = 3$. △

Here C_2 is a $\mathrm{Ham}_2(2)$ code, so this is an example of the following:

Corollary 5.7. *If a binary $[n, k, d]$ code C has d odd, then $\widehat{C}$ is a $[n + 1, k, d + 1]$ code.*

Proof. Using Proposition 5.6, we need only show for Theorem 4.11 that any d columns in the check-matrix for $\widehat{C}$ are linearly independent. But this is clear, as their last entries must add to 1 in $\mathbb{F}_2$. □

[1] What if the dependent set of columns of $\widehat{H}$ includes the last column? Then we have linear dependence for *one fewer* columns of H. It still follows that $d \leq d'$.

Corollary 5.8. *The extended binary Hamming code* $\widehat{\mathrm{Ham}}_2(r)$ *is a* $[2^r, 2^r - r - 1, 4]$ *code.*

Since 4 is even, $\widehat{\mathrm{Ham}}_2(r)$ is not perfect (see Q16). By Prop.1.2, $\widehat{\mathrm{Ham}}_2(r)$ can still only correct one symbol-error, but it can detect two. This is a gain worth having, particularly if we can ask for retransmission. (Strictly, it can detect three - but with three symbol-errors even unique-nearest-neighbour decoding may give the wrong answer.)

Suppose we have two equivalent codes, C and C', with check-matrices H and H', and we form the extended codes $\widehat{C}$ and $\widehat{C}'$, with check-matrices $\widehat{H}$ and $\widehat{H}'$. Will $\widehat{C}$ and $\widehat{C}'$ be equivalent? If the codes are binary then yes: H and H' have the same columns in a (possibly) different order, and so do $\widehat{H}$ and $\widehat{H}'$. But if $q \geq 3$, then some column of H' may be a multiple ($\lambda \neq 0, \neq 1$) of a column of H. With one added at the end of each, they are no longer multiples, so $\widehat{C}$ and $\hat{C}'$ may not be equivalent (Q70).

To find a generator-matrix for an extended code, we can use the $G \leftrightarrow H$ algorithm on $\widehat{H}$. But we can also get it directly from the original generator-matrix, by adding an overall check-digit to each row, and so to each codeword.

Notation: If $\mathbf{g} = (g_1, \ldots, g_n)$, we write $(\mathbf{g}, a)$ for $(g_1, \ldots, g_n, a)$.

Proposition 5.9. *If* C *has generator-matrix* G *with rows* $\mathbf{g}_i = (g_{i1}, \ldots, g_{in})$*, for* $\leq i \leq k$*, then* $\widehat{C}$ *has generator-matrix* $\widehat{G}$ *with rows* $\widehat{\mathbf{g}_i} = (\mathbf{g}_i, -\sum_{j=1}^{n} g_{ij})$

Example 43. Using Proposition 4.5, the code C_3 from Example 42 has generator matrix

$$G = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}.$$

Using Proposition 5.9, the generator matrix for $\widehat{C}_3$ is therefore

$$\widehat{G} = \begin{pmatrix} 2 & 2 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 & 2 \end{pmatrix},$$

and one can easily check that the rows of this matrix are orthogonal to the extended check matrix in Example 42. △

Proof. (Of These $\widehat{\mathbf{g}_i}$ are linearly independent, and there are k of them as required. We need only check that they are in $\widehat{C}$. Let the rows of C's check-matrix H be $\mathbf{h}_1, \ldots, \mathbf{h}_{n-k}$. Then $\widehat{C}$ has check-matrix $\widehat{H}$ with rows $\widehat{\mathbf{h}}_1 = (\mathbf{h}_1, 0), \ldots, \widehat{\mathbf{h}}_{n-k} = (\mathbf{h}_{n-k}, 0)$ and $\widehat{\mathbf{h}}_{n-k+1} = (1, \ldots, 1)$. Then

$$\widehat{\mathbf{g}_i} \cdot \widehat{\mathbf{h}_l} = (\mathbf{g}_i, -\sum_{j=1}^{n} g_{ij}) \cdot (\mathbf{h}_l, 0) \;=\; \mathbf{g}_i \cdot \mathbf{h}_l = 0 \quad \text{for } 1 \leq l \leq n-k,$$

$$\text{and } \widehat{\mathbf{g}_i} \cdot \widehat{\mathbf{h}}_{n-k+1} = (\mathbf{g}_i, -\sum_{j=1}^{n} g_{ij}) \cdot (1, \ldots, 1) = \sum_{j=1}^{n} g_{ij} - \sum_{j=1}^{n} g_{ij} = 0.$$

