Techniques Used in the Classification of Self-Dual $\mathbb{Z}_4$ Codes

Joe Fields and Vera Pless

Mathematics Department, University of Illinois at Chicago
Chicago, Illinois 60607

fields@math.uic.edu, VPless@uic.edu

1. Abstract

Recently [11], the self-dual $\mathbb{Z}_4$ codes of Type II and length 16 were completely classified. We discuss the methods used in this classification. We also present some of the codes with interesting properties that we found.

2. Introduction

Interest in $\mathbb{Z}_4$ codes has increased since it was shown [6] that "good" non-linear binary codes could be handled as Gray images of $\mathbb{Z}_4$ codes.

An interesting subclass of $\mathbb{Z}_4$ codes is the class of self-dual codes which exist at any length (in contrast to binary self-dual codes which can only exist at even lengths). There are classifications of all binary, ternary and quaternary self-dual codes of modest lengths. Such classifications are possible because there are "mass formulas" for all of these classes of codes. (A mass formula gives the total number of codes having a specific property.) Conway and Sloane [3] classified all self-dual $\mathbb{Z}_4$ codes of length up to 9 without a mass formula. Recently Gaborit [5] has demonstrated a mass formula for both the self-dual and the Type II $\mathbb{Z}_4$ codes. Using this information and a new code automorphism group program [7, 8], we have classified all Type II $\mathbb{Z}_4$ codes of length 16 [11]. Generator matrices for all of these codes are available on the world wide web at `http://www.math.uic.edu/~pless`

We found that there are 133 inequivalent Type II codes of length 16. As Type II codes can only exist at lengths divisible by 8, this is the next classification possible after the work of Conway and Sloane at length 8 – where there were just 4 Type II codes.

In our classification we were often in the situation where there were too many codes of a specific form to be distinguished by the automorphism group program. In fact there were up to 2^{21} codes of some types. We were able to reduce these numbers significantly so that at most a few hundred were run at a time with the code automorphism group program.

We discuss the methods used to make this reduction. Some involved finding representatives of certain $n \times n$ matrices under the action of S_n. Others involved finding elements in the dual of a certain subcode – these elements could be added to some of a code's generators producing the canonical form of the generator matrix for an equivalent code. We will discuss these methods more fully in Section 3. We will also discuss the linearity of the Gray images of the 5 inequivalent codes attaining the highest minimum Lee weight, 8.

3. Preliminaries

A $\mathbb{Z}_4$ code is a module over the ring $\mathbb{Z}/(4\mathbb{Z})$.

Two $\mathbb{Z}_4$ codes are *equivalent* if one can be obtained from the other by permuting coordinates and making sign changes in a subset of the coordinates.

For $\mathbb{Z}_4$ codes, the usual notion of dimension does not hold. Every $\mathbb{Z}_4$ code C is equivalent (indeed, equivalent via permutations only) to a code having a generator matrix of the form

$$\begin{bmatrix} I_{j\times j} & A & B \\ 0 & 2I_{k\times k} & C \end{bmatrix}$$

and C is said to have *type* 4^j2^k [3]. Furthermore, the number of vectors in such a code is 4^j2^k.

Orthogonality is defined as usual, and a code C is *self-dual* if $C = C^{\perp}$.

The *Lee weight* of a $\mathbb{Z}_4$ vector is an integer that counts 1 and 3 components as one, and 2 components as two. In computing the *Euclidean weight* of a vector, a 2 component counts as four. A $\mathbb{Z}_4$ code is of *Type II* if the Euclidean weight of any vector in it is divisible by 8 and it contains a vector equivalent to the all-ones vector.

The automorphism group of a length n code over $\mathbb{Z}_4$ is the group of all coordinate permutations and sign changes that fix C set-wise. These groups have natural permutation representations on $2n$ points. There is a normal subgroup consisting of sign changes only, and the quotient group acts naturally on n points.

