Chapter 9

The General Case

9

9

The General Case

9 The General Case

After the construction of codes with prescribed automorphism group, we are now attacking the general case, i.e. we will no longer make any assumption on the presence of nontrivial automorphisms. The main goal is the evaluation of a transversal of the isometry classes, for any given parameter set. Of course, this daunting task can only be solved for small parameters. Also, since we are mostly interested in good codes, we shall restrict attention to codes with minimum distance at least 3, the reason is that this restriction makes things much easier.

The main point is that the construction of a transversal of codes and the classification by isometry classes are not two separate issues but rather go hand in hand. We will see that the classification is best done already during the construction of codes. In fact, the construction of codes is supported by the classification part in that not too much overhead is constructed which otherwise would have to be deleted later. The corresponding algorithmic principle is that of *orderly generation of discrete structures.* The order refers to an order which we impose on the objects, for instance the lexicographical order given by the columns of the generator matrices. This leads to a central problem in the systematic construction of transversals of orbits, e.g. of isometry classes: We have to introduce a *normal form*, following the request of D. Slepian, who wrote in 1960 ([184]):

> *"The task of analyzing group codes would be greatly simplified if a canonical form could be found for each equivalence class of Ω-matrices*[1]*. That is, for a given n and k, we should like to be able to write down one generator matrix from each equivalence class. This would provide a simple means of describing each of the essentially different (n,k)-codes."*

The plan of this chapter is as follows. We first show how to reduce the computation of transversals of isometry classes to a problem in finite projective geometry. This will give us control over the minimum distance. The remaining problem of computing orbits can be solved using methods from Computational Group Theory. We will give a very brief introduction to this area, focusing mainly on fundamental algorithms for permutation groups. After that, we describe the method of orderly generation, and we apply this to the construction of optimal linear codes. The major issue is that of computing orbits of a group on subsets. We treat the permutation representation of the projective linear group. Finally, we present numerical data which was computed. We

[1] a group code is a linear code, an Ω-matrix is a generator matrix Γ

classify the isometry classes of optimal linear codes for small parameters and over small fields.

9.1

9.1 The Problem

We are faced with the following problem: For a given length n, dimension k, minimum distance d and field $\mathbb{F}_q$, we would like to determine the isometry classes of linear (n,k,d,q)-codes. For instance, this could be done by listing generator matrices for each such code. Before we embark on this mission, let us recall what we have learned in the earlier chapters.

9.1.1 **Remarks** In order to evaluate a transversal of the isometry classes of linear (n,k,d,q)-codes, we can use the following facts:

- In Chapter 1 we saw that a linear code C can be described both by a generator matrix Γ and by a check matrix Δ. The check matrix is a generator matrix of the dual code $C^\perp$. Moreover, as $(C^\perp)^\perp = C$, the mapping

  $$\perp \;: \mathcal{U}(n,k,q) \to \mathcal{U}(n,n-k,q),\ C \mapsto C^\perp$$

  is a bijection from the set of (n,k)-codes to the set of $(n,n-k)$-codes over $\mathbb{F}_q$. In fact, as the map $\perp$ is compatible with the various types of isometries, this map descends to a bijection of the corresponding isometry classes. This fact holds true both for linear and for semilinear isometry classes. In the following, when we speak of isometry classes (unqualified) we mean that the result holds regardless of whether the isometry classes under consideration are linear or semilinear. It remains to investigate the map $\perp$ further.

- As we are interested mainly in good codes, we may ignore codes with minimum distance at most 2. Such codes cannot correct a single error, so this restriction does not exclude anything which would be interesting. So, from now on we consider only codes C with minimum distance at least 3, for short: linear $(n,k,\geq 3,q)$-codes.

- From 1.3.9 we know that check matrices of such codes have pairwise linearly independent columns. In the language of 6.1.14, this means that such codes have projective duals. Conversely, a code whose check matrix is projective has minimum distance at least 3 (see Exercise 1.3.21). Therefore, the duality map may be restricted to induce bijections between the following isometry classes of codes:

  1. $(n,n-k)$ projective codes,

2. (n,k)-codes with minimum distance greater than or equal to three (or $(n,k,\geq 3)$-codes). ◇

Vector spaces are often difficult to handle with a computer. In part this results from the fact that there are usually many different bases for the same space. As far as Slepian's problem of computing transversals of linear codes is concerned, we have to consider orbits of the isometry group on vector spaces. This raises other issues, like how to represent these orbits on the Computer, when typically each orbit is very long and the orbit elements are vector spaces. In order to overcome these problems, we may look for different representations of codes. We take the approach indicated in the last item of the previous remark of looking at the projective dual code. We can build on ideas from Section 6.1. The fundamental result 6.1.13 identifies linear isometry classes of linear codes with certain orbits of groups on mappings into projective space. In 6.1.25, the result is specialized to injective functions, which can be identified with their image, since we have the symmetric group acting on the domain of the map. These results may be summarized and slightly generalized as

Theorem **9.1.2**

1. *There is a one-to-one correspondence between the linear isometry classes of projective $(n,\leq k,q)$-codes and the set of orbits*

$$\mathrm{PGL}_k(q)\backslash\!\!\backslash\binom{\mathrm{PG}_{k-1}(q)}{n}.$$

2. *There is a one-to-one correspondence between the semilinear isometry classes of projective $(n,\leq k,q)$-codes and the set of orbits*

$$\mathrm{P\Gamma L}_k(q)\backslash\!\!\backslash\binom{\mathrm{PG}_{k-1}(q)}{n}.$$

In both cases, the isometry classes of projective (n,i,q)-codes correspond to the orbits on n-subsets of $\mathrm{PG}_{k-1}(q)$ with the property that the n points span a vector space of dimension i, for $0\leq i\leq k$. □

In order to describe the underlying map between codes and orbits of points, we start with a generator matrix

$$\Gamma=(\gamma_{i,j})\in\mathbb{F}_q^{k\times n}$$

of a projective (n,k,q)-code. Then

$$P(\Gamma):=\Big\{P(\gamma_{*,0}),P(\gamma_{*,1}),\ldots,P(\gamma_{*,n-1})\Big\}\subseteq\mathrm{PG}_{k-1}(q), \qquad \mathbf{9.1.3}$$

is a set of n points in $\mathrm{PG}_{k-1}(q)$ with the property that these points span a vector space of dimension k. Here,

$$P(\gamma_{*,j}) = \langle \gamma_{*,j} \rangle$$

is the projective point whose homogeneous coordinates are listed in the j-th column of Γ.

From the definition of the map, it is clear that rearranging the columns of Γ does not change the set $P(\Gamma)$. The action of $\mathrm{GL}_k(q)$ on generator matrices is similar to the action of $\mathrm{PGL}_k(q)$ on n-sets of points in $\mathrm{PG}_{k-1}(q)$. This is because left-multiplying Γ by an invertible matrix A gives rise to the set $\{P(A \cdot \gamma_{*,j}) \mid j \in n\}$ which is the image of $P(\Gamma)$ under the projective transformation induced by A.

This shows that the map $\Gamma \mapsto P(\Gamma)$ descends to a map from the linear isometry classes of projective codes to the orbits of n-sets of points of $\mathrm{PG}_{k-1}(q)$ under the projective linear group $\mathrm{PGL}_k(q)$. This map is a one-to-one correspondence and preserves the dimension.

Furthermore, the action of $\Gamma\mathrm{L}_k(q)$ on generator matrices corresponds to the action of $\mathrm{P}\Gamma\mathrm{L}_k(q)$ on n-sets of points in $\mathrm{PG}_{k-1}(q)$. Thus, the given map descends to a map from the semilinear isometry classes of projective codes to the orbits of n-sets of points of $\mathrm{PG}_{k-1}(q)$ under the projective semilinear group $\mathrm{P}\Gamma\mathrm{L}_k(q)$. Again, the resulting map is a one-to-one correspondence and preserves the dimension.

In order to describe the inverse map, we introduce the following notation. To a set S of n different points $p_0, \ldots, p_{n-1}$ in $\mathrm{PG}_{k-1}(q)$ we associate the generator matrix

9.1.4
$$\Gamma(S) = (a_{i,j}) \in \mathbb{F}_q^{k \times n}$$

where $p_j = \langle a_{*,j} \rangle$ for $j \in n$. This construction is not unique for two reasons. At first, we are making a choice by ordering the points of the set. Furthermore, the vector $a_{*,j}$ with $p_j = \langle a_{*,j} \rangle$ is unique up to non-zero scalar multiples. Therefore, the matrix $\Gamma(S)$ is unique *up to order of its columns and multiplication of columns by nonzero scalars.* Changing to a different set

$$A \cdot S = \{P(A \cdot a_{*,j}) \mid j \in n\}$$

results in changing the generator matrix to $A \cdot \Gamma(S)$. Summarizing, the code generated by $\Gamma(S)$ is determined up to linear isometry.

Under the duality map, the previous result becomes

Corollary **9.1.5**

1. *There is a one-to-one correspondence between the linear isometry classes of $(n, \geq k, \geq 3, q)$-codes and the set of orbits*
$$\mathrm{PGL}_{n-k}(q)\backslash\!\!\backslash \binom{\mathrm{PG}_{n-k-1}(q)}{n}.$$
2. *There is a one-to-one correspondence between the semilinear isometry classes of projective $(n, \geq k, \geq 3, q)$-codes and the set of orbits*
$$\mathrm{P\Gamma L}_{n-k}(q)\backslash\!\!\backslash \binom{\mathrm{PG}_{n-k-1}(q)}{n}.$$

In both cases, the isometry classes of $(n, k+i, \geq 3, q)$-codes correspond to the orbits on n-subsets of $\mathrm{PG}_{n-k-1}(q)$ with the property that the n points span a vector space of dimension $k-i$, for for some i with $0 \leq i \leq k$. □

Here, if $S = \{p_0, \ldots, p_{n-1}\}$ is a set of n points in $\mathrm{PG}_{n-k-1}(q)$ we obtain a projective check matrix

$$\Delta(S) = (b_{i,j}) \in \mathbb{F}_q^{(n-k)\times n} \qquad \textbf{9.1.6}$$

where $p_j = \langle b_{*,j} \rangle$ for $j \in n$ (notice that this is a vector of length $n-k$). This matrix is well-defined up ordering of the columns and up to non-zero scalar multiples of the columns. Since we take this matrix as a representative of an isometry class of codes, this non-uniqueness does not bother us.

The last result is already very close to what we really want. Apart from codes with minimum distance 1 or 2, Slepian's problem of finding a transversal of all isometry classes of codes is solved (provided we can evaluate the orbits in question, this remains to be seen). But we can refine this approach a little, to better suit the application in coding theory. What if Slepian would have asked

> *"For a given n and k and $d_{\min}$, we should like to write down one generator matrix from each equivalence class of (n, k)-codes whose minimum distance is at least $d_{\min}$."*

That is, what if we are interested in codes with a given minimum distance. The point with codes is that we really are not interested all that much in the generality of *all* available codes. The focus is of course on "good" codes, i.e. codes whose minimum distance is high. That means, we wish to direct attention to finding only a subset of the set of all (n, k)-codes, namely those with minimum distance greater than or equal to $d_{\min}$, where $d_{\min}$ is some specified lower bound which we choose beforehand. Of course, in the spirit of Slepian we still want one generator matrix from each equivalence class, i.e. we still want to classify the codes exhaustively. In particular, if no such code exists, our construction procedure should prove this fact. As we will see shortly, it is possible to refine our approach and take into account the prescribed minimum distance $d_{\min}$ right from the start. Of course, this restriction will save us a lot of work since we can skip a whole lot of codes which do not meet the required minimum distance. In a sense, we are looking for the needle in the haystack.

Let us introduce the following terminology.

9.1.7 **Definition** In a projective space, a set of points $\langle v^{(0)}\rangle, \langle v^{(1)}\rangle, \dots, \langle v^{(r-1)}\rangle$ is said to be in *in general position* (or *independent*) if they generate a vector space of dimension r. That is, the points are independent in projective space if and only if the representing vectors $v^{(0)}, v^{(1)}, \dots, v^{(r-1)}$ are independent as vectors. ◇

It is clear that this property does not depend on the choice of the representing non-zero vectors $v^{(i)}$ out of their respective subspace $\langle v^{(i)}\rangle$.

Using this language, we can rephrase 1.3.10 as follows. The generator matrices of linear codes over $\mathbb{F}_q$ of length n, dimension at least k and with minimum distance at least $d_{\min}$ for some integer $d_{\min} \geq 3$ correspond (up to ordering of the columns and multiplication of columns by non-zero scalars) to the n-subsets of $\mathrm{PG}_{n-k-1}(q)$ with the property that any $d_{\min} - 1$ points are in general position.

In fact, this correspondence descends to a correspondence between isometry classes of codes and orbits of projective groups on sets of points in projective space.

9.1.8 **Theorem** *For any given $d_{\min} \geq 3$, we have the following:*

1. *The linear isometry classes of linear $(n, \geq k, \geq d_{\min}, q)$-codes correspond one-to-one to the subset of*
$$\mathrm{PGL}_{n-k}(q)\backslash\!\!\backslash\binom{\mathrm{PG}_{n-k-1}(q)}{n},$$
consisting of the orbits of n-sets whose $d_{\min} - 1$-subsets are all in general position.
2. *Correspondingly, the semilinear isometry classes of linear $(n, \geq k, \geq d_{\min}, q)$-codes correspond one-to-one to the subset of*
$$\mathrm{P\Gamma L}_{n-k}(q)\backslash\!\!\backslash\binom{\mathrm{PG}_{n-k-1}(q)}{n},$$
consisting of the orbits of n-sets whose $(d_{\min} - 1)$-subsets are all in general position.

In both cases, the true minimum distance d of these codes is determined by the size of the smallest set of points which are dependent. Also, the true dimension of such a code *is determined as $n - r$, where r is the vector space dimension of the space spanned by the n points.* □

The rest of this chapter is devoted to solving the problem of constructing and classifying codes algorithmically using Theorem 9.1.8. It involves techniques from Computational Group Theory. The major issue, namely that of computing orbits on sets is addressed in Sections 9.2 and 9.6. The following Section 9.2 handles the "base case", where the sets have size 1 and hence are in fact points. Section 9.6 treats the general case, building on the results of Section 9.2.

9.2 Computing with Permutation Groups

In this section we address the problem of explicit computations with permutation groups. Our main goal is to compute orbits of permutation groups on subsets. This is part of a rather new branch of mathematics called *Computational Group Theory*, or CGT for short. Our main references are the recent book by Holt, Eick and O'Brien [91], the book by Seress [177] and the one by Butler [35]. For more on combinatorial algorithms see the book by Kreher and Stinson [116]. Several computer algebra systems covering CGT are available. The two most prominent are GAP [63] and Magma [140].

Let G be a finite group acting on a finite set X. For technical reasons we prefer in this chapter actions from the right, i.e. mappings

$$X \times G \to X \colon (x, g) \mapsto xg,$$

such that $(xg)g' = x(gg')$ and $x1 = x$. But we still use the symbol $G(x)$ for the orbit of x and G_x for its stabilizer.

Let us assume that G acts faithfully, which means that only the identity element of G fixes every point in X. According to 1.4.5, G is isomorphic to the permutation group $\overline{G} = \delta(G)$ induced by G on X, a subgroup of the symmetric group S_X on X. Hence we can assume that G is a permutation group on X, i.e. that $G \leq S_X$. In this section, we are concerned with computational tasks like the following.

1. For $x \in X$, compute $G(x) = \{xg \mid g \in G\}$, the *orbit of x under G*.
2. For $x \in X$, compute $G_x = \{g \in G \mid xg = x\}$, the *stabilizer of x in G*.
3. For $x, y \in G$ with $y \in G(x)$, compute an element $g \in G$ with $xg = y$. We call such an element a *transporter element*.

A remark concerning the last problem is in order. The required element $g \in G$ with $xg = y$ may not be unique. In fact, by Lemma 3.4.1 the set of all elements $g \in G$ with this property forms a unique right coset of the stabilizer G_x, the stabilizer of x in G.

In order to get started, the group has to be specified in some concrete way. A very simple way is by a set of *generators*, i.e. a set S of elements of G which together generate G, i.e. $\langle S \rangle = G$. If G is finite, this means that each element $g \in G$ can be written as a word of finite length over the alphabet S (Exercise 9.2.1). This will suffice for the moment. A more sophisticated representation of a group will be presented in Section 9.7. So for now, let us always assume that G is given by a finite set of generators $S = \{s_0, \ldots, s_{r-1}\}$.

The first problem is that of computing the orbit of a point $x \in X$ under the group G. We start by introducing a graph which describes the action of G on the set X.

9.2.1 **Definition (action graph)** Let the group G act on the finite set X. Assume that G is generated by a set of generators $S = \{s_0, \ldots, s_{r-1}\}$. The *action-graph* of G on X with respect to the set S is the directed graph (digraph) $\mathcal{G} = (X, \mathcal{E})$. That is, the vertices of $\mathcal{G}$ are the elements of X. The edge set $\mathcal{E}$ consists of directed labeled edges. There is an edge from vertex x to vertex y labeled by s_j if

$$xs_j = y.$$

We write $x \longrightarrow y$ to indicate that there is an edge from x to y. A *directed path* is a sequence of $x_0, x_1, \ldots, x_{u-1}$ of vertices which are pairwise distinct (except possibly for x_0 and x_{u-1} which may coincide) such that $x_0 \longrightarrow x_1 \longrightarrow \ldots \longrightarrow x_{u-1}$. We write $x \rightsquigarrow y$ is there is a path from x to y. The length of a path is the number of edges used. We also define a cycle to be a path where the start and the endpoint coincide (i.e. with $x_0 = x_{u-1}$ in the above notation). A loop is a cycle of length 1, i.e. an edge from a vertex x to itself. ◇

The action graph may have loops, i.e. edges of the form (x, x) for some vertex $x \in X$. Also, it may have several edges from vertex x to vertex y, namely if there are several elements $s \in S$ with $xs = y$.

9.2.2 **Lemma** *Let the group G act on the finite set X. Let $\mathcal{G} = (X, \mathcal{E})$ be the action graph with respect to the generating set S of G. Then the orbits of G on X correspond one-by-one to the connected components of $\mathcal{G}$. In particular, the connected components of $\mathcal{G}$ are well-defined and independent of the choice of the generating set S of G.*

Proof: Without loss of generality, we can replace G by the finite group G/K, where K is the kernel of the action of G on X, i.e. the pointwise stabilizer of the whole set X. The fact that G/K is finite follows from the fact that X is finite. Thus we may assume that G is a finite group. By Exercise 9.2.1, each $g \in G$ can be written as a word $s_{i_0}s_{i_1}\dots s_{i_{u-1}}$ in the generators. Recall that we write $x \rightsquigarrow y$ if there is a directed path from x to y in $\mathcal{G}$. Such a path gives rise to a group element $g = s_{j_0}s_{j_1}\dots s_{j_{u-1}}$ with $xg = y$. If $g^{-1} = s_{i_0}s_{i_1}\dots s_{i_{v-1}}$, then there also is a path

$$y \to ys_{i_0} \to ys_{i_0}s_{i_1} \to \dots \to yg^{-1} = x$$

in $\mathcal{G}$, i.e. $y \rightsquigarrow x$. This means that

$$x \rightsquigarrow y \iff y \rightsquigarrow x.$$

In other words, the relation "$\rightsquigarrow$" is undirected, and we can replace it by the symmetric $x \sim y$ (so that "$\sim$" really is an equivalence relation on X). We conclude that the concept of a connected component is well-defined in action graphs. Also, we have shown that $x \sim y$ if and only if x and y belong to the same G-orbit. This means that the connected components of $\mathcal{G}$ correspond bijectively to the G-orbits on X. It remains to show that the connected components in the action graph depend only on the group G, and not on the choice of the generating set S for G. To this end, let $T = \{t_0, \dots, t_{s-1}\}$ be another generating set for G. Write $\mathcal{G}_S$ and $\mathcal{G}_T$ for the action graphs of G with respect to the generating sets S and T. We need to show that $x \rightsquigarrow y$ in $\mathcal{G}_S$ if and only if $x \rightsquigarrow y$ in $\mathcal{G}_T$. We note that $x \rightsquigarrow y$ in $\mathcal{G}_S$ implies that $xg = y$ for $g = s_{i_0}s_{i_1}\dots s_{i_{u-1}}$. The element g has an expression in terms of the second generating set, say $g = t_{j_0}t_{j_1}\dots t_{j_{v-1}}$. But then $x \rightsquigarrow y$ in $\mathcal{G}_T$. The converse follows by symmetry. □

Remark In Computer Science, a subset U of vertices in a directed graph is called strongly connected if both $x \rightsquigarrow y$ and $y \rightsquigarrow x$ hold for all $x, y \in U$. The maximal strongly connected subsets of a graph are called strongly connected components and there are algorithms to compute these for a given graph (see [42]). It follows from 9.2.2 that the connected components of an action graph are strongly connected components. Nevertheless, there is a difference. The reason is that the strongly connected components in general digraphs may still have edges between them. The connected components in action graphs do not have this property. ◇ **9.2.3**

Example Figure 9.1 shows action-graphs of S_6 with respect to two different generating systems. The left picture uses $s_0 = (0,1,2,3,4,5)$ and $s_1 = (0,1)$. **9.2.4**

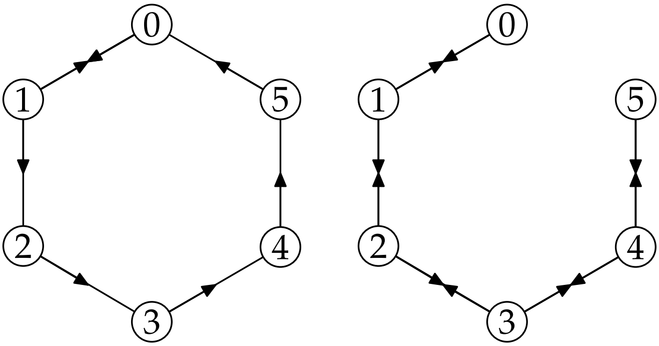


Fig. 9.1 Two action-graphs for S_6

The right picture is obtained by using the Coxeter generators $s_i = (i, i+1)$, where $i = 0, 1, \ldots, 4$. Edge labels and loops are not shown. ◇

To compute the orbit $G(x)$ of a point $x \in X$, we compute a *spanning tree* of the connected component of $\mathcal{G}$ containing x. This spanning tree is a cycle-free connected subgraph of $\mathcal{G}$, rooted at x, whose vertices are the elements of $G(x)$. This means that there is a unique directed path from x to any element y in $G(x)$. This spanning tree can be described by the following data structure:

9.2.5 **Definition (Schreier-tree)** Let G be a group acting on a finite set X. Let G be given by generators $s_0, \ldots, s_{r-1}$. Let $\mathcal{G} = (X, \mathcal{E})$ be the action graph for G acting on X. Let x be an element of X. A *Schreier-tree* for the orbit of x is a spanning tree for the connected component of $\mathcal{G}$ containing x. The tree is rooted at x and all edges are pointing away from x. ◇

We remark that a spanning tree for a connected component of a graph is in general not unique. For action graphs, this reflects the fact that there may be different ways to obtain a given element $y \in X$ as an image of x under group elements $g_1, g_2 \in G$. We will investigate these questions no further but we note, however, that the shape of the tree is important for performance considerations. For example, the average depth of a node should be small. There are special methods to build "shallow" Schreier-trees, see Seress [177]. The trick is to change the generating set S which is used for calculating the action graph beforehand.

The following basic orbit algorithm computes a Schreier-tree for the orbit of x under G. It uses a data structure called queue, which is similar to a waiting line. The new elements are appended to the end of the queue, and the elements are taken out in order. This means that the front-most element is processed first, then the second element and so forth until all elements are processed and the queue is empty.

Algorithm (orbits on points) 9.2.6

Input: A permutation group G acting on a finite set $X = \{x_1, \dots, x_n\}$, a generating set $S = \{s_0, \dots, s_{r-1}\}$ of G, a point $x \in X$.

Output: A Schreier-tree $T = (\mathcal{O}, \mathcal{E})$ for the orbit $\mathcal{O} = G(x)$.

```
let $Q$ be a queue holding the element $x$
let $\mathcal{O} := \{x\}, \mathcal{E} = \emptyset$, so that $T = (\{x\}, \emptyset)$ has only one node $x$
while $Q \neq \emptyset$ do
    let $y$ be the first element of $Q$ (remove $y$ from $Q$)
    for $i \in r$ do
        $z := ys_i$
        if $z \notin \mathcal{O}$ then
            append $z$ to $Q$, add $z$ to $\mathcal{O}$
            add the edge $(y, z)$ labeled by $s_i$ to $\mathcal{E}$
        end if
    end for
end while
```

□

Example Let G be the permutation group generated by 9.2.7

$$\begin{aligned}
s_0 &= (3,4)(9,14)(10,13)(11,12), \\
s_1 &= (3,9)(4,14)(10,11)(12,13), \\
s_2 &= (3,11)(4,12)(9,10)(13,14), \\
s_3 &= (2,3)(6,9)(7,10)(8,11), \\
s_4 &= (1,2)(5,6)(10,12)(11,13), \\
s_5 &= (0,1)(6,7)(9,10)(13,14).
\end{aligned}$$

The action-graph and a spanning Schreier-tree are shown in Fig. 9.2. It can be shown that $G \simeq \mathrm{PGL}_4(2)$. See also Examples 9.2.11,9.3.11 and 9.8.12 below. ◇

Let us now consider the problem of computing G_x, the stabilizer of x in G, for $x \in X$. The following result, due to Schreier, provides a set of generators for G_x, given generators for G.

Theorem (Schreier) *Let G be a finite group generated by a set of elements S. Let H be a subgroup of G and let $\mathcal{R}$ be a set of right coset representatives for H in G containing* 1. *For $r \in \mathcal{R}$ and $s \in S$, let $\overline{rs}$ be the unique element in $\mathcal{R}$ with $rs \in H\overline{rs}$. Then H is generated by all elements of the form $rs\overline{rs}^{-1}$, where $r \in \mathcal{R}$ and $s \in S$. Each such element is called a* Schreier-generator 9.2.8

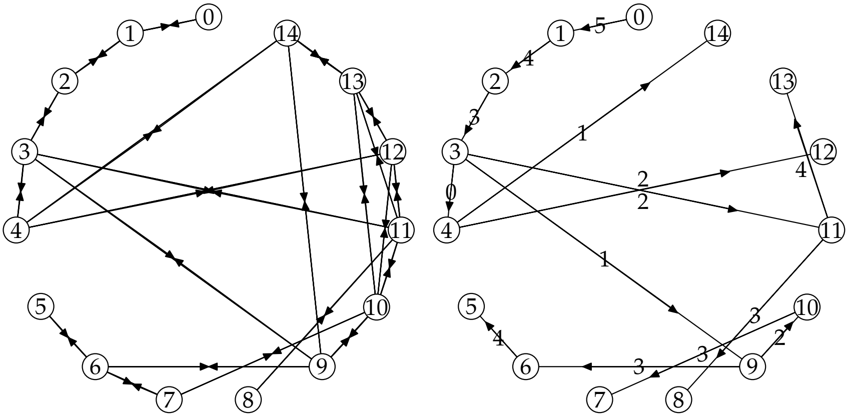


Fig. 9.2 Action-graph and Schreier-tree

Proof: The set $\mathcal{R}$ is a system of right coset representatives of H in G so that

$$G = \bigcup_{r \in \mathcal{R}} Hr.$$

We extend the function defined in the theorem to the whole group by letting $\overline{g}$, for $g \in G$, be the unique element in $\mathcal{R}$ with $g \in H\overline{g}$. Note that $\overline{hg} = \overline{g}$ if $h \in H$. Also, $\overline{\overline{g}} = \overline{g}$ for all $g \in G$. Finally, $\overline{g} = 1$ if and only if $g \in H$. Suppose $g = s_1 s_2 \cdots s_t \in G$ with each $s_i \in S$. Put

$$g_0 = 1,\ g_1 = s_1,\ g_2 = s_1 s_2,\ \ldots,\ g_t = s_1 s_2 \cdots s_t = g.$$

Write

$$u_0 = \overline{g_0} = 1,\ u_1 = \overline{g_1},\ \ldots,\ u_t = \overline{g_t} = \overline{g}.$$

Then

9.2.9 $$g u_t^{-1} = s_1 s_2 s_3 \cdots s_t u_t^{-1} = u_0 s_1 u_1^{-1} u_1 s_2 u_2^{-1} u_2 \cdots u_{t-1}^{-1} u_{t-1} s_t u_t^{-1},$$

which equals g if g is in H since then $u_t = \overline{g} = 1$. By definition of the function $g \mapsto \overline{g}$, we deduce from $\overline{g_{i-1}} = u_{i-1}$ that $g_{i-1} \in H u_{i-1}$. Hence there exists an element $h \in H$ with $g_{i-1} = h u_{i-1}$. Therefore $g_{i-1} s_i = h u_{i-1} s_i$, which implies $\overline{g_{i-1} s_i} = \overline{u_{i-1} s_i}$. It follows that for $i \geq 1$

$$u_i = \overline{u_i} = \overline{\overline{g_i}} = \overline{g_i} = \overline{g_{i-1} s_i} = \overline{u_{i-1} s_i},$$

which is an element of the form $\overline{rs}$ with $r \in \mathcal{R}$ and $s \in S$. Now let $g \in H$ and hence $u_t = 1$. Then 9.2.9 becomes

$$g = g u_t^{-1} = \prod_{i=1}^{t} u_{i-1} s_i u_i^{-1} = \prod_{i=1}^{t} u_{i-1} s_i \overline{u_{i-1} s_i}^{\,-1},$$

i.e. g can be written as a product of elements of the form $rs\overline{rs}^{-1}$, with $r \in \mathcal{R}, s \in S$. This finishes the proof. □

One particular instance of this is the computation of point stabilizers in permutation groups.