□

How can we return from $\widehat{C}$ to C? In terms of $\widehat{G}$, this is obvious: we just delete the last column, thus deleting the last symbol from each codeword of $\widehat{C}$. More generally:

Definition 5.10. To **puncture** a $[n, k, d]$ code C, we choose $1 \leq j \leq n$, and remove the j^{th} entry from every codeword.

Proposition 5.11. *Let C' be the $[n-1, k', d']$ code obtained by puncturing the $[n, k, d]$ code C in the i^{th} position.*

1. *If $d \geq 2$, then $k' = k$. If C has a codeword of minimum weight with a non-zero entry in the i^{th} position, then $d' = d - 1$. Otherwise, $d' = d$.*
2. *If $d = 1$, and C contains a weight 1 codeword whose non-zero element is in the i^{th} position, then $k' = k - 1$ (as long as $k > 1$) and $d' \geq 1$. Otherwise, $k' = k$ and $d' = 1$.*

Proof. Clearly we always have $n' = n - 1$.

If $d > 1$, then no two codewords can become the same by removing the i^{th} position, hence $q^{k'} = q^k$. If $c \in C$ is a codeword of minimum weight $w(c) = d$, with a non-zero entry in the i^{th} position, then in C' this becomes the codeword c' with $w(c') = d-1$. If no minimal weight codeword (with weight d) in C has a non-zero entry in the i^{th} position, then in C' all such words still have weight d, and removing the i^{th} position from all codewords cannot produce a word of weight less that d.

If $d = 1$, then there exists at least one codeword of weight one by Proposition 2.7. Since $\mathbb{F}_q$ is a field, there exist weight 1 words in C whose non-zero entry is a 1. Of course, there may be multiple such words whose 1s are in different positions. Then the generator matrix for C in RREF must contain each of these words as rows (as they cannot be written as the sum of multiple rows of the generator matrix, due to the leading ones in the rows of this matrix).

If C contains such a word whose 1 is in the i^{th} position, then the corresponding row of the generator matrix (in RREF) for C' is 0 in all positions, and so $k' < k$. Since no other row of this generator matrix contains a 1 in this position (as it is in RREF), the remaining rows must be linearly independent, and so $k' = k - 1$. The minimum distance of C' is then the minimum weight of the remaining words of C', so $d' \geq 1$.

If C contains a word of weight 1 whose non-zero entry is not in the i^{th} position, then this word still has weight 1 in C', and so $d' = 1$. If no weight 1 word in C has a non-zero entry in the i^{th} position, then by linearity no two words of C can differ in only the i^{th} position, and no two codewords can become the same by removing their i^{th} positions, so $q^{k'} = q^k$. □

Slightly more generally, given a code $C \subseteq \mathbb{F}_q^n$ and set of m coordinates $T = \{t_1, \ldots, t_m\}$, where $1 \leq t_i \leq n$ for $1 \leq i \leq m$, we can puncture C on each position in T to obtain the code C^T. The code obtained by puncturing C in position i as above, can then be written as $C^{\{i\}}$.

Proposition 5.12. *Let $C \subseteq \mathbb{F}_q^n$ be an $[n, k, d]$ code, and T be a coordinate set of size m as above. Then C^T is an $[n - m, k', d']$ code, with $k' \geq k - m$ and $d' \geq d - m$.*

Proof. Use Proposition 5.11 and induction. □

So it seems there could be n different single punctured versions of a code. These may or may not be equivalent; they may not even have the same minimum distance.