In [11] we showed that any Type II code C of length n has a generator matrix of the form

$$G = \begin{bmatrix} A & & \tilde{I}_k + 2B \\ & 2C & \end{bmatrix} \tag{1}$$

where A,B and C are binary matrices, $\tilde{I}_k = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & & & \\ \vdots & & I_{k-1} & \\ 0 & & & \end{bmatrix}$, and all of the following hold:

- $\begin{bmatrix} A & \tilde{I}_k \end{bmatrix}$ is a generator matrix for a doubly-even binary code C_1 of length n.
- $\begin{bmatrix} A & & \tilde{I}_k \\ & C & \end{bmatrix}$ generates $C_2 = C_1^{\perp}$.

- The matrix B is chosen to make C self-orthogonal over $\mathbb{Z}_4$ and so that every vector has Euclidean weight divisible by 8.

The code C has type $4^k 2^{n-2k}$.

We labeled a Type II code by describing the weight 4 vectors in C_1 and by giving its type. We recall that every self-orthogonal binary code generated by weight 4 vectors is a direct sum of d_{2n} $(n \geq 2)$, e_7 and e_8 [2]. If C_1 does not contain any weight 4 vectors, it receives the label f_t, where t is its dimension.

For each doubly-even code C_1 of length 16 and dimension k, there are as many $\mathbb{Z}_4$ Type II codes with C_1's label as there are ways to fill in the matrix $\tilde{I}_k + 2B$ in such a way as to satisfy the Type II constraints.

It is not difficult to find all the inequivalent doubly-even binary codes of length 16 and dimension k where $1 \leq k \leq 8$. For many of these doubly-even codes C_1, it is difficult to reduce the number of inequivalent $\mathbb{Z}_4$ codes C (with this C_1 and a given type) so that they can be handled by the group automorphism program. We describe how to do this reduction in the next section.

We use the canonical generator matrix (1) to see all of the choices and perform the reductions.

4. Methods

We call a doubly-even binary code which contains the all-ones vector *admissible*. Up to equivalence there are four admissible codes of length 16 and dimension 3. We label these codes $3-d_4$, $3-2d_4$, $3-d_6$ and $3-f_3$. These codes have (respectively) a single vector of weight four, two disjoint vectors of weight four, three mutually intersecting weight four vectors, and no weight four vectors at all.

From each of these admissible codes we obtain $\mathbb{Z}_4$ codes with the same basic label by extending the basis of the code to one for its dual (the vectors added in this extension are multiplied by 2) and filling in the off-diagonal positions (except those in the first row) in $\tilde{I}_2 + 2B$ in all possible ways (subject to the Type II constraints). In the proof of the mass formula, the positions on the diagonal are also taken to be free – since we are concerned with inequivalent codes, we fix these positions (a sign change on the column containing a 3 on the diagonal of $\tilde{I}_k$ would give us an equivalent code with a 1 there). Hence each of these admissible codes gives two $\mathbb{Z}_4$ codes.

For example, the admissible code $3-d_6$ has generator matrix

$$G_{3-d_6} = \left[\begin{array}{ccccccccccccc|ccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1 \end{array}\right].$$

From this code we get the following two $\mathbb{Z}_4$ codes of Type II:

$$G_{\mathbf{3-d_6a}} = \left[\begin{array}{ccccccccccccc|ccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&3&3\\
1&1&1&0&0&0&0&0&0&0&0&0&0&2&1&0\\
1&1&0&1&0&0&0&0&0&0&0&0&0&0&2&1\\
\hline
2&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
2&0&2&2&2&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&2&2&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&2&0&2&0&0&0&0&0&0&0&0&0\\
0&0&0&0&2&0&0&2&0&0&0&0&0&0&0&0\\
0&0&0&0&2&0&0&0&2&0&0&0&0&0&0&0\\
0&0&0&0&2&0&0&0&0&2&0&0&0&0&0&0\\
0&0&0&0&2&0&0&0&0&0&2&0&0&0&0&0\\
0&0&0&0&2&0&0&0&0&0&0&2&0&0&0&0\\
0&0&0&0&2&0&0&0&0&0&0&0&2&0&0&0
\end{array}\right]$$

and

$$G_{\mathbf{3-d_6b}} = \left[\begin{array}{ccccccccccccc|ccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&3&3\\
1&1&1&0&0&0&0&0&0&0&0&0&0&2&1&2\\
1&1&0&1&0&0&0&0&0&0&0&0&0&2&0&1\\
\hline
2&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
&&&&&&\vdots&&&&&&&&\vdots&\\
0&0&0&0&2&0&0&0&0&0&0&0&2&0&0&0
\end{array}\right].$$

A note on notation: admissible codes are binary, they are distinguished from their $\mathbb{Z}_4$ descendents by using boldface type for the labels of the latter.