Corollary *Let the group G act on the finite set X and let S be a set of generators for G. For $x \in X$, let $\mathcal{R} = \{r_1, r_2, \dots, r_\ell\}$ with $r_1 = 1$ be a set of elements such that the following holds: For each $y \in G(x)$ there is one and only one element $r \in \mathcal{R}$ with $xr = y$ (and therefore $|G(x)| = \ell$). Then* 9.2.10

$$G_x = \langle rs\overline{rs}^{-1} \mid r \in \mathcal{R}, s \in S \rangle. \qquad \square$$

The last of the three problems is that of computing transporter elements g such that $yg = z$ (provided y and z are in the same G-orbit, say $G(x)$, of course). Such transporter elements can be computed from the Schreier-tree of the orbit. Let $T = (G(x), \mathcal{E})$, and let y and z be elements of the orbit. Following the edge labels along the path from x to y and from x to z, respectively, we obtain group elements u and v with $xu = y$ and $xv = z$. Then $yu^{-1}v = xv = z$, so that $u^{-1}v$ is a transporter element.

Example (continuation of Example 9.2.7) An element $g \in G$ mapping 6 to 13, for example, can be determined directly from the tree: 9.2.11

$$g = s_3^{-1} s_1^{-1} s_2 s_4 = (1, 2, 12, 14, 10, 7, 3)(4, 11, 8, 9, 5, 6, 13).$$

The reader should carefully note that this product of permutations has to be read from the left to the right, since in this chapter we prefer actions from the right. ◇

We need further notation concerning the solution of orbit type problems.

Definition (orbit data structure) Let G be a group which acts on the finite set X. The triple 9.2.12

$$\text{orbit}(G, X) = (\mathcal{T}, \sigma, \varphi) := (\mathcal{T}, \sigma, \varphi)$$

is the *orbit data* for G acting on X provided that

1. $\mathcal{T}$ is a transversal of the G-orbits on X,
2. $\sigma\colon X \to L(G) : x \mapsto G_x$,
3. $\varphi\colon X \to G : x \mapsto g$ with $xg \in \mathcal{T}$.

Here, $L(G)$ denotes the lattice of subgroups of G as defined in 3.4.4. We call σ the *stabilizer map* and φ the *transporter map.* ◇

9.2.13 **Remarks**

1. It follows from 3.4.1 that the image of the map φ is unique only modulo elements of the stabilizer.
2. The orbit data structure is not a purely mathematical object. The point with the maps φ and σ is that we should be able to compute function values efficiently. When we say that the orbit data $(\mathcal{T}, \sigma, \varphi)$ for G on X is available, we mean that we have determined $\mathcal{T}$ and are able to evaluate the maps σ and φ with reasonably small effort.
3. It may be that $\sigma(y)$ is known only for elements of the transversal $\mathcal{T}$. If this is the case, and if in addition we are able to compute transporter elements, then for a given x in X we compute $g = \varphi(x)$, $y = xg$ and $\sigma(y) = G_y$. It follows from 3.4.3 that
$$\sigma(x) = G_x = g^{-1}G_y g = g^{-1}\sigma(y)g. \qquad \diamond$$

Exercises

E.9.2.1 **Exercise** Let G be a group, generated by a set $S = \{s_0, \dots, s_{r-1}\}$ of generators. Then each $g \in G$ has an expression of the form $g = s_{i_0}^{\epsilon_0} s_{i_1}^{\epsilon_1} \dots s_{i_{r-1}}^{\epsilon_{r-1}}$ with $i_k \in r$ and $\epsilon_k \in \{\pm 1\}$. Show that if G is finite, we can find an expression for g of this form with $\epsilon_k = 1$ for $k \in r$.

9.3

9.3 A Permutation Representation

To get back to codes, let us start by enumerating the points of finite projective spaces. This allows us to translate the action of the general linear group from that of a matrix group to that of a permutation group. We will do the case of $\mathbb{F}_q^k$ first, and then move on to the projective case.

Assume that $\kappa_0, \kappa_1, \kappa_2, \dots, \kappa_{q-1}$ are the elements of the field $\mathbb{F}_q$, where we always require that $\kappa_0 = 0$ is the zero element and $\kappa_1 = 1$ is the unit element in the field.

We start by ranking the points in $\mathbb{F}_q^k = \{\sum_{i=0}^{k-1} v_i e^{(i)} \mid v_i \in \mathbb{F}_q\}$. Recall from 1.3.5 that for every integer $q \geq 2$ we have the base q expression of an integer $m \in q^k$
$$m = \sum_{i \in k} a_i q^i,$$
which we abbreviate as $m = (a_{k-1}, \dots, a_0)_q$.

Lemma *Let q be a prime power. Let $m \in q^k$ be an integer with $m = (a_{k-1}, \dots, a_0)_q$. The map* **9.3.1**

$$\mathrm{rk}_{k,q}^{-1} : q^k \to \mathbb{F}_q^k : m \mapsto (\kappa_{a_0}, \dots, \kappa_{a_{k-1}}), \quad \textbf{9.3.2}$$

is a bijection, we call it the unrank function *for $\mathbb{F}_q^k$. Its inverse*

$$\mathrm{rk}_{k,q} : \mathbb{F}_q^k \to q^k : (\kappa_{a_0}, \dots, \kappa_{a_{k-1}}) \mapsto m, \quad \textbf{9.3.3}$$

is the rank function *for $\mathbb{F}_q^k$.* □

The proof is straightforward.

Example Let $\mathbb{F}_3 = \{\kappa_0, \kappa_1, \kappa_2\}$, with $\kappa_0 = \overline{0} = 0$, $\kappa_1 = \overline{1} = 1$, and $\kappa_2 = \overline{2} = 2$. We obtain the following unrank function for $\mathbb{F}_3^2$. **9.3.4**

$$\begin{aligned} &\mathrm{rk}_{2,3}^{-1}(0) = (0,0),\ \mathrm{rk}_{2,3}^{-1}(1) = (1,0),\ \mathrm{rk}_{2,3}^{-1}(2) = (2,0),\\ &\mathrm{rk}_{2,3}^{-1}(3) = (0,1),\ \mathrm{rk}_{2,3}^{-1}(4) = (1,1),\ \mathrm{rk}_{2,3}^{-1}(5) = (2,1),\\ &\mathrm{rk}_{2,3}^{-1}(6) = (0,2),\ \mathrm{rk}_{2,3}^{-1}(7) = (1,2),\ \mathrm{rk}_{2,3}^{-1}(8) = (2,2). \end{aligned}$$

Correspondingly

$$\mathrm{rk}_{2,3}((0,0)) = 0,\ \mathrm{rk}_{2,3}(e^{(0)}) = 1,\ \mathrm{rk}_{2,3}(e^{(1)}) = 3, \dots \qquad \diamond$$

Let us turn our attention to the projective space $\mathrm{PG}_d(q)$. We want to enumerate (i.e. label) the set of one-dimensional subspaces $\langle v \rangle$ of $\mathbb{F}_q^{d+1}$, where $v \neq 0$. Recall from Section 3.7 that we denote the number of points of $\mathrm{PG}_d(q)$ by

$$\theta_d(q) = \frac{q^{d+1} - 1}{q - 1} = |\,\mathrm{PG}_d(q)| = q^d + q^{d-1} + \dots + q + 1.$$

In order to enumerate the points of a projective space $\mathrm{PG}_d(q)$, we are going to choose nonzero representatives out of each one-dimensional subspace of $V = \mathbb{F}_q^{d+1}$. Let $e^{(0)}, \dots, e^{(d)}$ be the standard basis of V. We introduce the following notation. For

$$u = \langle u_0 e^{(0)} + \dots + u_d e^{(d)} \rangle \in \mathrm{PG}_d(q),$$

let $\mathrm{lc}(u)$ be the largest index i for which $u_i \neq 0$ (and hence $u_{i+1} = \cdots = u_d = 0$). We call $\mathrm{lc}(u)$ the *leading coefficient* of u. Notice that this definition depends on the labeling of the basis vectors, which is intentional. To label the one-dimensional subspaces of V we need to pick one nonzero vector out of each such subspace. A simple way to do this is to take as representatives the vectors

$$u = (u_0, \dots, u_d) \in \mathbb{F}_q^{d+1}$$

whose rightmost nonzero coordinate is one, i.e. with $u_k = 1, u_{k+1} = \cdots = u_d = 0$, where $k = \mathrm{lc}(u)$. Such vectors are called *standard.*

There is one more condition which we pose but which seems a little unmotivated at this point. We require that the unit vectors and the all-one vector get the smallest possible ranks, i.e. we ask that

$$\begin{aligned}
\mathrm{rk}(\langle e^{(0)}\rangle) &= 0,\\
\mathrm{rk}(\langle e^{(1)}\rangle) &= 1,\\
&\vdots\\
\mathrm{rk}(\langle e^{(d)}\rangle) &= d,\\
\mathrm{rk}(\langle e^{(0)}+\ldots+e^{(d)}\rangle) &= d+1.
\end{aligned}$$

The reason for this requirement will become clear in Section 9.8, when we exhibit a special property of these vectors (namely, they form a "base" in the sense of Section 9.7).

The remaining vectors are of the form

$$u = (u_0,\ldots,u_{k-1},1,0,\ldots,0)$$

with $(u_0,\ldots,u_{k-1}) \in \mathbb{F}_q^k \setminus \{0\}$, where $k = \mathrm{lc}(u)$. If $k = d$ we also have that $(u_0,\ldots,u_{d-1}) \neq (1,\ldots,1)$. We decide to order these vectors first according to the value of k (which can take any value from 1 to d). Among the vectors u for a given $k = \mathrm{lc}(u)$ we order according to the ranks of $(u_0,\ldots,u_{k-1})$ as points in $\mathbb{F}_q^k$ as given by 9.3.3. We skip the zero vector which cannot occur. If $k = d$ we also need to skip the all-one vector. This requires some additional effort. We will shift the rank before we apply 9.3.2 and conversely we will also shift the rank after application of 9.3.3. The all-one vector – as an element of $\mathbb{F}_q^d$ – has rank

$$1+q+q^2+\ldots+q^{d-1} = \frac{q^d-1}{q-1} = \theta_{d-1}(q).$$

Therefore, we need to increase all ranks which are greater than or equal to this number by one before calling 9.3.2. Conversely, if we are ranking a vector u with $\mathrm{lc}(u) = d$, we need to decrease all ranks of $(u_0,\ldots,u_{d-1}) \in \mathbb{F}_q^d$ by one if they happen to be greater than $\theta_{d-1}(q)$. To facilitate this we will introduce a shift function. Summarizing, we have the following unrank and rank functions for the points of $\mathrm{PG}_d(q)$. We remark that the if clauses are to be read in order, that is, the second and all following if clauses are to be understood as "otherwise if."

9.3.5 **Lemma** *We define the unrank function* $\mathrm{rk}_{d;q}^{-1}\colon \theta_d(q) \to \mathrm{PG}_d(q)$ *by*

9.3.6
$$\mathrm{rk}_{d;q}^{-1}(m) = \begin{cases} \langle e^{(m)}\rangle & \text{if } m \le d,\\ \langle \sum_{i=0}^{d} e^{(i)}\rangle & \text{if } m = d+1,\\ \langle \mathrm{rk}_{d,1;q}^{-1}(m-d-1)\rangle & \text{otherwise,} \end{cases}$$

where

$$\mathrm{rk}_{d,k;q}^{-1}(m) = \begin{cases} \mathrm{rk}_{d,*;q}^{-1}(m) & \textit{if } k = d \\ e^{(k)} + \mathrm{rk}_{k,q}^{-1}(m) & \textit{if } m < q^k \\ \mathrm{rk}_{d,k+1;q}^{-1}(m - q^k + 1) & \textit{otherwise.} \end{cases} \tag{9.3.7}$$

Here,

$$\mathrm{rk}_{d,*;q}^{-1}(m) = e^{(d)} + \mathrm{rk}_{d,q}^{-1}\big(\mathrm{shift}_{\theta_{d-1}(q)}(m)\big) \tag{9.3.8}$$

with

$$\mathrm{shift}_j(m) := \begin{cases} m & \textit{if } m < j, \\ m+1 & \textit{otherwise.} \end{cases} \tag{9.3.9}$$

This map $\mathrm{rk}_{d;q}^{-1}$ *is a bijection. Its inverse is the* rank function *for* $\mathrm{PG}_d(q)$*, denoted as* $\mathrm{rk}_{d;q}$*. For a point* $\langle u \rangle$ *with* $u = (u_0, u_1, \ldots, u_d) \in \mathbb{F}_q^{d+1} \setminus \{0\}$ *one has* $\mathrm{rk}_{d;q}(\langle u \rangle) =$

$$\begin{cases} k & \textit{if } \langle u \rangle = \langle e^{(k)} \rangle \\ d+1 & \textit{if } \langle u \rangle = \langle 1, \ldots, 1 \rangle \\ d + 2 - k + q\theta_{k-2}(q) + \mathrm{rk}_{k,q}\left(\frac{u_0}{u_k}, \ldots, \frac{u_{k-1}}{u_k}\right) & \textit{if } k = \mathrm{lc}(u) < d \\ 2 + q\theta_{d-2}(q) + \mathrm{shift}_{\theta_{d-1}(q)}^{-1}\left(\mathrm{rk}_{d,q}\left(\frac{u_0}{u_d}, \ldots, \frac{u_{d-1}}{u_d}\right)\right) & \textit{if } \mathrm{lc}(u) = d. \end{cases} \tag{9.3.10}$$

□

Example We have $\theta_2(2) = 2^2 + 2 + 1 = 7$, $\theta_2(3) = 3^2 + 3 + 1 = 13$ and $\theta_3(2) = 2^3 + 2^2 + 2 + 1 = 15$. Table 9.1 shows the labeling of points of $\mathrm{PG}_2(2)$, $\mathrm{PG}_2(3)$ and $\mathrm{PG}_3(2)$. Let us see some specific examples. We have **9.3.11**

$$\begin{aligned} \mathrm{rk}_{3;2}^{-1}(4) &= \langle 1,1,1,1 \rangle \quad \text{by 9.3.6,} \\ \mathrm{rk}_{3;2}^{-1}(5) &= \langle \mathrm{rk}_{3,1;2}^{-1}(1) \rangle \quad \text{by 9.3.6} \\ &= \langle e^{(1)} + \mathrm{rk}_{1,2}^{-1}(1) \rangle \quad \text{by 9.3.7} \\ &= \langle e^{(1)} + e^{(0)} \rangle = \langle 1,1,0,0 \rangle \quad \text{by 9.3.2,} \\ \mathrm{rk}_{3;2}^{-1}(14) &= \langle \mathrm{rk}_{3,1;2}^{-1}(10) \rangle \quad \text{by 9.3.6} \\ &= \langle \mathrm{rk}_{3,2;2}^{-1}(9) \rangle \quad \text{by 9.3.7} \\ &= \langle \mathrm{rk}_{3,3;2}^{-1}(6) \rangle \quad \text{by 9.3.7} \\ &= \langle \mathrm{rk}_{3,*;2}^{-1}(6) \rangle \quad \text{by 9.3.7} \\ &= \langle e^{(3)} + \mathrm{rk}_{3,2}^{-1}(\mathrm{shift}_7(6)) \rangle \quad \text{by 9.3.7} \\ &= \langle e^{(3)} + \mathrm{rk}_{3,2}^{-1}(6) \rangle \quad \text{by 9.3.9} \\ &= \langle e^{(3)} + e^{(2)} + e^{(1)} \rangle = \langle 0,1,1,1 \rangle \quad \text{by 9.3.2,} \\ \mathrm{rk}_{2;3}^{-1}(12) &= \langle \mathrm{rk}_{2,1;3}^{-1}(9) \rangle \quad \text{by 9.3.6} \\ &= \langle \mathrm{rk}_{2,2;3}^{-1}(7) \rangle \quad \text{by 9.3.7} \\ &= \langle \mathrm{rk}_{2,*;3}^{-1}(7) \rangle \quad \text{by 9.3.7} \end{aligned}$$

Table 9.1 Labeling $\mathrm{PG}_2(2)$, $\mathrm{PG}_2(3)$ and $\mathrm{PG}_3(2)$

m	$\mathrm{rk}_{2;2}^{-1}(m)$	$\mathrm{rk}_{2;3}^{-1}(m)$	$\mathrm{rk}_{3;2}^{-1}(m)$
0	$\langle 1,0,0\rangle$	$\langle 1,0,0\rangle$	$\langle 1,0,0,0\rangle$
1	$\langle 0,1,0\rangle$	$\langle 0,1,0\rangle$	$\langle 0,1,0,0\rangle$
2	$\langle 0,0,1\rangle$	$\langle 0,0,1\rangle$	$\langle 0,0,1,0\rangle$
3	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0,1\rangle$
4	$\langle 1,1,0\rangle$	$\langle 1,1,0\rangle$	$\langle 1,1,1,1\rangle$
5	$\langle 1,0,1\rangle$	$\langle 2,1,0\rangle$	$\langle 1,1,0,0\rangle$
6	$\langle 0,1,1\rangle$	$\langle 1,0,1\rangle$	$\langle 1,0,1,0\rangle$
7		$\langle 2,0,1\rangle$	$\langle 0,1,1,0\rangle$
8		$\langle 0,1,1\rangle$	$\langle 1,1,1,0\rangle$
9		$\langle 2,1,1\rangle$	$\langle 1,0,0,1\rangle$
10		$\langle 0,2,1\rangle$	$\langle 0,1,0,1\rangle$
11		$\langle 1,2,1\rangle$	$\langle 1,1,0,1\rangle$
12		$\langle 2,2,1\rangle$	$\langle 0,0,1,1\rangle$
13			$\langle 1,0,1,1\rangle$
14			$\langle 0,1,1,1\rangle$

$$\begin{aligned}
&= \langle e^{(2)} + \mathrm{rk}_{2,3}^{-1}(\mathrm{shift}_4(7))\rangle \quad \text{by 9.3.7}\\
&= \langle e^{(2)} + \mathrm{rk}_{2,3}^{-1}(8)\rangle \quad \text{by 9.3.9}\\
&= \langle e^{(2)} + 2e^{(1)} + 2e^{(0)}\rangle = \langle 2,2,1\rangle \quad \text{by 9.3.2.}
\end{aligned}$$

Conversely, we have

$$\begin{aligned}
\mathrm{rk}_{3;2}(\langle 1,1,1,1\rangle) &= 4 \quad \text{by 9.3.10,}\\
\mathrm{rk}_{3;2}(\langle 1,1,0,0\rangle) &= 3+2-1+\frac{0}{1}+\mathrm{rk}_{1,2}\big((1)\big) \quad \text{by 9.3.10}\\
&= 4+1=5 \quad \text{by 9.3.3,}\\
\mathrm{rk}_{3;2}(\langle 0,1,1,1\rangle) &= 2+\frac{6}{1}+\mathrm{shift}_6^{-1}\big(\mathrm{rk}_{3,2}\big((0,1,1)\big)\big) \quad \text{by 9.3.10}\\
&= 8+\mathrm{shift}_7^{-1}(6) \quad \text{by 9.3.3,}\\
&= 8+6=14 \quad \text{by 9.3.9,}\\
\mathrm{rk}_{2;3}(\langle 2,2,1\rangle) &= 2+\frac{6}{2}+\mathrm{shift}_4^{-1}\big(\mathrm{rk}_{2,3}\big((2,2)\big)\big) \quad \text{by 9.3.10}\\
&= 5+\mathrm{shift}_4^{-1}(8) \quad \text{by 9.3.3}\\
&= 5+7=12 \quad \text{by 9.3.9.}
\end{aligned}$$

◇

Example Using the ranks of the previous example, the permutations of Examples 9.2.7 and 9.2.11 can be written as matrices. Recall that we use row-vector convention, i.e. the images of a linear map are written in the rows of the corresponding matrix. For instance, the elements of the generating set S can be written as matrices as follows. **9.3.12**

$$s_0 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 1&1&1&1 \end{pmatrix},$$

since $0s_0 = 0, 1s_0 = 1, 2s_0 = 2, 3s_0 = 4$ and $0,1,2,4$ are the ranks of the projective points which are represented by the vectors in the rows of this matrix. Similarly, we obtain

$$s_1 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 1&0&0&1 \end{pmatrix}, s_2 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 1&1&0&1 \end{pmatrix}, s_3 = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix},$$

$$s_4 = \begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&1 \end{pmatrix}, s_5 = \begin{pmatrix} 0&1&0&0\\ 1&0&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix}.$$

From this we see that the group of Example 9.2.7 really is a projective linear matrix group. That it is the full group $\mathrm{PGL}_4(2)$ will follow from a result in Section 9.8, where a special generating set ("strong generators") for this group is exhibited. The permutation

$$(1,2,12,14,10,7,3)(4,11,8,9,5,6,13)$$

of Example 9.2.11 is in fact the matrix

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

and we have that

$$\begin{aligned} \mathrm{rk}_{3;2}^{-1}(6) \cdot A &= \langle 1,0,1,0 \rangle \cdot \begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&0&1&1\\ 0&1&0&0 \end{pmatrix} \\ &= \langle 1,0,1,1 \rangle = \mathrm{rk}_{3;2}^{-1}(13). \end{aligned}$$

◇

9.4

9.4 The Lexicographical Order

Let $(X, \leq)$ be a totally ordered set. In this section, we are concerned with the set of subsets of X, also known as the *power set* of X. In addition to our customary notation 2^X, we will introduce the notation $\mathcal{P}(X)$ for this set. That is,

$$\mathcal{P}(X) = \{A \mid A \subseteq X\}.$$

Clearly, the size of $\mathcal{P}(X)$ is $2^{|X|}$. Later on, we will also consider the set of subsets of size k of X, for some nonnegative integer $k \leq |X|$. We denote this set as

$$\mathcal{P}_k(X) = \{A \mid A \subseteq X,\ |A| = k\}.$$

We introduce the following notation. For a subset A of a totally ordered set X we write $A = \{a_0, a_1 \ldots, a_{m-1}\}_<$ to indicate that the elements of A are listed in order, i.e. that $a_0 < a_1 < \cdots < a_{m-1}$. The set $\mathcal{P}(X)$ can be ordered in a very natural way, using the ordering of elements of X. This is the *lexicographical order* which has already appeared in 3.4.20.

9.4.1 **Definition (the lexicographical order)** For subsets $A = \{a_0, a_1, \ldots, a_{m-1}\}_<$ and $B = \{b_0, b_1, \ldots, b_{n-1}\}_<$ of the totally ordered set X we put

$$A \preceq B \iff \begin{cases} \exists\, r < \min(m, n) : a_i = b_i \text{ for } i \in r \text{ and } a_r < b_r, & \text{or} \\ m \leq n \text{ and } a_i = b_i \text{ for } i \in m. \end{cases}$$

◇

9.4.2 **Example** Let $X = \{a, b, \ldots, z\}$ be the Roman alphabet with the usual ordering of letters. Then $\mathcal{P}(X)$ is ordered lexicographically as follows (we leave out set brackets and commas for simplicity).

$$\begin{aligned} &\emptyset \prec a \prec ab \prec abc \prec \cdots \prec abc\ldots wxyz \prec abc\ldots wxz \\ &\quad \prec abc\ldots wy \prec abc\ldots wyz \prec abc\ldots wz \prec \cdots \\ &\quad \prec b \prec bc \prec \cdots \prec bcd\ldots xyz \prec \cdots \prec y \prec yz \prec z. \end{aligned}$$

◇

Let $(X, \leq)$ be a totally ordered finite set. The lexicographical order on $\mathcal{P}(X)$ can be represented by a tree, the *order tree* $T_{(X,\preceq)}$ or simply $T_\preceq$. The nodes of $T_\preceq$ are the subsets of X, i.e. the elements of the power set $\mathcal{P}(X)$. The edges of $T_\preceq$ can be described as follows. For subsets A and B (of a totally ordered set X), we say that A is a *prefix* of B if $A \subseteq B$ and either $A = B$ or $\min(B \setminus A) > \max A$. In other words, the prefixes of a set $B = \{b_0, b_1, \ldots, b_{m-1}\}_<$ are just the sets $\{b_0, \ldots, b_i\}$ for $i \leq m - 1$. If A is a prefix of B then we say that B is a *descendant* (or *offspring*) of A or that A is an ancestor of B. We say that B is an *immediate descendant* of A if B is a descendant of A and $|B| = |A| + 1$, i.e. $B = A \cup \{\max B\}$. Two nodes are *siblings* if they are immediate descendants of

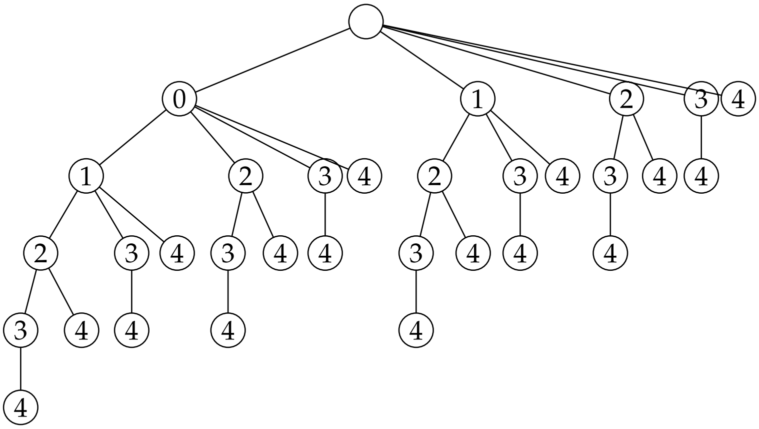


Fig. 9.3 Order tree of subsets of $\{0,1,2,3,4\}$

the same node. The edges of the tree $T_{\preceq}$ are between immediate descendants. One can think of these edges as being directed, pointing from the smaller to the larger set. In this sense, $T_{\preceq}$ is a rooted tree, with the empty set serving as root. A *leaf* is a node without descendants. An *inner node* is a node which is not a leaf. We say that a node has *distance i* from the root if the unique path from the root to that node has length i. The i-th *level* of the tree is the set of nodes at distance i from the root. A *common ancestor* of two sets A and B is an ancestor of both A and B. The *immediate common ancestor* of A and B is the ancestor of A and B which is largest in size. We arrange the siblings of a node according to their largest element, using the original ordering of elements of X. The siblings are drawn from left to right in increasing order. Thus $T_{\prec}$ is an ordered tree. In the Computer Science literature (cf. [42], for example) these trees are known as *binomial trees*. We also mention that it is ubiquitous in the Computer Science literature that trees grow top down. Thus the root node is the node on top of the drawing, whereas the leaves are the nodes at the bottom.