Example 44. Let $C \subset \mathbb{F}_2^5$ be the $[5, 2, 2]$ code generated by

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Using Proposition 4.5, this has check-matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

and so the extended code $\widehat{C}$ has check-matrix

$$\widehat{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

and by Proposition 5.9, $\widehat{C}$ has generator matrix

$$\widehat{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

If we puncture in the 6^{th} position, we clearly get back C, so $\widehat{C}^{\{6\}} = C$. However, if we puncture $\widehat{C}$ in the 3^{rd} position, we get a code $\widehat{C}^{\{3\}}$ with generator matrix

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix},$$

which is clearly a code of minimum distance 1, which is not even (permutation) equivalent to C.

The code $\widehat{C}^{\{1,3\}}$ has generator matrix

$$\tilde{G} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix},$$

and hence has dimension 1 and minimum distance 4. △

To summarise, extending takes a code C of dimension k in the space $\mathbb{F}_q^n$, and makes a new code $\widehat{C}$, still of dimension k, in $\mathbb{F}_q^{n+1}$. Puncturing $\widehat{C}$ (once) projects it back into $\mathbb{F}_q^n$, but may or may not give C again.

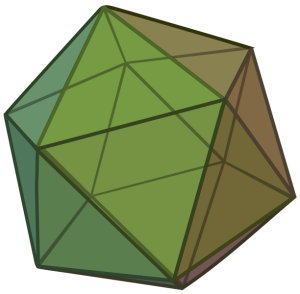
5.3 Golay Codes

The remainder of this chapter of the notes is non-examinable, and is provided only for your interest.

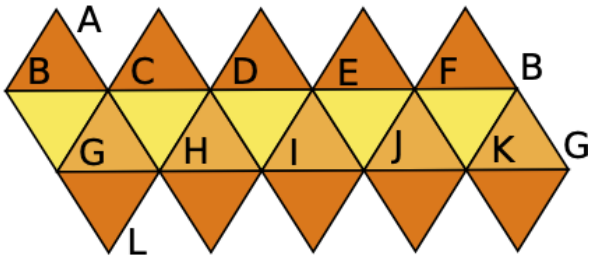
The perfect binary [23,12,7] code found by Golay is known as $\mathcal{G}_{23}$. We shall find it indirectly: assuming such a code exists, we shall first consider an extended version of it, and then puncture this. We call this code $\mathcal{G}_{24}$; by Corollary 5.7 it is a $[24, 12, 8]$ code, and is not perfect. We first describe how to construct a generator matrix for $\mathcal{G}_{24}$ and then show that the code generated from this is indeed a binary $[24, 12, 8]$ code. In fact, one can prove that a $[24, 12, 8]$ code is unique up to equivalence, a fact which we will use without proof.

5.3.1 The Extended Binary Golay Code $\mathcal{G}_{24}$

We construct our generator matrix using the adjacencies of the vertices of the icosahedron. We recall that an icosahedron is a twenty sided platonic solid with five equilateral triangles meeting at each vertex. The icosahedron is shown in Figure 5.1a. It is easiest to consider the adjacency of the icosahedron by studying its net (Figure 5.1b).



(a) The Icosahedron.



(b) The net of the icosahedron, with the vertices labelled A to L.

Figure 5.1: Images adapted from https://commons.wikimedia.org/wiki/File:Icosahedron.jpg and https://commons.wikimedia.org/wiki/File:Icosahedron_flat.svg respectively.

Definition 5.13. Two points of the icosahedron are *adjacent* if in the net there is a single line joining the two points.

Note that by the above definition, a point X is not adjacent to itself, since there is no single line joining any point to itself in the net of the icosahedron.

