Coding theory

Ilan Vardi*

Coding theory considers the problem of sending messages over noisy transmission lines. For example, when saying a name like "Vardi" over a phone line, the recipient might hear "Zardi" since "V" and "Z" can be confused. The usual way around this is to spell the words and replace each letter with a corresponding name. Thus

$$\begin{array}{l} \text{A} \mapsto \text{Alpha} \\ \text{B} \mapsto \text{Bravo} \\ \vdots \qquad \vdots \\ \text{Z} \mapsto \text{Zero} \end{array}$$

So "Vardi" become "Victor, Alpha, Romeo, David, Indio." This replacement has two features: redundancy, i.e., a single letter is replaced by a word, and separation, i.e., similar sounding letters are replaced by different sounding words.

To simplify the problem, consider the alphabet to consist of "0" and "1" instead of "A" to "Z". An error occurs when a 0 is changed to a 1 or a 1 changed to a 0. This alphabet is used in digital computers, for example, the ASCII representation of text

$$\begin{array}{r} \text{A} \mapsto 1000001 \\ \text{B} \mapsto 1000010 \\ \vdots \qquad \vdots \\ \text{Z} \mapsto 1011010 \\ \text{a} \mapsto 1100001 \\ \vdots \qquad \vdots \\ \text{z} \mapsto 1111010 \end{array}$$

is the usual encoding of letters into binary strings. Sending out information one bit at a time leads to a situation where errors go unnoticed, so coding theory comes into play. As before, there are two considerations: redundancy and separation.

*IHES, 35 route de Chartres, 91440 Bures–sur–Yvette, France, ilan@ihes.fr

Redundancy means that a message like "101" is coded in by a longer message. The simplest way to implement this for binary messages is to use a parity check, i.e., to add a 0 or 1 at then end of every message depending on whether the message has an even number of 1's or an odd number of 1's. For example

$$011 \mapsto 0110, \qquad 10011 \mapsto 100111\,.$$

This scheme will be able to detect if a single error has occurred in transmission, but will not be able to find the error, nor will it be able to detect if two errors have occurred. The advantage of this method is that it is easy to implement and easy to analyze mathematically. The reason for this is that the parity check digit is simply the sum of the other digits mod 2, i.e., the digits are added according to the rules

$$0+0=0, \qquad 0+1=1, \qquad 1+0=1, \qquad 1+1=0\,.$$

With multiplication mod 2 given by the usual rules

$$0\times 0=0, \qquad 0\times 1=1, \qquad 1\times 0=1, \qquad 1\times 1=1\,.$$

Since $1+1=0$, this means that all the bits added to the parity check bit should always return zero. One can therefore also think of the parity check as a verification that the inner product of a vector of all ones with the original message and the extra bit should return zero (the term parity check will be used in this way as well). For example

$$(0\ 1\ 1)\begin{pmatrix}1\\1\\1\end{pmatrix}=0, \qquad (1\ 0\ 0\ 1\ 1\ 1)\begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix}=0\,.$$

This approach allows the use of linear algebra mod 2 which is even simpler than usual. For example, in mod 2 algebra, the identity

$$(a+b)^2=a^2+b^2$$

is true. The most popular codes all rely on generalized parity checks and are called *linear codes.*

The second consideration is separation, i.e., two similar but different words should be much different after being encoded. This condition asks for *error correction,* i.e., that the code be able to find errors, not just detect them. The measure of separation will be in how many places two codewords differ. Thus 1111 and 1100 differ in two places, so are at *Hamming distance* two, while 1111 and 1011 are at Hamming distance one. (Hamming distance is most popularly used to denote distance from winning a lottery: "I was only two digits away from the one million dollar jackpot.")

Hamming distance is then used as follows: If a word is received that is not a codeword, then replace it with a codeword of minimum Hamming distance away. An good code will be one where any two codewords are far away, the farther the minimum distance, the more errors can be corrected. In fact, if the minimum distance between two words is d, then $(d-1)/2$ errors can be corrected since there will only be a single codeword at less than that distance.

An example is given by the $[7, 4]$ *Hamming code* which has minimum distance 3 so is able to correct a single error. This code consists of 7 digit codewords where the first 4 digits are the original word, and the last 3 digits are parity checks, i.e., there are 4 *information bits* and 3 *redundant bits*. The different messages that can be sent are

$$0000,\ 0001,\ 0010, \ldots,\ 1111$$

and three extra digits will be tacked by doing certain parity checks, i.e., linear combinations of the digits mod 2. If the word to be encoded is (a_1, a_2, a_3, a_4), then the encoded word is

$$(a_1, a_2, a_3, a_4, a_2 + a_3 + a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4)\,,$$

so the last 3 digits are generalized parity checks. The complete list of codewords is

$$\begin{array}{llll}
(0,0,0,0,0,0,0), & (0,0,0,1,1,1,1), & (0,0,1,0,1,1,0), & (0,0,1,1,0,0,1), \\
(0,1,0,0,1,0,1), & (0,1,0,1,0,1,0), & (0,1,1,0,0,1,1), & (0,1,1,1,1,0,0), \\
(1,0,0,0,0,1,1), & (1,0,0,1,1,0,0), & (1,0,1,0,1,0,1), & (1,0,1,1,0,1,0), \\
(1,1,0,0,1,1,0), & (1,1,0,1,0,0,1), & (1,1,1,0,0,0,0), & (1,1,1,1,1,1,1)\,.
\end{array}$$

It is fairly easy to see that any two codewords differ in 3 places at least.

Now the extra bits are linear combinations of the information bits, so encoding can be done by using matrix multiplication with a matrix H given by

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

The word $\mathbf{x}$ is encoded by matrix multiplication on the right $\mathbf{y} = \mathbf{x}H$, for example

$$(1\ 0\ 1\ 1)\,H = (1\ 0\ 1\ 1\ 0\ 1\ 0)\,,$$

which agrees with the list of codewords. A *Mathematica* implementation of the above starts with mod 2 arithmetic via the `Mod[]` function

```
In[1]:= Mod[1+1, 2]

Out[1]= 0
```

(a more efficient implementation of Mod 2 addition is to replace 0 and 1 by 1 and -1 and use ordinary multiplication, but this only saves a factor of about 25% in speed, so is not very significant).

Implementing the encoding is done easily

```
H= {{1, 0 , 0 , 0 , 0 , 1 , 1},
    {0, 1 , 0 , 0 , 1 , 0 , 1},
    {0, 0 , 1 , 0 , 1 , 1 , 0},
    {0, 0 , 0 , 1 , 1 , 1 , 1}}

encode[x_]:= Mod[x . H, 2]
```

The entire list of codewords is generated by the *Mathematica* program

```
In[1]:= Mod[Distribute[Table[{0, 1}, {4}], List] . H2, 2]

Out[1]= {{0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 1, 1, 1},

>    {0, 0, 1, 0, 1, 1, 0}, {0, 0, 1, 1, 0, 0, 1}, {0, 1, 0, 0, 1, 0, 1},

>    {0, 1, 0, 1, 0, 1, 0}, {0, 1, 1, 0, 0, 1, 1}, {0, 1, 1, 1, 1, 0, 0},

>    {1, 0, 0, 0, 0, 1, 1}, {1, 0, 0, 1, 1, 0, 0}, {1, 0, 1, 0, 1, 0, 1},

>    {1, 0, 1, 1, 0, 1, 0}, {1, 1, 0, 0, 1, 1, 0}, {1, 1, 0, 1, 0, 0, 1},

>    {1, 1, 1, 0, 0, 0, 0}, {1, 1, 1, 1, 1, 1, 1}}
```

As in [20], the code `Distribute[Table[{0,1}, {4}], List]` generates all 16 combinations of $0,1$ vectors, and these are each multiplied by H, but *Mathematica* enables one to do this in one single matrix multiplication.

The decoding is accomplished using a number of facts. The first is the use of a *parity check matrix* which is a matrix P which has the property that $\mathbf{c}$ is a codeword if and only if

$$P\,\mathbf{c} = \mathbf{0}\,, \quad \text{(the notation } \mathbf{0} \text{ means a vector of all zeroes)},$$

i.e., you are doing a number of parity checks at once. In linear algebra terms, this says that the code is the *null space* of the matrix P. In the case of the Hamming code, this matrix is

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

The second fact is that any two distinct codewords in the Hamming code differ in at least 3 places. Finally, it is an amazing but true fact that any length 7 vector of 0's and 1's differs from a codeword in at most one place. This yields the decoding algorithm:

1. Compute $\mathbf{z} = P\,\mathbf{y}$.
2. If $\mathbf{z} \neq \mathbf{0}$, then $\mathbf{z}$ is equal to a column of P. Change the bit of $\mathbf{y}$ corresponding to this column.
3. Return the first 4 digits of $\mathbf{y}$.

You only return the first 4 digits as these are the information bits that were originally transmitted. This algorithm can be implemented as

```
P = {{0, 1, 1}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}}

decode[y_]:= Take[Mod[y + (If[# === Mod[y . P, 2], 1, 0]& /@ P), 2], 4]
```

Given the above facts, one can see why the algorithm works. Assume that the word received is $\mathbf{y}$ and write as

$$\mathbf{y} = \mathbf{x} + \mathbf{e}\,,$$

where $\mathbf{x}$ is the intended codeword and $\mathbf{e}$ represents the error in transmission. Multiplying by P gives

$$P\,\mathbf{y} = P\,\mathbf{x} + P\,\mathbf{e} = P\,\mathbf{e}$$

since $\mathbf{x}$ is a codeword. Now the fact that any word is at most one digit off a codeword implies that $\mathbf{e}$ is a vector with all 0's or a single 1. In the first case, there was no error in transmission, while in the second case, the product $P\,\mathbf{e}$ consists of picking off the column of P where the lone 1 in $\mathbf{e}$ is. In other words, the error bit corresponds to the column of P which equals $P\,\mathbf{e}$.

The search for efficient codes

The Hamming code is the first non trivial example of an efficient code, and it is interesting to try to other efficient codes. A first improvement would be to get a multiple error correcting code. An example is the next most famous code the *Golay code*, which is a 3-error correcting code, i.e., has minimum distance 8. This code has 12 information bits and 12 parity check bits, so there are 2048 codewords in all, so giving a list of codewords is not the best way to analyze it. However, a direct definition is still useful, so one can use the generator matrix. A less cumbersome way to generate this matrix is to consider the polynomial

$$g(x) = x + x^2 + x^3 + x^4 + x^6 + x^8 + x^9 + x^{12} + x^{13} + x^{16} + x^{18}$$

To generate the words of the Golay code take the powers of this polynomial

$$g(x), g(x)^2, g(x)^3, \ldots, g(x)^{23}$$

and reduce all terms $x^n \bmod 23$, for example, $x^{50} \mapsto x^4$. Then take vector of 0's and 1's with 1's exactly where there is a power of x.

To make this more precise, first note that the reduction mod 23 is equivalent to doing a polynomial remainder with $x^{23} - 1$. Also, one has to keep in mind that these operations are also done mod 2.

The polynomial $g(x)$ may seem to have come from nowhere, but it turns out that it is a standard construction that characterizes the Golay code as a quadratic residue code. The thing to notice is that the exponents of $g(x)$ are exactly the perfect squares mod 23. The next point is that the Golay code is a cyclic code which means that if c is a codeword, then so is any cyclic shift of c. What this says is that the polynomials x^c form a subring of polynomials modulo $x^{32} - 1$. It turns out that these rings are all generated by a single polynomial, and for quadratic codes, this is the one coming from the polynomial of residues.

There are other constructions of the Golay code, and it is clear that all these approaches combine to make it one of the most interesting objects in mathematics.

Weight enumerators of codes

Let $\mathcal{C}_2$ be the set of all doubly–even codes containing the all ones vector and $\mathcal{C}_1$ the set of all doubly–even self–dual codes, so that $\mathcal{C}_1 \subset \mathcal{C}_2$ (see below for definitions). Codes in $\mathcal{C}_1$ are of special interest and those of length ≤ 32 have been classified (see [5]). On the other hand, the number of equivalence classes $H_1(n)$ of such codes of length n satisfies $\log H_1(n) \gg n^2$ as $n \to \infty$ with $8 \mid n$ (see [15, Chapter 19]). This makes it desirable to find class invariants which distinguish between different classes of a given length but span a finitely generated algebra.

Natural candidates for such invariants are multiple weight enumerators. For any binary linear code C of length n and $r \in \mathbf{Z}^+$ define the r–fold weight enumerator of C to be

$$W_r(x; C) = \sum_{(c_1,\ldots,c_r)\in C^r} \prod_{\alpha\in\mathbf{F}_2^r} x_\alpha^{\omega_\alpha(c_1,\ldots,c_r)}\,, \tag{1.1}$$

where $x = (x_\alpha)$ is a 2^r–tuple of variables with $\alpha \in \mathbf{F}_2^r$ and $\omega_\alpha(c_1, \ldots, c_r)$ is the number of occurrences of α as a row in the matrix of column vectors $(c_1, \ldots, c_r)$. The polynomial $W_r \in \mathbf{Z}[x]$ is homogeneous of degree n and can be written using multi–indices $u = (u_\alpha)$ as

$$W_r(x; C) = \sum_m A_u x^u\,,$$

where $\sum_\alpha u_\alpha = n$ and A_u is the number of r–tuples $(c_1, \ldots, c_r)$ from C which have u_α occurrences of α in their rows for each α. Thus $W_1(x; C) = W_1(x_0, x_1; C)$ is the ordinary weight enumerator of C, while $W_2(x; C) = W_2(x_{00}, x_{01}, x_{10}, x_{11}; C)$ is the biweight enumerator of C introduced in [14] (see also [15]). The variables $x_{00}, x_{01}, x_{10}, x_{11}$, will also be denoted by x_0, x_1, x_2, x_3 corresponding to the binary expansion of their subscripts.

Now $W_r(x; C)$ is a class invariant satisfying

$$W_r(x; C_1 \oplus C_2) = W_r(x; C_1)\, W_r(x; C_2)\,.$$

As $\mathcal{C}_N$ is closed under $\oplus$ for $N = 1$ or 2 the set $\mathcal{W}_r(N)$ of all finite $\mathbf{C}$–linear combinations of 1 and $W_r(x; C)$ for $C \in \mathcal{C}_N$ forms a (graded) subalgebra of $\mathbf{C}[x]$. In fact,

$$\mathcal{W}_r(N) = \bigoplus_{d \equiv 0 \pmod{4/N}} \mathcal{W}_r^d(N)\,,$$

where $\mathcal{W}_r^d(N)$ is the vector space of polynomials in $\mathcal{W}_r(N)$ which are homogeneous of degree d. Let

$$P(t) = P_{r,N}(t) = \sum_{d \equiv 0 \pmod{4/N}} \dim \mathcal{W}_r^d(N)\, t^d$$

be the associated "Poincaré series". Two basic problems are (i) to show that $\mathcal{W}_r(N)$ is finitely generated, (ii) to compute $P(t)$.