Example Consider the five element set $X := \{0,1,2,3,4\}_{<}$. Figure 9.3 displays the order tree $T_{(X,\preceq)}$. Here, we label the nodes by the largest element of the set which they represent (the root node is represented as an empty node). Clearly, the set corresponding to a node can be reconstructed by simply collecting all labels of nodes along the unique path from the root to the node. **9.4.3**

There are essentially two different ways of traversing the nodes of a tree. The *depth first search* strategy (or *pre-order traversal*) is to move down the tree, visiting a node and its offsprings recursively. This is done in such a way that the leftmost offspring and its whole branch is visited first. Then on the way back one moves to the right, visiting possible siblings and their whole subtrees recursively. This way all the siblings are dealt with before the procedure returns to its ancestor. One may imagine this procedure as follows. Think of the tree as a fence in the plane. Walk around that fence, starting from the root node in the direction to the leftmost offspring, keeping the fence to your left. A different way to visit the nodes of a tree is the *breadth first search* order. Here, the nodes at any given level are visited in order, starting from the root and going down to deeper levels.

The ordering of subsets is encoded in the tree. Namely, we encounter the sets in lexicographical order if we visit the nodes of the tree in depth first strategy. In the above example, the depth first search arranges the subsets of $\{0, \dots, 4\}$ in the following order, which is indeed lexicographical.

$$\begin{aligned} \emptyset &\prec 0 \prec 01 \prec 012 \prec 0123 \prec 01234 \prec 0124 \prec 013 \prec 0134 \\ &\prec 014 \prec 02 \prec 023 \prec 0234 \prec 024 \prec 03 \prec 034 \prec 04 \\ &\prec 1 \prec 12 \prec 123 \prec 1234 \prec 124 \prec 13 \prec 134 \prec 14 \\ &\prec 2 \prec 23 \prec 234 \prec 24 \prec 3 \prec 34 \prec 4. \end{aligned}$$

Using breadth first search, the nodes of the tree will be visited in the following order:

$$\begin{array}{l} \emptyset, \\ 0, 1, 2, 3, 4, \\ 01, 02, 03, 04, 12, 13, 14, 23, 24, 34, \\ 012, 013, 014, 023, 024, 034, 123, 124, 134, 234, \\ 0123, 0124, 0134, 0234, 1234, \\ 01234. \end{array}$$

◇

Let us collect fundamental properties of the order tree in the following lemma. The proofs are straightforward and therefore omitted.

9.4.4 **Lemma** *Let $X = \{x_0, x_1, \dots, x_{n-1}\}_<$ be a finite totally ordered set. Let $\preceq$ be the lexicographical order on $\mathcal{P}(X)$. Then the order tree $T_\preceq$ has the following properties.*

1. *For every node of the tree, the corresponding set is the union of the labels along the path leading to that node. Moreover, the labels are encountered in ascending order along this path.*

2. *The nodes at level i correspond to i-subsets of X, and hence there are $\binom{n}{i}$ of them.*

3. *For $A, B \subseteq X$, a common ancestor of A and B corresponds to a prefix of $A \cap B$ and vice-versa. The immediate common ancestor is the prefix of $A \cap B$ which is largest in size.*

4. *The tree has 2^{n-1} leaves corresponding to the subsets of X which contain x_{n-1}. The tree has 2^{n-1} inner nodes corresponding to the subsets of X which do not contain x_{n-1}.*

5. *The subtree rooted at a set $A \subseteq X$ consists of the subsets $B \subseteq X$ for which A is a prefix of B. If $\max A = x_i$, there are 2^{n-1-i} such nodes. In particular, the subtree whose root is $\{x_i\}$ (i.e. the tree which is rooted at the i-th descendant of the global root), contains all subsets $A \subseteq X$ with $\min A = x_i$. There are 2^{n-1-i} such sets.*

6. *Two subtrees rooted at sets A and B (with $A, B \subseteq X$) are equal in shape and labeling of the nodes if and only if $\max A = \max B$.*

7. *If the order tree is traversed in depth first search, the subsets are encountered in lexicographic order. That is, a subset A precedes a subset B in the lexicographical order if and only if A is reached first when traversing the tree $T_{(X,\preceq)}$ in depth first search. In terms of the common ancestor C of A and B we can say that $A \preceq B$ if and only if either $C = A$ or $\min A \setminus C < \min B \setminus C$. That is, $A \preceq B$ if and only if either B is a descendant of A or the branch containing B is to the right of the branch containing A among the siblings of the immediate common ancestor of A and B.*

8. *Let A be a subset with $\max A = x_i$ (put $i = -1$ if $A = \emptyset$). The leftmost leaf in the subtree rooted at A is the set $A \cup \{x_{i+1}, \dots, x_{n-1}\}$. The rightmost leaf in the subtree rooted at A is the set $A \cup \{x_{n-1}\}$.*

9. *Let A be a subset with $\max A = x_i$ (put $i = -1$ if $A = \emptyset$). The subtree rooted at A contains exactly $\binom{n-1-i}{k-|A|}$ sets of size k.* □

It is of course useful to have rank and unrank functions for the set of subsets of a finite set.

Lemma *Let $X = \{x_0, x_1, \dots, x_{n-1}\}_<$ be a totally ordered finite set of n elements. Define a function, the rank function, from $\mathcal{P}(X)$ to the set of integers 2^n as follows. For a set $A \subseteq X$ define* **9.4.5**

$$\mathrm{rk}_X \colon \mathcal{P}(X) \to 2^n : A \mapsto \begin{cases} 0 & \text{if } A = \emptyset, \\ |A| + \sum_{\substack{x_i \in X \setminus A \\ x_i < \max A}} 2^{n-1-i} & \text{otherwise.} \end{cases}$$

This function is one-to-one and onto. Its inverse is the unrank function, defined as

$$\mathrm{rk}_X^{-1}(r) := \mathrm{rk}_X^{-1}(r, 0),$$

where

$$\mathrm{rk}_X^{-1}(r, m) := \emptyset \quad \text{if } r = 0,$$

while for $0 < r < 2^{n-m}$ *we have*

$$\mathrm{rk}_X^{-1}(r,m) := \begin{cases} \{x_m\} \cup \mathrm{rk}_X^{-1}(r-1, m+1) & \text{if } 2^{n-1-m} \geq r, \\ \mathrm{rk}_X^{-1}\left(r - 2^{n-1-m}, m+1\right) & \text{if } 2^{n-1-m} < r. \end{cases}$$

□

9.4.6 **Example** For $X = \{0, \ldots, 4\}$ as above, we have

$$\mathrm{rk}_X(\{1,3,4\}) = 3 + 2^{5-1-0} + 2^{5-1-2} = 23,$$

which can be verified by counting the nodes in Fig. 9.3. The tree rooted at $\{0\}$ has 16 nodes, and the tree rooted at $\{1,2\}$ brings in another 4 nodes, so that the set $\{1,3,4\}$ has indeed rank 23. On the other hand, we have

$$\begin{aligned} \mathrm{rk}_X^{-1}(23) &= \mathrm{rk}_X^{-1}(23,0) \\ &\overset{16\leq 23}{=} \mathrm{rk}_X^{-1}(7,1) \\ &\overset{8\geq 4}{=} \{1\} \cup \mathrm{rk}_X^{-1}(6,2) \\ &\overset{4\leq 6}{=} \{1\} \cup \mathrm{rk}_X^{-1}(2,3) \\ &\overset{2\geq 2}{=} \{1,3\} \cup \mathrm{rk}_X^{-1}(1,4) \\ &\overset{1\geq 1}{=} \{1,3,4\} \cup \mathrm{rk}_X^{-1}(0,5) \\ &= \{1,3,4\}, \end{aligned}$$

which is the original set again. ◇

Sometimes we are only interested in the set $\mathcal{P}_k(X)$ of k-subsets of X, where k is some fixed integer with $0 \leq k \leq |X|$. The elements of $\mathcal{P}_k(X)$ can be ranked and unranked as well.

9.4.7 **Lemma** *Let* $X = \{x_0, x_1, \ldots, x_{n-1}\}_<$ *be a totally ordered finite set of n elements. Let k be an integer with* $0 \leq k \leq n$. *Define a function, the rank function of* $\mathcal{P}_k(X)$ *to the set of integers* $\binom{n}{k}$ *as follows. For a k-subset* $A = \{x_{a_0}, x_{a_1}, \ldots, x_{a_{k-1}}\}_<$, *put*

$$\mathrm{rk}_{X,k} \colon \mathcal{P}_k(X) \to \binom{n}{k} : A \mapsto \sum_{i=0}^{k-1} \sum_{j=a_{i-1}+1}^{a_i-1} \binom{n-1-j}{k-1-i},$$

where $a_{-1} := -1$. *The function* $\mathrm{rk}_{X,k}$ *is one-to-one and onto. Its inverse is the function* $\mathrm{rk}_{X,k}^{-1}$, *which is given by*

$$\mathrm{rk}_{X,k}^{-1}(r) = \mathrm{rk}_{X,k}^{-1}(r,0),$$

where

$$\mathrm{rk}_{X,k}^{-1}(r,m) := \emptyset \quad \text{if } k = 0,$$

whereas for $k > 0$

$$\mathrm{rk}_{X,k}^{-1}(r,m) = \begin{cases} \{x_m\} \cup \mathrm{rk}_{X,k-1}^{-1}(r,m+1) & \text{if } \binom{n-1-m}{k-1} > r, \\ \mathrm{rk}_{X,k}^{-1}\left(r - \binom{n-1-m}{k-1}, m+1\right) & \text{if } \binom{n-1-m}{k-1} \leq r. \end{cases}$$

□

Example For $X = \{0, \dots, 4\}$ as above, we have **9.4.8**

$$\mathrm{rk}_{X,3}(\{1,3,4\}) = \binom{5-1-0}{3-1-0} + \binom{5-1-2}{3-1-1} = \binom{4}{2} + \binom{2}{1} = 8,$$

which can of course be verified by counting nodes in Fig. 9.3. The tree rooted at $\{0\}$ contains six 3-subsets, and the tree rooted at $\{1,2\}$ brings in another two 3-subsets, so that the set $\{1,3,4\}$ has rank $6 + 2 = 8$. On the other hand, we have

$$\begin{aligned} \mathrm{rk}_{X,3}^{-1}(8) &= \mathrm{rk}_{X,3}^{-1}(8,0) \\ &\overset{\binom{5-1-0}{3-1}=6\leq 8}{=} \mathrm{rk}_{X,3}^{-1}(2,1) \\ &\overset{\binom{5-1-1}{3-1}=3>2}{=} \{1\} \cup \mathrm{rk}_{X,2}^{-1}(2,2) \\ &\overset{\binom{5-1-2}{2-1}=2\leq 2}{=} \{1\} \cup \mathrm{rk}_{X,2}^{-1}(0,3) \\ &\overset{\binom{5-1-3}{2-1}=1>0}{=} \{1,3\} \cup \mathrm{rk}_{X,1}^{-1}(0,4) \\ &\overset{\binom{5-1-4}{1-1}=1>0}{=} \{1,3,4\} \cup \mathrm{rk}_{X,0}^{-1}(0,5) \\ &= \{1,3,4\}, \end{aligned}$$

which is the original set again. ◇

Exercises

Exercise Compute the rank of $A = \{2,3,5,7\}$ as a subset of $\{0, \dots, 7\}$. **E.9.4.1**

Exercise Compute $\mathrm{rk}_X^{-1}(99)$ where $X = \{0, \dots, 7\}$. **E.9.4.2**

Exercise Compute the rank of $A = \{2,3,5,7\}$ as a 4-subset of $\{0, \dots, 7\}$. **E.9.4.3**

Exercise Compute $\mathrm{rk}_{X,4}^{-1}(66)$ where $X = \{0, \dots, 7\}$. **E.9.4.4**

E.9.4.5 **Exercise** If $X = \{apple, orange, pear, potato, banana, mango, lemon\}_<$, compute

1. $\mathrm{rk}_X(\{orange, potato, mango\})$,
2. $\mathrm{rk}_{X,3}(\{orange, potato, mango\})$,
3. $\mathrm{rk}_X^{-1}(79)$ and
4. $\mathrm{rk}_{X,3}^{-1}(27)$.

9.5

9.5 Orderly Generation of Codes

In order to construct linear codes, we need to direct attention to the technique of *orderly generation* of discrete structures. A discrete structure is simply a type of object which can be defined as an orbit of a group acting on a finite set. Examples in Combinatorics are graphs, codes, designs etc. When we speak of the construction of objects, we mean that we produce one object out of each isomorphism class. This object is called the representative, or the labeled object. In the 1970s, the technique of orderly generation has been invented independently by Read [165] and Faradžev [51, 52, 53] for the construction of graphs. The name comes from the fact that it generates representatives for the orbits in question in lexicographic order. A more refined version is described by McKay [146], who also presents an extensive literature list. McKay broadens the technique to general structures and introduces the concept of a canonical extension.

In the following, we will first discuss the technique of orderly generation in some detail and then come back to linear codes later. We start with an action of a group G on a finite set X, whose elements we call points. The group G also acts on subsets of X, via

$$\mathcal{P}(X) \times G \to \mathcal{P}(X) \colon (R, g) \mapsto Rg = \{xg \mid x \in R\}.$$

We call this the *induced action of G on* $\mathcal{P}(X)$. The *setwise stabilizer* of a set $R \subseteq X$ is the subgroup

$$G_R := \{g \in G \mid Rg = R\} = \{g \in G \mid \forall\, r \in R : \; rg \in R\}.$$

A related concept is the *pointwise stabilizer* of a set $R = \{r_0, \dots, r_{s-1}\}$, which is the subgroup

$$G_{r_0,\dots,r_{s-1}} := \{g \in G \mid r_i g = r_i \text{ for all } i \in s\} = \bigcap_{i \in S} G_{r_i}.$$

Occasionally, we will consider groups which are of mixed type. For instance, if we wish to stabilize the set R setwise, and in addition fix the point x, then we will write

$$G_{R,x} = G_R \cap G_x = \{g \in G \mid Rg = R, \text{ and } xg = x\}.$$

Here, the point x may or may not be a member of the set R. Another case is when the set R is enlarged by one further element x outside of R. The setwise stabilizer of $R \cup \{x\}$ is denoted as

$$G_{R\cup\{x\}} = \{g \in G \mid \forall r \in R : \ rg \in R \cup \{x\} \text{ and } xg \in R \cup \{x\}\,\}.$$

We would like to compute the orbits of G on the set of subsets of the finite set $X = \{x_0, \ldots, x_{n-1}\}_<$. The following problems arise.

1. Compute a *transversal* $\mathcal{T}$ for the G-orbits on subsets of X, which is a set of subsets of X such that
   (a) each orbit of G on $\mathcal{P}(X)$ is represented by one subset in $\mathcal{T}$, and
   (b) no such orbit is represented twice.
   The elements of the transversal are called *orbit representatives.*
2. For $S \subseteq X$, compute $\sigma(S) = G_S = \{g \in G \mid Sg = S\}$, the setwise stabilizer of S in G.
3. For $S \subseteq X$, determine an element $\varphi(S) = g \in G$ which maps S to its orbit representative in $\mathcal{T}$, i.e. a transporter element (such an element might not be unique).

Of course, in many applications one is not interested in the totality of subsets. Instead, often one has restrictions coming from the particular problem one is interested. This means that we are only interested in a subset of $\mathcal{P}(X)$, or even subsets of

$$\mathcal{P}_i(X) = \{S \subseteq X \mid |S| = i\},$$

the set of subsets of size i. To formalize this idea, we may indicate this condition by a function

$$f \ : \ \mathcal{P}(X) \to \{0,1\}, \ S \mapsto f(S)$$

where $f(S)$ is one if and only if the set S is *admissible*, i.e. satisfies the condition. We require that the condition is invariant under the action of the group, i.e. that

$$f(S) = f(Sg) \quad \forall\, g \in G, \,\forall\, S \subseteq X. \tag{9.5.1}$$

Also, we require that the condition is *hereditary*, i.e. that

$$f(S) = 1 \ \Rightarrow \ f(T) = 1 \quad \forall\, T \subseteq S \subseteq X. \tag{9.5.2}$$

In the following, we will assume that such a function $f : \mathcal{P}(X) \to \{0,1\}$ has been defined. This is no restriction as one can always define $f(S) = 1$ for all $S \subseteq X$. If f is such a test-function, we may restrict the action of G to the set

$$\mathcal{P}^{(f)}(X) := \mathcal{P}(X) \cap f^{-1}(\{1\}) = \{S \in \mathcal{P}(X) \mid f(S) = 1\}$$

or to one of the sets

$$\mathcal{P}_i^{(f)}(X) := \mathcal{P}_i(X) \cap f^{-1}(\{1\}) = \{S \in \mathcal{P}_i(X) \mid f(S) = 1\}.$$

There are many different ways to choose a transversal. One particular is the *canonical transversal*. It consists of *canonical orbit representatives*, which are the sets $R \subseteq X$ with

$$R \preceq Rg \quad \text{for all } g \in G.$$

Each orbit $G(S)$, $S \subseteq X$ is represented in this transversal by its least element,

9.5.3
$$\overline{S} = \min_{R \in G(S)} R = \min_{g \in G} Sg,$$

where the minimum is taken with respect to the lexicographical order. The function which takes a set S to its canonical orbit representative $\overline{S}$ can be thought of as a projection map. It satisfies the property that $\overline{\overline{S}} = \overline{S}$. The image of this function is the canonical transversal

$$\mathcal{T} = \{\overline{S} \mid S \in \mathcal{P}(X)\}.$$

It consists of the canonical subsets.

9.5.4 **Lemma** *Let X be a totally ordered finite set, and let G be a group acting on X. Let A be a canonical subset of X in the sense of 9.5.3. Then every prefix B of A is also canonical.*

Proof: Let $A = \overline{A} = \{a_0, a_1, \dots, a_{n-1}\}_<$ be a canonical subset of X. Let $B = \{a_0, \dots, a_{m-1}\}$ with $m - 1 \leq n - 1$ be a prefix of A. Assume that B is not canonical. Thus there exists an element $g \in G$ with $Bg = C = \{c_0, \dots, c_{m-1}\} \preceq B$ and $C \neq B$. Since $|C| = |B|$ it must be the case that there exists $r < m$ with $c_i = a_i$ for $i \in r$ and $c_r < a_r$. Also $Ag = C \cup \{a_m g, \dots, a_{n-1} g\}$. In order to compare Ag with A in the lexicographical order, put

$$d = \min_{i=m}^{n-1} a_i g.$$

If $d > c_r$ then $Ag = \{c_0, c_1, \dots, c_r, \dots\}_<$ and therefore $Ag \preceq A$ but $Ag \neq A$, contradicting the fact that A is canonical. Otherwise, let s be the least index such that $d < c_s$. Then $Ag = \{c_0, c_1, \dots, c_{s-1}, d, \dots\}_<$ and because $c_i = a_i$ for $i \in s$ and $d < c_s = a_s$, again we have the contradiction that $Ag \preceq A$ but $Ag \neq A$. We conclude that the assumption was incorrect and thus B is canonical. □

The method of orderly generation looks at all extensions of the form

$$S \cup \{x\},$$

called *extension sets.* Here, S is a member of the transversal of i-subsets and x is in $X \setminus S$. In fact, one requires that x is the least element in its G_S-orbit and that $x > \max S$. Then, one employs a test for whether a given set $S \subseteq X$ is canonical. Such a test is not easy to provide, as it involves a systematic search over the whole group G, to test whether the set Sg is lexicographically less than S for any given $g \in G$. If no Sg precedes S in the lexicographic order, S is canonical and will be output by the algorithm. The automorphism group of S is just the set of all elements $g \in G$ for which $Sg = g$, so

$$G_S = \{g \in G \mid Sg = S\}$$

can be computed at the same time. Of course, this backtrack procedure can be refined. One would try to avoid looking at every group element $g \in G$. This can be done by taking into account the subgroup structure of G. In fact, the automorphism group will be constructed by successively extending the known part of the group with new automorphisms found during the search. We omit the details here. The algorithm orderly generation can be summarized as follows. We do not state this algorithm as a theorem since we do not prove its correctness. Nevertheless, we mention that correctness can be proved using 9.5.4. We define

$$\mathcal{T}_{\leq i} = \bigcup_{j=0}^{i} \mathcal{T}_j.$$

Algorithm (orderly generation) **9.5.5**

Input: A group G acting on a set $X_<$, a test-function f, an integer i

Output: $\mathcal{T}_{\leq i}$, the canonical transversal for the G-orbits on admissible sets of size $\leq i$.

```
if $f(\emptyset) = 1$ then scan$(\emptyset, G)$ end if
end
```

Where the function scan is defined as follows.

```
scan$(S, A)$
compute $\mathcal{T}_S$, the canonical transversal of the $A$-orbits on $X \setminus S$.
    for each $x \in \mathcal{T}_S$ do
        if $x > \max S$ then
            if $f(S \cup \{x\}) = 1$ then
```

```
                **if** $S \cup \{x\}$ is canonical **then**
                    print $S \cup \{x\}$
                    **if** $|S| + 1 < i$ **then**
                        scan$(S \cup \{x\}, G_{S \cup \{x\}})$
                    **end if**
                **end if**
            **end if**
        **end if**
    **end for**
**end**
```

□

As already mentioned, testing whether a given set is the lexicographically least set among its G-orbit is a hard problem. It is actually easier to drop the requirement that the canonical element is the least among its orbit and replace it by some other kind of canonical form. This is McKay's variant. It relies on a function φ such that

$$R\varphi(R) = S\varphi(S) \quad \text{whenever} \quad R \sim_G S. \tag{9.5.6}$$

Such a function φ can be realized by a "partition backtrack" algorithm (cf. Leon's series of articles [128, 129, 130]). In addition, this algorithm computes the set-stabilizer of the set in question. If such a map φ is to be used for the orderly generation of orbits, the "scan" algorithm needs to change. This is because an extension $S \cup \{x\}$ where x is smallest among its G_S-orbit is not necessarily canonical with respect to the function φ. Also, the requirement that $x > \max S$ must be dropped. To make things work, one introduces another function

$$m \,:\, \mathcal{P}(X) \to X, \tag{9.5.7}$$

satisfying the two conditions

1. $m(R) \in R$, and
2. $m(Rg) \sim_{G_{Rg}} m(R)g$.

Such a function m is easily defined in terms of the map φ. For instance, one can take

$$m(R) = \Big(\min R\varphi(R)\Big)\varphi(R)^{-1}.$$

To see that this works, we argue as follows. It is clear that $m(R) \in R$. Since $\varphi(R)\varphi(Rg)^{-1}$ maps R to Rg, we deduce from 3.4.1 that there exists an element $h \in G_{Rg}$ such that

$$\varphi(R)\varphi(Rg)^{-1} = gh.$$

We conclude that

$$\begin{aligned} m(Rg) &= \Big(\min Rg\varphi(Rg)\Big)\varphi(Rg)^{-1} \\ &= \Big(\min R\varphi(R)\Big)\varphi(Rg)^{-1} \\ &= m(R)\varphi(R)\varphi(Rg)^{-1} \\ &= m(R)gh, \end{aligned}$$

which shows that $m(Rg)$ is in the same G_{Rg}-orbit as $m(R)g$.

We may summarize this algorithm as

Theorem (McKay [146]) *Let G act on the finite set $X_<$. Let $f : \mathcal{P}(X) \to \{0,1\}$ be a test-function on X which is G-invariant and hereditary (in the sense of 9.5.1 and 9.5.2). Let φ and m be functions as in 9.5.6 and 9.5.7, respectively. Then for any given integer $i \leq |X|$, Algorithm 9.5.9 computes a transversal $\mathcal{T}_{\leq i}$ of the orbits of G on admissible subsets of X of size at most i together with the corresponding stabilizers in G.* **9.5.8**

Algorithm (orderly generation by canonical augmentation) **9.5.9**

Input: A group G acting on a set $X_<$, a test-function f, an integer i, functions φ and m as in 9.5.6 and 9.5.7, respectively.

Output: $\mathcal{T}_{\leq i}$, a transversal for the G-orbits on admissible sets of size $\leq i$.

```
if $f(\emptyset) = 1$ then scan$(\emptyset, G)$ end if
end
```

Where the function scan is defined as follows.

```
scan$(S, A)$
  compute $\mathcal{T}_S$, a transversal of the $A$-orbits on $X \setminus S$.
  for each $x \in \mathcal{T}_S$ do
      if $f(S \cup \{x\}) = 1$ then
          compute $y := m(S \cup \{x\})$ and $B := G_{S\cup\{x\}}$
              if $x \sim_B y$ then
                  print $S \cup \{x\}$
                  if $|S| + 1 < i$ then
                      scan$(S \cup \{x\}, B)$
                  end if
              end if
          end if
    end if
```

```
(15) end for
(16) end                                                          □
```

Proof: We proceed by induction on j, the size of the subsets under consideration. If $j = 0$, the algorithm outputs $\emptyset$ and $G_\emptyset = G$, provided that $f(\emptyset) = 1$. Let us assume that $\mathcal{T}_i$, a transversal for the G-orbits on $\mathcal{P}_i^{(f)}(X)$ is computed correctly (together with the corresponding stabilizers in G). We need to show that each G-orbit on $(i+1)$-subsets is represented exactly once in the output of the algorithm. We proceed in two steps.

At first, we claim that each G-orbit on admissible $(i+1)$-subsets is represented *at least once* in the output. To see this, let R be an admissible $(i+1)$-subset of X. Since f is hereditary, the subset $R \setminus \{m(R)\}$ is again an admissible i-subset. By induction hypothesis, there exists an element $g \in G$ such that

$$(R \setminus \{m(R)\})g = S \in \mathcal{T}_i.$$

We define

$$z := m(R)g \in X \setminus S.$$

Since $\mathcal{T}_S$ is a transversal of the G_S-orbits on $X \setminus S$, there exists an element $h \in G_S$ such that

$$zh = x \in \mathcal{T}_S.$$

We conclude that

$$\begin{aligned} Rgh &= (R \setminus \{m(R)\})gh \cup \{m(R)gh\} \\ &= Sh \cup \{zh\} \\ &= S \cup \{x\} \end{aligned}$$

and $S \cup \{x\}$ is one of the extensions considered in lines (5)-(9). Since

$$y = m(S \cup \{x\}) = m(Rgh) \sim_{G_{S\cup\{x\}}} m(R)gh = zh = x,$$

the extension $S \cup \{x\}$ is accepted in line (7).

Secondly, we claim that each G-orbit on admissible $(i+1)$-subsets is represented *at most once* in the output. Assume the contrary. Let $R \sim_G S$ be two admissible $(i+1)$-subsets computed by the algorithm. Then

$$R = U \cup \{x\}, \quad S = V \cup \{y\}$$

with $U, V \in \mathcal{T}_i$, $x \in \mathcal{T}_U$, $y \in \mathcal{T}_V$. In addition, we know that

$$x \sim_{G_R} m(R), \quad \text{and} \quad y \sim_{G_S} m(S),$$

since $U \cup \{x\}$ and $V \cup \{y\}$ must both have been accepted in line (7) of the algorithm. Since $R \sim_G S$, there exists an element $g \in G$ such that $Rg = S$. Thus $G_S = g^{-1}G_R g$. Since $x \sim_{G_R} m(R)$, there exists an element $r \in G_R$ such that $xr = m(R)$, and so

$$xg(g^{-1}rg) = m(R)g,$$

i.e.

$$xg \sim_{G_S} m(R)g,$$

since $g^{-1}rg \in g^{-1}G_R g = G_S$. Thus

$$y \sim_{G_S} m(S) = m(Rg) \sim_{G_S} m(R)g \sim_{G_S} xg \in S.$$

This means that there exists an element $h \in G_S$ with

$$y = xgh.$$

Thus

$$Ugh = (R \setminus \{x\})gh = (Sg^{-1} \setminus \{x\})gh = Sh \setminus \{xgh\} = S \setminus \{y\} = V,$$

i.e. $U \sim_G V$. But U and V are elements of the transversal $\mathcal{T}_i$, and by induction hypothesis, this transversal contains exactly one representative of each G-orbit. It follows that

$$U = V,$$

and therefore $x \neq y$ (since $R \neq S$ by assumption). Thus

$$Ugh = V = U,$$

i.e. $gh \in G_U$. From $xgh = y$ we conclude that $x \sim_{G_U} y$, which is a contradiction to the fact that the algorithm considers in line (4) only representatives $x \in \mathcal{T}_U$ of the U-orbits on $X \setminus U$. The assumption must be wrong and the claim is proved. This finishes the proof of the fact that algorithm 9.5.9 is correct. □

Let us return to the problem of computing isometry classes of linear codes. Given a length n, a dimension k and a lower bound $d_{\min}$ for the minimum distance, let us now construct all (n,k)-codes over some finite field $\mathbb{F}_q$ whose minimum distance is at least $d_{\min}$ where $d_{\min} \geq 3$. From 9.1.5 we deduce the following. Depending on whether we want to compute linear or semilinear isometry classes, we are interested in the orbits of $G = \mathrm{PGL}_{n-k}(q)$ or $G = \mathrm{P\Gamma L}_{n-k}(q)$ on

$$\mathcal{P}_n(X), \quad X = \mathrm{PG}_{n-k-1}(q),$$

respectively.