Notice two things, these generator matrices are identical except for their upper-right 3×3 blocks, and furthermore they span equivalent codes. We can see this equivalence in two different ways.

First, note that the vector $\left[\begin{array}{ccccccccccccc|ccc} 2&0&0&0&0&0&0&0&0&0&0&0&0&2&2&2 \end{array}\right]$ is in both codes (this may not be immediately obvious, but one only needs to verify that it is orthogonal to the first three rows of each generator matrix). If we add this vector to each of the first three rows of (say) $G_{\mathbf{3-d_6a}}$, then make the appropriate sign changes (in columns 1, 14, 15 and 16) we have converted one generator matrix to the other.

The second way of seeing the equivalence of these codes is to perform row and column permutations (interchange columns 15 and 16, then interchange rows 2 and 3, finally interchange columns 3 and 4).

In the remainder of this paper we will not give full generator matrices for $\mathbb{Z}_4$ codes, we will only give the first k rows of a generator matrix (where k is the exponent of 4 in the type of the code). We call the $k \times k$ block at the right-hand side of these partial generator matrices the Gaborit square, or more succinctly, the G-square. The remainder (i.e. the part that is not shown) of the generator matrices we call the 2-part – it is completely determined by the binary code C_1. Thus the generator matrices of codes that are derived from the same binary code will only differ in their G-squares.

The two codes derived from $3-d_4$ have the following (partial) generator matrices:

$$\left[\begin{array}{ccccccccccccc|ccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&3&1\\ 1&1&1&0&0&0&0&0&0&0&0&0&0&2&1&0\\ 0&1&1&1&1&1&1&1&0&0&0&0&0&2&2&1 \end{array}\right]$$

and

$$\left[\begin{array}{ccccccccccccc|ccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&3&1\\ 1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&2\\ 0&1&1&1&1&1&1&1&0&0&0&0&0&2&0&1 \end{array}\right]$$

The vector $\left[\begin{array}{ccccccccccccc|ccc} 0&2&0&0&0&0&0&0&0&0&0&0&0&2&2&2 \end{array}\right]$ is orthogonal to the displayed portion of each of these generator matrices, and so is in both codes. The addition of this vector to all three rows of either matrix will transform it into a matrix equivalent to the other (sign changes must be made in columns 2, 14, 15 and 16).

The first case (that we encountered) where inequivalent $\mathbb{Z}_4$ codes are derived from the same admissible code is $3-2d_4$, these $\mathbb{Z}_4$ codes are thus given the labels $\mathbf{3-2d_4a}$ and $\mathbf{3-2d_4b}$. These codes have different symmetrized weight enumerators.

There are 8 admissible codes of dimension 4 and length 16 having labels: $4-2d_4$, $4-4d_4$, $4-d_4+d_6$, $4-d_4$, $4-d_6$, $4-d_8$, $4-e_7$ and $4-f_4$. We can construct 2^3 $\mathbb{Z}_4$ codes having each of these labels. Using the techniques described above we can show that up to equivalence there is exactly one $\mathbb{Z}_4$ code derived from each of the codes: $4-d_4$, $4-d_6$, $4-e_7$ and $4-f_4$. From the code $4-4d_4$, we get three $\mathbb{Z}_4$ codes, and from the remaining admissible codes ($4-2d_4$, $4-d_4+d_6$, $4-d_8$) we get two.

Let us consider the codes derived from $4-d_8$. We take a generator matrix for $4-d_8$ to be

$$G_{4-d_8} = \left[\begin{array}{cccccccccccc|cccc} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 1&1&0&1&0&0&0&0&0&0&0&0&0&0&1&0\\ 1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&1 \end{array}\right],$$

and note the symmetries of this matrix: the first row is distinct, but (up to equivalence) we can freely permute the last three rows. Since the inner products among these last three rows are all two, the relevant entries in the G-square must be skew-symmetric. Clearly S_3 acts on the last three rows and columns of the G-square, and so, we need to find representatives for the orbits of this action. We can find these representatives computationally – a brute force approach is to make a list of all possible G-squares, compute the orbit of the first element in the list, place this first element in a new list of orbit representatives, remove all elements of this orbit from the list and repeat until the list is exhausted.