When $r = N = 1$ these problems were solved by Gleason in 1970 [10] from which it follows that $W_1(1)$ is actually freely generated by

$$x_0^8 + 14\, x_0^4\, x_1^4 + x_1^8 \qquad \text{and} \qquad x_0^{24} + 759\, x_0^{16} x_1^8 + 2576\, x_0^{12} x_1^{12} + 759\, x_0^8 x_1^{16} + x_1^{24}\,.$$

These polynomials are the weight enumerators of the $[8,4]$ Hamming code and the $[24,12]$ Golay code, both of which are in $\mathcal{C}_1$. Thus, one also has

$$P_{1,1}(t) = \frac{1}{(1-t^8)\,(1-t^{24})}\,.$$

Gleason obtained this result by identifying $\mathcal{W}_1(1)$ with the algebra of invariant polynomials for a finite unitary reflection group of order 192 (see Section 2). Later proofs were given based on a connection with classical modular forms (see [3], [4] and [6]). This approach also leads to a solution of these problems for $r = 1$ and $N = 2$. It follows from [12, Theorem 9] that $W_1(2)$ is freely generated by

$$x_0^4 + x_1^4 \qquad \text{and} \qquad x_0^8 + x_1^8\,,$$

which are the weight enumerators of the $[4,1]$ and $[8,1]$ repetition codes and are in $\mathcal{C}_2$ and so

$$P_{1,2}(t) = \frac{1}{(1-t^4)\,(1-t^8)}\,.$$

It is not known whether $\mathcal{W}_r(N)$ is in general finitely generated for $r > 1$. The result for $r = 2$ is proved in [8] using a connection with Siegel modular forms. The result proved there is that $\mathcal{W}_2(2)$ is freely generated by the biweight enumerators of r_4, r_8, r_{12} and d_8 so that

$$P_{2,2}(t) = \frac{1}{(1-t^4)\,(1-t^8)^2\,(1-t^{12})}\,,$$

and $\mathcal{W}_2(1)$ is generated by the biweight enumerators of $d_8^+, d_{24}^+, d_{32}^+, d_{40}^+$ and g_{24} so

$$P_{2,1}(t) = \frac{1+t^{32}}{(1-t^8)\,(1-t^{24})^2\,(1-t^{40})}\,.$$

An important consquence of finding explicit generators for $\mathcal{W}_r(N)$ is that computation of weight enumerators of *extremal codes*, i.e., codes that have largest possible minimum distance is facilitated. In fact, we compute the weight enumerators for extremal codes of length $48, 72, 96, 120$, and 144, even though these are not known to exist in the last four cases.

The group associated to r–fold enumerators

The r–fold weight enumerators are invariant under a finite group G_r of $2^r \times 2^r$ matrices. It will be shown that this group is generated by the matrices

$$a_r = a_1^{\otimes r}$$

where

$$a_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and by

$$\left\{ b(S) = \operatorname{diag}\left(e\left(\frac{1}{4}\, {}^t\alpha S \alpha \right) \right);\quad \alpha \in \mathbf{F}_2^r,\ S = {}^tS \right\},$$

where

$$a^{\otimes r} \stackrel{\text{def}}{=} \overbrace{a_1 \otimes \cdots \otimes a_1}^{r \text{ times}},$$

and $e(x) = e^{2\pi i x}$. Here $A \otimes B$ is the tensor or Kronecker product or matrices given by

$$A \otimes B = \begin{pmatrix} AB_{1,1} & \dots & AB_{1,s} \\ \vdots & & \vdots \\ AB_{r,1} & \dots & AB_{r,s} \end{pmatrix}.$$

First, we show that a_r preserves the r–fold enumerators. In fact, this is just a generalization of the MacWilliams identity for ordinary weight enumerators [15].

Consider the function

$$w(c_1, \ldots, c_r) = \prod_{\alpha \in \mathbf{F}_2^r} x_\alpha^{\omega(c_1,\ldots,c_r)}$$

so that

$$w_2(x; C) = \sum_{c_1,\dots,c_r \in C} w(c_1, \dots, c_r).$$

If one defines

$$g(c_1, \dots, c_r) = \sum_{\gamma_1,\dots,\gamma_r \in \mathbf{F}_2^r} (-1)^{\Sigma\langle\gamma_i, c_i\rangle} w(\gamma_1, \dots, \gamma_r)\,,$$

then

$$\sum_{c_1,\dots,c_r} g(c_1, \dots, c_r) = \sum_{\gamma_1,\dots,\gamma_r \in \mathbf{F}_2^r} w(\gamma_1, \dots, \gamma_r) \sum_{c_1,\dots,c_r \in \mathbf{F}_2^r} (-1)^{\Sigma\langle\gamma_i, c_i\rangle}\,. \tag{2.1.1}$$

Now the inner sum is zero unless every $\langle c_i, \gamma_i\rangle = 0$, so this sum is

$$|C|^r \sum_{\gamma_1,\dots,\gamma_r \in C^\perp} f(\gamma_1, \dots, \gamma_r) = |C|^r\, W_r(x; C^\perp)\,.$$

On the other hand, looking at every term on the left of (2.1.1)

$$\begin{aligned} g(c_1, \dots, c_r) &= \sum_{\gamma_1,\dots,\gamma_r} (-1)^{\Sigma_i \gamma_i, c_i\rangle} \prod_{\alpha \in F_2^r} x_\alpha^{\omega_\alpha(\gamma_1,\dots,\gamma_r)} \\ &= \sum_{\gamma_1,\dots,\gamma_r \in \mathbf{F}_2^r} (-1)^{\Sigma_{i,j} \gamma_{ij} c_{ij}} \prod_{\alpha \in \mathbf{F}_2^r} x_\alpha^{\omega_\alpha(\gamma_1,\dots,\gamma_r)} \\ &= \sum_{\gamma'_1,\dots,\gamma'_n \in \mathbf{F}_2^r} (-1)^{\Sigma_j \langle\gamma'_j, c'_j\rangle} x_{\gamma'_1} \cdots x_{\gamma'_n}\,, \end{aligned}$$

where the r column vectors $c_1, \dots, c_r$ are considered to be a matrix (c_{ij}) with rows $c'_1, \dots, c'_n$ in $\mathbf{F}_2^r$. One gets the further reduction

$$\begin{aligned} &= \prod_{j=1}^{n} \left(\sum_{\gamma' \in \mathbf{F}_2^r} (-1)^{\langle\gamma', c'_j\rangle} x_{\gamma'} \right) \\ &= \prod_{\alpha \in \mathbf{F}_2^r} \left(\sum_{\beta \in \mathbf{F}_2^r} (-1)^{\langle\alpha,\beta\rangle} x_\beta \right)^{w_\alpha(c_1,\dots,c_r)}\,. \end{aligned}$$

It follows that

$$W_r(x; C^\perp) = W_r(Ax; C)\,,$$

for the matrix A given by

$$A(i,j) = \frac{1}{|C|^{r/n}} (-1)^{\langle\alpha_i, \alpha_j\rangle}\,,$$

where $\alpha_1, \dots, \alpha_{2^r}$ are the elements of $\mathbf{F}_2^r$ corresponding to the variables $x_{\alpha_1}, \dots, x_{\alpha_{2^r}}$.

If C is self dual, then $|C| = 2^{n/2}$, so

$$W_r(x; C) = W_r(a_r^* x; C)\,,$$

for the matrix a_r^* given by

$$a_r^*(i,j) = \frac{1}{2^{r/2}}\,(-1)^{\langle \alpha_i, \alpha_j\rangle}\,.$$

Ordering $\mathbf{F}_2^r$ corresponding to the binary expansion of $0,\ldots,2^r-1$ it is easily seen that $a_r^* = a_r$.

Next, we show that the $b(S)$ preserve the r–fold enumerators. It is clear that

$$b(S_1)\otimes b(S_2) = b(S_1\oplus S_2) \tag{2.1.2}$$

so one only needs to consider additive generators of symmetric matrices, i.e., matrices $S_{i,j}$ which are 1 at the (i,j) and (j,i) positions and 0 elswhere, and matrices S_i which are 1 at the (i,i) position and 0 elsewhere.

Now

$$W_r(b(S)\,x;C) = \sum_c \prod_\alpha x_\alpha^{\omega_\alpha(c)} e\left(\frac{1}{4}\ {}^t\alpha S\alpha\,\omega_\alpha(c)\right)$$

and for $\imath \neq j$ the value of ${}^t\alpha S_{i,j}\alpha$ is 2 or 0 depending on whether $\alpha_i = \alpha_j = 1$ or not. Similarly, the value of ${}^t\alpha S_i\alpha = 1$ exactly when $\alpha_i = 1$. Moreover, one has the identities

$$\sum_{\alpha\colon {}^t\alpha S_{i,j}\alpha = 2} \omega_\alpha(c_1,\ldots,c_r) = \omega_{11}(c_i,c_j)\,,$$

$$\sum_{\alpha\colon {}^t\alpha S_i\alpha = 1} \omega_\alpha(c_1,\ldots,c_r) = \omega_1(c_i)\,.$$

So if C is orthogonal, then ω_{11} and ω_1 are both even so for S even one has that

$$W_r(b(S)x;C) = W_r(x;C)\,. \tag{2.1.2}$$

If C is even, then $4\,|\,\omega_1$ so (2.1.2) holds for all S.

It will be shown that the group G_r contains all automorphisms of the r–fold weight enumerators.

Note that $a_r^2 = I_{2^r}$ and $b(S)^{-1} = b(-S)$. Let

$$U_\ell = \{e(j/\ell)\,I;\quad j \pmod{\ell}\}$$

be the cyclic group of scalar ℓth roots of unity. Let $G_0 = U_4$ and $b_1 = b((1))$,

Lemma 2.1 *(a) If r is odd then $U_8 \subset G_r$.*

(b) If r is even then $U_4 \subset G_r$.

Proof: (a) We have

$$(a_1b_1)^3 = e(1/8)\,I_2$$

so it follows from (2.1.1) that $b_1^{\otimes r} \in G_r$ hence G_r contains

$$(a_rb_1^{\otimes r})^3 = ((a_1b_1)^3)^{\otimes r} = e(r/8)\,I_{2^r}$$

and the result follows.

(b) We have from (2.1.1) that

$$b' = I_{2^{r-1}} \otimes b_1 \in G_r$$

so, since $a_1^2 = I_2$,

$$(a_r b')^6 = I_{2^{r-1}} \otimes (a_1 b_1)^6 = i\, I_{2^r} \in G_r$$

giving the result for $r \geq 2$. For $r = 0$ the result follows by definition. □

Let G_r' be the finite group of $2^r \times 2^r$ matrices generated by $a_r' = e(-3r/8)\, a_r$ and by the $b(S)$, as before, and let $G_0' = G_0$. By the above Lemma, $G_r' \subset G_r$ with equality if r is even. Equality does not hold for $r = 1$, as will be seen.

Lemma 2.1.2 *For any $r \geq 1$ there are two embeddings of G_{r-1}' in G_r' given by*

$$\gamma \mapsto \gamma \otimes I_2, \qquad \gamma \mapsto I_2 \otimes \gamma.$$

Proof: The case $r = 1$ is immediate from the Lemma. If $r \geq 2$ then by (??) it is enough to show for the first embedding that $a_{r-1}' \otimes I_2 \in G_r'$ and $b(S) \otimes I_2 \in G_r'$ for an integral symmetric $(r-1) \times (r-1)$ matrix S. The latter follows from (??), while in the former case one has that G_r' contains

$$(a_r'\,(I_{2^{r-1}} \otimes b_1))^9 = (e(-3(r-1)/8)\, a_{r-1})^9 \otimes (e(-3/8)\, a_1 b_1)^9 = (e(-3(r-1)/8)\, a_{r-1})^9\, a_{r-1} \otimes I_2$$

as required. The other embedding is similar. □

For $r \leq 2$ it turns out that G_r and G_r' are well known finite unitary reflection groups. This follows from

Lemma 2.1.3 *(a) G_1' is a group of order ?? generated by the unitary reflections*

$$u = \frac{e(-1/8)}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \qquad v = \begin{pmatrix} -i & 0 \\ 0 & 1 \end{pmatrix}.$$

(b) G_1 is a group of order 192 (group number ?? in [18]) generated by the unitary reflections

$$w = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & i \\ -i & 1 \end{pmatrix}, \qquad v.$$

(c) $G_2' = G_2$ is a group of order 46080 (group number 31 in [18]) generated by the five unitary

reflections

$$d = \mathrm{diag}\,(1,1,1,-1), \qquad e = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$g = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad h = \frac{1}{2}\begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix}$$

Remark: The invariant theory of this group was studied as early as 1887 by Maschke [13].

Proof: (a) One has

$$a_1' = v^2u^3, \qquad b_1 = u^2v^3u^2\,,$$

while

$$u = (a_1')^3\, b_1^2, \qquad v = (a_1')^6 b_1\,.$$

(b) Similar to (a).

(c) Write $b_2 = I_2 \otimes b_1$ and $b_3 = b_1 \otimes I_2$. It is easily checked that $G_2 = G_2'$ is generated by a_2, b_2, b_3, d. Now

$$a_2 = gdedgdhdg(de)^2g, \quad b_2 = gefed, \quad b_3 = gfged\,,$$

while

$$e = (da_2)^3 b_2 da_2 (b_3 d)^3 a_2 b_2 a_2, \quad f = (da_2)^3 b_2 b_3^3,$$

$$g = b_2 da_2 (b_3 d)^3 a_2 b_2 (da_2)^4, \quad h = dgdedga_1 g(ed)^2 gd,$$

which combined with (??) give the result. Note that the formulas for e and g were found using [9, p. 326] and checked by computer. □

We now identify certain subgroups of G_r. For example, since the weight enumerator is a sum of terms $w(c_1, \ldots, c_r)$, any map permuting all the r–tuples $(c_1, \ldots, c_r)$ leaves the weight enumerator invariant. It follows that the weight enumerator is invariant under a representation of $GL(r, \mathbf{F}_2^r)$ to $M(2^r, \mathbf{C})$, the set of $2^r \times 2^r$ matrices, and actually under "affine" transformations of the form $a + g$, where $a \in \mathbf{F}_2^r$ and $g \in GL(r, \mathbf{F}_2^r)$. Call this subgroup H_r.

To see how this group acts exactly, let $\alpha \in \mathbf{F}_2^r$ and $g \in GL(r, \mathbf{F}_2^r)$, with $\alpha = (\alpha_i)$ and $g = (g_{ij})$. To $\sigma = (\alpha, g) \in H_r$ associate a substitution $T_\sigma f(x)$ having the property that

$$T_\sigma w(c_1, \ldots, c_r) = w\left(\alpha_1\,\mathbf{1} + \sum g_{1j} c_i, \ldots, \alpha_r\,\mathbf{1} + \sum g_{rj} c_r\right)\,,$$

where $\alpha_i\,\mathbf{1} = (\alpha_i, \ldots, \alpha_i)$ is the vector of all α_i's (c contains $\mathbf{1}$ by assumption).