It remains to take the prescribed minimum distance $d_{\min}$ into account. In order to apply 9.1.8, we need to check whether the n-subset of $\mathrm{PG}_{n-k-1}(q)$

under consideration has the property that any $d_{\min}-1$ points are independent. If $S \subseteq \mathrm{PG}_{n-k-1}(q)$ is a set of size n, we put

$$f(S) = \begin{cases} 1 & \text{if any } d_{\min}-1 \text{ points of } S \text{ are independent,} \\ 0 & \text{otherwise.} \end{cases}$$

This function f is our test-function. We need to check if the function f satisfies the requirements listed in 9.5.1 and 9.5.2. Since both groups $\mathrm{PGL}_{n-k}(q)$ and $\mathrm{P\Gamma L}_{n-k}(q)$ preserve the linear structure of $\mathbb{F}_q^{n-k}$, the condition about linear independence of subsets is invariant under the action. It is clear that the condition is hereditary. Let us put

$$\Upsilon_{n,k,d_{\min},q} = \mathcal{P}_n^{(f)}(\mathrm{PG}_{n-k-1}(q)) = \Big\{ S \subseteq \mathrm{PG}_{n-k-1}(q) \,\Big|\, |S| = n,\ f(S) = 1 \Big\}.$$

The next result shows a connection between canonical orbit representatives and systematic generator matrices.

9.5.10 **Lemma** *Consider the action of $G \geq \mathrm{PGL}_k(q)$ on n-subsets of points of $X = \mathrm{PG}_{k-1}(q)$. Let X be totally ordered according to 9.3.5. Let*

$$A := \{\langle u^{(0)}\rangle, \ldots, \langle u^{(n-1)}\rangle\}_<$$

be a canonical orbit representative. Let

$$\Gamma(A) = \left(u^{(0)^\top} \,\middle|\, \cdots \,\middle|\, u^{(n-1)^\top} \right)$$

be the generator matrix corresponding to A. Then the following conditions are equivalent.

1. *The rank of the matrix $\Gamma(A)$ is r.*
2. *$\langle u^{(i)}\rangle = \langle e^{(i)}\rangle$, for $i \in r$ and $u^{(j)} \in \langle e^{(0)}, \ldots, e^{(r-1)}\rangle$ for $j = r, \ldots, n-1$.*
3. *$\langle u^{(i)}\rangle = \langle e^{(i)}\rangle$, for $i \in r$ and $\langle u^{(r)}\rangle \neq \langle e^{(r)}\rangle$.*

Proof:

1. $\Rightarrow$ 2.: Since G is transitive on r-dimensional subspaces, the rank condition implies that the orbit of A contains an element $B = \{\langle e^{(0)}\rangle, \ldots, \langle e^{(r-1)}\rangle, \ldots\}$. But $\{\langle e^{(0)}\rangle, \ldots, \langle e^{(r-1)}\rangle\}$ is the lexicographically least set of size r. Hence A contains this set, i.e.

$$A = \{\langle e^{(0)}\rangle, \ldots, \langle e^{(r-1)}\rangle, \langle u^{(r)}\rangle, \ldots, \langle u^{(n-1)}\rangle\},$$

and – also by the rank condition – each column of $\Gamma(A)$ is in the span of these vectors:

$$u^{(j)} \in \langle e^{(0)}, \ldots, e^{(r-1)}\rangle,\ r \leq j \leq n-1.$$

2. $\Rightarrow$ 3.: If $r = k$, there is nothing to show. Otherwise we have that $u^{(r)} \in \langle e^{(0)}, \dots, e^{(r-1)} \rangle$ which implies $\langle u^{(r)} \rangle \neq \langle e^{(r)} \rangle$.

3. $\Rightarrow$ 1.: Since $\{\langle u^{(0)} \rangle, \dots, \langle u^{(r-1)} \rangle\} = \{\langle e^{(0)} \rangle, \dots, \langle e^{(r-1)} \rangle\}$, the rank of $\Gamma(A)$ is $\geq r$. Now assume that the rank of $\Gamma(A)$ is strictly greater than r. This implies that there is a column of $\Gamma(A)$ which is linearly independent from the first r columns. Let this be column $i \geq r$. Then $u^{(i)} \notin \langle e^{(0)}, \dots, e^{(r-1)} \rangle$. Since G is transitive on subspaces of fixed dimension, there is an element $g \in G$ with

$$\{\langle e^{(0)} \rangle, \dots, \langle e^{(r-1)} \rangle, \langle u^{(i)} \rangle\} g = \{\langle e^{(0)} \rangle, \dots, \langle e^{(r)} \rangle\}.$$

But the latter set is the lexicographically least set of size $r + 1$. The prefix of length r of the canonical set A must therefore coincide with this set, i.e. we have shown that $\langle u^{(r)} \rangle = \langle e^{(r)} \rangle$, which contradicts 3. Thus the assumption that the rank of $\Gamma(A)$ is greater than r was incorrect. This means that the rank of $\Gamma(A)$ is indeed r. □

Corollary 9.5.11 *Let $A = \{\langle u^{(0)} \rangle, \dots, \langle u^{(n-1)} \rangle\}_<$ be a canonical representative for an orbit of $\mathrm{PGL}_k(q)$ acting on n-subsets of $\mathrm{PG}_{k-1}(q)$. If the vectors $u^{(i)}$ are standard, then the matrix*

$$\Gamma(A) = \left(u^{(0)^\top} \mid \cdots \mid u^{(n-1)^\top} \right)$$

is systematic. □

As described above, we compute the orbits of G on $Y_{(i,k,d_{\min},q)}$ where i goes from 0 to n. Here we choose $G = \mathrm{PGL}_{n-k}(q)$ or $G = \mathrm{P\Gamma L}_{n-k}(q)$, depending on whether we want to compute linear or semilinear isometry classes of codes. As described in 9.1.1, the sets in $Y_{(i,k,d_{\min},q)}$ give rise to $(i, \geq i - n + k, \geq d_{\min}, q)$-codes, which is sensible only for $i \geq n - k$. As pointed out before, we will have to go through all values $i \leq n$, since the orbits on $Y_{(i,k,d_{\min},q)}$ will be constructed inductively. At each step, the canonical transversal $\mathcal{T}_i$ for the orbits of G on the set $Y_{(i,k,d_{\min},q)}$ is computed, as well as some additional data. This additional data can be used to realize functions σ_i and φ_i with $(\mathcal{T}_i, \sigma_i, \varphi_i)$ as in 9.2.12. The union

$$\mathcal{T}_{\leq n} = \bigcup_{i=0}^{n} \mathcal{T}_i$$

of all canonical representatives is the *tree of canonical representatives.* The leaves at depth n comprise the isometry classes of codes. We follow the convention of labeling nodes by their largest element. We display the ranks of the projective points rather than the projective points themselves.

Given an orbit representative $A = \{\langle u^{(0)} \rangle, \dots, \langle u^{(s-1)} \rangle\}_<$, we construct the corresponding code as follows. As in 9.1.6, we form the check matrix $\Delta(A)$. At this point we come back to the initial remarks in Section 9.1 about the matrix $\Delta(A)$ being not uniquely defined. This non-uniqueness has two reasons. The first one lies in the fact that the elements of the set A may be rearranged freely. We have resolved that issue by requiring that the elements of A be ordered increasingly (according to their ranks as given by 9.3.5). The second problem lies in the choice of the representatives $u^{(i)}$ for the projective points. To this end, we simply require that $u^{(i)}$ is standard, i.e. that its rightmost nonzero coordinate is one. With these two conventions, the matrix $\Delta(A)$ becomes unique and we may take this matrix as a check matrix of a code. This $(n, \geq k, \geq d, q)$-code is a representative of an isometry class. More precisely, if the rank of $\Delta(A)$ is r, then we have found an $(n, n-r, \geq d, q)$-code. Here we have $n - r \geq k$ since $r \leq n - k$. In order to obtain a generator matrix, we proceed as follows.

By 9.5.10, $\Delta(A)$ is systematic provided that A is the lexicographically least element in its G-orbit. If r is determined as the index for which $\langle u^{(i)} \rangle = \langle e^{(i)} \rangle$ for $i = 0, \ldots, r-1$ and $\langle u^{(r)} \rangle \neq \langle e^{(r)} \rangle$, then the rank of $\Delta(A)$ is r. Thus, we can write

$$\Delta(A) = \left(\begin{array}{c|c} I_r & M \\ \hline 0 & 0 \end{array} \right)$$

for some $r \times (n-r)$-matrix M. By 1.3.9, a generator matrix of the code is

$$\Gamma(A) = \left(-M^\top \mid I_{n-r} \right).$$

Let us consider an example.

9.5.12 **Example** We wish to construct and classify binary $(8,4)$-codes with minimum distance at least $d_{\min} = 3$. For this we are looking for sets of 8 points in $\mathrm{PG}_3(2)$. Since $d_{\min} - 1 = 2$, and since two distinct points of a projective space are always linearly independent, any subset is admissible. In order to construct the codes, we compute the orbits of $\mathrm{PGL}_4(2)$ on $\mathcal{P}_{\leq 8}\big(\mathrm{PG}_3(2)\big)$ (here, $\mathcal{P}_{\leq i}(X) := \cup_{j=0}^{i} \mathcal{P}_i(X)$). The resulting tree of canonical orbit representatives is shown in Fig. 9.4 (see Example 9.3.11 for the ranks of points in $\mathrm{PG}_3(2)$). We find 6 leaves at level 8, which comprise all essentially distinct $(8, 4, \geq 3)$-codes. Table 9.2 shows the corresponding check and generator matrices. The last code is equivalent to the extended $(7,4)$-Hamming-code. Being the only code with distance 4, this is the optimal binary code with length 8 and dimension 4. Notice that the second to last code is decomposable. Its generator matrix contains a zero column. The code can thus be seen as the direct sum (in the sense of 2.2.11) of a $(6,4)$-codes with a (rather trivial) $(1,0)$-code. For a description of the algorithm to compute the orbits, we refer to the next section. In 9.6.12, we will pick this example up again and show more details of the actual computation. ◇

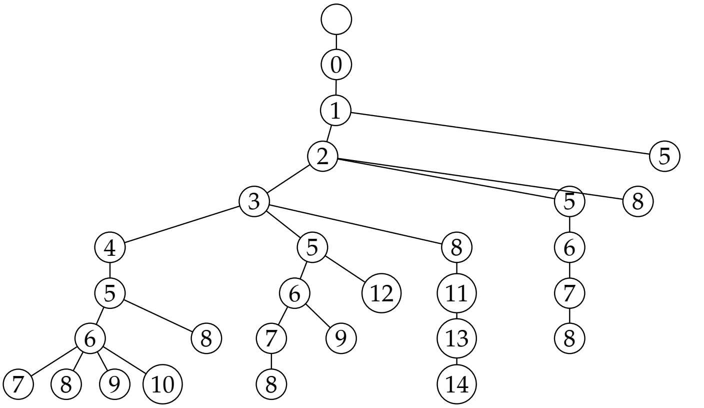


Fig. 9.4 Orbits of $\mathrm{PGL}_4(2)$ on $\mathcal{P}_{\leq 8}\big(\mathrm{PG}_3(2)\big)$

Table 9.2 Binary $(8,4,\geq 3)$-codes

A	$\Delta(A)$	$\Gamma(A)$	d
$\{0,1,2,3,4,5,6,7\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&1&0\\0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&1\\0&0&0&1&1&0&0&0\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&1&1&1&0&0&0\\1&1&0&0&0&1&0&0\\1&0&1&0&0&0&1&0\\0&1&1&0&0&0&0&1\end{array}\right)$	3
$\{0,1,2,3,4,5,6,8\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&1&1\\0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&1\\0&0&0&1&1&0&0&0\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&1&1&1&0&0&0\\1&1&0&0&0&1&0&0\\1&0&1&0&0&0&1&0\\1&1&1&0&0&0&0&1\end{array}\right)$	3
$\{0,1,2,3,4,5,6,9\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&1&1\\0&1&0&0&1&1&0&0\\0&0&1&0&1&0&1&0\\0&0&0&1&1&0&0&1\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&1&1&1&0&0&0\\1&1&0&0&0&1&0&0\\1&0&1&0&0&0&1&0\\1&0&0&1&0&0&0&1\end{array}\right)$	3
$\{0,1,2,3,4,5,6,10\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&1&0\\0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&0\\0&0&0&1&1&0&0&1\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&1&1&1&0&0&0\\1&1&0&0&0&1&0&0\\1&0&1&0&0&0&1&0\\0&1&0&1&0&0&0&1\end{array}\right)$	3
$\{0,1,2,3,5,6,7,8\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&0&1\\0&1&0&0&1&0&1&1\\0&0&1&0&0&1&1&1\\0&0&0&1&0&0&0&0\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&0&0&1&0&0&0\\1&0&1&0&0&1&0&0\\0&1&1&0&0&0&1&0\\1&1&1&0&0&0&0&1\end{array}\right)$	3
$\{0,1,2,3,8,11,13,14\}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&1&1&1&0\\0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&1\\0&0&0&1&0&1&1&1\end{array}\right)$	$\left(\begin{array}{cccc\|cccc}1&1&1&0&1&0&0&0\\1&1&0&1&0&1&0&0\\1&0&1&1&0&0&1&0\\0&1&1&1&0&0&0&1\end{array}\right)$	4

9.6

9.6 The Algorithm Snakes and Ladders

The two algorithms presented in the previous section rely on the fact that we are able to compute a canonical from of every subset. This is indeed a hard problem, and it can be tedious to provide a canonical form for a specific group action, since computing the canonical form depends very much of the nature of the group action under consideration. In this section, we will present an orbit algorithm which is general in the sense that it does not depend on the nature of the group in question. The algorithm proceeds in breadth first search, i.e. constructs the orbit representatives level by level. Also, it avoids backtracking as much as possible. The price one pays is that the amount of memory required correlates linearly to the number of orbits computed. In a sense, one trades computing time with memory. Of course, this is a limitation which may restrict the scope to which problems can be tackled. On the other hand, the speedup from the memory versus time tradeoff makes it realistic to tackle instances of hard problems, such as the computation of isometry classes of linear codes which is our main topic.

Essentially, this algorithm is due to Schmalz [171] ("Leiterspiel" loosely translated as "snakes and ladders"). Whereas Schmalz formulated his algorithm very much in the language of group theory, here we will stick to the concept of a group acting on a set. That is, we will describe the algorithm as computing orbits of a group G on subsets of a set X on which G acts. This is different from the approach taken by Schmalz, whose algorithm is formulated in the language of double cosets in finite groups. The name Leiterspiel ("ladder game") refers to the fact that the algorithm works along a sequence of subgroups which are alternately subgroups and overgroups of each other (what we will call the "down-and-up process").

We assume that orbits on points can be computed, for instance using the algorithms described in Section 9.2. The main goal is to provide a triple $(\mathcal{T}, \sigma, \varphi)$ which is a solution to the orbit problem for G acting on admissible subsets of X.

We will favor an inductive solution to the problem, namely by computing the orbits of G on $\mathcal{P}_i^{(f)}(X)$ for $i = 0, 1, \ldots$. In the search tree, this corresponds to a breadth-first search. Let

$$\operatorname{orbit}\big(G, \mathcal{P}_i^{(f)}(X)\big) = (\mathcal{T}_i, \sigma_i, \varphi_i),$$

be a solution to the orbit problem on i-subsets. Note that the case $i = 0$ is trivial, whereas $i = 1$ is the basic orbit problem on points, which we are able to solve using the methods provided in Section 9.2.

Assume that a transversal $\mathcal{T}_i$ of orbits of G on sets $\mathcal{P}_i^{(f)}(X)$, i.e. on admissible sets of size i has already been computed. Often this will be the canonical transversal, i.e. the transversal consisting of all sets which are canonical with respect to some ordering. For sake of simplicity, we will say that a set R is canonical if it belongs to one of the transversals $\mathcal{T}_i$ for some i.

In order to compute $\mathcal{T}_{i+1}$, we consider extensions of sets in $\mathcal{T}_i$. An *extension* is a set of the form

$$R \cup \{x\} \in \mathcal{P}_{i+1}^{(f)}(X),$$

where R is in $\mathcal{T}_i \subseteq \mathcal{P}_i^{(f)}(X)$ and x is in $X \setminus R$. There are four major tasks involved in computing the "next level" of orbits on $(i+1)$-sets:

Problem 1 Ensure that each G-orbit on admissible $(i+1)$-sets is reached.

Problem 2 Determine when two extensions $R \cup \{x\}$ and $S \cup \{y\}$ are isomorphic (i.e. belong to the same G-orbit). Note that here R and S are canonical, i.e. elements of the transversal $\mathcal{T}_i$.

Problem 3 Compute the stabilizer in G of an extension set $R \cup \{x\}$. Here we assume that the stabilizer of the canonical set R is known.

Problem 4 Provide a transporter map φ_{i+1} for $(i+1)$-sets. That is, given an $(i+1)$-subset $F \subseteq X$, compute an element $g \in G$ such that $Fg \in \mathcal{T}_{i+1}$.

Problem 1 is addressed easily. Let F be an admissible $(i+1)$-subset of X. Let $z := \max F$ and put $H := F \setminus \{z\}$, which is admissible since f is hereditary. Thus $H \in G(R)$ for some $R \in \mathcal{T}_i$ and $Hg = R$ for $g = \varphi(H)$. Let $x := zg \in X \setminus R$. This shows that $F \sim_G R \cup \{x\}$, which is one of the candidate sets which we considered. We note that later on, we will reduce the number of candidate sets further (see 9.6.2).

Problem 2 amounts to determining when two extensions $R \cup \{x\}$ and $S \cup \{y\}$ with $R, S \in \mathcal{T}_i$ belong to the same G-orbit. The following "exchange lemma" gives a necessary and sufficient condition for deciding that question (cf. Fig. 9.5).

Lemma *Assume that* $\mathrm{orbit}(G, \mathcal{P}_i(X)) = (\mathcal{T}_i, \sigma_i, \varphi_i)$. *For* $R, S \in \mathcal{T}_i$, $x \in X \backslash R$, $y \in X \backslash S$, *we have* $R \cup \{x\} \sim_G S \cup \{y\}$ *if and only if one of the following two conditions holds* **9.6.1**

1. $R = S$ *and* $x \sim_{G_S} y$ *or*
2. *there exists an* $r \in R$ *such that*

$$((R \backslash \{r\}) \cup \{x\})t = S \quad \textit{and} \quad rt \sim_{G_S} y$$

where $t = \varphi_i((R \backslash \{r\}) \cup \{x\})$.

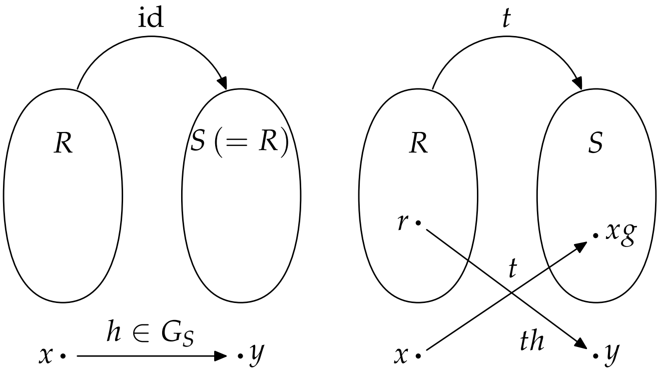


Fig. 9.5 The two cases of Lemma 9.6.1

Proof: Necessity: Assume that $R \cup \{x\} \sim_G S \cup \{y\}$. Then there exists an element $g \in G$ with

$$(R \cup \{x\})g = S \cup \{y\}.$$

We must show that 1. or 2. holds. Assume that $Rg = S$ and hence $xg = y$. Since R and S are both orbit representatives in $\mathcal{T}_i$, we must have $R = S$. But then $Sg = Rg = S$, i.e. $g \in G_S$ and hence $x \sim_{G_S} y$, which is 1. Otherwise we have $S \neq Rg \subseteq S \cup \{y\}$, and hence $xg \in S$. Let $r = yg^{-1} \in R$. Hence

$$((R \setminus \{r\} \cup \{x\})g = S.$$

But also

$$((R \setminus \{r\} \cup \{x\})t = S,$$

where $t = \varphi_i((R\backslash\{r\}) \cup \{x\})$ is the transporter element mapping $(R\backslash\{r\}) \cup \{x\}$ onto the canonical representative S. Thus g is contained in the left coset tG_S, i.e. $g = th$ for some $h \in G_S$. Now $rth = rg = y$, i.e. $rt \sim_{G_S} y$, which is 2.
Sufficiency: If 1. is valid, and if $g \in G_S$ maps x to y, then clearly $(R \cup \{x\})g = Rg \cup \{xg\} = S \cup \{y\}$. If 2. holds with $h \in G_S$ mapping rt to y then

$$(R \cup \{x\})th = ((R\backslash\{r\}) \cup \{x\})th \cup \{r\}th = Sh \cup \{rth\} = S \cup \{y\}. \qquad \square$$

The first part of this result has an important implication for the candidate set of extensions (Problem 1):

9.6.2 **Corollary** *It suffices to consider only extensions of the form $R \cup \{x\}$ where R is a canonical i-set and $x \in X \setminus R$ is canonical under the stabilizer of R in G.* □

$$\begin{tikzcd}
G_R \arrow[rr, dashed] \arrow[dr] & & G_{R\cup\{x\}} \\
& G_{R,x} \arrow[ur] &
\end{tikzcd}$$

Fig. 9.6 The "down-and-up" process

From now on, we consider only extensions $R \cup \{x\}$ of the form described in the previous corollary.

Let us now describe the problem of computing the stabilizer of extension sets (Problem 3). If the extension is $R \cup \{x\}$, then this amounts to computing

$$G_{R\cup\{x\}},$$

the setwise stabilizer of $R \cup \{x\}$. We assume that G_R, the setwise stabilizer of the canonical set R is known. The difficulty is that there is no relationship between the groups G_R and $G_{R\cup\{x\}}$ (meaning that neither is a subgroup of the other in general). However, they share a common subgroup, namely the group $G_{R,x}$ which is the set of elements of G which stabilize R setwise and x pointwise. The idea is to first go down from G_R to $G_{R,x}$ (the "downstep"), which is relatively straightforward. Generators for $G_{R,x}$ can be computed from generators for G_R by means of 9.2.10. The difficulty is to compute the group $G_{R\cup\{x\}}$ from the subgroup $G_{R,x}$ (i.e. the "upstep"). In the following, we will address this problem first (we will call it the "down-and-up" process, cf. Fig. 9.6). Afterwards, we will present the algorithm to compute orbits on sets which combines all the ideas developed so far.

Recall that for a subgroup V of G we have

$$X_V = \{x \in X \mid \forall v \in V : \; xv = x\}.$$

The following result is a consequence of 3.4.1.

Lemma *Let the group H act on a set X. Let $V = H_x$ be the stabilizer of a point $x \in X$. In addition, let $\mathcal{R}$ be a set of elements of H such that for each $y \in H(x)$ there exists one and only one $g \in \mathcal{R}$ with $xg = y$. Then $\mathcal{R}$ is a set of right coset representatives of V in H.* □ **9.6.3**

In the situation of the lemma, we call H the *extension of V w.r.t. the coset representatives $\mathcal{R}$* and we write

$$H = \operatorname{Ext}(V, \mathcal{R}, x) = \bigcup_{r \in \mathcal{R}} Vr,$$

where the last union is over disjoint cosets. For $R \in \mathcal{T}_i$ and $x \in X \setminus R$, let us define

9.6.4 $$R^*(x) := \left\{ r \in R \;\middle|\; \begin{array}{l} (R \setminus \{r\}) \cup \{x\} \in G(R) \text{ and } rt \in G_R(x), \\ \text{where } t = \varphi_i((R \setminus \{r\}) \cup \{x\}) \end{array} \right\}$$

and put $R(x) = R^*(x) \cup \{x\}$. The next result describes the stabilizer of extension sets.

9.6.5 **Lemma** *Let G act on the finite set X. Assume that* $\operatorname{orbit}(G, \mathcal{P}_i(X)) = (\mathcal{T}_i, \sigma_i, \varphi_i)$. *For $R \in \mathcal{T}_i$ let* $\operatorname{orbit}(G_R, X \setminus R) = (\mathcal{T}_R, \sigma_R, \varphi_R)$. *Fix $x \in \mathcal{T}_R$. Then the orbit of x under $G_{R\cup\{x\}}$ is $R(x)$. In particular $G_{R\cup\{x\}} = \operatorname{Ext}(G_{R,x}, \mathcal{R}, x)$ where*

$$\mathcal{R} = \{1\} \;\cup\; \big\{ t \cdot \varphi_R(rt) \;\big|\; r \in R^*(x),\;\; t = \varphi_i((R \setminus \{r\}) \cup \{x\}) \big\}.$$

Here, $G_{R,x} = G_R \cap G_x = \{g \in G \mid Rg = R,\; xg = x\}$ and $G_{R\cup\{x\}}$ is the setwise stabilizer of the set $R \cup \{x\}$.

Proof: Let $\mathcal{O}_x = G_{R\cup\{x\}}(x)$ be the orbit of x under $G_{R\cup\{x\}}$. We claim that $\mathcal{O}_x = R(x) = R^*(x) \cup \{x\}$.

Consider $r \in R$. Let $g \in G_{R\cup\{x\}}$ with $rg = x$. Since g maps $R \cup \{x\}$ onto itself, we have

$$R \cup \{x\} = (R \cup \{x\})g^{-1} = Rg^{-1} \cup \{xg^{-1}\} = Rg^{-1} \cup \{r\}.$$

This implies that $Rg^{-1} = (R \setminus \{r\}) \cup \{x\}$ and therefore

$$((R \setminus \{r\}) \cup \{x\})g = R \in \mathcal{T}_i,$$

i.e. $(R \setminus \{r\}) \cup \{x\} \in G(R)$. By definition, the group element $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$ maps $(R \setminus \{r\}) \cup \{x\}$ to the canonical representative in $\mathcal{T}_i$ of its G-orbit, which must be the set R. Thus

$$((R \setminus \{r\}) \cup \{x\})t = R = ((R \setminus \{r\}) \cup \{x\})g.$$

We conclude that t and g belong to the same left coset of G_R, i.e. there is an element $h \in G_R$ such that $g = th$. Thus

$$((R \setminus \{r\}) \cup \{x\})t = ((R \setminus \{r\}) \cup \{x\})gh^{-1} = Rh^{-1} = R.$$

Also, $rt = rgh^{-1} = xh^{-1} \sim_{G_R} x$ and therefore $r \in R^*(x)$.