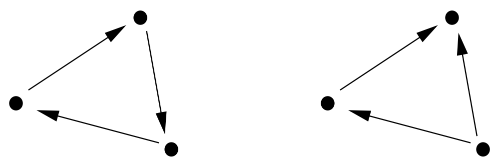
A slightly less brutal approach is to take advantage of a one-to-one correspondence between G-squares and digraphs.

Essentially, to every binary admissible code, we associate a set of graphs the points of which correspond to basis elements for the code. That is, the points correspond to rows of a generator matrix in canonical form (aside from the first row, where we always have the all-ones vector). Edges come in two classes: if the inner-product of the vectors corresponding to the endpoints is singly-even the edge must be directed, and if the inner-product is doubly-even the edge must be undirected. Different graphs that correspond to a given admissible code have the same directed/undirected 'signature'. Certain choices are made to determine a particular graph: when an edge must be directed, we choose the direction, and when an edge must be undirected we choose whether to put it in or not. These graphs look like multi-partite graphs in which the 'parts' consist of complete directed graphs, and some subset of the possible (undirected) edges between parts.

The G-square is a matrix of the form $\tilde{I}_k + 2B$. Consider the sub-matrix B' which consists of the last $k-1$ rows and columns of B. B' is the adjacency matrix of this directed graph. A 1 in the ith row and jth column of B' indicates that there is an edge going from vertex j to vertex i.

How many essentially distinct complete digraphs are there on three points?

$$\begin{matrix} 1 & 1 & 1 & 1 \\ * & 1 & 0 & 2 \\ * & 2 & 1 & 0 \\ * & 0 & 2 & 1 \end{matrix}$$



$$\begin{matrix} 1 & 1 & 1 & 1 \\ * & 1 & 0 & 0 \\ * & 2 & 1 & 0 \\ * & 2 & 2 & 1 \end{matrix}$$

FIGURE 1. There are just two complete digraphs on 3 points.

This sort of situation occurs many times and so we have found the G-squares which correspond to the 4 complete digraphs on 4 points, the 12 complete digraphs on 5 points and the 56 complete digraphs on 6 points. For instance, the code $5 - d_{10}$ has (a priori) 2^6 $\mathbb{Z}_4$ descendents, but we can reduce this immediately to the four representatives of S_4 acting on skew-symmetric matrices by row and column permutations. Interestingly, it turns out that there are only two inequivalent $\mathbb{Z}_4$ codes descendant from $5 - d_{10}$, but our purpose here was only to reduce the number of codes that need to be distinguished by the automorphism group program.

It would appear that the component code e_7 would, like d_8, have two $\mathbb{Z}_4$ descendents. This is not so – and provides an illustration of our other main technique. First, let us consider d_8 again. Since there is a column of 1's in the generator matrix for d_8, there is a special element in $2C_2$ (Recall that C_2 is $C_1^{\perp}$), namely

$$s = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2].$$

If we add s to every row of a generator matrix for some $\mathbb{Z}_4$ descendant, and then make a sign change in the first column, we have a generator matrix (back in canonical form) for an equivalent code. In the 'graphs' context, s gives us the freedom to (simultaneously) reverse all the arrows. Such an element (i.e. one that gives us the "reverse-all-arrows" action) exists for all d_{2n} codes. While this action doesn't collapse any orbits for d_8, it does for d_{10} (and larger codes, see Figure 2) – where certain digraphs become equivalent, as one is the result of reversing all of the arrows in the other.