Let us label the vertices of the Icosahedron as v_i for $1 \leq i \leq 12$. We now define a matrix A, where the entry in row i, column j, is 0 if the two points indexed by the position in the matrix are adjacent, and 1 if not. That is, we let $A \in M_{12,12}(\mathbb{F}_2)$ be given by

$$A_{i,j} = \begin{cases} 0 & \text{if } v_i \text{ is adjacent to } v_j \\ 1 & \text{if } v_i \text{ is not adjacent to } v_j. \end{cases}$$

Explicitly, we have

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that this matrix is symmetric, since v_i is adjacent to $v_j \iff v_j$ is adjacent to v_i.

Definition 5.14. A generator matrix G for the **Extended Binary Golay Code** $\mathcal{G}_{24}$ is then given in standard form as $(I_{12} \mid A)$,

$$G = \begin{pmatrix} 1 & & & & & & & & & & & & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & & & & & & & & & & & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ & & 1 & & & & & & & & & & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ & & & 1 & & & & & 0 & & & & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ & & & & 1 & & & & & & & & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ & & & & & 1 & & & & & & & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ & & & & & & 1 & & & & & & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ & & & & & & & 1 & & & & & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ & & & & 0 & & & & 1 & & & & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ & & & & & & & & & 1 & & & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & & & & & & & & 1 & & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Since G is in standard form, every row in G can be written as $(\mathbf{e}_i, \mathbf{a}_i)$, where the $\mathbf{e}_i$ are standard basis vectors, and the $\mathbf{a}_i$ are rows of A. The code generated by G is clearly a binary code with $n = 24$ and $k = 12$. In order to show that $\mathcal{G}_{24}$ is a binary $[24, 12, 8]$ code, we 'just' need to show that the code generated by G has $d = 8$. We do not use Theorem 4.11 here, as we would need to show that no combination of 7 or less columns of G are linearly dependent. Instead, we show this in several steps by considering the weights of the codewords directly.

Definition 5.15. For $\mathbf{x}$ and $\mathbf{y} \in \mathbb{F}_2^n$, the *intersection* of $\mathbf{x}$ and $\mathbf{y}$, written as $\mathbf{x} \cap \mathbf{y}$, is coordinatewise product of $\mathbf{x}$ and $\mathbf{y}$.

This intersection of $\mathbf{x}$ and $\mathbf{y}$ has 1s in exactly {places where $\mathbf{x}$ has a 1} $\cap$ {places where $\mathbf{y}$ has a 1}, and zeros elsewhere.

Lemma 5.16. *For* $\mathbf{x}$ *and* $\mathbf{y}$ *in* $\mathbb{F}_2^n$*:* i) $w(\mathbf{x} + \mathbf{y}) = w(\mathbf{x}) + w(\mathbf{y}) - 2w(\mathbf{x} \cap \mathbf{y})$.
ii) $\mathbf{x} \cdot \mathbf{y} = 0 \in \mathbb{F}_2$ if and only if $w(\mathbf{x} \cap \mathbf{y})$ is even.
iii) $\mathbf{x} \cdot \mathbf{x} = 0 \in \mathbb{F}_2$ if and only if $w(\mathbf{x})$ is even.

Proof. Easy - see Q75. □

Example 45. $\mathbf{a}_2 = (0,1,0,1,1,0,0,1,1,1,0,1)$, $\mathbf{a}_3 = (0,0,1,0,1,1,0,0,1,1,1,1)$. Then $\mathbf{a}_2 + \mathbf{a}_3 = (0,1,1,1,0,1,0,1,0,0,1,0)$, $\mathbf{a}_2 \cap \mathbf{a}_3 = (0,0,0,0,1,0,0,0,1,1,0,1)$. The weights are $w(\mathbf{a}_2) = w(\mathbf{a}_3) = 7$, $w(\mathbf{a}_2 + \mathbf{a}_3) = 6$, and $w(\mathbf{a}_2 \cap \mathbf{a}_3) = 4$.

In fact, we have $w(\mathbf{a}_i) = 7$ and $w(\mathbf{g}_i) = 8$ for $i = 1, \ldots, 12$, since each point on the icosahedron is not adjacent to exactly 7 other points. △

Proposition 5.17.

$$d(\mathcal{G}_{24}) = 8.$$

As noted in Example 45, there certainly exist codewords in the Extended Binary Golay Code of weight 8. We show there are no codewords with $w(\mathbf{c}) < 8$ via four lemmas. The first two concern the code as a whole.