Since x is clearly an element of its own orbit under $G_{R\cup\{x\}}$, it remains to show that the elements $r \in R^*(x)$ lie in $\mathcal{O}_x$. If $r \in R^*(x)$, we have that $((R \setminus \{r\}) \cup \{x\})t = R$ and $rt \sim_{G_R} x \in \mathcal{T}_R$ where $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$. The second condition implies that $rth = x$ where $h = \varphi_R(rt) \in G_R$. The equation

$$(R \cup \{x\})th = ((R \setminus \{r\}) \cup \{x\})th \cup \{r\}th = Rh \cup \{r\}th = R \cup \{x\}$$

shows that $g = th$ stabilizes $R \cup \{x\}$. In addition, since $xg^{-1} = r$ we have that $r \in \mathcal{O}_x$. This proves the claim.

We are now able to show that $G_{R\cup\{x\}} = \operatorname{Ext}(G_{R,x}, \mathcal{R}, x)$. It is clear that $G_{R,x}$ is a subgroup of $G_{R\cup\{x\}}$. Next, $G_{R\cup\{x\},x} = G_{R\cup\{x\}} \cap G_x = G_{R,x}$. Hence the cosets of $G_{R,x}$ in $G_{R\cup\{x\}}$ are in one-to-one correspondence with the distinct images of x under $G_{R\cup\{x\}}$. If we revisit the proof of the claim above, we notice that for $r \in R^*(x)$ the element $t\varphi_R(rt)$ where $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$ is in $G_{R\cup\{x\}}$ and maps $r \in R^*(x) = \mathcal{O}_x \setminus \{x\}$ to x. Since the identity element (denoted as 1) is trivially contained in $G_{R\cup\{x\}}$ and maps x to x, the union $\mathcal{R}$ of 1 and all elements $t\varphi_R(rt)$ as above forms a transporter set for the distinct elements of $\mathcal{O}_x$ in $G_{R\cup\{x\}}$. By the standard argument alluded to above we have that $G_{R\cup\{x\}}$ is the union of the right cosets of $G_{R,x}$ with respect to the elements of $\mathcal{R}$. Therefore by 9.6.3, $G_{R\cup\{x\}}$ is the extension of $G_{R,x}$ with respect to the point x and the transversal $\mathcal{R}$. □

The following result is helpful in computing the set $R^*(x)$. It may reduce the number of $r \in R$ which have to be tested.

Lemma *Let G act on the finite set X. Assume that* $\operatorname{orbit}(G, \mathcal{P}_i(X)) = (\mathcal{T}_i, \sigma_i, \varphi_i)$. *For $R \in \mathcal{T}_i$ let* $\operatorname{orbit}(G_R, X \setminus R) = (\mathcal{T}_R, \sigma_R, \varphi_R)$. *Fix $x \in \mathcal{T}_R$ and $r \in R$. Then* **9.6.6**

1. *If $r \in R^*(x)$ then $rs \in R^*(x)$ for all $s \in G_{R\cup\{x\}}$.*
2. *If $r \notin R^*(x)$ then $rs \notin R^*(x)$ for all $s \in G_{R\cup\{x\}}$.*

Proof: Let $r \in R^*(x)$ and $s \in G_{R\cup\{x\}}$. Then $(R \setminus \{r\}) \cup \{x\} \in G(R)$ and $rt_r \in G_R(x)$ for $t_r = \varphi_i((R \setminus \{r\}) \cup \{x\})$. The latter condition means that there is an element $h \in G_R$ such that

$$rt_rh = x.$$

We will now show that $rs \in R^*(x)$. Using the fact that $s \in G_{R\cup\{x\}}$ we obtain

$$\begin{aligned}((R \setminus \{rs\}) \cup \{x\})s^{-1}t_r &= ((R \cup \{x\}) \setminus \{rs\})s^{-1}t_r\\ &= (R \cup \{x\})s^{-1}t_r \ \setminus \ \{rs\}s^{-1}t_r\\ &= (R \cup \{x\})t_r \ \setminus \ \{r\}t_r\\ &= ((R \setminus \{r\}) \cup \{x\})t_r\\ &= R,\end{aligned}$$

i.e. $(R \setminus \{rs\}) \cup \{x\} \in G(R)$. Thus with $t_{rs} = \varphi_i((R \setminus \{rs\}) \cup \{x\})$ we have

$$((R \setminus \{rs\}) \cup \{x\})t_{rs} = R.$$

Putting the two equations together we see that $s^{-1}t_r$ and t_{rs} differ only by an element $u \in G_R$, i.e.

$$t_{rs} = s^{-1}t_r u.$$

Thus

$$rs \cdot t_{rs} = rss^{-1}t_r u = rt_r u = (rt_r h)h^{-1}u = xh^{-1}u \in G_R(x),$$

since h and u are both elements of G_R. This proves the first part. For the second part, assume that $r \notin R^*(x)$ but $rs \in R^*(x)$. Then by the first part we deduce that $r = rss^{-1} \in R^*(x)$, which is a contradiction. □

9.6.7 Remarks

1. In the previous result, the group $G_{R \cup \{x\}}$ may be replaced by the subgroup $G_{R,x}$. The reason for doing this is that the group $G_{R \cup \{x\}}$ may not be known initially, whereas the smaller group $G_{R,x}$ may be known. In fact, we have

$$G_{R \cup \{x\}} = \mathrm{Ext}(G_{R,x}, \mathcal{R}, x)$$

   where $\mathcal{R}$ is defined in terms of $R^*(x)$. Thus when testing elements r for membership in $R^*(x)$ we cannot use $G_{R \cup \{x\}}$. Since $G_{R,x}$ is simply a point stabilizer in the known group G_R, we may start with this group instead. Later on, when non-trivial elements

$$g^{(0)}, \ldots, g^{(i-1)}$$

   in $\mathcal{R}$ have been found, we may form the overgroup

$$H^{(i)} = \langle G_{R,x}, g^{(0)}, \ldots, g^{(i-1)} \rangle \leq G_{R \cup \{x\}}$$

   and apply 9.6.6 to $s \in H^{(i)}$.
2. To apply 9.6.6, one computes the orbits of $H = G_{R \cup \{x\}}$ (or any known subgroup thereof, see the previous remark) on the elements of R. For each orbit $H(r)$, only the representative r needs to be tested for membership in $R^*(x)$. If $r \in R^*(x)$ then $H(r) \subseteq R^*(x)$. Otherwise $H(r) \cap R^*(x) = \emptyset$. ◇

Summarizing, we have seen in Lemma 9.6.1 how to decide whether or not two extensions $R \cup \{x\}$ and $S \cup \{y\}$ are in the same G-orbit, i.e. isomorphic. This is the main tool for reducing isomorphic copies. It is now time to take the lexicographical order into account. We use the following tie breaker. If two extensions are isomorphic then we always keep the lexicographically smaller one of the two and we discard the other one. So, if $R \preceq S$ then we keep $R \cup \{x\}$.

Or, if $R = S$ but $x < y$ then we keep $R \cup \{x\}$ and discard $R \cup \{y\}$. Essentially, we do a breadth first search in the tree of canonical orbit representatives. This step comprises the isomorph rejection.

We assume that all representatives of i-orbits are available, i.e. that

$$(\mathcal{T}_i, \sigma_i, \varphi_i)$$

has been computed. Next we examine the sets $R \in \mathcal{T}_i$ in lexicographically increasing order. For each such set R, we compute the orbits of its stabilizer $G_R = \sigma_i(R)$ on the remaining points $X \setminus R$. Let

$$(\mathcal{T}_R, \sigma_R, \varphi_R)$$

be the resulting orbit data. As usual, we assume that $\mathcal{T}_R$ is the canonical transversal. This means that the elements of $\mathcal{T}_R$ (which are just points) are the least among their respective G_R-orbit. Next, we consider the extensions of the form $R \cup \{x\}$ where $x \in \mathcal{T}_R$ (in increasing order). Since G_R is known by assumption, the stabilizer $G_{R,x}$ can be computed. Recall that

$$G_{R,x} = G_R \cap G_x$$

is the pointwise stabilizer of x in G_R. Actually,

$$G_{R,x} = \sigma_R(x)$$

is part of the orbit data which has been computed in the previous step. Next, we compute the set $R^*(x)$ of 9.6.4. For this, we try all $r \in R$ and see if the set $(R \setminus \{r\}) \cup \{x\}$ is contained in the G-orbit of R. This can be done by computing

$$t = \varphi_i((R \setminus \{r\}) \cup \{x\})$$

and testing whether

$$((R \setminus \{r\}) \cup \{x\})t = R.$$

If this is the case then we have to test the second condition, which requires that rt is in the same G_R-orbit as x. For this, we simply compute $h = \varphi_R(rt)$ and test if $rth = x$. If all these conditions hold then $r \in R^*(x)$, otherwise we proceed to test the next element in R.

Assume that $r \in R^*(x)$ has been found. Then

$$((R \setminus \{r\}) \cup \{x\})t = R, \text{ and } rth = x$$

where t and h are as above. Thus

$$(R \cup \{x\})th = ((R \setminus \{r\}) \cup \{x\})th \cup \{r\}th = Rh \cup \{x\} = R \cup \{x\},$$

i.e. $a := a_r := th$ is an automorphism of the extension set $R \cup \{x\}$. This automorphism a has the property that $ra = x$, i.e. $xa^{-1} = r$. In other words, this automorphism is a coset representative for the subgroup $G_{R,x}$ in $G_{R\cup\{x\}}$. If $\mathcal{R}$ is the collection of all a_r for $r \in R^*(x)$ together with the identity, then $\mathcal{R}$ is a transversal of the cosets of $G_{R,x}$ in $G_{R\cup\{x\}}$. In other words,

$$G_{R\cup\{x\}} = \mathrm{Ext}(G_{R,x}, \mathcal{R}, x).$$

As remarked above, once the first automorphism a_r has been found, we can immediately form the group $H^{(1)} := \langle G_{R,x}, a_r\rangle$, which is a subgroup of $G_{R\cup\{x\}}$. Later on, we may use $H^{(1)}$ to reduce the number of $r \in R$ which need to be tested for membership in $R^*(x)$. We proceed by induction on $i = 1, 2, \ldots$. Whenever another automorphism generator a_r has been found while testing an element $r \in R$, we define the group extension

$$H^{(i+1)} = \langle H^{(i)}, a_r\rangle.$$

Of course, once an element $r \in R$ has been proven to be outside of $R^*(x)$, we can eliminate the whole orbit $H^{(i)}(r) \subseteq R$ from the search. All this follows from 9.6.6.

What happens if $r \in R$ does not lie in $R^*(x)$? Then we have found a group element $g = th$ with $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$ and $h \in G_S$ such that

$$((R \setminus \{r\}) \cup \{x\})t = S,$$

and $rth = y$. Thus

$$(R \cup \{x\})th = ((R \setminus \{r\}) \cup \{x\})th \cup \{r\}th = Sh \cup \{y\} = S \cup \{y\}.$$

This means that the extension $R \cup \{x\}$ is isomorphic to $S \cup \{y\}$, i.e.

$$R \cup \{x\} \sim_G S \cup \{y\}.$$

Here, we use the word isomorphic as a synonym for "being in the same G-orbit." In this language, we can say that the element th is an isomorphism between the two extensions. Note that $R = S$ is still possible (but then $x < y$). We claim that $R \cup \{x\}$ precedes $S \cup \{y\}$. To see this, recall that we proceed in a breadth first search fashion, i.e. we process the extensions at any given level in lexicographically increasing order. Hence, if $S \cup \{y\}$ were less than $R \cup \{x\}$ then we would have detected the fact that $R \cup \{x\} \sim_G S \cup \{y\}$ earlier, and we would have discarded $R \cup \{x\}$. So, at this point we decide to eliminate the extension $S \cup \{y\}$, since it is not canonical. However, we will not totally delete the extension from the search tree. Instead, we decide to save the isomorphism th which maps $R \cup \{x\}$ to $S \cup \{y\}$. Actually, we decide to store the inverse,

$$\psi_S(y) := (th)^{-1}$$

and call this a *fusion element*. Also, we introduce a *fusion node* for the extension $S \cup \{y\}$. The fusion node serves as a means of recoding the information which we gained about isomorphic sets. If $S \cup \{y\}$ is a fusion node, we always have that

$$(S \cup \{y\})\psi_S(y) = R \cup \{x\} \text{ is canonical.} \tag{9.6.8}$$

The fusion nodes will help to speed up the algorithm when it comes to computing transporter elements, as we will see in the next paragraph. Summarizing, we have seen how to construct the canonical transversal $\mathcal{T}_{i+1}$ of orbits on sets of size $i+1$ together with the respective stabilizers.

Let us now address the problem of defining the transporter map φ_{i+1} (since this map is needed for the induction). More specifically, given a set F of size $i+1$, the question is to find the canonical representative

$$R \cup \{x\} \in \mathcal{T}_{i+1}$$

with $F \sim_G R \cup \{x\}$. In particular, we wish to determine an element $g \in G$ with $Fg = R \cup \{x\}$. This problem can be solved recursively. The set F is split into $z := \max F$ and $Z = F \setminus \{z\}$. By induction, we can compute an element $t := \varphi_i(Z)$. Then $S := Zt$ is a canonical orbit representative. Using the orbit data, we compute $h \in G_S$ such that $zth = y$ is canonical under G_S. If $S \cup \{y\}$ is canonical under G, we return th. Otherwise, if $S \cup \{y\}$ is a fusion node, then we have a fusion element $\psi_S(y)$ such that

$$(S \cup \{y\})\psi_S(y) = R \cup \{x\}$$

is canonical by 9.6.8 and we return $th\psi_S(y)$. This finishes the description of the algorithm. Let us present the algorithm as

Theorem *Let G act on the finite set X. Assume that we can compute stabilizers, group extensions and orbits on points for subgroups of G. Furthermore, let $f : \mathcal{P}(X) \to \{0,1\}$ be a test function which is G-invariant and hereditary (in the sense of 9.5.1 and 9.5.2). Then Algorithm 9.6.10 computes the orbits of G on $\mathcal{P}^{(f)}(X) = \mathcal{P}(X) \cap f^{-1}(\{1\})$, the set of admissible subsets of X.* □ 9.6.9

Algorithm (orbits on subsets) 9.6.10

Input: orbit$\big(G, \mathcal{P}_i^{(f)}(X)\big) = (\mathcal{T}_i, \sigma_i, \varphi_i)$

Output: orbit$\big(G, \mathcal{P}_{i+1}^{(f)}(X)\big) = (\mathcal{T}_{i+1}, \sigma_{i+1}, \varphi_{i+1})$

```
for $R \in \mathcal{T}_i$ do
    compute orbit$(G_R, X \setminus R) := (\mathcal{T}_R, \sigma_R, \varphi_R)$
end for
$\mathcal{T}_{i+1} := \emptyset$
```

```
for $R \in \mathcal{T}_i$ (in increasing order) do
    for $x \in \mathcal{T}_R$ (in increasing order) with $f(R \cup \{x\}) = 1$
            and for which $\psi_R(x)$ has not yet been defined do
        $G_{R,x} := \sigma_R(x)$
        $H := G_{R,x}$
        for all $r \in R$ which are least in their $H$-orbit do
            $t := \varphi_i((R \setminus \{r\}) \cup \{x\})$
            $S := ((R \setminus \{r\}) \cup \{x\})t$
            $h := \varphi_S(rt)$
            $y := rth$
                (now: $(R \cup \{x\})th = S \cup \{y\},\ S \in \mathcal{T}_i,\ y \in \mathcal{T}_S$)
            if $S = R$ and $y = x$ then   (case 1 of 9.6.1)
                $H := \langle H, th \rangle$
                    ($th$ is an automorphism of $R \cup \{x\}$)
            else   (case 2 of 9.6.1)
                $\psi_S(y) := (th)^{-1}$
                    ($th$ is an isomorphism from $R \cup \{x\}$ to $S \cup \{y\}$)
            end if
        end for
        append $R \cup \{x\}$ to $\mathcal{T}_{i+1}$
        $\sigma_{i+1}(R \cup \{x\}) := H\ (= G_{R \cup \{x\}})$
    end for
end for
return $(\mathcal{T}_{i+1}, \sigma_{i+1}, \varphi_{i+1})$
```

Where the function φ_{i+1} is defined as follows.

```
function $\varphi_{i+1}(F)$
    $z := \max F$, $Z := F \setminus \{z\}$ (a set of size $i$)
    $t := \varphi_i(Z)$
    $S := Zt$
    $h := \varphi_S(zt), y := zth$
    if $\psi_S(y)$ has been defined then
        return $th\psi_S(y)$
    else
        return $th$
    end if
end function
```

□

Proof: The proof is by induction. The orbits of subsets of size 0 are trivially known. The orbits of subsets of size 1 are known by assumption. Now assume that $\operatorname{orbit}(G, \mathcal{P}_i(X)) = (\mathcal{T}_i, \sigma_i, \varphi_i)$ has already been computed. In order to prove correctness of Algorithm 9.6.10, we verify that $\mathcal{T}_{i+1}$ is a transversal for the orbits of G on $\mathcal{P}_{i+1}(X)$, and that $\sigma_{i+1}(R) = G_R$ for $R \in \mathcal{T}_{i+1}$ and that $\varphi_{i+1}(S) = t$ such that $St \in \mathcal{T}_{i+1}$ for all $S \in \mathcal{P}_{i+1}(X)$.

First of all, each $(i+1)$-subset S can be written as $S = S' \cup \{y\}$ where S' is an i-subset and $y \in X \setminus S'$. Putting $g := \varphi_i(S')$ we get $S \sim_G Sg = S'g \cup \{yg\}$ where $S'g$ is an orbit representative in $\mathcal{T}_i$. Hence we get representatives of all G-orbits on $(i+1)$-sets from the extensions of the form $R \cup \{x\}$ where $R \in \mathcal{T}_i$ and $x \in X \setminus R$. In lines (0) and $(4), (5)$ these extensions of orbit representatives are considered. In line (1), the orbits of G_R on $X \setminus R$ are computed for $R \in \mathcal{T}_i$. The result is $(\mathcal{T}_R, \sigma_R, \varphi_R)$, where

1. $\mathcal{T}_R$ is a transversal of the orbits of G_R on $X \setminus R$,
2. $\sigma_R : X \setminus R \to L(G)$ is such that $\sigma_R(x) = G_{R,x} = (G_R)_x$ is the stabilizer of x in G_R, and
3. $\varphi_R : X \setminus R \to G$ is a map with $\varphi_R(y) = g \in G$ such that $yg \in \mathcal{T}_R$.

The candidate set is the set of extensions $R \cup \{x\}$ where $R \in \mathcal{T}_i$ and $x \in \mathcal{T}_R$. In lines (4) and (5), the extensions $R \cup \{x\}$ are considered again.

We must now show that the extensions which are added to $\mathcal{T}_{i+1}$ in line (20) are pairwise not in the same G-orbit. Let $R \cup \{x\}$ and $S \cup \{y\}$ be two arbitrary distinct extension sets (with $R, S \in \mathcal{T}_i$ and $x \in \mathcal{T}_R$ and $y \in \mathcal{T}_S$). By 9.6.1, $R \cup \{x\} \sim_G S \cup \{y\}$ if and only if either $R = S$ and $x \sim_{G_R} y$, or for one $r \in R$ the equations

$$((R \setminus \{r\}) \cup \{x\})t = S \quad \text{and} \quad rt \sim_{G_S} y \tag{9.6.11}$$

hold for $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$. First, consider the case $R = S$ and $x \sim_{G_R} y$. Since $R \cup \{x\}$ is different from $S \cup \{y\}$, we must have $x \neq y$. But x and y are different elements of the transversal $\mathcal{T}_R$ of G_R orbits, which contradicts $x \sim_{G_R} y$. Hence we must be in the second case, i.e. 9.6.11 holds true for some $r \in R$. Without loss of generality, we assume that

$$R \cup \{x\} \preceq S \cup \{y\},$$

i.e. that $R \cup \{x\}$ has been considered before $S \cup \{y\}$ in lines (4) and (5). By 9.6.11, there is an element $r \in R$ for which $((R \setminus \{r\}) \cup \{x\})t = S$ with $t = \varphi_i((R \setminus \{r\}) \cup \{x\})$ and $rth = y \in \mathcal{T}_S$ with $h = \varphi_S(rt) \in G_S$. In this case, the fusion element $\psi_S(y) = (th)^{-1}$ will be defined in line (16) which prevents the extension $S \cup \{y\}$ from being considered in lines (4) and (5). This proves that the computed set $\mathcal{T}_{i+1}$ intersects each G-orbit at most once. From the above, we already know that $\mathcal{T}_{i+1}$ contains elements from every orbit, and hence $\mathcal{T}_{i+1}$ is a transversal of the $(i+1)$-orbits of G, as required. The fact that $G_{R\cup\{x\}} = \mathrm{Ext}(G_{R,x}, \mathcal{R}, x)$ has been shown in 9.6.5. The transversal $\mathcal{R}$ is never explicitly computed. Instead, in line (7) the group H is initialized to be $H = G_{R,x} = \sigma_R(x)$. The if clause in line (13) evaluates to true if and only if $r \in R^*(x)$, which means that th is an element of $\mathcal{R}$. Therefore, the group H is extended by th in line (14). Line (8) reduces the number of $r \in R$ which have to be tested. According to 9.6.7, we require that $r \in R$ be minimal in its H-orbit. At the end of the loop, in line (18), the full group $G_{R\cup\{x\}}$ has been computed in H. In lines (19) and (20), the new canonical representative $R \cup \{x\}$ is added to $\mathcal{T}_{i+1}$ and the stabilizer $G_{R\cup\{x\}}$ is stored as $\sigma_{i+1}(R \cup \{x\})$. At the end of the for loops in lines (21) and (22), the transversal $\mathcal{T}_{i+1}$ is complete.

It remains to show that $\varphi_{i+1}(F)$ is an element $g \in G$ with $Fg \in \mathcal{T}_{i+1}$. In line (25), F is written as a union of an i-set Z and the element z. For $t = \varphi_i(Z)$ we then have $Zt = S \in \mathcal{T}_i$ in line (27). Hence the orbit data for the set S has been computed and we can evaluate $h = \varphi_S(zt)$ and define $y = zth$ in line (28). We now have

$$Fth\psi_S(y) = (Z \cup \{z\})th\psi_S(y) = S \cup \{y\}.$$

If $\psi_S(y)$ has not been defined then $S \cup \{y\}$ is canonical and we return th. Otherwise, there has been an extension $R \cup \{x\}$ and an element $r \in R \setminus R^*(x)$ such that $((R \setminus \{r\}) \cup \{x\})t' = S$, with $t' := \varphi((R \setminus \{r\}) \cup \{x\})$, and $y = xt'h'$ for $h' = \varphi_S(xt') \in G_{\{S\}}$. Since r is not in $R^*(x)$, the if clause in (13) did not hold and the element $(t'h')^{-1}$ has been stored as $\psi_S(y)$. Hence

$$\begin{aligned} Fth\psi_S(y) &= (Z \cup \{z\})th\psi_S(y) \\ &= (S \cup \{y\})\psi_S(y) \\ &= (S \cup \{y\})(t'h')^{-1} \\ &= (R \cup \{x\}) \in \mathcal{T}_{i+1}. \end{aligned}$$

This proves that in each case $\varphi_{i+1}(F)$ is an element that maps F to its canonical orbit representative, as required. This completes the proof that the algorithm computes the orbit data for G acting on subsets. □

9.6.12 **Example (continuation of Example 9.5.12)** Let us consider the binary (8,4)-codes again. The generation tree is shown in Fig. 9.7. A node A is represented by a box, with the label $\max A$ indicated in a circle right above the box. The circled numbers immediately below the box are the possible extensions. Inside the box, information on the stabilizer is given. The first number is the order of the stabilizer. After that, the orbits of the stabilizer on points are indicated. Inside each orbit, the numbers are arranged in increasing order. Hence

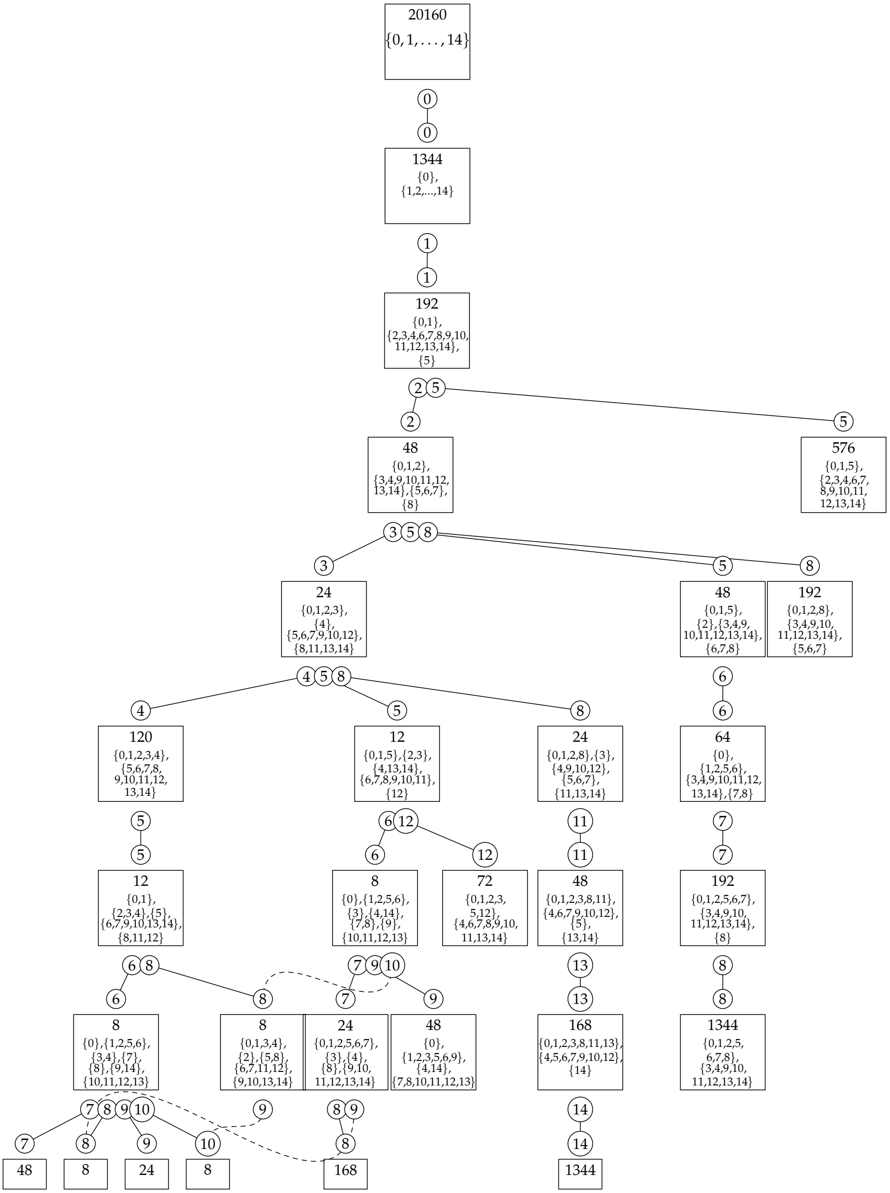


Fig. 9.7 Generation tree of $(8, 4, \geq 3, 2)$-codes

the first number is the least orbit representative. Not every orbit leads to an extension. The rank condition must be satisfied for possible extensions (since $d = 3$, the rank condition is always satisfied in this example). The solid lines stand for extensions leading to canonical sets, i.e. to new orbit representatives at depth one step further down the tree. These lines always connect circles with equal numbers. The three dashed and somewhat curly lines are related to fusion nodes. Recall that fusion nodes stand for extension sets which are not canonical. Each fusion node is connected by a curly line to the corresponding canonical node, which is to the left. Associated with every curly line is a fusion element. The three fusion nodes are

$$\{0,1,2,3,4,5,8,9\}, \{0,1,2,3,5,6,10\}, \text{ and } \{0,1,2,3,5,6,7,9\}.$$

The corresponding fusion elements are (in matrix form and as permutations of the points of $\mathrm{PG}_3(2)$, respectively)

$$\psi_{\{0,1,2,3,4,5,8\}}(9) = \begin{pmatrix} 1&0&1&0\\ 1&1&1&1\\ 0&0&0&1\\ 1&0&0&0 \end{pmatrix} = (0,6,13,12,9,2,3)(1,4,5,10,14,7,8),$$

$$\psi_{\{0,1,2,3,5,6\}}(10) = \begin{pmatrix} 1&1&1&0\\ 1&1&0&0\\ 0&0&0&1\\ 0&1&0&0 \end{pmatrix} = (0,8,12,10)(1,5,2,3)(4,14,9,6)(7,11),$$

and

$$\psi_{\{0,1,2,3,5,6,7\}}(9) = \begin{pmatrix} 1&1&1&0\\ 1&0&1&0\\ 0&0&1&0\\ 1&1&1&1 \end{pmatrix} = (0,8,7)(1,6,5)(3,4,9)(11,13,12).$$