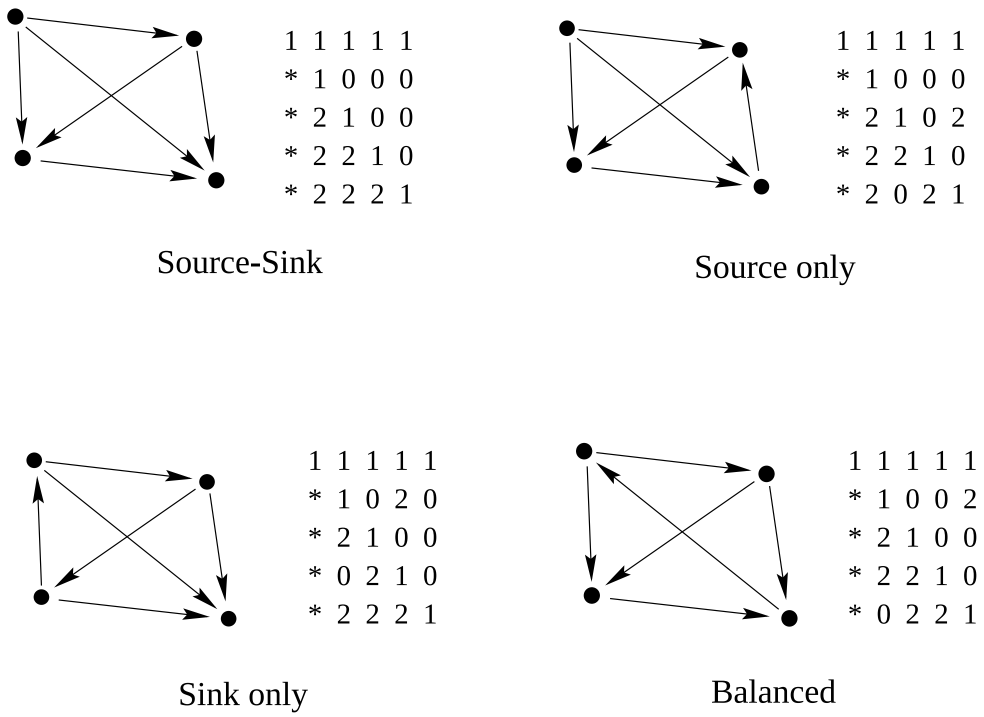


FIGURE 2. The graphs labeled "source only" and "sink only" are equivalent under the 'reverse-all-arrows' action.

Back to e_7, here there are other elements of $2C_2$ that can be used to "reverse any arrow". For instance the vector

$$s = [0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2]$$

can be added to the 1st, 3rd and 4th rows of a generator matrix for a descendant code:

$$G_{\mathbf{4-e_7*}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & * \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & * & 1 \end{bmatrix}$$

and after appropriate sign changes on columns, we have a generator matrix for an equivalent code (in canonical form) where a different choice for the starred positions has been made. There are vectors in $2C_2$ that give us such freedom for all the positions in the G-square of an e_7 descendant code – thus there is only one descendant (up to equivalence).

Another situation that occurs frequently can be illustrated with the admissible code $6 - d_6 + e_7$. A generator matrix for this code is

$$G_{6-d_6+e_7} = \left[\begin{array}{cccc|cccccc|c|cc|ccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right],$$

which has been partitioned to show the block structure. From the analysis done previously, we would expect that there could only be one component subcode corresponding to the d_6 block (the number of distinct complete digraphs on two points). We have just seen that there is only one possibility for the 3×3 block that corresponds to the e_7 component subcode. But what about the 3×2 and 2×3 off-diagonal blocks? Clearly if these are zero, we have a direct sum, and the all-ones vector serves to 'glue' the two components together. What other possibilities are allowed?

Since (up to equivalence) we are free to permute the rows and columns of the G-square that correspond to a particular component code amongst themselves, we have an action of $S_2 \times S_3$ on the square.

Notice that vectors from different components have inner-product zero – so the entries in these off-diagonal blocks must be symmetric. In the context of our digraph analogy, this means that edges go in both directions between points that are in different blocks, thus we can think of these edges as being undirected.

We are free to permute the points that correspond to basis elements of a component subcode amongst themselves, but not between different component subcodes. The action of $S_2 \times S_3$ on the graph respects this partition.

It is a tedious exercise to determine the inequivalent graphs that have a certain directed/undirected signature, an example is given in Figure 3.