Lemma 5.18. *The code $\mathcal{G}_{24}$ is self-dual, that is, $\mathcal{G}_{24}^{\perp} = \mathcal{G}_{24}$.*

Proof. We know that $\dim(\mathcal{G}_{24}^{\perp}) = n - k = 12$, and so $|\mathcal{G}_{24}^{\perp}| = |\mathcal{G}_{24}|$. It is therefore enough to show that $\mathcal{G}_{24} \subset \mathcal{G}_{24}^{\perp}$. Recalling that $\mathcal{G}_{24}^{\perp} = \{\mathbf{x} \in \mathbb{F}_2^{24} \mid \mathbf{x} \cdot G^t = \mathbf{0}\}$, we therefore have $\mathcal{G}_{24} \subset \mathcal{G}_{24}^{\perp} \iff \mathbf{c} \cdot G^t = \mathbf{0} \ \forall \ \mathbf{c} \in \mathcal{G}_{24}$. By linearity, it is enough to check this for a basis for $\mathcal{G}_{24}$. We therefore check that for any two rows $(\mathbf{e}_i, \mathbf{a}_i)$ and $(\mathbf{e}_j, \mathbf{a}_j)$ of G, we have $(\mathbf{e}_i, \mathbf{a}_i) \cdot (\mathbf{e}_j, \mathbf{a}_j) = 0$.

By Lemma 5.16, this is true whenever we have $i = j$, since all rows of G have even weight. For $i \neq j$,

$$\begin{aligned}(\mathbf{e}_i, \mathbf{a}_i) \cdot (\mathbf{e}_j, \mathbf{a}_j) &= \mathbf{e}_i \cdot \mathbf{e}_j + \mathbf{a}_i \cdot \mathbf{a}_j \\ &= \mathbf{a}_i \cdot \mathbf{a}_j.\end{aligned}$$

In terms of our icosahedron, $\mathbf{a}_i \cdot \mathbf{a}_j$ is just the number of points on the icosahedron not adjacent to v_i *and* not adjacent to v_j modulo 2. By the rotational symmetry of the icosahedron, without loss of generality we can assume that v_i is $v_1 = A$, and v_j is one of $v_2 = B$, $v_7 = G$ or $v_{12} = L$. If we let $E_i = \{\text{points in the icosahedron adjacent to } v_i\}$, then by the Inclusion-Exclusion principle, we have

$$\begin{aligned}\mathbf{a}_i \cdot \mathbf{a}_j &= 12 - |E_i \cup E_j| \\ &= 12 - |E_i| - |E_j| + |E_i \cap E_j|.\end{aligned}$$

We can now consider the three possibilities for v_j in turn:

- If v_j is $v_{12} = L$, then $\mathbf{a}_i \cdot \mathbf{a}_j = 12 - 5 - 5 + 0 \pmod 2 = 0$.
- If v_j is $v_7 = G$, then $\mathbf{a}_i \cdot \mathbf{a}_j = 12 - 5 - 5 + 2 \pmod 2 = 0$.
- If v_j is $v_2 = B$, then $\mathbf{a}_i \cdot \mathbf{a}_j = 12 - 5 - 5 + 2 \pmod 2 = 0$.

We therefore see that

$$\mathbf{a}_i \cdot \mathbf{a}_j = w(\mathbf{a}_i \cap \mathbf{a}_j) \pmod 2 = 0 \quad \forall\, 1 \leq i, j \leq 12. \tag{5.1}$$

This shows that every basis vector is orthogonal to every other basis vector, and hence by linearity that $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$. □

Lemma 5.19. $(A \mid I_{12})$ *is another generator-matrix for* $\mathcal{G}_{24}$.