For instance, the fusion node $\{0,1,2,3,5, 6,10\}$ is connected to the canonical node $\{0,1,2,3,4,5,8\}$. This is because application of the fusion element maps one set onto the other:

$$\{0,1,2,3,5,6,10\}(0,8,12,10)(1,5,2,3)(4,14,9,6)(7,11) = \{8,5,3,1,2,4,0\}.$$

Let us trace the computation of the automorphism group of the extended Hamming code, for example. As pointed out in Example 9.5.12, the $(8,4)$ extended Hamming code is the rightmost leaf at level 8, i.e. the set

$$\{0,1,2,3,8,11,13,14\}.$$

Essentially, the computation consists of 8 repetitions of the "down-and-up" process described earlier in this section. For each prefix R of the set in question, we compute from the given group G_R the groups $G_{R,x}$ and $G_{R\cup\{x\}}$. We

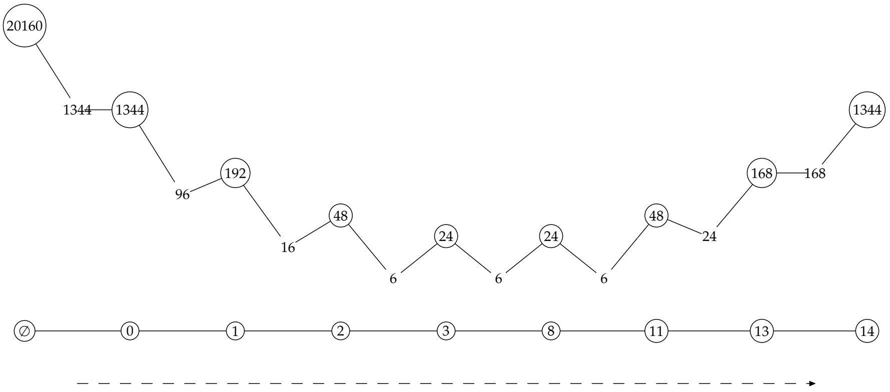


Fig. 9.8 Computing the automorphism group of the $(8,4)$ extended Hamming code

proceed by induction on the size of the prefix R, i.e. we start with $R = \emptyset$, then consider $R = \{0\}$, after that $R = \{0,1\}$ and so forth. This means that we are moving from the left to the right in Fig. 9.8, which shows the elements of R at the bottom. Above, the order of the groups G_R (circled) and $G_{R,x}$ is plotted on a logarithmic scale. The computation starts with the empty set, whose automorphism group is $G = \mathrm{PGL}(4,2)$ of order 20160 (this is the root node in Fig. 9.7). Then the point 0 is chosen. Since 0 is in an orbit of length 15,

$$|G_0| = 20160/15 = 1344.$$

Next, we add the point 1 to the set. Since 1 is in an orbit of G_0 of length 14, we have

$$|G_{0,1}| = 1344/14 = 96.$$

The upstep results in an automorphism which interchanges 0 and 1, so that $G_{0,1}$ is of index 2 in the set stabilizer $G_{\{0,1\}}$, which must therefore be of order $2 \cdot 96 = 192$. Then the point 2 is added from an orbit of $G_{\{0,1\}}$ of length 12, so that $G_{\{0,1\},2}$ has order $192/12 = 16$ (recall that $G_{\{0,1\},2}$ denotes the intersection of the set stabilizer of $\{0,1\}$ with the point stabilizer of 2. The following upstep detects that 2 is in an orbit of length 3 under $G_{\{0,1,2\}}$, so that the next set stabilizer is $G_{\{0,1,2\}}$ of order $16 \cdot 3 = 48$. The computation continues in this way. Eventually, the automorphism group of the extended Hamming code is computed to be the set stabilizer

$$G_{\{0,1,2,3,8,11,13,14\}}$$

of order 1344. ◇

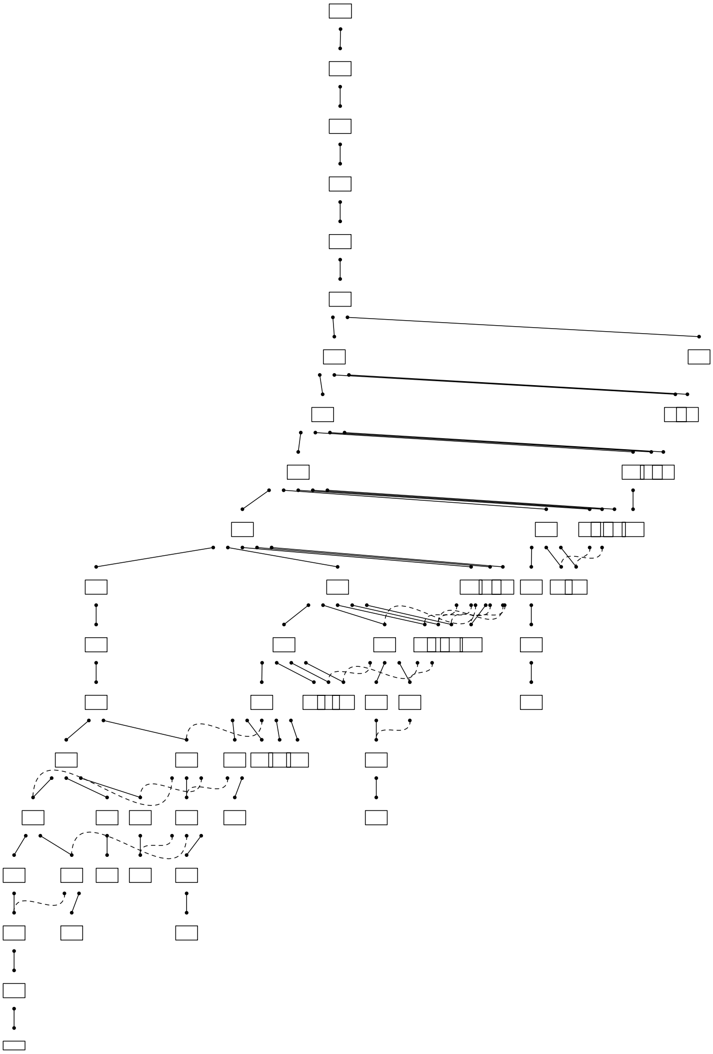


Fig. 9.9 The binary $(18, 9, 6)$-codes

Figure 9.9 shows the generation tree for the unique binary $(18, 9, 6)$-code. The sole purpose of this example is to give a rough idea of the nature of such trees. We suppress all labels and automorphism group order information.

9.7 Base and Strong Generating Sets

9.7

The orbit algorithm as described above depends on the availability of good algorithms to work with permutation groups. In particular, point stabilizer subgroups and extension overgroups need to be computed (as well as orbits on points). It turns out that our first attempt at these algorithms, based on sets of generators, does not perform well for large examples. The reason is that the number of generators produced by 9.2.10 may become too large, which in turn deteriorates the performance of the orbit algorithm on points.

In this section, we will overcome this bottleneck by introducing a better data structure for permutation groups. This data structure, introduced by Sims [181, 182], is called a stabilizer chain. It represents the group by means of a chain of subgroups terminating in the trivial group. Each group in the chain is the stabilizer of a point in the previous group.

To begin with, let G be a group acting faithfully on a set finite, totally ordered set $X = \{x_0, \dots x_{n-1}\}_<$. A subset $B = \{b_0, \dots, b_{r-1}\} \subset X$ is called *base* for G on X if the *pointwise stabilizer* $G_{b_0,\dots,b_{r-1}} = 1$, i.e. if only the identity of G fixes all the points of B. An *ordered base* for G on X is a *sequence* $(b_0, \dots, b_{r-1})$ such that the corresponding set $\{b_0, \dots, b_{r-1}\}$ is a base for G. An ordered base B gives rise to a chain of subgroups

$$G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(r)} = \langle 1 \rangle, \qquad 9.7.1$$

where

$$G^{(i+1)} = G^{(i)}_{b_i} \qquad 9.7.2$$

is the stabilizer of b_i in $G^{(i)}$. This is called the *stabilizer chain* (or *Sims chain*) for G with respect to B. The base is called *irredundant* if no two (consecutive) terms of the sequence of subgroups coincide.

The images of the base points determine a permutation in the following sense.

Lemma *Let G be a permutation group with base $B = (b_0, \dots, b_{r-1})$. Let g and h be two elements of G. Then $g = h$ if and only if $b_i g = b_i h$ for $i \in r$. In other words, knowing the images of all base points determines a permutation uniquely.* 9.7.3

Proof: The condition $b_i g = b_i h$ for $i \in r$ is equivalent to $b_i g h^{-1} = b_i$ for all $i \in r$, which in turn is equivalent to $gh^{-1} \in G^{(r)} = \langle 1 \rangle$, using the fact that B is a base. Thus $gh^{-1} = 1$, i.e. $g = h$. □

By 3.4.1, the cosets of $G^{(i+1)}$ in $G^{(i)}$ correspond to the different elements in the orbit

$$\mathcal{O}^{(i)} = G^{(i)}(b_i),$$

which we call the *i-th basic orbit*. In particular, since $G^{(i+1)}$ is a point stabilizer in $G^{(i)}$ by 9.7.2, the index satisfies

$$\left|G^{(i)}\right| / \left|G^{(i+1)}\right| = |\mathcal{O}^{(i)}| =: \ell_i$$

and hence by 9.7.1

9.7.4
$$|G| = \prod_{i \in r} \left|G^{(i)}\right| / \left|G^{(i+1)}\right| = \prod_{i \in r} \ell_i.$$

For $i \in r$, we choose *coset representatives* $\sigma_{i,0}, \ldots, \sigma_{i,\ell_i - 1}$ for $G^{(i+1)}$ in $G^{(i)}$, so that

9.7.5
$$G^{(i)} = \bigcup_{j \in \ell_i} G^{(i+1)} \sigma_{i,j}$$

is the decomposition of $G^{(i)}$ into cosets of $G^{(i+1)}$. We require that $\sigma_{i,0} = 1$, the identity element of G, for all $i \in r$. A *strong generating set* for G relative to B is a set S of elements of G with the property that

9.7.6
$$\langle S \cap G^{(i)} \rangle = G^{(i)} \quad \text{for } i \in r.$$

9.7.7 **Example** Consider the symmetric group $G = S_n$ acting on the set n. An ordered base for G is $B = (0, 1, \ldots, n-2)$. $G^{(i)}$ is isomorphic to S_{n-i} (acting on the set $\{i, \ldots, n-1\}$). The basic orbits are of length $\ell_i = n - i$. Coset representatives are $\sigma_{i,j} = (i, i+j)$ for $j \in \ell_i$ and $i \in n-1$. The sets

$$U = \{(0, 1, \ldots, n-1), (0, 1)\}$$

and

$$V = \{(0,1), (1,2), \ldots, (n-2, n-1)\}$$

both generate S_n. For $n \geq 3$, U is not a strong generating set (for example the group $G^{(1)}$, which is the symmetric group acting on $\{1, \ldots, n-1\}$ contains none of the generators). On the contrary, the generating set V is a strong generating set for all n. This is because $V \cap G^{(i)} = \{(i, i+1), \ldots, (n-2, n-1)\}$ generates S_{n-i} acting on the set $\{i, \ldots, n-1\}$. In fact, for each $n \geq 2$, the permutations $(i, i+1)$ for $i \in n-2$ form a strong generating set for S_n. ◇

The point of knowing a strong generating set S for a permutation group G is that the basic orbits $G^{(i)}(b_i)$ can be computed easily. Namely, it is straightforward to compute the subsets

$$S^{(i)} := S \cap G^{(i)}, \quad i \in r,$$

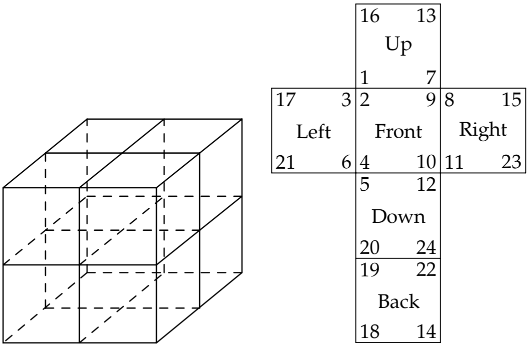


Fig. 9.10 Rubik's $2 \times 2 \times 2$ cube

which for fixed $i \in r$ contain those generators which fix the first i base points $b_0, \ldots, b_{i-1}$. Using the orbit algorithm of Section 9.2, one computes the corresponding basic orbit. From this orbit, coset representatives $\sigma_{i,0}, \ldots, \sigma_{i,\ell_i-1}$ can be determined (they are just the transporter elements of Section 9.2). The point is that the basic orbits and the corresponding Schreier trees can be constructed easily from the strong generating set. This is not the case for arbitrary generating sets, where one has to go through more complex algorithms, like the Schreier–Sims algorithm described in [91], for example. The difficulty lies in the fact that the basic orbits $\mathcal{O}^{(i)} = G^{(i)}(b_i)$ can only be computed when generators for $G^{(i)}$ are known. This explains why the set $S^{(i)}$ which generates $G^{(i)}$ is so valuable.

One further remark concerning the Schreier tree is in order. Recall that we require that $\sigma_{i,0} = 1$. This condition is automatically satisfied for Schreier trees, since the path from the root to itself corresponds to the empty word, which by definition is the identity element in the group. Let us consider another example.

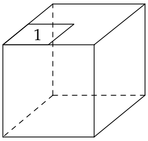
Example Figure 9.10 shows Rubik's cube in the simplified version with sides of length 2 instead of three. We label the faces with the integers in $\{1, \ldots, 24\}$ as indicated beneath. Here, we start labeling points from 1, since many current computer algebra systems have permutations act on $1, 2, 3, \ldots$ We will follow this convention throughout this example, for the sake of allowing the reader to verify the claims made by using a standard software package. **9.7.8**

Consider the group G which is generated by the rotations of the sides. We follow the widely accepted notation due to Singmaster (cf. [183]), which denotes the quarter turns in clockwise direction of the left, right, front, back, up and down side of the cube by L, R, F, B, U and D, respectively. However, we stick to the notation $A^{-1}, A^{-2}, \ldots$ for the inverse, the square of the inverse etc. of the element A (as opposed to using A' for the inverse of A which is sometimes used). The permutations which correspond to these six generators are

$$\begin{aligned}
R &= (7,14,24,10)(8,15,23,11)(9,13,22,12)\\
B &= (13,17,20,23)(14,18,19,22)(16,21,24,15)\\
D &= (4,11,22,21)(5,12,24,20)(6,10,23,19),\\
L &= (1,4,20,18)(2,5,19,16)(3,6,21,17),\\
F &= (1,8,12,6)(2,9,10,4)(3,7,11,5),\\
U &= (1,16,13,7)(2,17,14,8)(3,18,15,9).
\end{aligned}$$

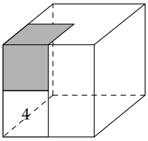
An ordered base for the group G is $(1,4,7,10,13,16,19)$. We get the following stabilizer chain, where we indicate the length of the fundamental orbit in parenthesis and where the grey area in the pictures indicates faces which have been stabilized.

$$G = G^{(0)}$$



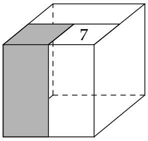
(24)

$$\geq \ \left(G^{(1)} = \right) G_1$$



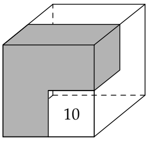
(21)

$$\geq \ \left(G^{(2)} = \right) G_{1,4}$$



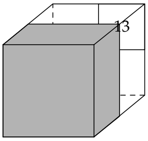
(18)

$$\geq \ \left(G^{(3)} = \right) G_{1,4,7}$$



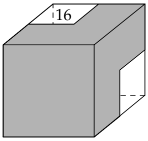
(15)

$$\geq \ \left(G^{(4)} = \right) G_{1,4,7,10}$$



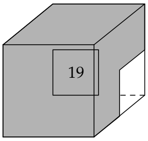
(12)

$$\geq \ \big(G^{(5)} = \big)G_{1,4,7,10,13}$$



(9)

$$\geq \ \big(G^{(6)} = \big)G_{1,4,7,10,13,16}$$



(6)

$$\geq \ \big(G^{(7)} = \big)G_{1,4,7,10,13,16,19} = 1.$$

Hence by 9.7.4, the order of G (i.e. the number of positions) is

$$24 \times 21 \times 18 \times 15 \times 12 \times 9 \times 6 = 88\,179\,840.$$

Note that the generating set $\{L, R, F, B, U, D\}$ for G is *not* strong. A strong generating set can be found by considering moves which fix the grey part and permute the remaining faces among themselves. The point is that these moves may bring the grey part into disarray for a while. However, at the end of the move the grey faces are brought back into place. By computing Schreier trees it can be checked that

$$S = \{\nu, \tau, \delta, B, \omega, R, D, L\}$$

is a strong generating set, where

$$\begin{aligned}
\tau &= (BLFRD)^3 = (19,24)(20,22)(21,23),\\
\rho &= DFU^{-1}R^{-1}UFD^{-1}F^{-1} = (4,5,6)(7,9,8)(10,12,11)(19,21,20),\\
\nu &= \rho^2 B^{-1}\rho B = (19,20,21)(22,24,23),\\
\delta &= B\tau B^{-1} = (16,20)(17,19)(18,21),\\
\omega &= DBD^{-1}B^{-1} = (10,23,11,22,12,24)(16,19,18,20,17,21).
\end{aligned}$$

We find that

$$\begin{aligned}
G^{(6)} = G_{1,4,7,10,13,16} &= \langle \nu, \tau \rangle,\\
G^{(5)} = G_{1,4,7,10,13} &= \langle \nu, \tau, \delta \rangle,\\
G^{(4)} = G_{1,4,7,10} &= \langle \nu, \tau, \delta, B \rangle,\\
G^{(3)} = G_{1,4,7} &= \langle \nu, \tau, \delta, B, \omega \rangle,\\
G^{(2)} = G_{1,4} &= \langle \nu, \tau, \delta, B, \omega, R \rangle,\\
G^{(1)} = G_{1} &= \langle \nu, \tau, \delta, B, \omega, R, D \rangle,\\
G^{(0)} = G &= \langle \nu, \tau, \delta, B, \omega, R, D, L \rangle,
\end{aligned}$$

which are groups of order 6, 54, 648, 9720, 174 960, 3 674 160 and 88 179 840, respectively. More details on the group of Rubik's cube (in particular, the version with sides of length 3) can be found in the books by Neumann, Stoy and Thompson [158] and in the above-mentioned book by Singmaster [183]. ◇

Our next goal is to identify group elements with integers, using a known stabilizer chain for the permutation group. This serves two purposes. Firstly, it is convenient, as integers are often easier to handle in computer programs. Secondly, this enables us to pick group elements uniformly at random, which is useful for randomized algorithms for permutation groups. To begin with, let us introduce the multibase representation of an integer.

9.7.9 **Lemma** *Let $L = (\ell_0, \dots, \ell_{r-1})$ be a sequence of positive integers and define $m = \prod_{i \in r} \ell_i$. Any integer n in $m = \{0, \dots, m-1\}$ has a unique representation of the form*

$$n = \sum_{i \in r} a_i \prod_{j \in i} \ell_j$$

with integers $a_i \in \ell_i$ for $i \in r$ (here, an empty product is defined to be 1). We write

$$n = (a_{r-1}, \dots, a_0)_L$$

and call this the multibase representation *of n with respect to B.*

Proof: Put $m_i = \prod_{j \in i} \ell_j$ for $i \in r+1$, i.e. $m_r = m$.
Existence: If $r = 1$ we may put $a_0 = n$ and we are finished. Thus let us assume that $r \geq 2$. Given $n = n_{r-1}$ with $n \in m$, integral division yields unique integers $a_{r-1} \geq 0$ and n_{r-2} with

$$n = n_{r-1} = a_{r-1} m_{r-1} + n_{r-2} \quad \text{with } n_{r-2} \in m_{r-1}.$$

Here we have $a_{r-1} = \lfloor n_{r-1}/m_{r-1} \rfloor$, and since $n_{r-1} = n < m = m_{r-1}\ell_{r-1}$ we have $a_{r-1} \in \ell_{r-1}$. If $r \geq 3$, we may repeat this argument for n_{r-2} and obtain an equation of the form

$$n_{r-2} = a_{r-2} m_{r-2} + n_{r-3} \quad \text{with } n_{r-3} \in m_{r-2} \text{ and } a_{r-2} \geq 0.$$

Here we have $a_{r-2} = \lfloor n_{r-2}/m_{r-2} \rfloor$, and since $n_{r-2} < m_{r-1} = m_{r-2}\ell_{r-2}$ we have $a_{r-2} \in \ell_{r-2}$. If we proceed in this way, we define integers $a_i \in \ell_i$ and $n_{i-1} \in m_i$. Eventually we arrive at an equation of the form

$$n_1 = a_1 m_1 + n_0 \quad \text{with } n_0 \in m_1 \text{ and } a_1 \in \ell_1.$$

Note that by definition $m_1 = \ell_0$, so that we may simply put $a_0 = n_0 \in m_1 = \ell_0$. Thus, we have written n as

$$\begin{aligned} n &= n_{r-1} \\ &= a_{r-1} m_{r-1} + n_{r-2} \\ &= a_{r-1} m_{r-1} + a_{r-2} m_{r-2} + n_{r-3} \\ &\vdots \\ &= \sum_{i \in r} a_i m_i. \end{aligned}$$

Uniqueness: Let

$$(a_{r-1},\dots,a_0)_L = n = (b_{r-1},\dots,b_0)_L$$

be two expressions for n. Subtraction yields

$$0 = \sum_{i\in r}(b_i - a_i)m_i.$$

Put $\Delta_i := b_i - a_i$. Let j be such that $\Delta_j \neq 0$ (such an index j exists if we assume that the expressions are distinct). Therefore

$$\Delta_j m_j = -\sum_{\substack{i\in r\\ i\neq j}} \Delta_i m_i. \qquad \textbf{9.7.10}$$

Notice that

$$|\Delta_i| \leq b_i < \ell_i, \quad i \in r. \qquad \textbf{9.7.11}$$

If $j < r-1$, we may consider 9.7.10 modulo m_{j+1} to get

$$\Delta_j m_j \equiv -\sum_{i\in j-1} \Delta_i m_i \bmod m_{j+1}. \qquad \textbf{9.7.12}$$

Using 9.7.11 we get that

$$|\Delta_i| m_i \leq (\ell_i - 1)m_i = \ell_i m_i - m_i = m_{i+1} - m_i.$$

Therefore, over the integers, the right hand side of 9.7.12 is bounded above by

$$\left|\sum_{i\in j-1} \Delta_i m_i\right| \leq \sum_{i\in j-1} |\Delta_i| m_i \leq \sum_{i\in j-1} \big(m_{i+1} - m_i\big) = m_j - m_0 = m_j - 1 < m_j.$$

But $\Delta_j \neq 0$, which means that 9.7.12 has no solution modulo m_{j+1}. Hence $\Delta_j \neq 0$ is impossible. If $j = r-1$, 9.7.10 becomes

$$\Delta_{r-1} m_{r-1} = -\sum_{i\in r-1} \Delta_i m_i.$$

The same argument as before shows that the absolute value of the right hand side of this equation is bounded above by m_{r-1}, which contradicts the fact that Δ_{r-1} is nonzero. These contradictions show that the multibase representation is unique. □

We introduce some more notation. For a sequence $L = (\ell_0,\dots,\ell_{r-1})$, let

$$\overleftarrow{L} = (\ell_{r-1},\dots,\ell_0)$$

be the *reversed sequence*. The following result enables us to identify group elements with integers.

9.7.13 **Lemma** *Let the group G be of order $|G|$ with base $B = (b_0, \dots, b_{r-1})$ and basic orbits of lengths $|G^{(i)}(b_i)| = \ell_i$, $i \in r$. Furthermore, assume that coset representatives $\sigma_{i,j}$ for $j \in \ell_i$, $i \in r$ have been chosen. Put $L = (\ell_0, \dots, \ell_{r-1})$. Define a map*

$$\mathrm{rk}^{-1} \colon |G| \to G \colon n \mapsto \sigma_{r-1,a_0}\sigma_{r-2,a_1} \cdots \sigma_{0,a_{r-1}},$$

where $(a_{r-1}, \dots, a_0)_{\overleftarrow{L}}$ is the multibase representation of n with respect to $\overleftarrow{L}$. This map is bijective, we call it the unrank *function for G. Its inverse is the* rank *function for G.*

Proof: By 3.4.1, each element $g \in G^{(0)} = G$ can be written as

$$g = g^{(1)}\sigma_{0,a_{r-1}},$$

for a uniquely determined coset representative $\sigma_{0,a_{r-1}}$, $a_{r-1} in \ell_0$ and a unique element $g^{(1)} \in G^{(1)}$. Repeating this argument for $g^{(1)}$ yields a unique coset representative $\sigma_{1,a_{r-2}}$, $a_{r-2} \in \ell_1$, and a unique element $g^{(2)} \in G^{(2)}$ such that

$$g^{(1)} = g^{(2)}\sigma_{1,a_{r-2}}.$$

If $r > 2$, we may proceed in this fashion. In the i-th step we find an equation of the form

$$g^{(i)} = g^{(i+1)}\sigma_{i,a_{r-1-i}},$$

for some unique elements $g^{(i+1)}$ and $\sigma_{i,a_{r-1-i}}$, $a_{r-1-i} \in \ell_i$. This process terminates once we reach

$$g^{(r-1)} = g^{(r)}\sigma_{r-1,a_0},$$

with $a_0 \in \ell_{r-1}$, since then $g^{(r)} \in G^{(r)} = 1$, the trivial group, i.e. $g^{(r)} = 1$. Back-substituting the equations into each other gives

$$\begin{aligned} g &= g^{(1)}\sigma_{0,a_{r-1}} \\ &= g^{(2)}\sigma_{1,a_{r-2}}\sigma_{0,a_{r-1}} \\ &\vdots \\ &= \sigma_{r-1,a_0}\sigma_{r-2,a_1} \cdots \sigma_{1,a_{r-2}}\sigma_{0,a_{r-1}}, \end{aligned}$$

with $a_i \leq \ell_{r-1-i}$ for $i \in r$. This means that we are able to write the given group element g in a unique way as a product of coset representatives. In the literature, the indicated process is known as the *sift algorithm*. To turn this representation into a number, we simply consider the sequence $a_0, \dots, a_{r-1}$ as multibase representation

$$(a_{r-1}, a_{r-2}, \dots, a_0)_{\overleftarrow{L}} = n$$

of some integer $n \in |G| = \prod_{i \in r} \ell_i$. This process defines a rank function on the set of group elements. In fact, this function is bijective because different group elements give different factorizations and hence different multibase representations of numbers. The inverse process gives the unrank function. □

Table 9.3 Unranking the elements of S_3

n	$(a_1, a_0)_{(2,3)}$	$\mathrm{rk}^{-1}(n) = \sigma_{1,a_0}\sigma_{0,a_1}$
0	$(0,0)$	$1 = 1 \cdot 1$
1	$(0,1)$	$(1,2) = (1,2) \cdot 1$
2	$(1,0)$	$(0,1) = 1 \cdot (0,1)$
3	$(1,1)$	$(0,1,2) = (1,2) \cdot (0,1)$
4	$(2,0)$	$(0,2) = 1 \cdot (0,2)$
5	$(2,1)$	$(0,2,1) = (1,2) \cdot (0,2)$

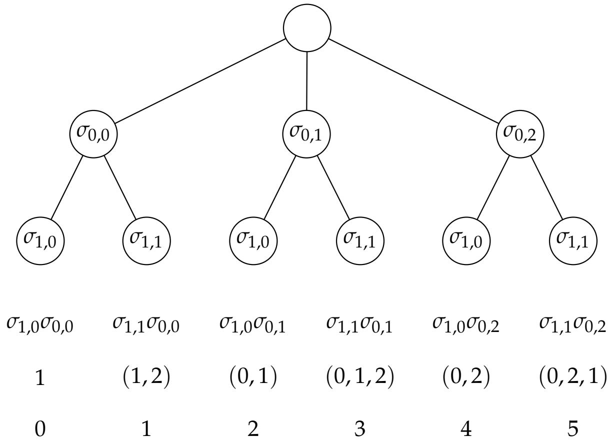


Fig. 9.11 The elements of S_3 by rank

We remark that the order of the terms in the function rk^{-1} of 9.7.13 matters, since we do not require the group to be abelian.