It should be pointed out that there are in fact only three $\mathbb{Z}_4$ descendents of $6-d_6+e_7$, so the correspondence with digraphs is not exact. That is, there are equivalent codes which correspond to different digraphs. Nevertheless, 13 (see Figure 3) is a significant improvement over the 1024 codes that would have had to have been checked for isomorphisms a priori.

5. Five codes with minimum Lee weight 8

The highest minimum Lee weight in a Type II code of length 16 is 8, and there are exactly 5 codes with this minimum weight.

We see that these codes have types $4^5 2^6$, $4^6 2^4$, $4^7 2^2$ and 4^8. The two codes of type 4^8 have identical symmetrized weight enumerators and their groups have the same orders

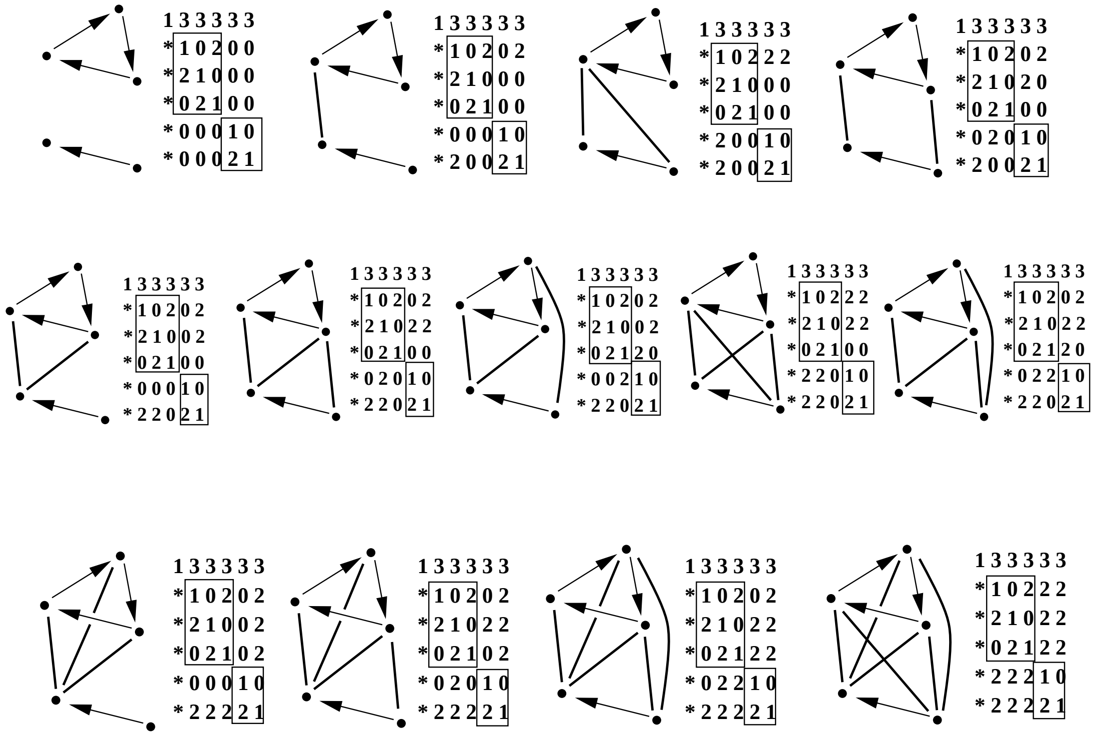


FIGURE 3. There are 13 distinct graphs having the directed/undirected signature of $6 - d_6 + e_7$.

FIGURE 4. Symmetrized weight enumerators for the five codes having minimum Lee weight 8. (Reproduced from [11])

label	swe coefficients										
	1 c^{16}	c^4 c^{12}	c^6 c^{10}	c^8	b^4c^3 b^4c^9	b^4c^5 b^4c^7	b^8 b^8c^8	b^8c^2 b^8c^6	b^8c^4	$b^{12}c$ $b^{12}c^3$	b^{16}
5_f5	1	140	448	870	0	0	480	13440	33600	0	2048
6_4d4c	1	76	192	486	512	1536	480	11904	30528	2048	1024
7_2d8g	1	44	64	294	768	2304	480	11136	28992	3072	512
8_d16g	1	28	0	198	896	2688	480	10752	28224	3584	256
8_2e8j	1	28	0	198	896	2688	480	10752	28224	3584	256

and orbit structure. But, two codes with different labels cannot be equivalent. The admissible code $5-f$ is actually the unique $[16,5,8]$ code $R(1,4)$. From the table above we see that the Gray image C of $5-f5$ contains only vectors of weights divisible by 4.