It follows that $T_\sigma\, f(x) = f(x\sigma)$ where $(x_\gamma)\sigma = (x_{\gamma\sigma})$, $\gamma \in \mathbf{F}_2^r$, and

$$\gamma\sigma = \alpha + g\gamma\,.$$

Note that this uses the expression of the variables in the weight enumerators in terms of elements of $\mathbf{F}_2^r$.

Theorem 2.1.1 *The group H_r acts as a permutation group on the variables $x_0, \ldots, x_{2^r-1}$ and is three transitive for $r \geq 2$. It is the largest permutation subgroup of G_r.*

Corollary 2.1.1 *The biweight enumerators are symmetric functions.*

Invariant theory of G_2

Since all biweight enumerators are symmetric functions one introduces a linear symmetrizing operator on polynomials defined on terms x^u by

$$\operatorname{Sym} x^u = \frac{1}{|\operatorname{Stab}(u)|} \sum_{\alpha \in S_n} x^{u\alpha}\,,$$

where $\operatorname{Stab}(u) = \{\alpha \in S_k :\ u\alpha = u\}|$ for $u\alpha = (u_{0\alpha}, u_{1\alpha}, \ldots, u_{(n-1)\alpha})$.

For example, let d_8^+ be the $[8,4]$ Hamming code, then

$$W_2(x; d_8^+) = \operatorname{Sym}\,(x^{(0,0,0,8)} + 14\, x^{(0,0,4,4)} + 168\, x^{(2,2,2,2)})\,.$$

For d_n, d_n^+ one has the identities

$$W_2(x; d_n) = \frac{1}{4} \sum_{\beta \in \mathbf{F}_2^2} \left(\sum_{\alpha \in \mathbf{F}_2^2} (-1)^{\langle \alpha, \beta \rangle}\, x_\alpha^2 \right)^{n/2}, \qquad \text{for } n \equiv 0 \pmod 4\,,$$

$$W_2(x; d_n^+) = \frac{1}{4} \sum_{\beta, \gamma \in \mathbf{F}_2^2} \left(\sum_{\alpha \in \mathbf{F}_2^2} (-1)^{\langle \alpha, \beta \rangle}\, x_{\alpha+\gamma} x_\alpha \right)^{n/2}, \qquad \text{for } n \equiv 0 \pmod 8\,,$$

which can be used to compute the weight enumerators efficiently. Explicit forms for the generators are found in the tables.

In [11] Huffman finds free generators $P_8, P_{12}, P_{20}, P_{24}$ of degrees $8, 12, 20, 24$ for the ring of invariants $\mathcal{A}$ of G_2. Huffman also shows that the subalgebra of invariants $\mathcal{A}^{(8)}$ consisting of polynomials in $\mathcal{A}$ with homogeneous terms of degree divisible by 8 is given by

$$\mathbf{C}[P_8, P_{12}^2, P_{24}] \oplus (P_{12}P_{20})\mathbf{C}[P_8, P_{12}^2, P_{24}]\,.$$

Let

$$W_{12}(x) = \mathrm{Sym}\,(x^{(0,0,0,12)} - 33\,x^{(0,0,4,8)} + 330\,x^{(0,4,4,4)} + 792\,x^{(2,2,2,6)})$$

$$\begin{aligned} W_{20}(x) &= (x_0x_1 + x_2x_3)(x_0x_2 + x_1x_3)(x_0x_3 + x_1x_2)(x_0x_1 - x_2x_3)(x_0x_2 - x_1x_3)(x_0x_3 - x_1x_2) \\ &\qquad \times (x_0^2 + x_1^2 + x_2^2 + x_3^2)(x_0^2 - x_1^2 + x_2^2 - x_3^2)(x_0^2 + x_1^2 - x_2^2 - x_3^2)(x_0^2 - x_1^2 - x_2^2 + x_3^2) \\ &= \mathrm{Sym}\,(-\,x^{(0,4,4,12)} + 2\,x^{(0,4,8,8)} +\, x^{(2,2,2,14)} -\, x^{(2,2,6,10)} - 14\,x^{(2,6,6,6)} + 13\,x^{(4,4,4,8)}) \end{aligned}$$

Then one has

$$\begin{aligned} P_8 &= \frac{1}{16}\,W_2(x;d_8^+) \\ P_{12} &= \frac{1}{576}\,W_{12}(x) \\ P_{20} &= \frac{1}{48}\,W_2(x;d_8^+)W_{12}(x) + 36W_{20}(x) \\ P_{24} &= \frac{1}{56}\,W_2(x;d_8^+)^3 + \frac{1}{88}\,W_{12}(x)^2 - \frac{9}{308}\,W_2(x;g_{24}^+) \\ 14W_{12}(x)^2 &= -121W_2(x;d_8^+)^3 + 63W_2(x;d_{24}^+) + 72W_2(x;g_{24}) \\ P_{24} &= \frac{9}{112}\,W_2(x;d_8^+)^3 - \frac{9}{176}\,W_2(x;d_{24}^+) - \frac{9}{308}\,W_2(x;g_{24}) \\ P_{12}P_{20} &= \frac{3}{16384}\,W_2(x;d_{32}) + \frac{1}{59136}\,W_2(x;d_8^+)W_2(x;d_{24}^+) - \frac{7}{11264}\,W_2(x;d_8^+)W_2(x;g_{24}) \\ &\qquad + \frac{1417}{3096576}\,W_2(x;d_8^+)^4 \\ P_{20}^2 &= \frac{27}{13120}\,W_2(x;d_{40}^+) - \frac{603}{83968}\,W_2(x;d_8^+)W_2(x;d_{32}^+) + \frac{1721}{230912}\,W_2(x;d_8^+)^2W_2(x;d_{24}^+) \\ &\qquad + \frac{9}{28864}\,W_2(x;d_8^+)^2W_2(x;g_{24}) - \frac{8341}{3778560}\,W_2(x;d_8^+)^5 \end{aligned}$$

It follows that the algebra of invariants of G_2 is also generated by $W_2(x;d_8^+), W_{12}(x), W_{20}(x), W_2(x;g_{24})$ (or $W_2(x;g_{24})$ replaced by $W_2(x;d_{24}^+)$. Furthermore, $\mathcal{A}^{(8)}$ is given by

$$\mathbf{C}[W_2(x;d_8^+), W_2(x;d_{24}^+), W_2(x;g_{24}), W_2(x;d_{40}^+)] \oplus W_2(x;d_{32}^+)\mathbf{C}[W_2(x;d_8^+), W_2(x;d_{24}^+), W_2(x;g_{24}), W_2(x;d_{40}^+)].$$

This implies that $W_2(x;d_{32})^2$ is be expressible in terms of the other generators. In fact, the generators satisfy the relation

$$\begin{aligned} f_{64}(y) &= 98451045y_4^2 - 81962496y_2y_5 - 382258800y_1y_2y_4 + 837202240y_1^2y_2^2 \\ &\quad + 157420032y_1^3y_5 - 57248730y_1^4y_4 - 1012059664y_1^5y_2 + 446026933y_1^8 \\ &\quad - 93671424y_3y_5 + 345058560y_1y_3y_4 - 413378560y_1^2y_2y_3 + 169773824y_1^5y_3 - 13352960y_1^2y_3^2\,, \end{aligned}$$

i.e., this polynomial is zero under the substitutions

$$y_1 \mapsto W_2(x; d_8^+), \quad y_2 \mapsto W_2(x; d_{24}^+), \quad y_3 \mapsto W_2(x; g_{24}^+), \quad y_4 \mapsto W_2(x; d_{32}^+), \quad y_5 \mapsto W_2(x; d_{40}^+), .$$

One checks directly that the generators do not satisfy satisfy a relation of smaller degree.

Invariants for higher r

As before, let $x = (x_\alpha)$, $\alpha \in \mathbf{F}_2^r$, be a vector of 2^r variables. Let

$$\mathcal{A}_r = \{f \in \mathbf{C}[x] : \ f(x\gamma) = f(x) \text{ for all } \gamma \in G_r\}$$

$$\mathcal{A}'_r = \{f \in \mathbf{C}[x] : \ f(x\gamma) = f(x) \text{ for all } \gamma \in G'_r\}$$

be the polynomial invariants of G_r and G'_r. Clearly $\mathcal{A}'_r$ is an algebra and $\mathcal{A}_r \subset \mathcal{A}'_r$ is a subalgebra with equality if r is even. It is easily seen that

$$\mathcal{A}'_r = \bigoplus_{m \geq 0} \mathcal{A}'_{r,m} ,$$

where $\mathcal{A}'_{r,m}$ consists of the homogeneous elements of degree m. Now $\mathcal{A}'_{r,m}$ is a finite dimensional vector space which by Lemma ?? has dimension zero unless $4 \,|\, m$. Also

$$\mathcal{A}_r = \bigoplus_{m \geq 0} \mathcal{A}_{r,m} ,$$

where $\mathcal{A}_{r,m} \subset \mathcal{A}'_{r,m}$ is the subspace of G_r –invariants. By Lemma ?? it is seen that $\mathcal{A}_r \subset \mathcal{E}_r$ if r is odd??.

Lemma 2.2.1 *Suppose that $f \in \mathcal{A}'_{r,m}$ and $r = r_1 - r_2$ with $r_1, r_2 \geq 0$. Then for any fixed $y \in \mathbf{C}^{2^{r_1}}$ one has*

$$f(x \otimes y), f(y \otimes x) \in \mathcal{A}'_{r_2,m} .$$

Proof: This follows easily from Lemma ??. □

Now suppose that $r \geq 1$. By Lemma ?? there is a map $\varphi : \mathcal{A}'_{r,m} \to \mathcal{A}'_{r-1,m}$ given by

$$\varphi f(x) = f(x \otimes (1, 0)) .$$

It is clear that φ is linear and that a space of *cups invariants* can be defined by $\mathcal{B}'_{r,m} = \ker \varphi$. Now φ extends to an algebra homomorphism $\varphi : \mathcal{A}'_r \to \mathcal{A}'_{r-1}$ with kernel

$$\mathcal{B}'_r = \bigoplus_{m \geq 0} \mathcal{B}'_{r,m} ,$$

an ideal of $\mathcal{A}'_r$. Define the subalgebra

$$\mathcal{E}_r = \bigoplus_{8\,|\,m} \mathcal{A}_{r,m}$$

and note that by Lemma ??, $\mathcal{A}_r = \mathcal{E}_r$ when r is odd.

Let $\mathcal{W}_r = \mathcal{W}^r_{\mathcal{C}_2}$ be the subalgebra of finite linear combinations of genus r weight enumerators of even self–dual codes. By Lemma ?? and ?? one has

Proposition 2.2.? *$\mathcal{W}_r$ is a subalgebra of $\mathcal{E}_r$ for $r \geq 0$.*

From ??, one has that if $W_r(x; C) \in \mathcal{W}_r$ with $r \geq 1$, then

$$\varphi W_r(x; C) = W_{r-1}(x; C)\,.$$

Since $W_{r-1} \neq 0$, it is seen that W_r is never a cusp invariant.

Extremal codes

One can use the explicit generators to compute the biweight enumerator of an extremal code in $\mathcal{C}_1$ of length 48 as well as the biweight enumerator that extremal codes of length 72 and 96, if they exist, would have. The computation shows that these enumerators are in fact unique. The computation for length 120 and 144 finds exactly two linearly independent generators, thus a one parameter family of extremal enumerators (recall that the coefficent of $\operatorname{Sym} x^{0^{n-1}n^1}$ is 1). In all the cases, enumerators with nonnegative coefficients were found, so existence of such extremal codes cannot be ruled out by this computation. Moreover, they satisfy a further condition given by the following.

The Assmus–Mattson theorem [2] states that extremal codes give designs, so this leads to conditions on the coefficients of the biweight enumerators since these correspond to counting intersections numbers of blocks in the designs.

First, a general fact is proved about designs

Proposition 2.3.1 *(a) Consider a t-(n, k, λ) design such that the total number of pairs of unequal blocks intersecting at m elements is I_m, then*

$$\binom{\lambda}{2}\binom{n}{t} = \sum_{m \geq t} I_m \binom{m}{t}. \tag{2.3.1}$$

(b) Consider two designs D_1, D_2 with parameters t-(n, k_1, λ_1) and t-(n, k_2, λ_2) such that D_1, D_2 have no blocks in common, and let I'_m be the total number of pairs of blocks with the first from D_1 and the second from D_2 which intersect at m places. Then

$$\lambda_1\,\lambda_2 \binom{n}{t} = \sum_{m \geq t} I'_m \binom{m}{t}.$$

Proof: (a) The left hand side of (2.3.1) counts the number of pairs of blocks whose intersection contains a given t–tuple, taken over all t–tuples in the design. The right hand side counts the same number by considering each pair of blocks according to how many points there are in their intersection, where the binomial factor comes from the fact that each pair is counted $\binom{m}{t}$ times for every t–subset in the intersection.

(b) The proof is similar to (a). □

Applying this to the biweight enumerator gives

Theorem 2.3.1 *(a) If the weight k codewords of a code C of length n form a t–design, then the coefficients of $W_2(x;C)$ satisfy*

$$\binom{A_{(n-k,k,0,0)}\ \binom{k}{t}/\binom{n}{t}}{2}\binom{n}{t}=\frac{1}{2}\sum_{m\geq t}A_{(n-2k+m,k-m,k-m,m)}\binom{m}{t}. \tag{2.3.2}$$

(b) If the weight k_1 and weight k_2 codewords both form t–designs, where $k_1\neq k_2$, then

$$A_{(n-k_1,k_1,0,0)}A_{(n-k_2,k_2,0,0)}\binom{k_1}{t}\binom{k_2}{t}\Big/\binom{n}{t}=\sum_{m\geq t}A_{(n-k_1-k_2+m,k_1-m,k_2-m,m)}\binom{m}{t}. \tag{2.3.3}$$

Proof: (a) Since the number of length k codewords is $A_{(n-k,k,0,0)}$, the coefficient of $x_0^{n-k}x_1^k$ in $W_(x;C)$ (the variables are written using the binary notation), the design property implies that

$$\lambda=A_{(n-k,k,0,0)}\binom{k}{t}\Big/\binom{n}{t}.$$

Also $A_{(n-2k+m,k-m,k-m,m)}$, is the coefficient of $x_0^{n-2k+m}x_1^{k-m}x_2^{k-m}x_3^m$ in $W_2(x;e_n)$, and counts the number of ordered pairs of codewords C_k, C'_k intersecting at m places, i.e., is equal to twice I_m. The result follows from Proposition 2.3.1.