9.7.14 **Example** Consider the symmetric group S_3 acting on $\{0,1,2\}$ with base $B = (0,1)$. The basic orbits are of length $\ell_0 = 3$ and $\ell_1 = 2$. Hence $\overleftarrow{L}=\overleftarrow{(3,2)}= (2,3)$. Coset representatives are

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

The unrank function lists the elements in the order indicated in Table 9.3. The ordering may be visualized as in Fig. 9.11. The coset representatives are shown as the nodes of a tree. The leaves stand for elements of the group. The corresponding permutations and their ranks are shown at the bottom. ◇

We are now in a position to define another important graph associated to a group. If G is a group and if S is a set of elements of G, the *Cayley-graph* of G

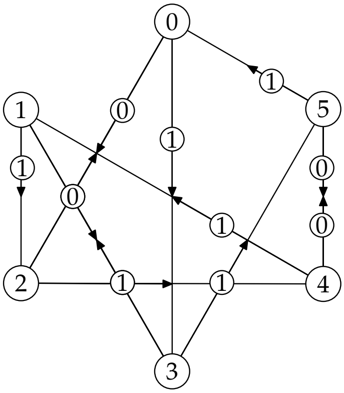


Fig. 9.12 The Cayley-graph of S_3

with respect to S is the action-graph whose vertices are the elements of G and whose edges are defined by the right-multiplication by elements $s \in S$. That is, the Cayley-graph of G with respect to S has an edge from x to y labeled by $s_i \in S$ if $xs_i = y$ holds in G. Figure 9.12 shows the Cayley graph of S_3 with respect to the generating set $S = \{s_0, s_1\}$ where $s_0 = (0, 1, 2)$ and $s_1 = (0, 1)$. Cayley graphs are often used to investigate combinatorial problems theoretically, and they can also be useful for studying the concepts defined in this section.

9.7.15 **Example** Consider the Cayley graph of Rubik's cube. As noted above, for the $2 \times 2 \times 2$ cube, we may assume that one corner is fixed, for instance the front-top-left corner $1, 2, 3$. That leaves only the generators R, D and B as well as their inverses. We consider the Cayley graph of $G^{(1)} = G_1$ (of order 3 674 160, see above) with respect to these 6 generators. Cayley graphs admit the defining group as vertex transitive automorphism group. Therefore, in order to compute the diameter of the graph it suffices to compute the distance of the vertex furthest away from a given vertex. If Γ_i is the set of vertices at distance i from the identity node, then Jianyi Yao, a student at Colorado State University, reports the following numbers:

i	$\|\Gamma_i\|$
0	1
1	6
2	27
3	120
4	534

i	$\|\Gamma_i\|$
5	2256
6	8969
7	33058
8	114149
9	360508

i	$\|\Gamma_i\|$
10	930588
11	1350852
12	782536
13	90280
14	276

In particular, this means that there are 276 "worst case" positions, i.e. positions which can be restored with no less than 14 quarter turns. This agrees with results obtained by Cooperman et al. [41], which report that the diameter of this graph is 14. ◇

Summarizing, the concept of base and strong generating set defines a new data structure for permutation groups. This data structure is based on the stabilizer chain corresponding to the base. To represent that chain, one needs one Schreier tree for each basic orbit. At any particular level, one obtains coset representatives for the next subgroup in the chain from the Schreier tree. Using a fixed ordering of these representatives, 9.7.13 allows one to access group elements numerically. For further details on working with stabilizer chains, we refer to the above-mentioned books by Holt [91] and by Seress [177]. We only mention that randomization plays a key role in those algorithms.

Exercises

Exercise Let G be a finite group and let S be a subset of G. Show the following. **E.9.7.1**

1. The Cayley-graph of G with respect to S is connected if and only if S generates G.
2. The Cayley-graph is undirected (i.e., (u, v) is an edge whenever (v, u) is an edge) if and only if S is closed under inverses (i.e., $s \in S \Leftrightarrow s^{-1} \in S$). In particular, this is the case if S consists of involutions, i.e. elements of order 2.

9.8 The Projective Linear Group

9.8

The goal of this section is to describe a stabilizer chain for $\mathrm{PGL}_k(q)$, the projective linear group of $\mathrm{PG}_{k-1}(q)$. We will find a base for this group, and we will list the coset representatives $\sigma_{i,j}$ explicitly. This leads us to determine a strong generating set. In the same vein, we will also treat the projective semilinear group $\mathrm{P\Gamma L}_k(q)$ in the following section.

For $-1 \leq s < d$, define the set

$$\mathrm{PG}_{d\backslash s}(q) = \{\langle u \rangle \in \mathrm{PG}_d(q) \mid \mathrm{lc}(u) > s\}.$$

We also put

$$\theta_{d\backslash s}(q) = |\,\mathrm{PG}_{d\backslash s}(q)| = \theta_d(q) - \theta_s(q) = \frac{q^{d+1} - q^{s+1}}{q-1},$$

with $\theta_{-1}(q) = 0$. As usual, we rank and unrank the elements of this set.

9.8.1 **Lemma** *Let d, s and q be given, where q is a prime power and $-1 \leq s < d$. Define a map $\mathrm{rk}^{-1}_{d\setminus s;q}$ from $\theta_{d\setminus s}(q)$ to $\mathrm{PG}_{d\setminus s}(q)$ by*

$$\mathrm{rk}^{-1}_{d\setminus s;q}(n) = \begin{cases} \langle e^{(s+1+n)} \rangle & \text{if } n \leq d-s-1 \\ \langle \sum_{i=0}^{d} e^{(i)} \rangle & \text{if } n = d-s \\ \langle \mathrm{rk}^{-1}_{d,s+1;q}(n-d+s) \rangle & \text{otherwise,} \end{cases}$$

where $\mathrm{rk}^{-1}_{d,s+1;q}$ is the function of 9.3.7. The map $\mathrm{rk}^{-1}_{d\setminus s;q}$ is a bijection, we call it the unrank function *for $\mathrm{PG}_{d\setminus s}(q)$. Its inverse is the rank function for $\mathrm{PG}_{d\setminus s}(q)$, denoted as $\mathrm{rk}_{d\setminus s;q}$. For a point $\langle u \rangle \in \mathrm{PG}_{d\setminus s}(q)$, with $u = (u_0, u_1, \ldots, u_d) \in \mathbb{F}_q^{d+1} \setminus \{0\}$ one has $\mathrm{rk}_{d\setminus s;q}(\langle u \rangle) =$*

$$\begin{cases} k & \text{if } \langle u \rangle = \langle e^{(s+1+k)} \rangle \\ d-s & \text{if } \langle u \rangle = \langle 1, \ldots, 1 \rangle \\ d+1-k+\frac{q^k-q^{s+1}}{q-1} + \mathrm{rk}_{k,q}\left(\frac{u_0}{u_k}, \ldots, \frac{u_{k-1}}{u_k}\right) & \text{if } k = \mathrm{lc}(u) < d \\ 1+\frac{q^d-q^{s+1}}{q-1} + \mathrm{shift}^{-1}_{\theta_{d-1}(q)}\left(\mathrm{rk}_{d,q}\left(\frac{u_0}{u_d}, \ldots, \frac{u_{d-1}}{u_d}\right)\right) & \text{if } \mathrm{lc}(u) = d. \end{cases} \quad (9.8.2)$$

For $s = -1$, we get the ordinary unrank function back, i.e.

$$\mathrm{rk}_{d\setminus -1;q} = \mathrm{rk}_{d;q} \text{ and } \mathrm{rk}^{-1}_{d\setminus -1;q} = \mathrm{rk}^{-1}_{d;q}. \qquad \square$$

9.8.3 **Example** We have $\theta_{2\setminus -1}(3) = 13$, $\theta_{2\setminus 0}(3) = 13-1 = 12$, $\theta_{2\setminus 1}(3) = 13-4 = 9$. Table 9.4 shows the functions $\mathrm{rk}_{2\setminus s;3}(x)$ for $-1 \leq s \leq 1$. ◇

9.8.4 **Example** We have $\theta_{3\setminus -1}(2) = 15$, $\theta_{3\setminus 0}(2) = 14$, $\theta_{3\setminus 1}(2) = 12$ and $\theta_{3\setminus 2}(2) = 8$. Table 9.5 shows the functions $\mathrm{rk}_{3\setminus s;2}(x)$ for $-1 \leq s \leq 2$. ◇

Let us introduce some notation for special kinds of matrices. We denote by $F_{n,i}$ the $(n \times (n+1))$ matrix which is obtained from the identity matrix I_{n+1} by removing the i-th row. In other words, we put

$$F_{n,i} = \left(\begin{array}{c|c|c} I_i & \mathbf{0}_i^\top & 0 \\ \hline 0 & \mathbf{0}_{n-i}^\top & I_{n-i} \end{array}\right) = \left(\begin{array}{ccc|c|ccc} 1 & & & 0 & & & \\ & \ddots & & \vdots & & 0 & \\ & & 1 & 0 & & & \\ \hline & & & 0 & 1 & & \\ & 0 & & \vdots & & \ddots & \\ & & & 0 & & & 1 \end{array}\right),$$

a matrix whose i-th column is zero. In addition, let $E_{u,v}$ be the $k \times k$ matrix whose only nonzero entry is in the (u,v)-position, with value one. Formally

$$E_{u,v} = (\delta_{i,u}\delta_{v,j})_{i \in k, j \in k}.$$

Table 9.4 The functions $\mathrm{rk}_{2\backslash s;3}(\langle x\rangle)$ for $\langle x\rangle \in \mathrm{PG}_2(3)$

	$\mathrm{rk}_{2\backslash s;3}(\langle x\rangle)$		
$\langle x\rangle \in \mathrm{PG}_2(3)$	$s=-1$	$s=0$	$s=1$
$\langle 1,0,0\rangle$	0		
$\langle 0,1,0\rangle$	1	0	
$\langle 0,0,1\rangle$	2	1	0
$\langle 1,1,1\rangle$	3	2	1
$\langle 1,1,0\rangle$	4	3	
$\langle 2,1,0\rangle$	5	4	
$\langle 1,0,1\rangle$	6	5	2
$\langle 2,0,1\rangle$	7	6	3
$\langle 0,1,1\rangle$	8	7	4
$\langle 2,1,1\rangle$	9	8	5
$\langle 0,2,1\rangle$	10	9	6
$\langle 1,2,1\rangle$	11	10	7
$\langle 2,2,1\rangle$	12	11	8

Table 9.5 The functions $\mathrm{rk}_{3\backslash s;2}(\langle x\rangle)$ for $\langle x\rangle \in \mathrm{PG}_3(2)$

	$\mathrm{rk}_{3\backslash s;2}(\langle x\rangle)$			
$\langle x\rangle \in \mathrm{PG}_3(2)$	$s=-1$	$s=0$	$s=1$	$s=2$
$\langle 1,0,0,0\rangle$	0			
$\langle 0,1,0,0\rangle$	1	0		
$\langle 0,0,1,0\rangle$	2	1	0	
$\langle 0,0,0,1\rangle$	3	2	1	0
$\langle 1,1,1,1\rangle$	4	3	2	1
$\langle 1,1,0,0\rangle$	5	4		
$\langle 1,0,1,0\rangle$	6	5	3	
$\langle 0,1,1,0\rangle$	7	6	4	
$\langle 1,1,1,0\rangle$	8	7	5	
$\langle 1,0,0,1\rangle$	9	8	6	2
$\langle 0,1,0,1\rangle$	10	9	7	3
$\langle 1,1,0,1\rangle$	11	10	8	4
$\langle 0,0,1,1\rangle$	12	11	9	5
$\langle 1,0,1,1\rangle$	13	12	10	6
$\langle 0,1,1,1\rangle$	14	13	11	7

Lastly, we introduce the 2 × 2-matrix

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

The next result describes a base and strong generating set for the projective linear group $\mathrm{PGL}_k(q)$ in the standard action on $\mathrm{PG}_{k-1}(q)$. For sake of simplicity, we do not distinguish in our notation between the matrices and the induced permutations on the projective space. Also we let group elements be denoted either by matrices or by the corresponding permutations.

9.8.5 **Theorem (base and strong generating set for $\mathrm{PGL}_k(q)$)** *Let $q = p^h$ with p prime and h a positive integer. Let $\mathrm{PG}_{k-1}(q)$ be the one-dimensional subspaces of the vector space $V = \mathbb{F}_q^k$ with basis $e^{(0)}, \ldots, e^{(k-1)}$. Assume that $\kappa_0, \kappa_1, \ldots, \kappa_{q-1}$ are the elements of the field $\mathbb{F}_q$, ordered in such a way that $\kappa_0 = 0$ and $\kappa_1 = 1$.*

1. *For $i \in k+1$, let*

$$b_i := \begin{cases} \langle e^{(i)} \rangle & \textit{if } i < k, \\ \langle \sum\limits_{i \in k} e^{(i)} \rangle & \textit{if } i = k. \end{cases}$$

The sequence $B = (b_0, \ldots, b_k)$ is a base for $\mathrm{PGL}_k(q)$ acting on $\mathrm{PG}_{k-1}(q)$. The corresponding stabilizer chain has basic orbits of lengths

$$\ell_i = \begin{cases} \theta_{k-1 \backslash i-1}(q) & \textit{for} \quad i \in k, \\ (q-1)^{k-1} & \textit{for} \quad i = k. \end{cases}$$

2. *Coset representatives can be chosen as follows.*
   (a) For $i \in k$, and for $j \in \ell_i$, let

$$\sigma_{i,j} = \left(\begin{array}{c|c} I_i & 0 \\ \hline \multicolumn{2}{c}{v} \\ \hline 0 & F_{k-i-1,s-i} \end{array} \right),$$

   where $\langle v \rangle = \mathrm{rk}^{-1}_{k-1 \backslash (i-1);q}(j)$ and $s = \mathrm{lc}(v) \geq i$.

   (b) For $j \in \ell_k$, define

$$\sigma_{k,j} = \mathrm{diag}(1, \kappa_{a_0+1}, \ldots, \kappa_{a_{k-2}+1}),$$

   where $j = (a_{k-2}, \ldots, a_0)_{q-1}$ is the base $(q-1)$ representation of j.

   If $q = 2$, the base point b_k is redundant.

3. *A strong generating set for $\mathrm{PGL}_k(q)$ is the set*

9.8.6 $$S = \Big\{ \mathcal{P}_0, \ldots, \mathcal{P}_{k-2}, \mathcal{E}_{r,j}, \mathcal{D}_1, \ldots, \mathcal{D}_{k-1} \;\Big|\; r \in h,\ j \in k-1 \Big\},$$

where

$$\begin{aligned}\mathcal{P}_i &= \begin{pmatrix} I_i & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & I_{k-2-i} \end{pmatrix}, && \textbf{9.8.7}\\ \mathcal{E}_{r,j} &= I_k + \beta_r E_{k-1,j}, && \textbf{9.8.8}\\ \mathcal{D}_i &= I_k + (\alpha - 1)E_{i,i}. && \textbf{9.8.9}\end{aligned}$$

Here, $(\beta_0, \ldots, \beta_{h-1})$ is an $\mathbb{F}_p$-basis for $\mathbb{F}_q$ (as vector space over $\mathbb{F}_p$) and α is a primitive element for $\mathbb{F}_q$, i.e. a generator of the multiplicative group $\mathbb{F}_q^*$. If $q = 2$, the elements $\mathcal{D}_i$ of 9.8.9 are all equal to I_k and may be omitted from the set S.

Proof: The pointwise stabilizer in $\mathrm{GL}_k(q)$ of the unit vectors $e^{(0)}, \ldots, e^{(k-1)}$ consists of the diagonal matrices with nonzero determinant. These are just the diagonal matrices whose diagonal entries are all nonzero. The stabilizer of the unit vectors and the vector $e^{(0)} + \ldots + e^{(k-1)}$ are the matrices of the center $\mathcal{Z}_k$, defined in 3.7.5, i.e. the matrices of the form λI_k where $\lambda \in \mathbb{F}_q^*$. Hence in the factor group $\mathrm{PGL}_k(q) = \mathrm{GL}_k(q)/\mathcal{Z}_k$, only the identity element stabilizes

$$b_0 = \langle e^{(0)} \rangle, \ldots, b_{k-1} = \langle e^{(k-1)} \rangle, \text{ and } b_k = \langle e^{(0)} + \ldots + e^{(k-1)} \rangle.$$

This shows that B is a base. The statement about the lengths of the basic orbits will follow once we have verified that the given coset representatives are a transversal for $G^{(i+1)}$ in $G^{(i)}$. For $i = 0$, we consider matrices of the form

$$\sigma_{0,j} = \left(\frac{v}{F_{k-1,s}} \right), \quad j \in \ell_0,$$

where $\langle v \rangle = \mathrm{rk}_{k-1;q}^{-1}(j)$ and where $s = \mathrm{lc}(v)$. Developing the determinant of $\sigma_{0,j}$ along the nonzero entries of the matrix $F_{k-1,s}$ leaves a nonzero one by one matrix as last term. Thus $\sigma_{0,j}$ is an element of $\mathrm{PGL}_k(q)$. The fact that we can put any element $\langle v \rangle$ of $\mathrm{PG}_{k-1}(q)$ into the first row of the coset representative means that $\mathrm{PGL}_k(q)$ is transitive on the set of points of $\mathrm{PG}_{k-1}(q)$. Thus

$$\ell_0 = \theta_{k-1}(q) = \theta_{k-1\backslash -1}(q) = \frac{q^k - 1}{q - 1}.$$

Next, consider the case where $0 < i < k$. Elements in $G^{(i)}$ stabilize pointwise the base points $b_0, \ldots, b_{i-1}$, which means that they fix the subspaces

$$\langle e^{(0)} \rangle, \ldots, \langle e^{(i-1)} \rangle$$

spanned by the first i unit vectors. Since the diagonal matrices are in $G^{(i+1)}$, we may choose these unit vectors themselves for the first i rows of $\sigma_{i,j}$, so that

$$\sigma_{i,j} = \left(\begin{array}{c|c} I_i & 0 \\ \hline * & * \end{array}\right).$$

The i-th row of $\sigma_{i,j}$ is the image $\langle v \rangle$ of $b_i = \langle e^{(i)} \rangle$ under $\sigma_{i,j}$. In order to make $\sigma_{i,j}$ invertible, v must not lie in the span of $e^{(0)}, \ldots, e^{(i-1)}$. Thus $\mathrm{lc}(v) \geq i$, i.e. $\langle v \rangle \in \mathrm{PG}_{k-1\backslash i-1}(q)$. For $j \in \theta_{k-1\backslash i-1}(q) = \ell_i$ we may take

$$\langle v \rangle = \mathrm{rk}^{-1}_{k-1\backslash i-1;q}(j),$$

so that

$$\sigma_{i,j} = \left(\begin{array}{c|c} I_i & 0 \\ \hline \multicolumn{2}{c}{v} \\ \hline 0 & F_{k-i-1,s-i} \end{array}\right).$$

By computing the determinant one verifies that this matrix $\sigma_{i,j}$ is invertible, provided that $s = \mathrm{lc}(v)$. This shows that the given set of matrices $\sigma_{i,j}$ form coset representatives for $G^{(i+1)}$ in $G^{(i)}$. Also, the lengths of the basic orbits are $\ell_i = \theta_{k-1\backslash i-1}(q)$.

For $i = k$ we need coset representatives for $G^{(k+1)}$ in $G^{(k)}$. Recall that $G^{(k)}$ is the group of diagonal matrices (modulo scalars, i.e. modulo $\mathcal{Z}_k$) whereas $G^{(k+1)}$ is the identity modulo $\mathcal{Z}_k$. Thus coset representatives for $G^{(k+1)}$ in $G^{(k)}$ are diagonal matrices with nonzero elements on the diagonal. Modulo $\mathcal{Z}_k$, we may choose representatives of the form

$$\mathrm{diag}(1, \lambda_1, \ldots, \lambda_{k-1}),$$

where $\lambda_1, \ldots, \lambda_{k-1}$ are nonzero field elements which can be chosen independently. This shows that $\ell_k = (q-1)^{k-1}$. We consider the map which takes an integer $j \in (q-1)^{k-1}$ to the matrix

$$\mathrm{diag}(1, \kappa_{a_0+1}, \ldots, \kappa_{a_{k-2}+1}) \in \mathrm{PGL}_k(q),$$

where

$$j = (a_{k-2}, \ldots, a_0)_{q-1}$$

is the base $q-1$ representation of j. Since $\kappa_{a_i+1} \neq 0$ (recall that we require that $\kappa_0 = 0$ and $\kappa_u \neq 0$ for $u > 0$), this map is a bijection onto the mentioned set of coset representatives for $G^{(k+1)}$ in $G^{(k)}$. This finishes the proof of the first two parts of the theorem.

Let us now verify that the set given in 9.8.6 is a strong generating set for $\mathrm{PGL}_k(q)$. This is proved inductively, going from the small groups to the larger ones in the stabilizer chain, i.e. from the large indices to the smaller ones. Recall that we have set

$$S^{(i)} = S \cap G^{(i)}$$

for $i \in k+1$. Showing that the generating sets $S^{(i)}$ for $G^{(i)}$ are strong can be done by induction. We put

$$H^{(i)} = \langle S^{(i)} \rangle \leq G^{(i)}, \quad i \in k+1,$$

and then show that $H^{(i)} = G^{(i)}$. In each step we need to show that

$$|H^{(i)}(b_i)| = \ell_i = |G^{(i)}(b_i)|,$$

since then by 3.4.1 and by induction hypothesis,

$$|H^{(i)}| = |H^{(i+1)}| \cdot \ell_i = |G^{(i+1)}| \cdot \ell_i = |G^{(i+1)})|$$

and therefore $H^{(i)} = G^{(i)}$.

The statement is clear for $i = k+1$, since $S^{(k+1)} = \emptyset$ and hence $H^{(k+1)} = G^{(k+1)} = 1$. For $i = k$,

$$S^{(k)} = S \cap G^{(k)} = \{\mathcal{D}_j \mid 1 \leq j < k\}.$$

Modulo $\mathcal{Z}_k$, every diagonal matrix can be written as a product of (powers of) suitable $\mathcal{D}_j$. This shows that $G^{(k)} = H^{(k)} = \langle S^{(k)} \rangle$.

The set $S^{(k-1)} = S \cap G^{(k-1)}$ is

$$S^{(k-1)} \ = \ S^{(k)} \cup \{\mathcal{E}_{r,j} \mid r \in h, \ j \in k-1\},$$

with $\mathcal{E}_{r,j}$ as in 9.8.8. Written out, we have

$$\mathcal{E}_{r,j} = \left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline v' & 1 \end{array} \right),$$

with

$$v' = \beta_r e^{(j)} \in \mathbb{F}_q^{k-1} \quad \text{for } r \in h, \ j \in k-1.$$

Now consider the basic orbit $G^{(k-1)}(b_{k-1})$. This is just the set

$$\mathrm{PG}_{k-1\backslash k-2}(q) = \big\{ \langle v \rangle \in \mathrm{PG}_{k-1}(q) \mid \mathrm{lc}(v) = k-1 \big\}.$$

Thus,

$$v = (v_0, \ldots, v_{k-2}, 1) = (v', 1)$$

with $v' = (v_0, \ldots, v_{k-2}) \in \mathbb{F}_q^{k-1}$ arbitrary. Notice that if $w = (w', 1)$ is another vector with $w' = (w_0, \ldots, w_{k-2}) \in \mathbb{F}_q^{k-1}$, then the corresponding coset representatives multiply as follows

$$\left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline v' & 1 \end{array} \right) \cdot \left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline w' & 1 \end{array} \right) = \left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline v' + w' & 1 \end{array} \right).$$

This shows that in the factor group $G^{(k)}$ modulo $G^{(k+1)}$, multiplication of coset representatives

$$\sigma_{k,j} = \left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline \multicolumn{2}{c}{v} \end{array}\right) = \left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline v' & 1 \end{array}\right)$$

results in addition of the first $k-1$ components of the vectors in the last rows. In particular, the coset representatives form a group by themselves (i.e. a "complement" of $G^{(k)}$ in $G^{(k-1)}$). It is clear that the first $k-1$ components form an additive group $\mathbb{F}_q^{k-1}$. Furthermore, since $\mathbb{F}_q \simeq \mathbb{F}_p^h$ (as additive groups), we have the isomorphism from the group of coset representatives onto $\mathbb{F}_q^{k-1} \simeq \mathbb{F}_p^{h(k-1)}$. Therefore, a basis for the group of coset representatives is given by the matrices $\mathcal{E}_{r,j}$, where $r \in h$ and $j \in k-1$. But these are exactly the elements of $S^{(k-1)} \setminus S^{(k)}$. This shows that the elements of $S^{(k-1)}$ generate the full basic orbit $G^{(k-1)}(b_{k-1})$, and hence by the remark that $\langle S^{(k-1)} \rangle = H^{(k-1)} = G^{(k-1)}$.