Proof. By Proposition 4.5, $(-A^t \mid I_{12})$ is a generator-matrix for $\mathcal{G}_{24}^{\perp}$. But $-A^t = A$, and by Lemma 5.18, $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$. We therefore have that $(A \mid I_{12})$ is a generator matrix for $\mathcal{G}_{24}$ also. □

The next two Lemmas are about individual codewords of $\mathcal{G}_{24}$.

Lemma 5.20. *If* $\mathbf{c} \in \mathcal{G}_{24}$*, then* $w(\mathbf{c})$ *is divisible by 4.*

Proof. Let $\mathbf{c}_1$, $\mathbf{c}_2 \in \mathcal{G}_{24}$ have weights divisible by 4. Since $\mathcal{G}_{24}$ is self-dual by Lemma 5.18, $\mathbf{c}_1 \cdot \mathbf{c}_2 = 0$, so $w(\mathbf{c}_1 \cap \mathbf{c}_2)$ is even by Lemma 5.16. So, $w(\mathbf{c}_1 + \mathbf{c}_2) = w(\mathbf{c}_1) + w(\mathbf{c}_2) - 2w(\mathbf{c}_1 \cap \mathbf{c}_2)$ is also divisible by 4. Since any $\mathbf{c} \in \mathcal{G}_{24}$ is a sum of rows of G, all of which have weight 8, then all such $\mathbf{c}$ must have weight divisible by 4. □

Lemma 5.21. *No* $\mathbf{c} \in \mathcal{G}_{24}$ *has weight 4.*

Proof. We prove this via a contradiction. Suppose $\mathbf{c} \in \mathcal{G}_{24}$, $w(\mathbf{c}) = 4$. Since $(I_{12} \mid A)$ and $(A \mid I_{12})$ are both generator matrices for $\mathcal{G}_{24}$, then $\mathbf{c}$ can be written as a sum of $(\mathbf{e}_i, \mathbf{a}_i)$ *and* as a sum of $(\mathbf{a}_i, \mathbf{e}_i)$. Let $\mathbf{l} = (c_1, \ldots, c_{12})$, $\mathbf{r} = (c_{13}, \ldots, c_{24})$, such that $\mathbf{c} = (\mathbf{l}, \mathbf{r})$. Then $w(\mathbf{l}) + w(\mathbf{r}) = 4$ and so we can consider the different cases satisfying this:

- If $w(\mathbf{l}) = 0$, then we have no $(\mathbf{e}_i, \mathbf{a}_i)$, and so $w(\mathbf{c}) = 0$.
- If $w(\mathbf{r}) = 0$, then we have no $(\mathbf{a}_i, \mathbf{e}_i)$, and so $w(\mathbf{c}) = 0$.
- If $w(\mathbf{l}) = 1$, then we have one $(\mathbf{e}_i, \mathbf{a}_i)$, and so $w(\mathbf{c}) = 8$.
- If $w(\mathbf{r}) = 1$, then we have one $(\mathbf{a}_i, \mathbf{e}_i)$, and so $w(\mathbf{c}) = 8$.
- If $w(\mathbf{l}) = w(\mathbf{r}) = 2$, then we have two $(\mathbf{e}_i, \mathbf{a}_i)$, and so

$$\begin{aligned} w(\mathbf{c}) &= 2 + w(\mathbf{a}_i + \mathbf{a}_j) \\ &= 2 + w(\mathbf{a}_i) + w(\mathbf{a}_j) - 2w(\mathbf{a}_i \cap \mathbf{a}_j) \\ &= 16 - 2w(\mathbf{a}_i \cap \mathbf{a}_j) \end{aligned} \tag{5.2}$$

However, by Equation (5.1), we see that we have already shown that $w(\mathbf{a}_i \cap \mathbf{a}_j) \neq 6$ for any i, j in the proof of Lemma 5.18, and therefore no $\mathbf{c} \in \mathcal{G}_{24}$ has $w(\mathbf{c}) = 4$. □

Lemmas 5.20 and 5.21 together complete the proof of Proposition 5.17 – we have shown that the weights of all codewords are divisible by 4, no word of weight 4 exists and there does exist at least one (in fact there are 759) words of weight 8. The Extended Binary Golay Code is therefore a binary $[24, 12, 8]$ code as claimed.