The proof of part (b) is similar. □

The extremal polynomials for $n=40,48,72,96$ were computed. For the case $n=40$, two linearly independent extremal solutions invariant under G_2 exist. In [19] a list of 79 inequivalent extremal codes of length 40 is given. The codes corresponding to the 2–(19, 10, 5) designs III and QR of [19, Table 1][1] are given in Table ??, where the code is generated by

$$(I_{20},A)$$

[1]There is a misprint in [19, Table 1] and the (4, 11) position of the matrix corresponding to design IV should be 0 and not 1.

where A is the matrix given in the table, and their biweight enumerators are given in table ??. Since these are linearly independent, they generate the extremal invariants of length 40.

The biweight enumerators for the 78 other codes is given in Table ?? in abbreviated form, i.e., by giving the coefficient of $x_0^{28}x_1^4x_2^4x_3^4$ (the number of pairs of weight 8 codewords intersecting at 4 places) since this determines the enumerator uniquely. The codes are derived by taking one of the matrices A_{III}, A_{IV}, A_{QR} and inverting the columns c_i given in the table[2] by $1 - c_i$ (i.e., exchange 0's and 1's in the column), one then inverts the rows which have new row sum $\equiv 1 \pmod 4$.

In fact, the computational method yields the following improvement on [16, Theorem 3]

Theorem ??. *Assume that the self dual–code C has generating matrix (I, A) and r–fold linear combinations such that all r–fold linear combinations of A have weight $\geq d - r$, for $1 \leq r \leq d/2$, and all r–fold linear combinations of the columns of A also have weight $\geq d - r$, $1 \leq r < d/2$, then C has minimum weight d.*

As stated above in the cases $n = 72$, and $n = 96$ the enumerator was unique and the coefficients were also found to satisfy (2.3.2) and (2.3.3) corresponding to 5–designs. For $n = 120$ and $n = 144$ there were two independent solutions which also satisfied the conditions (2.3.2) and (2.3.3). These computations provide evidence that the computation was correct. The fact that the coefficients are positive for $n = 72, 96, 120, 144$, gives positive evidence that extremal codes exist in these cases.

The polynomials for $n = 40, 48, 72, 96$ are given in the tables.

Higher weight enumerators

The situation for the third weight enumators is more complicated. For example, the polynomials are no longer symmetric. As well certain identies no longer hold, thus one has that

$$W_2(x; d_8^+ \otimes d_8^+) = W_2(x; d_8^+)^2 = W_2(x; d_{16}^+)$$

since $W_2(x; d_8^+)$ is the only generator of length < 24). Computing the corresponding third weight enumerators directly gives

$$\begin{aligned} W_3(x; d_8^+) = \operatorname{Sym}\,(&x^{(0,0,0,0,0,0,0,8)} + 14\,x^{(0,0,0,0,0,0,4,4)} + 1344\,x^{(1,1,1,1,1,1,1,1)}) \\ &+ 168\,(x^{(0,0,0,0,2,2,2,2)} + x^{(0,0,2,2,0,0,2,2)} + x^{(0,0,2,2,2,2,0,0)} + x^{(0,2,0,2,0,2,0,2)} + x^{(0,2,0,2,2,0,2,0)} \\ &\quad + x^{(0,2,2,0,0,2,2,0)} + x^{(0,2,2,0,2,0,0,2)} + x^{(2,0,0,2,0,2,2,0)} + x^{(2,0,0,2,2,0,0,2)} + x^{(2,0,2,0,0,2,0,2)} \\ &\quad + x^{(2,0,2,0,2,0,2,0)} + x^{(2,2,0,0,0,0,2,2)} + x^{(2,2,0,0,2,2,0,0)} + x^{(2,2,2,2,0,0,0,0)}), \end{aligned}$$

[2] In [19, Table II] the columns are written $0, 1, \ldots, 19$ instead of $1, 2, \ldots, 20$ here. Moreover, there is a typographical error in entry 60 of the table and we have used the inverted columns $\{2, 3, 11, 14\}$ (in Tonchev's notation this is $\{1, 2, 10, 13\}$) which also leads to a code different from the other 79 in the list.

and

$$W_3(x;d_{16}^+) = \mathrm{Sym}\,(x^{(0,0,0,0,0,0,0,8)} + 28\,x^{(0,0,0,0,0,4,12)} + 198\,x^{(0,0,0,0,0,8,8)} +$$

$$420\,x^{(0,0,0,0,0,4,4,8)} + 43008\,x^{(1,1,1,1,1,1,5,5)} + 2298240\,x^{(2,2,2,2,2,2,2,2)})$$

$$336\,(x^{(0,0,0,0,2,2,2,10)}, 56) + 4704\,(x^{(0,0,0,0,2,2,6,6)}, 84) + 2520\,(x^{(0,0,0,0,4,4,4,4)}, 56)$$

$$+\,29400\,(x^{(0,0,0,0,4,4,4,4)}, 14) + 1680\,(x^{(0,0,0,2,2,2,2,8)}, 56) + 3360\,(x^{(0,0,0,2,2,2,4,6)}, 224)$$

$$+\,63840\,(x^{(0,0,2,2,2,2,4,4)}, 84) + 430080\,(x^{(1,1,1,1,3,3,3,3)}, 14)\,,$$

and the difference is

$$W_3(x;d_8^+)^2 - W_3(x;d_{16}^+) = \mathrm{Sym}\,(2688\,x^{(1,1,1,1,1,1,1,9)} - 5376\,x^{(1,1,1,1,1,1,5,5)} - 96768\,x^{(2,2,2,2,2,2,2,2)})$$

$$-\,1344\,(x^{(0,0,0,0,4,4,4,4)}, 56) - 1344\,(x^{(0,0,0,2,2,2,2,8)}, 56) + 1344\,(x^{(0,0,0,2,2,2,4,6)}, 224)$$

$$-\,2688\,(x^{(0,0,2,2,2,2,4,4)}, 84) + 21504\,(x^{(1,1,1,1,3,3,3,3)}, 14)\,.$$

where the notation (x^u, k) means that k distinct permutations of the vector u appear. The third weight and fourth weight enumerators of Golay are given in the tables.

3. Computational results

3.1 A direct computation of $W_r(x;C)$

The r–fold weight enumerator $W_r(x;C)$ is a sum of $2^{nr/2}$ terms of the form $w(c_1,\ldots,c_r)$, for $c_1,\ldots,c_r \in C$, where

$$w(c_1,\ldots,c_r) = \prod_{\alpha\in\mathbf{F}_2^r} x_\alpha^{\omega_\alpha(c_1,\ldots,c_r)}\,.$$

For computational purposes it is more useful to consider the variables using the binary representation of F_2^r. One then constructs the function w as follows: If a word $\gamma = (\gamma_0,\ldots,\gamma_{n-1}) \in \mathbf{Z}^n$ is given, let $w(\gamma) = \prod_{k=0}^{n-1} x_{\gamma_k}$, and one extends w to r arguments by taking

$$w(c_1,\ldots,c_r) = w\left(\sum_{k=1}^{r} 2^{r-k} c_k\right)\,,$$

where addition is taken over $\mathbf{Z}$.

The computations use the fact that if $g(x) = w(c_1,\ldots,c_m)$ is known, then

$$w(v_{1\pi},\ldots,v_{m\pi}),\quad w(v_1,v_2,\ldots,1-v_j,\ldots,v_m),\quad w(v_1,\ldots,v_1+v_j,\ldots,v_m)$$

can be computed from $g(x)$ by a substitution of variables as in Section 2.1.1??.

In general, assume that the code is generated by $c_1, c_2, \ldots, c_n$, where $c_n = 1$, the vector of all ones. One then chooses k initial vectors $c_1, \ldots, c_k$, and computes

$$W_{r,k,\mathbf{p}}(x) = \sum_{\substack{v_{i_1},\ldots,v_{i_\ell} \in V_{r,k} \\ i_1 < i_2 < \cdots < i_\ell}} w(c_1, \ldots, c_k, \overbrace{v_{i_1}, \ldots, v_{1}}^{p_1}, \ldots, \overbrace{v_{i_\ell}, \ldots, v_{i_\ell}}^{p_\ell})$$

for each $\mathbf{p} = (p_1, \ldots, p_\ell)$ such that $p_1 + \cdots + p_\ell = r$, and where $V_{r,k} = \{v_i : \ 0 \le i < |V|/2^{k+1}\}$ is the space generated by $c_{k+1}, \ldots, c_{n-1}$.

Letting $T_\pi f(x)$ be the operator having the property that

$$T_\pi w(v_1, v_2, \ldots, v_r) = w(v_{\pi(1)}, \ldots, v_{\pi(r)})$$

when $\pi \in S_r$, one computes

$$W_{r,*}(x) = \sum_{\substack{\mathbf{p}=(p_1,\ldots,p_\ell) \\ p_1+\cdots+p_\ell = r}} \frac{1}{p_1! \ldots p_\ell!} \sum_{\pi \in S_r} T_\pi W_{r,k,\mathbf{p}}(x)\,,$$

(but instead of dividing by the factorial factor, one chooses for each $\mathbf{p}$ only the elements of S_r that give rise to distinct permutations of the variables).

One then defines an operator $T_{i,j}$ for each $1 \le i \le k$, $1 \le j \le r$, by

$$T_{i,j} w(c_1, \ldots, c_k, v_1, \ldots, v_r) = w(c_1, \ldots, c_k, v_1, \ldots, c_i + v_j, \ldots, v_r)\,.$$

It is clear that this operator can be extended to $J \subset \{1, \ldots, r\}$ by

$$T_{i,J} = \sum_{j \in J} T_{i,j}\,.$$

One also defines an operator T_j by

$$T_j w(v_1, \ldots, v_r) = w(v_1, \ldots, 1 - v_j, \ldots, v_r)$$

which is similarly extended to subsets of $\{1, \ldots, r\}$. Finally, one defines a projection operator

$$T w(c_1, \ldots, c_k, v_1, \ldots, v_r) = w(v_1, \ldots, v_r)\,.$$

It follows that

$$W_r(V) = T \left(\sum_{I \subset \{1,\ldots,r\}} T_I \left(\sum_{i=1}^{k} \sum_{J \subset \{1,\ldots,r\}} T_{i,J}\, W_{r,*}(x) \right)\right)\,.$$

The point is that all the functions on $W_{r,*}$ act as substitutions of variables and are thus easy to compute. The time consuming part of the computation is evaluating the sums $W_{r,k,\mathbf{p}}(x)$ which takes $2^{(n-k-1)r}/r!$ steps to compute. For small r, k, this method saves a factor $2^{(k+1)\,r}\, r!$ steps over direct evaluation.

Computation of extremal weight enumerators

Computations for biweigth enumerators are simplified due to the fact that the polynomials are symmetric (see above). For example, if u, v are two partitions of length n, one has the identity

$$\mathrm{Sym}\,(x^u)\,\mathrm{Sym}\,(x^v) = \frac{n!}{\mathrm{Stab}\,(u)}\, x^u\, \mathrm{Sym}\,(x^v)\,.$$

Computations of the biweight enumerators of extremal codes of length 40 was done along the lines of [16] and [19]. Thus, all 285 codewords of weight 8 were computed from the generating matrix and the number of pairs intersecting at four elements was then computed directly.

Generating the codewords of weight 8 can be done efficiently by using the fact that the code is self–dual and the code orthogonal to the one generated by (I, A) is generated by (A^t, I). So to compute all minimum weight codewords, one looks at all r–fold linear combinations A having weight $8-r$, for $1 \le r \le d/2$, and then one computes all r–fold linear combinations of A^t having weight $8-r$, for $1 \le r < d/2$. This method proves Theorem ??.

The Golay code

For the Golay code computation of the weight enumerators can be simplified due to its large symmetry group M_{24}. For example, it is known [6] that M_{24} is transitive on codewords with 0, 8, 12, 16, or 24 ones of which there are 1, 759, 2576, 759, and 1, respectively. So if c_i, $i = 0, 8, 12, 16, 24$, are codewords with these numbers of 1's, then

$$W_2(g_{24}, x) = \sum_{v \in g_{24}} \left(w(c_0, v) + 759\, w(c_8, v) + 2576\, w(c_{12}, v) + 759\, w(c_{16}, v) + w(c_{24}, v) \right).$$

This reduces the computation from order $|g_{24}|^2$ steps to about $|g_{24}|$ steps.

Computation of $W_3(g_{24}, x)$ and $W_4(g_{24}, x)$ uses the fact that M_{24} is also transitive on *intersections* of codewords. In fact, writing $|v|$ for the number of 1's in a codeword v, one has

Proposition 3.3.1. *Let $c, c' \in g_{24}$ be codewords with $|c \cap c'| = k$, then*

$$\{\alpha(c') :\ \alpha \in \mathrm{Stab}(c)\} = \{d \in g_{24} :\ |d| = |c'| \text{ and } |d \cap c| = k\}\,.$$

In other words, the stabilizer of c (as a set) takes is transitive on codewords a fixed Hamming distance from c.

Remark: Proposition 3.3.1 says that the codewords of the Golay code of a given weight form a *two point homogeneous space* with respect to M_{24} and Hamming distance, see [6, page 250].

Proof: One first characterizes all the intersection types that occur. The list can be derived from the expression for $W_2(g_{24}, x)$ and as above c_i will indicate an arbitrary codeword with $|c_i| = i$, where c_8's are also called *octads* and c_{12}'s are called dodecads.

(1) $|c_0 \cap c_i'| = i$,

(2) $|c_8 \cap c_8'| = 0, 2, 4, 8$,

(3) $|c_8 \cap c_{12}'| = 2, 4, 6$,

(4) $|c_8 \cap c_{16}'| = 0, 4, 6, 8$,

(5) $|c_{12} \cap c_{12}'| = 0, 4, 6, 8, 12$,

(6) $|c_{12} \cap c_{16}'| = 6, 8, 10$,

(7) $|c_{16} \cap c_{16}'| = 8, 10, 12, 16$,

(8) $|c_{24} \cap c_i'| = 24 - i$.

The proof uses theorems about the action of M_{24} on g_{24} as in [6, Chapter 10]. One says that a group is $a+b$ transitive on sets A, B if it can take any a elements of A to any other A elements of A while taking any b elements of B to any b elements of B.

Theorem 1. *M_{24} is transitive on intersections $|c_8 \cap c_8'| = 2$. The stabilizer of an octad is $6+1$, $3+2$, and $1+3$ transitive on the octad and its complement. The stabilizer of a dodecad is $5+0$, $3+1$, $1+3$, and $0+5$ transitive on the dodecad and its complement. Finally, for a given octad c_8, there is a point $i \notin c_8$ such that the stabilizer of c_8 is transitive on the octads disjoint from c_8 not containing i, and transitive on the octads disjoint from c_8 that contain i.*

These are Theorems 14, 15, and 18 of [6, Chapter 10]. Since dodecads are exactly the sum of two octads which intersect at two elements, this implies that M_{24} is transitive on dodecads. Since M_{24} is 5–transitive and an octad is determined by five elements, it follows that the group is also transitive on octads.

Continuation of proof: Since all c_{16}'s are $c_{16} = 1 - c_8'$ as c_8 runs through all octads, the proof of transitivity for these will follow from transitivity of any octad appearing as an intersection with another octad or dodecad.