We will need the following lemma:

Lemma *If C is a self-dual $\mathbb{Z}_4$ code whose Gray image $G(C)$ consists entirely of doubly-even vectors, then $G(C)$ is a linear binary code.*

Proof. It is known [6] that the Gray map is an isometry – if x and y are $\mathbb{Z}_4$ codewords so that the Lee weight of $x-y$ is d, then the Hamming weight of $G(x)-G(y)$ is also d.

Consider two vectors $x, y \in G(C)$, say $x = G(a)$ and $y = G(b)$ for $a, b \in C$. The distance between a and b (i.e. the Lee weight of $a-b$) is doubly-even since $G(a-b) \in$

$G(C)$. The distance between x and y (i.e. the Hamming weight of $x+y$) must also be doubly-even. Using the formula

$$|x+y| = |x| + |y| - 2(x \cap y)$$

for the weight of the sum of two vectors, we can deduce that $x \cap y$ is even, hence x and y are orthogonal. So $G(C)$ is a set of mutually orthogonal vectors, and so the linear code that $G(C)$ generates is self-orthogonal. But $G(C)$ is already as big as is possible for a self-orthogonal subspace of $\mathbb{F}_2^{2n}$.

□

Because the binary code $G(5-f5)$ has only doubly-even vectors (see Figure 4), by the lemma it is linear. There are 5 inequivalent doubly-even $[32,16,8]$ codes [2]. It is not hard to see that $G(5-f5)$ is $R(2,5)$.

If we compute the number of singly-even and doubly-even vectors in the Gray images of the other 4 codes it is easy to determine that they do not satisfy the conditions in [10] for these codes to be linear. Hence the Gray images of the last four codes listed above are $(32, 2^{16}, 8)$ non-linear binary codes.

References

[1] A. Bonnecaze, P. Sole, Quaternary type II codes, to appear.

[2] J.H. Conway, V. Pless, On the Enumeration of Self-Dual Codes *J. Comb. Theory* **A 28** (1980), 26-53.

[3] J.H. Conway, N.J.A. Sloane, Self-Dual Codes over the Integers Modulo 4, *J. Comb. Theory* **A 62** (1993), 31-45.

[4] J. Fields, V. Pless, J. Leon, The Self-Dual $\mathbb{Z}_4$ Codes of Length Up To 15, preprint.

[5] P. Gaborit, Mass Formulas for Self-Dual Codes Over $\mathbb{Z}_4$ and $\mathbb{F}+u\mathbb{F}_q$ Rings, *IEEE Trans. Inform. Theory* **IT42 n. 4** (1996), 1222-1228.

[6] A.R. Hammons, Jr., P.V. Kumar, A.R. Calderbank, N.J.A. Sloane and P. Sole, The $\mathbb{Z}_4$ Linearity of Kerdock, Preperata, Goethals and Related Codes, *IEEE Trans. Inform. Theory* **IT40** (1994), 301-319.

[7] Jeffrey S. Leon, Permutation Group Algorithms Based on Partitions, I: Theory and Algorithms, *J. Symbolic Computation.*

[8] Jeffrey S. Leon, Partitions, Refinements, and Permutation Group Computations, *Proc. DIMACS Workshop on Groups and Computation*, to appear.

[9] V. Pless, A Classification of Self-Orthogonal Codes over GF(2), *Discrete Math.* **3** (1972), 209-246.

[10] V. Pless, Parents, Children, Neighbors and the Shadow, *Contemporary Math.* **168** (1994), 279-290.

[11] V. Pless, J. Leon, J. Fields, All $\mathbb{Z}_4$ Type II Codes of Length 16 are Known, to appear in JCT(A).