5.3.2 The Perfect Binary Golay Code $\mathcal{G}_{23}$

We can now puncture the code by deleting the last column of G, to get a generator matrix for the Golay code $\mathcal{G}_{23}$. Since G contained a minimum weight codeword with a non-zero entry in the final position, then by Proposition 5.11, $\mathcal{G}_{23}$ is a [23, 12, 7] code. (It is true, though not obvious, that deleting any other column of G would give a permutation equivalent code which we could also call $\mathcal{G}_{23}$.)

We can quickly confirm that $\mathcal{G}_{23}$ is perfect: since $d = 7$, spheres $S(\mathbf{c}, 3)$ around each of the 2^{12} codewords do not intersect, and since $2^{12}\left(1 + 23 + \binom{23}{2} + \binom{23}{3}\right) = 2^{23}$, they fill $\mathbb{F}_2^{23}$ as required.

5.3.3 The Ternary Golay Codes $\mathcal{G}_{12}$ & $\mathcal{G}_{11}$

We now look briefly at the perfect Ternary Golay Code $\mathcal{G}_{11}$, with parameters $[11, 6, 5]$. We can find it in the same way as the perfect Binary Golay Code, by first considering the Extended Ternary Golay Code $\mathcal{G}_{12}$. One possible generator matrix for $\mathcal{G}_{12}$ is

$$G = [I_6 \mid A] = \begin{pmatrix} 1 & & & & & & 0 & 1 & 1 & 1 & 1 & 1 \\ & 1 & & & 0 & & 1 & 0 & 1 & 2 & 2 & 1 \\ & & 1 & & & & 1 & 1 & 0 & 1 & 2 & 2 \\ & & & 1 & & & 1 & 2 & 1 & 0 & 1 & 2 \\ & 0 & & & 1 & & 1 & 2 & 2 & 1 & 0 & 1 \\ & & & & & 1 & 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}.$$

As before we can confirm that this gives parameters $[12, 6, 6]$, and then delete the last column to get a generator-matrix for $\mathcal{G}_{11}$. Alternatively, in this slightly smaller case we can construct a check-matrix $H \in M_{5,11}(\mathbb{F}_3)$ for $\mathcal{G}_{11}$ directly, with any 4 columns linearly independent. (See Q76 and Q77.)

5.3.4 Other Constructions for $\mathcal{G}_{24}$

Aside from the construction of $\mathcal{G}_{24}$ discussed above in terms of the geometry of the icosahedron, there are many other interesting constructions of the code. Here, I would like to very briefly mention two other (related) constructions.

Firstly, the Extended Binary Golay Code can be constructed using a so called *Lexicographic* construction. In this construction, we start with a set C containing only the vector $\mathbf{c}_0 = (0, \ldots, 0) \in \mathbb{F}_2^{24}$. We now consider each non-zero element of $\mathbb{F}_2^{24}$ in turn, ordered increasingly by their values when read as binary numbers. This is known as a lexicographical ordering of the elements of $\mathbb{F}_2^{24}$. That is, we first consider $(0, \ldots, 0, 1)$, then $(0, \ldots, 0, 1, 0)$, and so on. Every time we consider a vector with distance at least 8 from every element already in our set C, we add it to the set C. So the first element we add to C is the vector $\mathbf{c}_1 = (0, \ldots, 0, 1, 1, 1, 1, 1, 1, 1, 1)$, as this is the first element of $\mathbb{F}_2^{24}$ in the lexicographical ordering with distance at least 8 from $\mathbf{c}_0$. The next element of $\mathbb{F}_2^{24}$ to be added is $\mathbf{c}_2 = (0, \ldots, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1)$, as this is the first element in the lexicographical ordering with distance at least 8 from the two words we have already added to C, namely $\mathbf{c}_0$ and $\mathbf{c}_1$.