(1) $|c_0 \cap c_i| = i$: Transitivity in this case follows from the fact just noted that M_{24} is transitive on codewords of any given weight.

(2a) $|c_8 \cap c_8'| = 0$: As in the last part of Theorem 1, the stabilizer of a c_8 is transitive on octads c_8' which contain or do not contain a certain $i \notin c_8$. Now the stabilizer of c_8 is transitive on the complement of c_8, so there is a map π stabilizing c_8 and $\pi(j) = i$, where $j \in c_8'$for a c_8' satisfying $|c_8 \cap c_8'| = 0$. The map π therefore takes the octad c_8' not containg i to an octad contaning i. This shows that the stabilizer is transitive on all octads disjoint from c_8.

(2b) $|c_8 \cap c_8'| = 2$: This is given in Theorem 1.

(2c) $|c_8 \cap c_8'| = 4$: Assume that $|c_8 \cap c_8''| = 4$, then by Theorem 1, there is a $\pi \in M_{24}$ such that π takes 3 elements of $c_8 \cap c_8'$ to 3 elements of $c_8 \cap c_8''$, and takes 2 elements of $c_8' \backslash c_8$ to 2 elements of $c_8'' \backslash c_8$ (here $A \backslash B = A \cap B^c$ is the part of A disjoint from B). It follows that $|\pi(c_8') \cap c_8''| \geq 5$, and so $\pi(c_8') = c_8''$.

(3c) $|c_{12} \cap c_8| = 2$: Let c_8' be such that $|c_{12} \cap c_8'| = 2$. Then Theorem 1 gives a $\pi \in M_{24}$ which takes 5 elements of $c_8 \backslash c_{12}$ to 5 elements of $c_8' \backslash c_{12}$. It follows that $\pi(c_8) = c_8'$.

(3b) $|c_{12} \cap c_8| = 4$: Let c_8' be such that $|c_{12} \cap c_8'| = 4$. Theorem 1 gives a map $\pi \in M_{24}$ taking the 4 elements of $c_8 \cap c_{12}$ to the 4 elements of $c_8' \cap c_{12}$. In order that $\pi(c_8) \neq c_8'$ there must be exactly four elements in the intersection. This would say that $|(c_{12} \cap \pi(c_8)) \cap (c_{12} \cap c_8')| = 0$, and so $|c_{12} + \pi(c_8) + c_8' \pmod 2| = 4$ which is impossible. This would also imply that

$$w(c_{12}, \pi(c_8), c_8') = x^{(4,4,4,0,8,0,4)}$$

but no such term appears in $W_3(g_{24}, x)$. In any case, one has that $\pi(c_8) = c_8'$.

(3c) $|c_{12} \cap c_8| = 6$: This is proved as in (3a).

(4a) $|c_{12} \cap c_{12}'| = 0$: There is only one such c_{12}'.

(4b) $|c_{12} \cap c_{12}'| = 8$: Let c_{12}'' be such that $|c_{12} \cap c_{12}''| = 8$. By Theorem 1 choose $\pi \in M_{24}$ taking the 4 elements in $c_{12}' \backslash c_{12}$ to the 4 elements in $c_{12}'' \backslash c_{12}$. Now $c_{12} \cap \pi(c_{12}') \cap c_{12}''$ has at least 4 elements, and must have exactly 4 elements if $\pi(c_{12}) \neq c_{12}''$. This cannot happen as then $c_{12} + \pi(c_{12}') + c_{12}'' \pmod 2$ would have 4 elements. Alternatively, one would have

$$w(c_{12}, \pi(c_{12}'), c_{12}) = x^{(8,0,0,4,0,4,4,4)}$$

which does not occur.

(4c) $|c_{12} \cap c_{12}'| = 4$: Follows from (4b) by letting $c_{12} \mapsto 1 - c_{12}$.

(4d) $|c_{12} \cap c_{12}'| = 6$: There are $408034/2576 = 1584$ dodecads intersecting a given dodecad at six places. It must be shown that the stabilizer of the dodecad is transitive on these. We will use the notation of [6, Chapter 10].

This can be proved directly using results of the next section, where the complete stabilizer of a dodecad is constructed explicitly, and so one merely checks that the orbit of c_{12}' under the stabilizer of c_{12} contains all 1584 elements. Moreover, this computation shows that the stabilizer of two dodecads intersecting at six places is A_5. This computation is easily done since the stabilizer of a dodecad has 95040 elements, a number which is easily accessible by computer.

First, consider a fixed dodecad u. Then the set of octads that intersect u at six places form a Steiner system $(12, 6, 5)$, since any 5 points of u are contained in a unique octad which must intersect u at exactly 6 places. It follows that the stabilizer of u as a set is M_{12}, the automorphism group of this Steiner system. In particular, there are $2576 = 244823040/95040$ dodecads, where $|M_{24}| = 244823040$, $|M_{12}| = 95040$. The $132 = \binom{12}{5} \big/ \binom{6}{5}$ blocks of 6 elements are called *special hexads*, and the $792 = \binom{12}{6} - 132$ others *ordinary hexads*.

Our problem is really about M_{24} so to proceed with the proof, one must first understand how M_{12} embeds in M_{24}.

An explicit embedding of M_{12} in M_{24}

As in [6, Chapter 10] consider a prime $p \equiv 3 \pmod 4$ and the sets

$$\Omega_p = \{\infty, 0, 1, 2, 3, \dots, p-1\}$$

$$Q_p = \{x^2 : x \in \mathbf{F}_p\}$$

$$N_p = \Omega_p \backslash Q\,,$$

where ∞ is a symbol satisfying $n + \infty = \infty$, $n \cdot \infty = \infty$, $1/\infty = 0$, and $1/0 = \infty$. So Ω_{p+1} is the finite field with p elements adjoined ∞, and is called $PL(p)$, the projective line of order p. One now considers the permutations of Ω_p given by

$$x\alpha_p = x + 1, \quad x\beta_p = kx, \quad x\gamma_p = -1/x\,,$$

where $k \in Q$ and $k \neq 0$. If $p = 11$, then $k = 3$ and if $p = 23$, then $k = 2$. (Note that all group actions are on the right.)

One further defines

$$\delta_{23}(x) = \begin{cases} x^3/9, & x \in Q, \\ 9x^3, & x \in N. \end{cases}$$

It is then shown in [6, Chapter 10] that the maps $\alpha_{23}, \beta_{23}, \gamma_{23}, \delta_{23}$ generate the group M_{24}.

As above M_{12} can be considered as a subgroup of M_{24}. Let $n = N_{23}, q = N_{23}$, then n, q are invariant dodecads under β_{23}, δ_{23} so one has the inclusions

$$\iota_n : \Omega_{11} \to \Omega_{23}, \qquad \iota_q : \Omega_{11} \to \Omega_{23}$$

given by

$$\iota_n\{\infty, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} = \{\infty, 15, 7, 14, 5, 10, 20, 17, 11, 22, 21, 19\}\,,$$

$$\iota_q\{\infty, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} = \{0, 3, 6, 12, 1, 2, 4, 8, 16, 9, 18, 13\}\,.$$

The special hexads on Ω_{11} are then given by ι_n^{-1} of intersections of octads with n. In other words, a permutation belongs to our copy of M_{12} exactly if it leave these special hexads invariant. It is seen that $\alpha_{11}, \beta_{11}, \gamma_{11}$ satisfy this, but they do not generate M_{12}.[3] In [7, p. 426] various generators of M_{12} are given and it turns out that

$$\pi_{11} = (\infty, 5, 3)\,(0, 7, 8)\,(1, 9, 6)\,(2, 10, 4)\,,$$

(denoted as π_2 in [7]) stabilizes our hexads. It follows from [7] that β_{11}, π_{11} generate our copy of M_{12} (β_{11} corresponds to the a of [7]).

Furthermore, let $\theta : M_{12} \to M_{12}$, written as g^θ, be the canonical outer automorphism of M_{12} (M_{12} is known to have automorphism group $2M_{12}$). This automorphism is essentially equivalent to knowing an embedding of $M_{12} \to M_{24}$. To see this, let $\tau \mapsto \tau^*$ be a map $M_{12} \to M_{24}$ such that τ^* stabilizes our dodecads n, q. This map is then given by

$$\tau^* = \iota_n^{-1} \tau \iota_n \, \iota_q \tau^\theta \iota_q\,.$$

Let M_{12}^* be the image of this map. Then θ in M_{12}^* is just conjugation by γ_{24}. Moreover, the embedding of M_{12} allows one to compute g^θ quite easily for $g \in M_{12}$.

One computes g^θ as follows: One first stores once and for all the set S of 132 octads that intersect n in 6 places. Now given $g \in M_{12}$ computing g^θ is equivalent to knowing how g^* acts on q, the complement of n. So let $j = i\iota_q \notin n$ for some $i \in \Omega_{12}$. One considers the set S' of $s \in S$ which contain j. This set is non empty since M_{12}^* is transitive on the complement of n, in fact S' contains 22 octads. Next, for each $s \in S'$, let $t = s \cap n$ be the hexad where s intersects n and T be the set of corresponding hexads $t' = t\iota_n^{-1}$ in Ω_{11}. These hexad are permuted to special hexads by g and so correspond to a set of octads S'' in Ω_{23}. The intersection of all the octads in S'' (in particular the intersection of the 2–sets consisting of the part of each octad not in n) consists of exactly one element which is clearly $k = i^\theta \iota_q$. The image of i under g^θ is thus $k\iota_q^{-1}$.

In this way, one gets that

$$\beta_{11}^\theta = \beta_{11}\,,$$

$$\pi_{11}^\theta = (\infty, 10, 8)\,(11, 9, 3)\,(5, 6, 2)\,(7, 4, 1)\,,$$

and the permutations

$$\beta_{11}^* = \delta_{23} = (\infty)\,(15)\,(14, 17, 11, 19, 22)\,(20, 10, 7, 5, 21)\,(0)\,(3)\,(8, 16, 13, 9, 12)\,(18, 4, 2, 6, 1)\,,$$

$$\pi_{11}^* = (\infty, 20, 5)\,(14, 19, 10)\,(21, 17, 7)\,(22, 15, 11)\,(0, 13, 9)\,(3, 18, 1)\,(6, 16, 2)\,(8, 12, 4)\,,$$

[3] In [6, Chapter 10] the permutation $\delta_{11} = (\infty)\,(0)\,(1)\,(8)\,(2, 10)\,(3, 4)\,(5, 9)\,(6, 7)$ is given as a generator of M_{12} but it turns out not to be consistent with the embedding of M_{12} into M_{24} considered here. In particular, it does not preserve the special hexads generated by octads intersecting n.

generate M_{12}^*. It is easy to check by computer that these generate M_{12}^* since one merely verifies that they generate more than 7920 distinct elements, since this number is known to be the order of the largest maximal subgroup of M_{12}, e.g., see [1, p. 33].

Note as well that $\langle \pi_{11}, \pi_{11}^\theta \rangle$ (the subgroup generated by these two elements) is A_4 since

$$(\pi_{11})^3 = (\pi_{11}^\theta)^3 = (\pi_{11}\,\pi_{11}^\theta)^2 = 1\,.$$

Transitivity for $|c_{12} \cap c_{12}'| = 6$

The transitivity that we seek is for dodecads intersecting our dodecad n at six places. Since M_{24} is transitive on dodecads, we can fix our dodecad to be n as above. Since M_{12}^* is five transitive on n, one can fix a 5–tuple of points on n, i.e., consider only the dodecads that intersect n at a set of 5 points. There are 12 such dodecads, 2 for each for each hexad arising from the intersection of an dodecad with our dodecad n. It is clear that these hexads cannot be special, so a first step in the proof is that M_{12} is transitive on ordinary hexads.

We choose the 5 element set $\{2,6,7,8,10\}$ and consider the 6 ordinary hexads containing it. Now let β_{11} be as above, then its cycle structure is

$$\beta_{11} = (\infty)\,(0)\,(1,3,9,5,4)\,(2,6,7,10,8)\,.$$

Note that the only special hexad containing this set is $\{0,2,6,7,8,10\}$, so replacing 0 with an element of $\{1,3,4,5,9\}$ yields an ordinary hexad, and these are permuted by β_{11}. The element

$$\gamma_{11}\pi_{11}\beta_{11}\alpha_{11} = (2,10,6,8,7,12,11,5,9,4,1)\,(3)$$

takes $\{1,2,6,7,8,10\}$ to $\{\infty,2,6,7,8,10\}$, so this shows transitivity on ordinary hexads.

Finally, we show that M_{12}^* is transitive on the two dodecads

$$\{\infty,3,8,9,11,12,13,14,16,17,19,22\}, \quad \{\infty,1,2,4,6,11,14,17,18,19,22,23\}\,.$$

intersecting n at the hexad

$$\{\infty,2,6,7,8,10\}\iota_n = \{\infty,11,14,17,19,22\}\,.$$

A permutation achieving this is

$$\begin{aligned}&(\alpha_{11}^*)^2\,\beta_{11}^*\gamma_{11}^*\alpha_{11}^*\beta_{11}^*\pi_{11}^*\\ &\quad = (\infty)\,(5)\,(15),(19)\,(0,3)\ \ (6,9),(8,2,16,1)\,(10,21,20,7)\,(12,18,13,4)\,(17,22,14,11)\,,\end{aligned}$$

and the result is proved.

Reducing the intersection types

In terms of computing weight enumerators $w(c,c')$ the list of intersection types can be reduced using our operations

$$T_{21}w(c,c') = w(c',c),\ T_{12_*} = w(c,1-c'),\ T_{12_s}w(c,c') = w(c,c+c'),\ T_{02}w(c,c') = w(0,c')\,.$$

Proposition 2. *The weight enumerator of any intersection type is an image of the following five under the T's.*

$$|c_8 \cap c_8'| = 0, \tag{1}$$

$$|c_8 \cap c_8'| = 2, \tag{2}$$

$$|c_8 \cap c_8'| = 4, \tag{3}$$

$$|c_8 \cap c_{12}'| = 4, \tag{4}$$

$$|c_{12} \cap c_{12}'| = 6. \tag{5}$$

Proof: This is shown by explicit calculation. Let $W_{i,j,k}$ denote a $w(c_i,c_j)$ where $|c_i \cap c_j| = k$, so that $W_{8,8,0}, W_{8,8,2}, W_{8,8,4}, W_{8,12,4}$, and $W_{12,12,6}$ are assumed to be known. Note that $T_{12_*}W_{i,j,k} = W_{i,24-j,i-k}$.

$$\begin{gathered} W_{0,8,0} = T_{0,2}W_{8,8,0}, \quad W_{0,12,0} = T_{0,2}W_{12,12,6}, \quad W_{0,16,0} = T_{12_*}W_{0,8,0}, \\ W_{0,0,0} = T_{0,2}T_{21}W_{0,8,0}, \quad W_{0,24,0} = T_{12_*}W_{0,0,0}\,. \end{gathered} \tag{1}$$

$$W_{8,8,8} = T_{12_s}T_{21}W_{0,8,0}\,. \tag{2}$$

$$W_{8,12,2} = T_{12_*}T_{12_s}W_{8,8,2}, \quad W_{8,12,6} = T_{12_s}W_{8,8,2}\,. \tag{3}$$

$$\begin{gathered} W_{8,16,0} = T_{12_*}T_{12_s}T_{21}W_{0,8,0} \quad W_{8,16,4} = T_{12_*}W_{8,8,4}, \\ W_{8,16,6} = T_{12_*}W_{8,8,2}, \quad W_{8,16,8} = T_{12_s}W_{8,8,0}\,. \end{gathered} \tag{4}$$

$$\begin{gathered} W_{12,12,12} = T_{12_s}T_{21}W_{0,12,0}, \quad W_{12,12,4} = T_{12_*}T_{12_s}T_{21}W_{8,12,4}, \\ W_{12,12,8} = T_{12_*}W_{12,12,4}, \quad W_{12,12,0} = T_{12_*}W_{12,12,12}\,. \end{gathered} \tag{5}$$

$$W_{12,16,i} = T_{12_*} W_{12,8,12-i} \,. \tag{6}$$

$$W_{16,16,i} = T_{12_*} T_{21} T_{12_*} W_{8,8,i-8} \tag{7}$$

$$W_{24,i,i} = T_{21} T_{12_*} T_{21} W_{0,i,0} \,. \tag{8}$$

The above results show that $W_r(g_{24}, x)$ can be computed by evaluating

$$\sum_{v_2,\ldots,v_r \in g_{24}} w(c_i, c_j, v_2, \ldots, v_r)$$

where (c_i, c_j) are as in Proposition 2, where the evaluation of the sum can be speeded up as in the last section, and then performing various substitutions as in the proof of Proposition 2.

This shows that the running time of $W_r(g_{24}, x)$ is on the order of the running time of a direct evaluation of $W_{r-2}(g_{24}, x)$. Note that the proof of Proposition 1 required knowing $W_3(g_{24}, x)$, so this was computed only using the fact that M_{24} is transitive on codewords of a given weight.

References

[1] J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker, R.A. Wilson, *ATLAS of Finite Groups*, Oxford University Press, New York 1985.

[2] E.F. Assmus and H.F. Mattson, *new 5–designs*, J. Comb. Theory **6** (1969), 122–135.

[3] E.R. Berlekamp, F.J. MacWilliams, and N.J.A. Sloane, *Gleason's theorem on self–dual codes*, IEEE Trans. on Inf. Theory **18** (1972), 409–414.

[4] M. Broué and M. Enguehard, *Polynomes de poids de certains codes et fonction thêta de certain réseaux*, Ann. Sc. École Norm. Sup. **6** (1973), 17–52.

[5] J.H. Conway, V. Pless, and N.J.A. Sloane, *The binary self–dual codes of length up to 32, a revised enumeration*, J. Comb. Th. A **60** (1992), 183–195.

[6] J.H. Conway and N.J. Sloane, *Sphere Packings, Lattices and Groups*, Springer Verlag, New York 1988.

[7] R.T. Curtis, *Natural constructions of the Mathieu groups*, Math. Proc. Camb. Phil. Soc. **106** (1989), 423–429.

[8] W. Duke, *On Codes and Siegel modular forms*, to appear, Duke Math. Journal.

[9] E. Freitag, *Siegelsche Modulfunktionen*, Springer Verlag 1983.

[10] A.M. Gleason, *Weight polynomials of self–dual codes and the MacWilliams identities,* in Actes Congrès Inter. de Math. **3** (1970), Gauthier–Villars, Paris 1971, 211–215.

[11] W.C. Huffman, *The biweight enumerator of self–dual orthogonal binary codes,* Discrete Math. **26** (1979), 129–143.

[12] D.P. Maher, *Modular forms from codes,* Canadian J. Math. **32** (1980), 40–58.

[13] H. Maschke, *Über die quaternäre, endliche, lineare Substitutionsgruppe der Borchadt'schen Moduln,* Math. Ann. **80** (1887), 496–515.

[14] F.J. MacWilliams, C.L. Mallows, and N.J.A. Sloane, *Generalizations of Gleason's theorem on weight enumerators of self–dual codes,* IEEE Trans. Inform. Theory **18** (1972), 794–805.

[15] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes,* North Holland, Amsterdam, 2nd printing, 1978.

[16] M. Ozeki, *Hadamard Matrices and Doubly Even Self–Dual Error–Correcting Codes,* J. Comb. Th. A **44** (1987), 274–287.

[17] V. Pless and N.J. Sloane, *On the Classification and Enumeration of Self–Dual Codes,* J. Comb. Theory (A) **18** (1975), 313–335.

[18] G.C. Shepard and J.A. Todd, *Finite unitary reflection groups,* Canadian J. of Math. **6** (1954), 274–304.

[19] V. D. Tonchev, *Self–Orthogonal Designs and Extremal Doubly even Codes,* J. Comb. Th. A **52** (1989), 197–205.

[20] I. Vardi, *Computational recreations in Mathematica,* Addison–Wesley, Reading, MA, 1991.

Explicit biweight generators

$$W_2(x;d_8^+) = \mathrm{Sym}\,(x^{(0,0,0,8)} + 14\,x^{(0,0,4,4)} + 168\,x^{(2,2,2,2)})\,,$$

$$\begin{aligned}W_2(x;g_{24}) = \mathrm{Sym}\,(&x^{(0,0,0,24)} + 759\,x^{(0,0,8,16)} + 2576\,x^{(0,0,12,12)} + 22770\,x^{(0,8,8,8)}\\ &+ 340032\,x^{(2,6,6,10)} + 212520\,x^{(4,4,4,12)} + 1275120\,x^{(4,4,8,8)} + 4080384\,x^{(6,6,6,6)})\,,\end{aligned}$$

$$\begin{aligned}W_2(x;d_{24}^+) = \mathrm{Sym}\,(&x^{(0,0,0,24)} + 66\,x^{(0,0,4,20)} + 495\,x^{(0,0,8,16)} + 2972\,x^{(0,0,12,12)}\\ &+ 2970\,x^{(0,4,4,16)} + 13860\,x^{(0,4,8,12)} + 34650\,x^{(0,8,8,8)} + 1320\,x^{(2,2,2,18)}\\ &+ 15840\,x^{(2,2,6,14)} + 168432\,x^{(2,2,10,10)} + 110880\,x^{(2,6,6,10)} + 83160\,x^{(4,4,4,12)}\\ &+ 1221660\,x^{(4,4,8,8)} + 6046656\,x^{(6,6,6,6)})\,,\end{aligned}$$

$$\begin{aligned}W_2(x;d_{32}^+) = \mathrm{Sym}\,(&x^{(0,0,0,32)} + 120\,x^{(0,0,4,28)} + 1820\,x^{(0,0,8,24)} + 8008\,x^{(0,0,12,20)}\\ &+ 45638\,x^{(0,0,16,16)} + 10920\,x^{(0,4,4,24)} + 120120\,x^{(0,4,8,20)} + 360360\,x^{(0,4,12,16)}\\ &+ 900900\,x^{(0,8,8,16)} + 1681680\,x^{(0,8,12,12)} + 3360\,x^{(2,2,2,26)} + 87360\,x^{(2,2,6,22)}\\ &+ 480480\,x^{(2,2,10,18)} + 4755840\,x^{(2,2,14,14)} + 1601600\,x^{(2,6,6,18)} + 5765760\,x^{(2,6,10,14)}\\ &+ 12108096\,x^{(2,10,10,10)} + 720720\,x^{(4,4,4,20)} + 5405400\,x^{(4,4,8,16)} + 69727840\,x^{(4,4,12,12)}\\ &+ 25225200\,x^{(4,8,8,12)} + 19219200\,x^{(6,6,6,14)} + 302766464\,x^{(6,6,10,10)} + 1328235480\,x^{(8,8,8,8)})\,,\end{aligned}$$

$$\begin{aligned}W_2(x;d_{40}^+) = \mathrm{Sym}\,(&x^{(0,0,0,40)} + 190\,x^{(0,0,4,36)} + 4845\,x^{(0,0,8,32)} + 38760\,x^{(0,0,12,28)}\\ &+ 125970\,x^{(0,0,16,24)} + 709044\,x^{(0,0,20,20)} + 29070\,x^{(0,4,4,32)} + 581400\,x^{(0,4,8,28)}\\ &+ 3527160\,x^{(0,4,12,24)} + 8314020\,x^{(0,4,16,20)} + 8817900\,x^{(0,8,8,24)} + 38798760\,x^{(0,8,12,20)}\\ &+ 62355150\,x^{(0,8,16,16)} + 116396280\,x^{(0,12,12,16)} + 6840\,x^{(2,2,2,34)} + 310080\,x^{(2,2,6,30)}\\ &+ 3255840\,x^{(2,2,10,26)} + 12093120\,x^{(2,2,14,22)} + 118090320\,x^{(2,2,18,18)} + 10852800\,x^{(2,6,6,26)}\\ &+ 84651840\,x^{(2,6,10,22)} + 221707200\,x^{(2,6,14,18)} + 465585120\,x^{(2,10,10,18)} + 798145920\,x^{(2,10,14,14)}\\ &+ 3488400\,x^{(4,4,4,28)} + 52907400\,x^{(4,4,8,24)} + 232792560\,x^{(4,4,12,20)} + 2914306260\,x^{(4,4,16,16)}\\ &+ 581981400\,x^{(4,8,8,20)} + 1745944200\,x^{(4,8,12,16)} + 3259095840\,x^{(4,12,12,12)} + 282172800\,x^{(6,6,6,22)}\\ &+ 1551950400\,x^{(6,6,10,18)} + 22981889280\,x^{(6,6,14,14)} + 5587021440\,x^{(6,10,10,14)} + 4364860500\,x^{(8,8,8,16)}\\ &+ 74192298960\,x^{(8,8,12,12)} + 302328806208\,x^{(10,10,10,10)})\,.\end{aligned}$$

Extremal weight enumerators

$$A_{III} = \begin{pmatrix}
0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1\\
1&0&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1&1&1\\
1&0&0&0&1&0&1&1&1&1&0&1&1&1&1&0&0&0&0&1\\
1&0&0&1&1&1&0&0&1&1&1&0&0&1&1&0&0&1&1&0\\
1&0&1&0&1&1&0&0&1&1&1&1&1&0&0&1&1&0&0&0\\
1&0&1&1&0&0&1&1&0&1&1&0&1&0&1&0&1&0&1&0\\
1&0&1&1&0&0&1&1&1&0&1&1&0&1&0&1&0&1&0&0\\
1&0&1&1&1&1&0&1&0&0&0&0&1&1&0&1&0&0&1&1\\
1&0&1&1&1&1&1&0&0&0&0&1&0&0&1&0&1&1&0&1\\
1&1&0&0&1&1&1&1&0&0&1&0&0&1&1&1&1&0&0&0\\
1&1&0&1&0&1&0&1&0&1&0&1&1&0&1&1&0&1&0&0\\
1&1&0&1&0&1&1&0&1&0&0&1&1&1&0&0&1&0&1&0\\
1&1&0&1&1&0&0&1&1&0&1&0&1&0&0&0&1&1&0&1\\
1&1&0&1&1&0&1&0&0&1&1&1&0&0&0&1&0&0&1&1\\
1&1&1&0&0&1&0&1&1&0&1&1&0&0&1&0&0&0&1&1\\
1&1&1&0&0&1&1&0&0&1&1&0&1&1&0&0&0&1&0&1\\
1&1&1&0&1&0&0&1&0&1&0&1&0&1&0&0&1&1&1&0\\
1&1&1&0&1&0&1&0&1&0&0&0&1&0&1&1&0&1&1&0\\
1&1&1&1&0&0&0&0&1&1&0&0&0&1&1&1&1&0&0&1
\end{pmatrix} \qquad A_{IV} = \begin{pmatrix}
0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1\\
1&0&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1&1&1\\
1&0&0&0&1&0&1&1&1&1&0&1&1&1&1&0&0&0&0&1\\
1&0&0&1&1&1&0&0&1&1&1&0&0&1&1&0&0&1&1&0\\
1&0&1&0&1&1&0&0&1&1&1&1&1&0&0&1&1&0&0&0\\
1&0&1&1&0&0&1&1&0&1&1&0&1&0&1&0&1&0&1&0\\
1&0&1&1&0&0&1&1&1&0&1&1&0&1&0&1&0&1&0&0\\
1&0&1&1&1&1&0&1&0&0&0&0&1&1&0&1&0&0&1&1\\
1&0&1&1&1&1&1&0&0&0&0&1&0&0&1&0&1&1&0&1\\
1&1&0&0&1&1&1&1&0&0&1&0&1&1&0&0&1&1&0&0\\
1&1&0&1&0&1&0&1&0&1&0&1&0&1&1&1&1&0&0&0\\
1&1&0&1&0&1&1&0&1&0&1&0&1&0&1&1&0&0&0&1\\
1&1&0&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&1\\
1&1&0&1&1&0&1&0&0&1&0&1&1&0&0&1&0&1&1&0\\
1&1&1&0&0&1&0&1&1&0&0&1&1&0&1&0&0&1&1&0\\
1&1&1&0&0&1&1&0&0&1&1&1&0&1&0&0&0&0&1&1\\
1&1&1&0&1&0&0&1&0&1&1&0&0&0&1&1&0&1&0&1\\
1&1&1&0&1&0&1&0&1&0&0&0&0&1&1&1&1&0&1&0\\
1&1&1&1&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&1
\end{pmatrix},$$

$$A_{QR} = \begin{pmatrix}
0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&1&1&0&0&0&0&1&0&1&0&1&1&1&1&0&0&1&1\\
1&1&0&1&1&0&0&0&0&1&0&1&0&1&1&1&1&0&0&1\\
1&1&1&0&1&1&0&0&0&0&1&0&1&0&1&1&1&1&0&0\\
1&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1&1&1&1&0\\
1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1&1&1&1\\
1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1&1&1\\
1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1&1\\
1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1\\
1&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0\\
1&0&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1\\
1&1&0&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0\\
1&0&1&0&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&1\\
1&1&0&1&0&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0\\
1&0&1&0&1&0&1&1&1&1&0&0&1&1&0&1&1&0&0&0\\
1&0&0&1&0&1&0&1&1&1&1&0&0&1&1&0&1&1&0&0\\
1&0&0&0&1&0&1&0&1&1&1&1&0&0&1&1&0&1&1&0\\
1&0&0&0&0&1&0&1&0&1&1&1&1&0&0&1&1&0&1&1\\
1&1&0&0&0&0&1&0&1&0&1&1&1&1&0&0&1&1&0&1\\
1&1&1&0&0&0&0&1&0&1&0&1&1&1&1&0&0&1&1&0
\end{pmatrix}$$