In this construction, the fact that the code being constructed has $n = 24$ and $d = 8$ is clear by construction. However, it is not immediately clear that this code is linear, or that it has dimension $k = 12$. As an exercise, you may wish to code the algorithm described above on a computer, and run it to see how many elements of $\mathbb{F}_2^{24}$ end up getting added to C. You should find that you end up with 4096 elements of C, which implies that *if* the code is linear, it has dimension $k = 12$. Since we stated (without proof) that a $[24, 12, 8]$ code is unique up to equivalence, this implies that this is a lexicographical construction of $\mathcal{G}_{24}$.

A related construction is via a two-player impartial game known as Mogul. The rules of this game are defined as follows:

Definition 5.22. Mogul is a game, played by two players with a row of twenty-four coins.

- To begin with, the twenty-four coins are all placed heads up.
- The two players then take turns to turn over between one and seven coins (inclusive), with the condition that the right-most coin turned is turned from heads to tails. Any other coins turned, if there are any others turned, may be turned from heads to tails or tails to heads.
- The winner is the player who makes the last possible move.

A possible first move for Mogul is shown in Figure 5.2. One can prove that under the assumption that both players play optimally, the player who makes the first move (P1) will always lose. We therefore say that P2 has a winning strategy, and this strategy consists of P2 always moving to one of a predefined set of positions on his turn. This set of positions, from which P1 can never hope to win, is exactly the Extended Binary Golay Code.

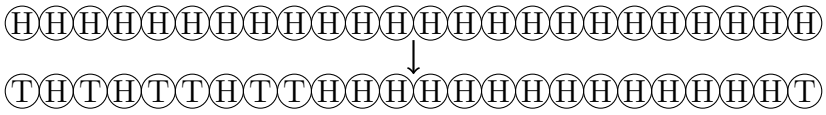


Figure 5.2: A move in the the mathematical game of Mogul

5.3.5 $\mathcal{G}_{24}$ and Moonshine

Perhaps the most interesting thing about $\mathcal{G}_{24}$ is its connection to Mathematical Physics through a phenomenon known as Mathieu Moonshine. This moonshine connects the physics of superstring theory to the mathematics of modular forms and the representation theory of a sporadic group called Mathieu 24 (M_{24}).

In Section 3.4, we saw that two binary codes are permutation equivalent if applying a permutation to the n positions of all codewords maps the set of codewords of one code to the other. The set of permutations which map a code to itself forms a group (under composition of permutations) known as the permutation automorphism group of the code C, $\mathrm{PAut}(C)$. For the Extended Binary Golay Code, this automorphism group can be shown to be the simple sporadic group $M_{24} \subset A_{24}$. The automorphism groups of the other 3 Golay codes are related to three other simple sporadic groups, M_{11}, M_{12} and M_{23}.

Those of you who are also studying representation theory will know that an algebraic object such as a group can be studied via its representations. Roughly speaking, these are maps from the group to a vector space on which the group can be seen to have a well defined action. That is, for a group G, a representation is a homomorphism from G to $GL(V)$, the group of invertible linear transformations for some vector space V. We say that such a representation has a *dimension* equal to the dimension of the vector space V. The representations of the group M_{24} appear to play a role in the physics of superstring theory, via an important (mock-)modular form

$$H_2^1(q) = 2q^{-1/8}\left(-1 + 45q + 231q^2 + 770q^3 + 2277q^4 + 5796q^5 + \ldots\right).$$

Each of the coefficients of q in this form is the dimension of a representation of M_{24}. However, the appearance of M_{24} in this context is still not fully understood. Due to similarities with a mathematical phenomenon known as Monstrous Moonshine, this appearance of the representation theory of M_{24} in the mock-modular form H_2^1 is known as Mathieu Moonshine.

The Extended Ternary Golay also has a connection to moonshine via its automorphism group. This moonshine is one of a collection of 23 instances of Moonshine known as Umbral Moonshine, in which the two Extended Golay Codes, as well as two other codes which you may meet in this course known as the Tetracode and the Hexacode, all appear.