$$
\begin{aligned}
W_2(x;e_{40,III}) =& \mathrm{Sym}\,(x^{(0,0,0,40)}+285\,x^{(0,0,8,32)}+21280\,x^{(0,0,12,28)}+239970\,x^{(0,0,16,24)}+525504\,x^{(0,0,20,20)} \\
&+25020\,x^{(0,8,8,24)}+479040\,x^{(0,8,12,20)}+1326030\,x^{(0,8,16,16)}+4947360\,x^{(0,12,12,16)}+48000\,x^{(2,6,6,26)} \\
&+3941760\,x^{(2,6,10,22)}+28696320\,x^{(2,6,14,18)}+140382720\,x^{(2,10,10,18)}+430444800\,x^{(2,10,14,14)}+7920\,x^{(4,4,4,28)} \\
&+1596000\,x^{(4,4,8,24)}+34402320\,x^{(4,4,12,20)}+91417920\,x^{(4,4,16,16)}+249458400\,x^{(4,8,8,20)}+2445547200\,x^{(4,8,12,16)} \\
&+9027162720\,x^{(4,12,12,12)}+56432640\,x^{(6,6,6,22)}+1972222080\,x^{(6,6,10,18)}+6009907200\,x^{(6,6,14,14)}+28759503360\,x^{(6,10,10,14)} \\
&+17360299620\,x^{(8,8,8,16)}+63637701120\,x^{(8,8,12,12)}+136675350528\,x^{(10,10,10,10)})\,,
\end{aligned}
$$

$$
\begin{aligned}
W_2(x;e_{40,IV}) =& \mathrm{Sym}\,(x^{(0,0,0,40)}+285\,x^{(0,0,8,32)}+21280\,x^{(0,0,12,28)}+239970\,x^{(0,0,16,24)}+525504\,x^{(0,0,20,20)} \\
&+22140\,x^{(0,8,8,24)}+490560\,x^{(0,8,12,20)}+1308750\,x^{(0,8,16,16)}+4935840\,x^{(0,12,12,16)}+53760\,x^{(2,6,6,26)} \\
&+3924480\,x^{(2,6,10,22)}+28707840\,x^{(2,6,14,18)}+140313600\,x^{(2,10,10,18)}+430617600\,x^{(2,10,14,14)}+5040\,x^{(4,4,4,28)} \\
&+1596000\,x^{(4,4,8,24)}+34428240\,x^{(4,4,12,20)}+91371840\,x^{(4,4,16,16)}+249631200\,x^{(4,8,8,20)}+2445374400\,x^{(4,8,12,16)} \\
&+9026166240\,x^{(4,12,12,12)}+56340480\,x^{(6,6,6,22)}+1972078080\,x^{(6,6,10,18)}+6010368000\,x^{(6,6,14,14)}+28760954880\,x^{(6,10,10,14)}+ \\
&+17359807140\,x^{(8,8,8,16)}+63635120640\,x^{(8,8,12,12)}+136679682048\,x^{(10,10,10,10)})\,.
\end{aligned}
$$

The following 3 tables gives weight enumerators of codes as in [19, Table II], where in (a,b,c) the value of a represents the numbering of [19, Table II], b is a list of the inverted columns, and c is the coefficient of $x_0^{28}x_1^4x_2^4x_3^4$ of the biweight enumerator.

Codes derived from A_{III}

(1,{2,3,4,5},7200),(2,{2,3,4,7},5760),(3,{2,3,6,7},6480),(4,{2,3,6,11},9360),(5,{2,3,6,12},6120),(6,{2,3,6,16},6360),

(7,{2,3,7,14},4680),(8,{2,3,7,17},5640),(9,{2,3,16,17},7200),(10,{2,3,16,18},5640),(11,{2,3,16,19},6480),(12,{2,3,4,5,6,7,8,9},7920),

(13,{2,3,4,5,6,7,12,16},5400),(14,{2,3,4,5,6,7,12,17},6480)

Codes derived from A_{IV}.

(15,{2,3,4,5},5760),(16,{2,3,4,9},6480),(17,{2,3,4,11},6480),(18,{2,3,4,12},5400),(19,{2,3,4,20},5760),(20,{2,3,5,11},5220),

(21,{2,3,5,12},4860),(22,{2,3,5,14},5280),(23,{2,3,5,17},4860),(24,{2,3,9,12},4920),(25,{2,3,11,17},5280),(26,{2,3,12,13},4920),

(27,{2,3,12,14},4740),(28,{2,3,12,15},4920),(29,{2,3,12,16},5040),(30,{2,3,12,17},5040),(31,{2,3,13,19},4860),

(32,{2,3,4,5,6,7,8,9},5040),(33,{2,3,4,5,6,7,12,16},4860),(34,{2,3,4,5,6,7,12,20},5040),

(35,{2,3,4,5,6,7,13,17},5220),(36,{2,3,4,5,6,11,12,16},6000),(37,{2,3,4,5,6,11,12,18},5760),

(38,{2,3,4,5,6,11,13,20},5100),(39,{2,3,4,5,6,12,13,18},5580),(40,{2,3,4,5,6,7,8,9,12,16,18,19},5220)

Codes derived from A_{QR}.

(41,{2,3,4,5},4860),(42,{2,3,4,6},4860),(43,{2,3,4,8},5040),(44,{2,3,4,9},4680),(45,{2,3,4,10},4680),(46,{2,3,4,11},4680),

(47,{2,3,4,12},5100),(48,{2,3,4,14},4860),(49,{2,3,4,16},5100),(50,{2,3,4,17},4860),(51,{2,3,4,18},5040),(52,{2,3,5,9},4500),

(53,{2,3,5,11},4860),(54,{2,3,5,14},4680),(55,{2,3,5,16},4860),(56,{2,3,5,17},4860),(57,{2,3,5,19},4560),(58,{2,3,6,7},5100),

(59,{2,3,6,15},4680),(60,{2,3,11,12},4860),(61,{2,3,4,5,6,7,8,9},4860),(62,{2,3,4,5,6,7,8,11},4920),

(63,{2,3,4,5,6,7,8,13},4920),(64,{2,3,4,5,6,7,8,14},4680),(65,{2,3,4,5,6,7,9,11},4500),

(66,{2,3,4,5,6,7,9,13},4680),(67,{2,3,4,5,6,7,9,19},4860),(68,{2,3,4,5,6,7,10,11},4500),

(69,{2,3,4,5,6,7,11,19},4920),(70,{2,3,4,5,6,7,12,18},4680),(71,{2,3,4,5,6,7,13,17},4680),

(72,{2,3,4,5,6,7,13,18},4500),(73,{2,3,4,5,6,7,8,9,10,11,12,15},5100),(74,{2,3,4,5,6,7,8,9,10,11,12,17},4680),

(75,{2,3,4,5,6,7,8,9,10,11,13,16},4740),(76,{2,3,4,5,6,7,8,9,10,11,14,17},4500),(77,{2,3,4,5,6,7,8,9,10,11,16,17},4380),

(78,{2,3,4,5,6,7,8,9,10,11,17,18},4380),(79,{2,3,4,5,6,7,8,9,10,12,13,19},4500),(80,{},3420)

$$
\begin{aligned}
W_2(x;e_{48}) = \mathrm{Sym}\,(& x^{(0,0,0,48)} + 17296\,x^{(0,0,12,36)} + 535095\,x^{(0,0,16,32)} + 3995376\,x^{(0,0,20,28)} \\
& + 7681680\,x^{(0,0,24,24)} + 10896480\,x^{(0,12,12,24)} + 59930640\,x^{(0,12,16,20)} + 153037170\,x^{(0,16,16,16)} \\
& + 143833536\,x^{(2,10,10,26)} + 2054764800\,x^{(2,10,14,22)} + 4954266240\,x^{(2,10,18,18)} + 14383353600\,x^{(2,14,14,18)} \\
& + 128422800\,x^{(4,8,8,28)} + 5094104400\,x^{(4,8,12,24)} + 29845458720\,x^{(4,8,16,20)} + 104279313600\,x^{(4,12,12,20)} \\
& + 279876088800\,x^{(4,12,16,16)} + 15981504\,x^{(6,6,6,30)} + 1917780480\,x^{(6,6,10,26)} + 29006429760\,x^{(6,6,14,22)} \\
& + 69040097280\,x^{(6,6,18,18)} + 131367962880\,x^{(6,10,10,22)} + 963684691200\,x^{(6,10,14,18)} + 2970162518400\,x^{(6,14,14,14)} \\
& + 34224676200\,x^{(8,8,8,24)} + 743439589200\,x^{(8,8,12,20)} + 1970391020400\,x^{(8,8,16,16)} + 7304226541920\,x^{(8,12,12,16)} \\
& + 4631359951680\,x^{(10,10,10,18)} + 14095686528000\,x^{(10,10,14,14)} + 26737956967680\,x^{(12,12,12,12)})\,,
\end{aligned}
$$

$$
\begin{aligned}
W_2(x;e_{72}) = \mathrm{Sym}\,(& x^{(0,0,0,72)}+249849\,x^{(0,0,16,56)}+18106704\,x^{(0,0,20,52)}+462962955\,x^{(0,0,24,48)}+4397342400\,x^{(0,0,28,44)} \\
&+16602715899\,x^{(0,0,32,40)}+25756721120\,x^{(0,0,36,36)}+1269732618\,x^{(0,16,16,40)}+22778233632\,x^{(0,16,20,36)}+126684686205\,x^{(0,16,24,32)} \\
&+222505525440\,x^{(0,16,28,28)}+265027826448\,x^{(0,20,20,32)}+898816786560\,x^{(0,20,24,28)}+1740663655920\,x^{(0,24,24,24)}+21107243520\,x^{(2,14,14,42)} \\
&+738753523200\,x^{(2,14,18,38)}+7481558407680\,x^{(2,14,22,34)}+23196860628480\,x^{(2,14,26,30)}+17034787123200\,x^{(2,18,18,34)} \\
&+108399766970880\,x^{(2,18,22,30)}+198576947036160\,x^{(2,18,26,26)}+407496443397120\,x^{(2,22,22,26)}+30851354520\,x^{(4,12,12,44)} \\
&+2197791731520\,x^{(4,12,16,40)}+41607214054560\,x^{(4,12,20,36)}+230509568076480\,x^{(4,12,24,32)}+404936970253200\,x^{(4,12,28,28)} \\
&+105512471951040\,x^{(4,16,16,36)}+1287259545337920\,x^{(4,16,20,32)}4354305609931200\,x^{(4,16,24,28)}+9589029965555040\,x^{(4,20,20,28)}+ \\
&+18464294745832320\,x^{(4,20,24,24)}+8618791104\,x^{(6,10,10,46)}+1348225179840\,x^{(6,10,14,42)}+49471860936960\,x^{(6,10,18,38)} \\
&+499028004921600\,x^{(6,10,22,34)}+1548124495682688\,x^{(6,10,26,30)}144056937024000\,x^{(6,14,14,38)}+3488024759788800\,x^{(6,14,18,34)} \\
&+22100107147937280\,x^{(6,14,22,30)}+40521332563367040\,x^{(6,14,26,26)}+53047567531580160\,x^{(6,18,18,30)}+198648827753967360\,x^{(6,18,22,26)} \\
&+405711319383659520\,x^{(6,22,22,22)}+577151190\,x^{(8,8,8,48)}+207774428400\,x^{(8,8,12,44)}+15604436724030\,x^{(8,8,16,40)} \\
&+294023902233600\,x^{(8,8,20,36)}+1630411711166700\,x^{(8,8,24,32)}+2863014214309920\,x^{(8,8,28,28)}+54653909088240\,x^{(8,12,12,40)} \\
&+2752651033957440\,x^{(8,12,16,36)}+33453622200398400\,x^{(8,12,20,32)}+113220025974871200\,x^{(8,12,24,28)}+89421974655025320\,x^{(8,16,16,32)} \\
&+663291753095442240\,x^{(8,16,20,28)}+1278377852247688500\,x^{(8,16,24,24)}+2801815487240837760\,x^{(8,20,20,24)}+6121804195584\,x^{(10,10,10,42)} \\
&+689497132040640\,x^{(10,10,14,38)}+16614960738647040\,x^{(10,10,18,34)}+105368235250658688\,x^{(10,10,22,30)} \\
&+193120424863382016\,x^{(10,10,26,26)}+50999406221998080\,x^{(10,14,14,34)}+772631922519959040\,x^{(10,14,18,30)} \\
&+2894800250820804480\,x^{(10,14,22,26)}+6915375298715304960\,x^{(10,18,18,26)}+14136542400036672000\,x^{(10,18,22,22)} \\
&+10162380189392760\,x^{(12,12,12,36)}+328559909158514880\,x^{(12,12,16,32)}+2439334397238021360\,x^{(12,12,20,28)} \\
&+4699539095722964640\,x^{(12,12,24,24)}+6489302054567048640\,x^{(12,16,16,28)}+27425750712290219520\,x^{(12,16,20,24)} \\
&+60163781995963112544\,x^{(12,20,20,20)}+2360284030535777280\,x^{(14,14,14,30)}+21144805020742233600\,x^{(14,14,18,26)} \\
&+43207625104317527040\,x^{(14,14,22,22)}+103312739390888966400\,x^{(14,18,18,22)}+73026669232692522270\,x^{(16,16,16,24)} \\
&+160135406236191998304\,x^{(16,16,20,20)}+246931575051320248320\,x^{(18,18,18,18)})\,.
\end{aligned}
$$

$$W_2(x;e_{96}) = \mathrm{Sym}\,(x^{(0,0,0,96)} + 3217056\,x^{(0,0,20,76)} + 369844880\,x^{(0,0,24,72)} + 18642839520\,x^{(0,0,28,68)} + 422069980215\,x^{(0,0,32,64)}$$

$$+4552866656416\,x^{(0,0,36,60)} + 24292689565680\,x^{(0,0,40,56)} + 65727011639520\,x^{(0,0,44,52)} + 91447669224080\,x^{(0,0,48,48)}$$

$$+104570405280\,x^{(0,20,20,56)} + 3435574528800\,x^{(0,20,24,52)} + 45556569163200\,x^{(0,20,28,48)} + 246470321988000\,x^{(0,20,32,44)}$$

$$+568004855035200\,x^{(0,20,36,40)} + 83346818959200\,x^{(0,24,24,48)} + 794404045065600\,x^{(0,24,28,44)} + 2997851667244560\,x^{(0,24,32,40)}$$

$$+4651857925222080\,x^{(0,24,36,36)} + 5284328657688960\,x^{(0,28,28,40)} + 13424126177766240\,x^{(0,28,32,36)} + 21983815972782450\,x^{(0,32,32,32)}$$

$$+2278640764800\,x^{(2,18,18,58)} + 125867316000000\,x^{(2,18,22,54)} + 2671275179053440\,x^{(2,18,26,50)} + 22440852081638400\,x^{(2,18,30,46)}$$

$$+78967680646752000\,x^{(2,18,34,42)} + 119741412119846400\,x^{(2,18,38,38)} + 5181497071176960\,x^{(2,22,22,50)} + 80076409528919040\,x^{(2,22,26,46)}$$

$$+476672933995361280\,x^{(2,22,30,42)} + 1150514305046668800\,x^{(2,22,34,38)} + 876867729678420480\,x^{(2,26,26,42)} + 3580240835070950400\,x^{(2,26,30,38)}$$

$$+5698903675033094400\,x^{(2,26,34,34)} + 9640391377643688960\,x^{(2,30,30,34)} + 5024542828320\,x^{(4,16,16,60)} + 481482729070560\,x^{(4,16,20,56)}$$

$$+16695199636380000\,x^{(4,16,24,52)} + 220679070232644000\,x^{(4,16,28,48)} + 1194154127956296000\,x^{(4,16,32,44)} + 2751990922968143040\,x^{(4,16,36,40)}$$

$$+34753182992055360\,x^{(4,20,20,52)} + 888264199119880800\,x^{(4,20,24,48)} + 8438724664991712000\,x^{(4,20,28,44)} + 31858367550603410880\,x^{(4,20,32,40)}$$

$$+49426785836903448960\,x^{(4,20,36,36)} + 16317687149881776000\,x^{(4,24,24,44)} + 108153665252316835200\,x^{(4,24,28,40)}$$

$$+274877876243464200000\,x^{(4,24,32,36)} + 482587536794478777600\,x^{(4,28,28,36)} + 790561958260747723200\,x^{(4,28,32,32)}$$

$$+2550803702400\,x^{(6,14,14,62)} + 442695215692800\,x^{(6,14,18,58)} + 25760664067334400\,x^{(6,14,22,54)}$$

$$+544759756196981760\,x^{(6,14,26,50)} + 4578181015194633600\,x^{(6,14,30,46)} + 16109163665666611200\,x^{(6,14,34,42)}$$

$$+24427472554755878400\,x^{(6,14,38,38)} + 58448134620243200\,x^{(6,18,18,54)} + 2534980057714863360\,x^{(6,18,22,50)}$$

$$+39038415514935459840\,x^{(6,18,26,46)} + 232467912661636807680\,x^{(6,18,30,42)} + 561063598255123084800\,x^{(6,18,34,38)}$$

$$+80065646774519051520\,x^{(6,22,22,46)} + 873667669793448545280\,x^{(6,22,26,42)} + 3568504178598806039040\,x^{(6,22,30,38)}$$

$$+5679757648257510182400\,x^{(6,22,34,34)} + 6539059877390193484800\,x^{(6,26,26,38)} + 17613665217337550054400\,x^{(6,26,30,34)}$$

$$+29808530176327588733184\,x^{(6,30,30,30)} + 373258922400\,x^{(8,12,12,64)} + 124416729364320\,x^{(8,12,16,60)}$$

$$+12557746761874560\,x^{(8,12,20,56)} + 433943664910934400\,x^{(8,12,24,52)} + 5737917137454324000\,x^{(8,12,28,48)}$$

$$+31047706070721439200\,x^{(8,12,32,44)} + 71551902119862551040\,x^{(8,12,36,40)} + 31729187903141520\,x^{(8,16,16,56)}$$

$$+2413140849526085280\,x^{(8,16,20,52)} + 61460714829064395600\,x^{(8,16,24,48)} + 584111712216884263200\,x^{(8,16,28,44)}$$

$$+2205019063006317649440\,x^{(8,16,32,40)} + 3421074244547483144640\,x^{(8,16,36,36)} + 1352698558286439873 60\,x^{(8,20,20,48)}$$

$$+24762464930333035558400\,x^{(8,20,24,44)} + 16418497070195805075840\,x^{(8,20,28,40)} + 41726146627488774548160\,x^{(8,20,32,36)}$$

$$+31624815527632216440000\,x^{(8,24,24,40)} + 141164663010158793916800\,x^{(8,24,28,36)} + 231233487795113812293600\,x^{(8,24,32,32)}$$

$$+406138206409326316958400\,x^{(8,28,28,32)} + 17629466880\,x^{(10,10,10,66)} + 11540866694400\,x^{(10,10,14,62)}$$

$+2117648987980800\,x^{(10,10,18,58)}+122747482005964800\,x^{(10,10,22,54)}+2596850937767886336\,x^{(10,10,26,50)}$

$+21822274581903072000\,x^{(10,10,30,46)}+76787656541530041600\,x^{(10,10,34,42)}+116436860319069388800\,x^{(10,10,38,38)}$

$+6144170967532800\,x^{(10,14,14,58)}+854362726669132800\,x^{(10,14,18,54)}+36927064913678300160\,x^{(10,14,22,50)}$

$+568871134950895338240\,x^{(10,14,26,46)}+3387357381386033498880\,x^{(10,14,30,42)}+8175513518391333427200\,x^{(10,14,34,38)}$

$+88585965878230369920\,x^{(10,18,18,50)}+2788055692317391319040\,x^{(10,18,22,46)}+30434385976192085045760\,x^{(10,18,26,42)}$

$+124300910713475507604480\,x^{(10,18,30,38)}+197846912191394918188800\,x^{(10,18,34,34)}+62177306312170801105920\,x^{(10,22,22,42)}$

$+465540671298029325696000\,x^{(10,22,26,38)}+1253915680217921329351680\,x^{(10,22,30,34)}+2298710559029677379112960\,x^{(10,26,26,34)}$

$+3889914262220217370083840\,x^{(10,26,30,30)}+434173652566080\,x^{(12,12,12,60)}+117071510832813600\,x^{(12,12,16,56)}$

$+8869295866098608640\,x^{(12,12,20,52)}+225990993274631340000\,x^{(12,12,24,48)}+2147604807767097470400\,x^{(12,12,28,44)}$

$+8107410578244675354720\,x^{(12,12,32,40)}+12578413493498305761280\,x^{(12,12,36,36)}+23694028828805190240\,x^{(12,16,16,52)}$

$+1323601928293600902240\,x^{(12,16,20,48)}+24238293930901651960800\,x^{(12,16,24,44)}+160700737156118207425440\,x^{(12,16,28,40)}$

$+408411228356328029520960\,x^{(12,16,32,36)}+53139636413472003413760\,x^{(12,20,20,44)}+678915683888730619968000\,x^{(12,20,24,40)}$

$+3030286393060516843349760\,x^{(12,20,28,36)}+4963862949040589865929280\,x^{(12,20,32,32)}+5839358223475199824291200\,x^{(12,24,24,36)}$

$+16799225313237413890132800\,x^{(12,24,28,32)}+29503754451400659871708800\,x^{(12,28,28,28)}+2620939671593712000\,x^{(14,14,14,54)}$

$+270702844404274022400\,x^{(14,14,18,50)}+8523464275904018486400\,x^{(14,14,22,46)}+93034526381098508563200\,x^{(14,14,26,42)}$

$+379983981235954467196800\,x^{(14,14,30,38)}+604802818584748818432000\,x^{(14,14,34,34)}+20368054453259288755200\,x^{(14,18,18,46)}$

$+454393132788226257561600\,x^{(14,18,22,42)}+3401995763559779356723200\,x^{(14,18,26,38)}+9163247069750392054656000\,x^{(14,18,30,34)}$

$+6953252080915883809670400\,x^{(14,22,22,38)}+34330843476377162717414400\,x^{(14,22,26,34)}+58096716011865154768423680\,x^{(14,22,30,30)}$

$+106495793828343904674009600\,x^{(14,26,26,30)}+3522250814603444421480\,x^{(16,16,16,48)}+141471290228655727542240\,x^{(16,16,20,44)}$

$+1807301013360943520604240\,x^{(16,16,24,40)}+8066956190453872991114400\,x^{(16,16,28,36)}+13214153783171548077502320\,x^{(16,16,32,32)}$

$+3964002010905760614735840\,x^{(16,20,20,40)}+34092545490716156388070560\,x^{(16,20,24,36)}+98081813885345815384578240\,x^{(16,20,28,32)}$

$+188988268097212468492284000\,x^{(16,24,24,32)}+331920630721173914714203200\,x^{(16,24,28,28)}+1086305572325313142512000\,x^{(18,18,18,42)}$

$+16621611828436208352864000\,x^{(18,18,22,38)}+82069352514044691433142400\,x^{(18,18,26,34)}+138880627186007069373378560\,x^{(18,18,30,30)}$

$+167725898337549940348665600\,x^{(18,22,22,34)}+520299048574710021878784000\,x^{(18,22,26,30)}+953773550830271320798279680\,x^{(18,26,26,26)}$

$+74770057770515243859199680\,x^{(20,20,20,36)}+414490427024726119115593440\,x^{(20,20,24,32)}+727959902804963642950179840\,x^{(20,20,28,28)}$

$+1402701258105770242985395200\,x^{(20,24,24,28)}+1063367844120806686912304640\,x^{(22,22,22,30)}+1949257967639374424005816320\,x^{(22,22,26,26)}$

$+2702815908351654992712480000\,x^{(24,24,24,24)})$

Higher weight enumerators

$$W_3(x;g_{24}) = \mathrm{Sym}\,(x^{(0,0,0,0,0,0,0,0,24)} + 759\,x^{(0,0,0,0,0,0,0,8,16)} + 2576\,x^{(0,0,0,0,0,0,0,12,12)} + 22770\,x^{(0,0,0,0,0,0,8,8,8)}$$

$$+28052640\,x^{(0,0,4,4,4,4,4,4)} + 15301440\,x^{(2,2,2,2,2,2,2,10)} + 153014400\,x^{(2,2,2,2,2,2,6,6)} + 2285015040\,x^{(3,3,3,3,3,3,3,3)})$$

$$+340032\,(x^{(0,0,0,0,2,6,6,10)},168) + 212520\,(x^{(0,0,0,0,4,4,4,12)},56) + 1275120\,(x^{(0,0,0,0,4,4,8,8)},84) + 4080384\,(x^{(0,0,0,0,6,6,6,6)},14)$$

$$+637560\,(x^{(0,0,0,4,4,4,4,8)},56) + 5100480\,(x^{(0,0,2,2,4,4,6,6)},336) + 15301440\,(x^{(0,2,2,2,2,4,4,8)},168) + 61205760\,(x^{(0,2,2,2,4,4,4,6)},224)$$

$$+24482304\,(x^{(1,1,1,1,5,5,5,5)},56) + 13601280\,(x^{(1,1,1,3,3,3,3,9)},56) + 40803840\,(x^{(1,1,1,3,3,3,5,7)},224) + 244823040\,(x^{(1,1,3,3,3,3,5,5)},84)$$

$$+795674880\,(x^{(2,2,2,2,4,4,4,4)},14)$$

$$W_4(x;g_{24}) = (x^{0^{15}24^{1}},16) + 759\,(x^{0^{14}8^{1}16^{1}},240) + 2576\,(x^{0^{14}12^{2}},120) + 22770\,(x^{0^{13}8^{3}},560)$$

$$+340032\,(x^{0^{12}2^{1}6^{2}10^{1}},1680) + 212520\,(x^{0^{12}4^{3}12^{1}},560) + 1275120\,(x^{0^{12}4^{2}8^{2}},840) + 4080384\,(x^{0^{12}6^{4}},140)$$

$$+637560\,(x^{0^{11}4^{4}8^{1}},1680) + 5100480\,(x^{0^{10}2^{2}4^{2}6^{2}},10080) + 1275120\,(x^{0^{10}4^{6}},448) + 28052640\,(x^{0^{10}4^{6}},840)$$

$$+15301440\,(x^{0^{9}2^{4}4^{2}8^{1}},5040) + 61205760\,(x^{0^{9}2^{3}4^{3}6^{1}},6720) + 24482304\,(x^{0^{8}1^{4}5^{4}},1680) + 13601280\,(x^{0^{8}3^{3}4^{3}9^{1}},1680)$$

$$+40803840\,(x^{0^{8}1^{3}3^{3}5^{1}7^{1}},6720) + 244823040\,(x^{0^{8}1^{2}3^{4}5^{2}},2520) + 15301440\,(x^{0^{8}2^{7}10^{1}},240) + 153014400\,(x^{0^{8}2^{6}6^{2}},840)$$

$$+15301440\,(x^{0^{8}2^{4}4^{4}},10080) + 795674880\,(x^{0^{8}2^{4}4^{4}},420) + 2285015040\,(x^{0^{8}3^{8}},30) + 15301440\,(x^{0^{7}2^{8}8^{1}},240)$$

$$+30602880\,(x^{0^{7}2^{6}4^{1}6^{1}},13440) + 61205760\,(x^{0^{7}2^{6}4^{3}},26880) + 40803840\,(x^{0^{6}1^{4}2^{1}3^{4}6^{1}},13440) + 81607680\,(x^{0^{6}1^{4}3^{4}4^{2}},6720)$$

$$+122411520\,(x^{0^{6}2^{9}6^{1}},4480) + 306028800\,(x^{0^{6}2^{8}4^{2}},20160) + 397837440\,(x^{0^{6}2^{8}4^{2}},840) + 122411520\,(x^{0^{5}1^{5}2^{2}3^{2}4^{1}5^{1}},80640)$$

$$+367234560\,(x^{0^{5}1^{4}2^{2}3^{4}4^{1}},40320) + 1468938240\,(x^{0^{5}2^{10}4^{1}},2688) + 122411520\,(x^{0^{4}1^{8}4^{4}},1680) + 122411520\,(x^{0^{4}1^{6}2^{4}3^{1}7^{1}},13440)$$

$$+244823040\,(x^{0^{4}1^{6}2^{3}5^{2}},6720) + 244823040\,(x^{0^{4}1^{6}2^{3}3^{2}6^{1}},26880) + 489646080\,(x^{0^{4}1^{6}2^{2}3^{2}4^{2}},40320) + 612057600\,(x^{0^{4}1^{5}2^{4}3^{2}5^{1}},40320)$$

$$+1713761280\,(x^{0^{4}1^{4}2^{4}3^{4}},13440) + 1958584320\,(x^{0^{4}1^{4}2^{4}3^{4}},5040) + 8446394880\,(x^{0^{4}2^{12}},140) + 122411520\,(x^{0^{3}1^{8}2^{4}8^{1}},1680)$$

$$+367234560\,(x^{0^{3}1^{8}2^{3}4^{1}6^{1}},6720) + 979292160\,(x^{0^{3}1^{7}2^{4}4^{1}5^{1}},13440) + 2693053440\,(x^{0^{3}1^{6}2^{4}3^{2}4^{1}},26880) + 4161991680\,(x^{0^{2}1^{8}2^{4}4^{2}},2520)$$

$$+13710090240\,(x^{0^{2}1^{6}2^{6}3^{2}},3360) + 244823040\,(x^{1^{15}9^{1}},16) + 3427522560\,(x^{1^{14}5^{2}},120) + 25461596160\,(x^{1^{12}3^{2}},140) + 102825676800\,(x^{1^{8}2^{8}},30)$$

Note that $W_4(x;g_{24})$ written out fully has $465,908$ terms.