For $i \in k-1$, the only strong generator in $S^{(i)} \setminus S^{(i+1)}$ is the matrix $\mathcal{P}_i$ of 9.8.7. This matrix "swaps" the coefficients of the basis vectors $e^{(i)}$ and $e^{(i+1)}$. We claim that a Schreier-tree for the basic orbit $G^{(i)}(b_i)$ can be obtained from $S^{(i)} = S^{(i+1)} \cup \{P_i\}$. The points of $G^{(i)}(b_i)$ which are not in $G^{(i+1)}(b_{i+1})$ are the points of the set $\mathrm{PG}_{k-1\setminus i-1}(q)$ which are not contained in $\mathrm{PG}_{k-1\setminus i}(q)$. They are the elements of the form

$$\langle (v_0, \ldots, v_{i-1}, 1, 0, \ldots, 0) \rangle = \langle v_0 e^{(0)} + \ldots + v_{i-1} e^{(i-1)} + e^{(i)} \rangle$$

for arbitrary $v_0, \ldots, v_{i-1} \in \mathbb{F}_q$. Since

$$b_i \mathcal{P}_i = \langle e^{(i)} \rangle \mathcal{P}_i = \langle e^{(i+1)} \rangle = b_{i+1},$$

the points of $G^{(i+1)}(b_{i+1})$ can be reached from b_i using $\mathcal{P}_i$ and generators from $S^{(i+1)}$. The equation

$$\langle v_0 e^{(0)} + \ldots + v_{i-1} e^{(i-1)} + e^{(i+1)} \rangle \mathcal{P}_i = \langle v_0 e^{(0)} + \ldots + v_{i-1} e^{(i-1)} + e^{(i)} \rangle$$

shows that all other points of $G^{(i)}(b_i) \setminus G^{(i+1)}(b_{i+1})$ can be reached as well. Hence $\langle S^{(i)} \rangle = H^{(i)} = G^{(i)}$. This finishes the proof of the theorem. □

9.8.10 **Corollary** *The order of* $\mathrm{PGL}_k(q)$ *is*

$$(q-1)^{k-1} \prod_{i \in k} \theta_{k-1\setminus i-1}(q) = \frac{1}{q-1} \prod_{i \in k} (q^k - q^i). \qquad \square$$

9.8.11 **Example** A stabilizer chain for $\mathrm{PGL}_3(3)$ is obtained from the ordered base $(\langle e^{(0)}\rangle, \langle e^{(1)}\rangle, \langle e^{(2)}\rangle, \langle e^{(0)} + e^{(1)} + e^{(2)}\rangle)$. Coset representatives are $\sigma_{0,0} = I_3$, $\sigma_{0,1} = \begin{pmatrix} 010 \\ 100 \\ 001 \end{pmatrix}$, $\sigma_{0,2} = \begin{pmatrix} 001 \\ 100 \\ 010 \end{pmatrix}$, $\sigma_{0,3} = \begin{pmatrix} 111 \\ 100 \\ 010 \end{pmatrix}$, $\sigma_{0,4} = \begin{pmatrix} 110 \\ 100 \\ 001 \end{pmatrix}$, $\sigma_{0,5} = \begin{pmatrix} 210 \\ 100 \\ 001 \end{pmatrix}$, $\sigma_{0,6} = \begin{pmatrix} 101 \\ 100 \\ 010 \end{pmatrix}$, $\sigma_{0,7} = \begin{pmatrix} 201 \\ 100 \\ 010 \end{pmatrix}$, $\sigma_{0,8} = \begin{pmatrix} 011 \\ 100 \\ 010 \end{pmatrix}$, $\ldots$, $\sigma_{0,12} = \begin{pmatrix} 221 \\ 100 \\ 010 \end{pmatrix}$, $\sigma_{1,0} = I_3$, $\sigma_{1,1} = \begin{pmatrix} 100 \\ 001 \\ 010 \end{pmatrix}$, $\sigma_{1,2} = \begin{pmatrix} 100 \\ 111 \\ 010 \end{pmatrix}$, $\sigma_{1,3} = \begin{pmatrix} 100 \\ 110 \\ 001 \end{pmatrix}$, $\ldots$, $\sigma_{1,11} = \begin{pmatrix} 100 \\ 221 \\ 010 \end{pmatrix}$, $\sigma_{2,0} = I_3$, $\sigma_{2,1} = \begin{pmatrix} 100 \\ 010 \\ 111 \end{pmatrix}$, $\sigma_{2,2} = \begin{pmatrix} 100 \\ 010 \\ 101 \end{pmatrix}$, $\ldots$, $\sigma_{2,8} = \begin{pmatrix} 100 \\ 010 \\ 221 \end{pmatrix}$, $\sigma_{3,0} = I_3$, $\sigma_{3,1} = \mathrm{diag}(1,2,1)$, $\sigma_{3,2} = \mathrm{diag}(1,1,2)$, $\sigma_{3,3} = \mathrm{diag}(1,2,2)$.

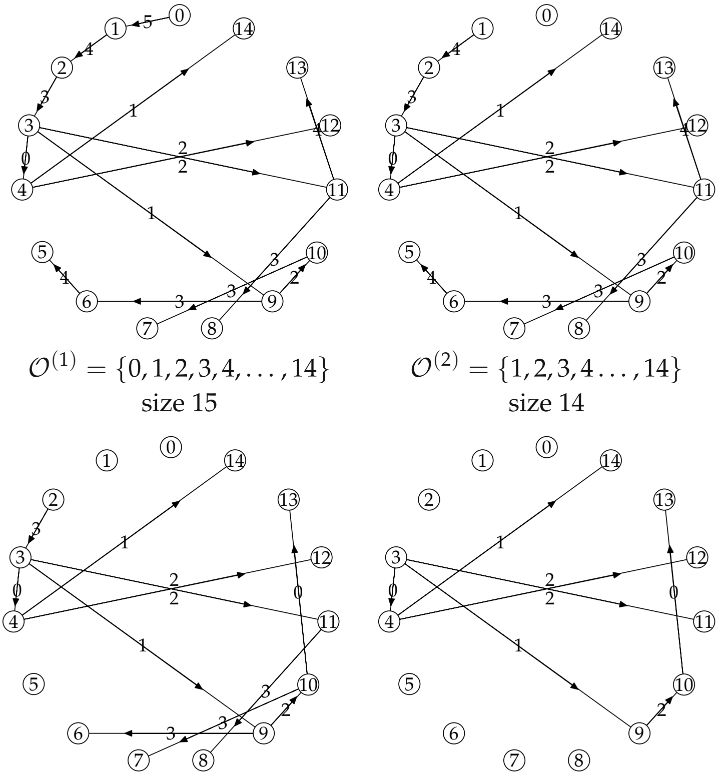
Strong generators are $\mathcal{P}_1 = \sigma_{0,1}$, $\mathcal{P}_2 = \sigma_{1,1}$, $\mathcal{E}_{0,0} = \sigma_{2,2} = \begin{pmatrix} 100 \\ 010 \\ 101 \end{pmatrix}$, $\mathcal{E}_{0,1} = \sigma_{2,4} = \begin{pmatrix} 100 \\ 010 \\ 011 \end{pmatrix}$, $\mathcal{D}_2 = \sigma_{3,1} = \mathrm{diag}(1,2,1)$, $\mathcal{D}_3 = \sigma_{3,2} = \mathrm{diag}(1,1,2)$. ◇

9.8.12 **Example** As pointed out in 9.3.12, the elements $s_0, \ldots, s_5$ listed in 9.2.7 are generators for $G = \mathrm{PGL}_4(2)$. In fact, they are strong generators for G with respect to the base (b_0, b_1, b_2, b_3), where $b_i = \mathrm{rk}_{3;2}^{-1}(i)$. In the following, to keep the notation simple we will identify projective points with their ranks. Thus, we would say that the base is $(0, 1, 2, 3)$. Let $G^{(i)} = G_{b_0,\ldots,b_{i-1}} = G_{0,\ldots,i-1}$ be the stabilizer of the first i base points. Then

$$S^{(i)} = S \cap G^{(i)} = \begin{cases} \{s_0, s_1, s_2, s_3, s_4, s_5\} & \text{if } i = 0, \\ \{s_0, s_1, s_2, s_3, s_4\} & \text{if } i = 1, \\ \{s_0, s_1, s_2, s_3\} & \text{if } i = 2, \\ \{s_0, s_1, s_2\} & \text{if } i = 3. \end{cases}$$

The basic orbits $\mathcal{O}^{(i)}$ and the corresponding Schreier-trees are shown in Fig. 9.13. From the Schreier-trees, coset representatives can be determined easily. For instance, an element of $G^{(2)}$ mapping $b_2 = 2$ to 10 (which is the 8-th element in the orbit $\mathcal{O}^{(3)}$) is

$$\sigma_{2,7} = s_3 s_1 s_2$$



$\mathcal{O}^{(1)} = \{0, 1, 2, 3, 4, \ldots, 14\}$
size 15

$\mathcal{O}^{(2)} = \{1, 2, 3, 4 \ldots, 14\}$
size 14

$\mathcal{O}^{(3)} = \{2, 3, 4, 6, 7, \ldots, 14\}$
size 12

$\mathcal{O}^{(4)} = \{3, 4, 9, 10, 11, 12, 13, 14\}$
size 8

Fig. 9.13 The basic orbits $\mathcal{O}^{(i)}$ for $\mathrm{PGL}_4(2)$

$$\begin{aligned} &= (2,3)(6,9)(7,10)(8,11) \\ &\quad \cdot (3,9)(4,14)(10,11)(12,13) \\ &\quad \cdot (3,11)(4,12)(9,10)(13,14) \\ &= (2,10,7,3)(4,13)(6,11,8,9)(12,14) \end{aligned}$$

Also, the group order is the product of the lengths of the basic orbits, which is $15 \cdot 14 \cdot 12 \cdot 8 = 20160$. It is now easy to access group elements numerically. For instance the group element 1777 (the birth year of Gauss) can be determined as follows. We write $1777 = ((14+4)12+6)8+1$, i.e. the multibase representation is $1777 = (1, 4, 6, 1)_{8,12,14,15}$. Therefore we need coset representatives mapping $b_0, \ldots, b_3$ to the second, 5-th, 7-th and second orbit element, respectively. That is, we need coset representatives $\sigma_{i,j}$ such that

$$\sigma_{0,1}(0) = 1,\ \sigma_{1,4}(1) = 5,\ \sigma_{2,6}(2) = 9,\ \sigma_{3,1}(3) = 4.$$

From the Schreier-trees, we obtain that

$$\begin{aligned}\sigma_{0,1} &= s_5,\\ \sigma_{1,4} &= s_4s_3s_1s_3s_4,\\ \sigma_{2,6} &= s_3s_1,\\ \sigma_{3,1} &= s_0,\end{aligned}$$

so that the group element 1777 is

$$\begin{aligned}\sigma_{3,1}\sigma_{2,6}\sigma_{1,4}\sigma_{0,1} &= s_0s_3s_1s_4s_3s_1s_3s_4s_5\\ &= (0,1,5)(2,10,12,6,3,4)(7,9,13,8,11,14).\end{aligned}$$

It is also possible to compute the coset representatives $\sigma_{i,j}$ directly using 9.8.5 and the labeling of points as indicated in Table 9.5. This gives

$$\begin{aligned}\sigma_{3,1}\sigma_{2,6}\sigma_{1,4}\sigma_{0,1} &= \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\1&1&1&1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\1&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\\ &= \begin{pmatrix}0&1&0&0\\1&1&0&0\\0&1&0&1\\1&1&1&1\end{pmatrix}\end{aligned}$$

This matrix sends the standard basis $0,1,2,3$ to $1,5,10,4$, respectively. Since group elements are the same whenever they have the same effect on all base points, this must be the same as the permutation

$$(0,1,5)(2,10,12,6,3,4)(7,9,13,8,11,14)$$

from above. Lastly, Fig. 9.14 depicts the coset representatives according to the 4 subgroups in the stabilizer chain of $\mathrm{PGL}_4(2)$. The numbers shown are the actual elements in the basic orbits $\mathcal{O}^{(i)}$, each corresponding to one coset representative $\sigma_{i,j}$. ◇

Exercises

Exercise Verify the statement of 9.8.1 that $\mathrm{rk}_{d\backslash -1;q} = \mathrm{rk}_{d;q}$ and that $\mathrm{rk}^{-1}_{d\backslash -1;q} = \mathrm{rk}^{-1}_{d;q}$. **E.9.8.1**

Exercise It was noted after 9.2.5 that the Schreier-trees are not unique, for instance they depend on the choice of the generating set. On the other hand, **E.9.8.2**

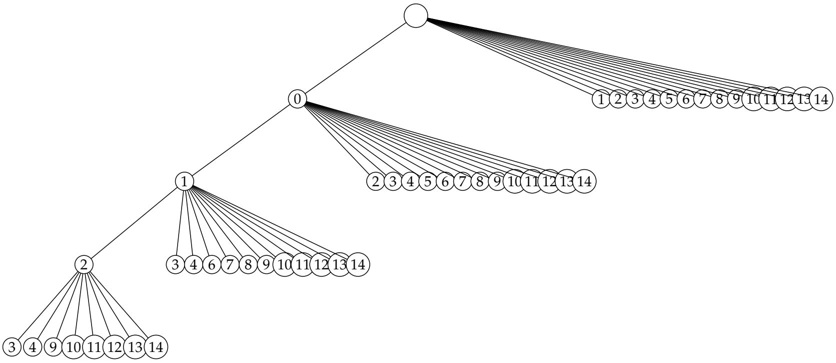


Fig. 9.14 The coset representatives for $\mathrm{PGL}_4(2)$

shortly before 9.7.9 it was noted that a stabilizer chain can be used to access group elements numerically. Convince yourself that the labeling of group elements using a stabilizer chain does not depend on the chosen generating set provided the elements of each of the fundamental orbits are ordered lexicographically. Therefore, a different choice of Schreier-trees in Example 9.8.12 would still yield the same group element with number 1777 as long as the elements of the basic orbits are listed in order.

E.9.8.3 **Exercise** Compute the position 999 999 of Rubik's $2 \times 2 \times 2$ cube, following the ideas developed in Exercise 9.8.2.

9.9

9.9 The Projective Semilinear Group

The next result describes a base and strong generating set for $\mathrm{P\Gamma L}_k(q)$. The proof of this result follows easily from 3.7.11 and is omitted.

9.9.1 **Theorem (base and strong generating set for $\mathrm{P\Gamma L}_k(q)$)** *Let $q = p^h$ with p prime and h a positive integer. Let $\mathrm{PG}_{k-1}(q)$ be the one-dimensional subspaces of the vector space $V = \mathbb{F}_q^k$ with basis $e^{(0)}, \ldots, e^{(k-1)}$. If q is prime then $\mathrm{P\Gamma L}_k(q) \simeq \mathrm{PGL}_k(q)$ and 9.8.5 applies. Otherwise, if $q = p^h$ with $h > 1$, choose a primitive element α for $\mathbb{F}_q$. For $i \in k+1$, let b_i be as in 9.8.5. Put $b_{k+1} = \langle \alpha e^{(0)} + e^{(1)} \rangle$.*

1. *The sequence* $B = (b_0, \dots, b_k, b_{k+1})$ *is an ordered base for* $\mathrm{P\Gamma L}_k(q)$ *acting on* $\mathrm{PG}_{k-1}(q)$*. The corresponding stabilizer chain has basic orbits of lengths*

$$\ell_i = \begin{cases} \theta_{k-1\backslash i-1}(q) & \textit{for} \quad i \in k, \\ (q-1)^{k-1} & \textit{for} \quad i = k, \\ h & \textit{for} \quad i = k+1. \end{cases}$$

2. *Coset representatives* $\gamma_{i,j}$*,* $i \in k+2$*,* $j \in \ell_i$ *can be chosen in the following way.*
   (a) *For* $i \in k+1$*, and for* $j \in \ell_i$*, let*

$$\gamma_{i,j} = \left(\sigma_{i,j}, 0\right)$$

   with $\sigma_{i,j}$ *as described in 9.8.5.*

   (b) *For* $j \in \ell_{k+1}$*, let*

$$\gamma_{k+1,j} = \left(I_k, j\right).$$

3. *A strong generating set for* $\mathrm{P\Gamma L}_k(q)$ *is given by the elements*

$$(\sigma, 0),$$

where σ *runs through all elements of a strong generating set of* $\mathrm{PGL}_k(q)$ *as described in 9.8.5, together with the element*

$$(I_k, 1). \qquad \square$$

Corollary *The order of* $\mathrm{P\Gamma L}_k(q)$ *is* **9.9.2**

$$h(q-1)^{k-1} \prod_{i \in k} \theta_{k-1\backslash i-1}(q) = \frac{h}{q-1} \prod_{i \in k} (q^k - q^i). \qquad \square$$

Example The field $\mathbb{F}_8$ is generated over $\mathbb{F}_2$ by a root α of the polynomial $X^3 + X^2 + 1$ (so that $\alpha^3 = 1 + \alpha^2$). In the additive labeling, the field elements are **9.9.3**

$$\begin{aligned} \kappa_0 &= 0, \\ \kappa_1 &= 1, \\ \kappa_2 &= \alpha, \\ \kappa_3 &= \alpha + 1, \\ \kappa_4 &= \alpha^2, \\ \kappa_5 &= \alpha^2 + 1, \\ \kappa_6 &= \alpha^2 + \alpha, \\ \kappa_7 &= \alpha^2 + \alpha + 1. \end{aligned}$$

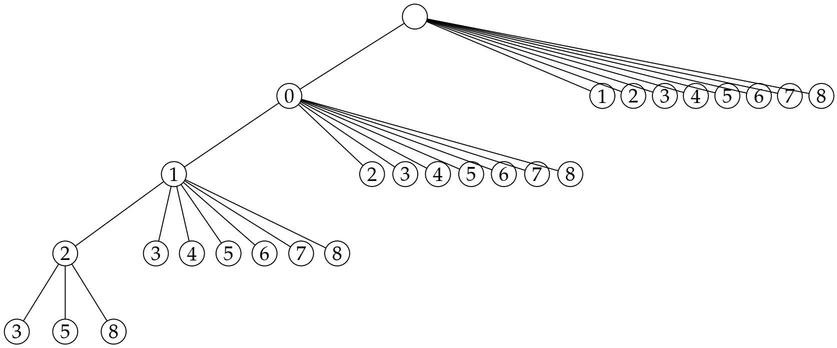


Fig. 9.15 The coset representatives for $\mathrm{P\Gamma L}_2(8)$

Using the rank function of 9.3.5, the 9 points of the projective line $\mathrm{PG}_1(8)$ are numbered as

$$\begin{aligned}
0 &= \langle(1,0)\rangle,\\
1 &= \langle(0,1)\rangle,\\
2 &= \langle(1,1)\rangle,\\
3 &= \langle(\kappa_2,1)\rangle,\\
4 &= \langle(\kappa_3,1)\rangle,\\
5 &= \langle(\kappa_4,1)\rangle,\\
6 &= \langle(\kappa_5,1)\rangle,\\
7 &= \langle(\kappa_6,1)\rangle,\\
8 &= \langle(\kappa_7,1)\rangle.
\end{aligned}$$

A base for $\mathrm{P\Gamma L}_2(8)$ is $(0,1,2,3)$. Strong generators are

$$\begin{aligned}
s_0 &= \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, 1\right) = (3,5,8)(4,6,7),\\
s_1 &= \left(\begin{pmatrix} 1 & 0 \\ 0 & \kappa_2 \end{pmatrix}, 0\right) = (2,7,4,8,6,5,3),\\
s_2 &= \left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, 0\right) = (1,2)(3,4)(5,6)(7,8),\\
s_3 &= \left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, 0\right) = (0,1)(3,7)(4,5)(6,8).
\end{aligned}$$

The basic orbits have length $9, 8, 7$, and 3, respectively. We conclude that the group $\mathrm{P\Gamma L}_2(8)$ has 1512 elements. Figure 9.15 depicts the coset representatives according to the 4 subgroups in the stabilizer chain of $\mathrm{P\Gamma L}_2(8)$. The numbers shown are the elements in the basic orbits $\mathcal{O}^{(i)}$, each corresponding to one coset representative $\sigma_{i,j}$. ◇

Exercises

Exercise Compute a base and stabilizer chain for $\mathrm{P\Gamma L}(3,4)$ using 9.9.1. List the coset representatives. **E.9.9.1**

9.10 Numerical Data

9.10

Let us now present numerical data concerning the classification of isometry classes of linear indecomposable codes for small finite fields. In all cases, we classify the semilinear isometry classes over $\mathbb{F}_q$. If q is a prime, then of course the semilinear isometry classes are the same as the linear isometry classes. We present results for the fields $\mathbb{F}_q$ with $q \in \{2,3,4,5,8,9,16,25,27\}$ in Tables 9.6-9.24. For a given length n and dimension k, the corresponding entry in the table lists the number of semilinear isometry classes of (n,k)-codes with a given minimum distance. For instance, an entry of the form

$$d^x e^y f^z$$

indicates that there are x classes of codes with minimum distance d, y classes with minimum distance e and z classes with minimum distance f. The minimum distances are ordered decreasingly, and the first value, d, is the optimal minimum distance in that parameter case. Exponents whose value is 1 are omitted. Underlined entries indicate non-trivial MDS-codes.

Table 9.6 Optimal indecomposable $\mathbb{F}_2$ codes

$n\backslash k$	1	2	3	4	5	6	7
4	4						
5	5	3					
6	6	$4\,3$	3				
7	7	$4^2 3$	$4\,3^3$	3			
8	8	$5\,4^2$	$4^3 3^6$	$4\,3^4$			
9	9	$6\,5^2 4^2$	4^8	$4^4 3^{18}$	3^5		
10	10	$6^2 5^2 4^2$	$5^2 4^{18}$	4^{19}	$4^4 3^{36}$	3^4	
11	11	$7\,6^3 5^2$	$6\,5^8 4^{29}$	$5\,4^{66}$	4^{30}	$4^2 3^{58}$	3^3
12	12	$8\,7^2 6^3 5^2$	$6^6 5^{19}$	$6\,5^{12} 4^{201}$	4^{214}	4^{41}	$4^2 3^{84}$
13	13	$8^2 7^3 6^3$	$7\,6^{16} 5^{37}$	$6^6 5^{72}$	$5^{15} 4^{1159}$	4^{580}	4^{45}
14	14	$9\,8^3 7^3$	$8\,7^5 6^{37}$	$7\,6^{39} 5^{292}$	$6^6 5^{261}$	$5^{11} 4^{6704}$	4^{1488}
15	15	$10\,9^2 8^4 7^3$	$8^3 7^{17}$	$8\,7^5 6^{195}$	$7\,6^{91} 5^{2547}$	$6^5 5^{995}$	$5^6 4^{41037}$
16	16	$10^2 9^3 8^4$	$8^{12} 7^{41}$	$8^4 7^{37}$	$8\,7^5 6^{1145}$	$6^{180} 5^{29826}$	$6^3 5^{4010}$
17	17	$11\,10^3 9^4 8^4$	$9^2 8^{32}$	$8^{18} 7^{241}$	$8^4 7^{84}$	7^3	6^{377}
18	18	$12\,11^2 10^4 9^4$	$10\,9^{11} 8^{71}$	8^{108}	$8^{34} 7^{1777}$	$8^2 7^{108}$	7^2
19		$12^2 11^3 10^5$	$10^6 9^{33}$	$9^7 8^{550}$	8^{411}	$8^{28} 7^{19021}$	$8\,7^{81}$
20			$11\,10^{21}$	$10^3 9^{81}$	$9^3 8^{6480}$	8^{1833}	8^{26}
21				10^{27}	$10^2 9^{178}$		
22					10^{37}	9^{248}	
23						10^{29}	9^{29}
24							10^6

Table 9.7 Optimal indecomposable $\mathbb{F}_2$ codes (cont.)

$n\backslash k$	8	9	10	11	12	13	14	15	16	17	18	19
12	3^{2}											
13	$4\,3^{109}$	3										
14	4^{48}	$4\,3^{126}$	3									
15	4^{3473}	4^{43}	$4\,3^{142}$	3								
16	$5\,4^{268258}$	4^{7456}	4^{47}	$4\,3^{143}$								
17	$6\,5^{13757}$	5	4^{14390}	4^{39}	3^{129}							
18	6^{918}	$6\,5^{29371}$		4^{25024}	4^{33}	3^{113}						
19	7	6^{1700}	5^{31237}		4^{39302}	4^{25}	3^{91}					
20	$8\,7^{33}$	7	6^{1682}	5^{14135}			4^{24}	3^{67}				
21	8^{12}	$8\,7^{20}$	7	6^{739}	5^{2373}			4^{16}	3^{50}			
22		8^{9}	$8\,7^{15}$	7	6^{128}	5^{128}			4^{15}	3^{34}		
23			8^{8}	$8\,7^{15}$	7	6^{8}	5			4^{9}	3^{21}	
24				8^{9}	$8\,7^{11}$		6				4^{8}	3^{14}
25					8^{7}							4^{5}

Table 9.8 Optimal indecomposable $\mathbb{F}_2$ codes (cont.)

$n\backslash k$	20	21	22	23	24	25	26
25	3^{9}						
26	4^{4}	3^{5}					
27		4^{2}	3^{3}				
28			4^{2}	3^{2}			
29				4	3		
30					4	3	
31						4	3
32							4

Table 9.9 Optimal indecomposable $\mathbb{F}_3$ codes

$n\backslash k$	1	2	3	4	5	6	7
3	3						
4	4	$\underline{3}$					
5	5	3^2					
6	6	$4^2 3^2$	3^4				
7	7	$5\,4^3 3^2$	$4^2 3^{12}$	3^4			
8	8	$6\,5^3 4^3$	$5\,4^{13} 3^{25}$	$4^3 3^{36}$	3^3		
9	9	$6^3 5^4$	$6\,5^8 4^{40}$	$5\,4^{41} 3^{185}$	$4\,3^{87}$	3^3	
10	10	$7^2 6^5$	$6^6 5^{39}$	$6\,5^{19} 4^{403}$	$5\,4^{134} 3^{1205}$	$4\,3^{195}$	3^2
11	11	$8\,7^4$	$7\,6^{35}$	$6^7 5^{452}$	$6\,5^{34} 4^{4840}$	$5\,4^{354} 3^{8297}$	3^{399}
12	12	$9\,8^4 7^6$	$8\,7^{15}$	6^{353}	$6^8 5^{8550}$	$6\,5^{36} 4^{73941}$	$4^{844} 3^{61060}$
13		$9^3 8^6$	$9\,8^7 7^{107}$	7^{72}	6^{5037}	$6^9 5^{191851}$	5^6
14			$9^3 8^{72}$	$8^{14} 7^{5221}$	7^{236}	6^{47674}	6
15				9^3		7^{22}	
16					9		

Table 9.10 Optimal indecomposable $\mathbb{F}_3$ codes (cont.)

$n\backslash k$	8	9	10	11	12	13	14	15	16
11	3								
12	3^{805}	3							
13	$4^{1532} 3^{457485}$	3^{1503}	3						
14	5	4^{2020}	3^{2658}						
15			4^{1778}	3^{4304}					
16				4^{1019}	3^{6472}				
17					4^{337}	3^{8846}			
18						4^{90}	3^{11127}		
19							4^{20}	3^{12723}	
20								4^9	3^{13358}

Table 9.11 Optimal indecomposable $\mathbb{F}_3$ codes (cont.)

$n\backslash k$	17	18	19	20	21	22	23	24	25	26	27	28
21	3^{12723}											
22		3^{11127}										
23			3^{8846}									
24				3^{6472}								
25					3^{4304}							
26						3^{2659}						
27							3^{1505}					
28								3^{807}				
29									3^{402}			
30										3^{201}		
31											3^{94}	
32												3^{47}

Table 9.12 Optimal indecomposable $\mathbb{F}_3$ codes (cont.)

$n\backslash k$	29	30	31	32	33	34	35	36
33	3^{23}							
34		3^{12}						
35			3^{6}					
36				3^{4}				
37					3^{2}			
38						3		
39							3	
40								3

Table 9.13 Optimal indecomposable $\mathbb{F}_4$ codes

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11
3	3										
4	4	$\underline{3}$									
5	5	$\underline{4}\,3^2$	$\underline{3}$								
6	6	$4^3 3^2$	$\underline{4}\,3^6$								
7	7	$5^2 4^4$	$4^7 3^{19}$	3^{10}							
8	8	$6^2 5^4$	$5^3 4^{38}$	$4^{16} 3^{96}$	3^{13}						
9	9	$7\,6^5$	$6^3 5^{39}$	$5^4 4^{326}$	$4^{19} 3^{466}$	3^{17}					
10	10	$8\,7^4$	6^{45}	$6^2 5^{642}$	$5^4 4^{4189}$	$4^{23} 3^{2380}$	3^{18}				
11		8^4	7^{25}	6^{841}	$6\,5^{19418}$	$5\,4^{66475}$	$4^{15} 3^{13080}$	3^{18}			
12			8^{16}	7^{275}	6^{19181}			4^{13}	3^{17}		
13				8^{30}	7^{452}				4^4	3^{13}	
14					8^6	7^{14}				4^2	3^{10}
15						8^3	7^4				4
16							8^2	7^3			
17								8^2	7^2		
18									8		

Table 9.14 Optimal indecomposable $\mathbb{F}_4$ codes (cont.)

$n\backslash k$	12	13	14	15	16	17	18
15	3^8						
16	4	3^5					
17		4	3^3				
18				3^2			
19					3		
20						3	
21							3

Table 9.15 Optimal indecomposable $\mathbb{F}_5$ codes

$n\backslash k$	1	2	3	4	5	6	7	8
3	3							
4	4	$\underline{3}$						
5	5	$\underline{4}\,3^{2}$	$\underline{3}$					
6	6	$\underline{5}\,4^{4}3^{2}$	$\underline{4}\,3^{9}$	$\underline{3}$				
7	7	5^{3}	$4^{17}3^{29}$	3^{21}				
8	8	$6^{3}5^{7}$	5^{16}	$4^{92}3^{344}$	3^{42}			
9		$7^{2}6^{8}$	$6^{16}5^{248}$	5^{134}	$4^{387}3^{4570}$	3^{92}		
10			$7^{7}6^{486}$	6^{93}	5^{558}	$4^{1568}3^{62846}$	3^{174}	
11					6^{60}	5^{503}	$4^{4089}3^{814405}$	3^{296}
12						6^{31}	5^{36}	4^{7062}

Table 9.16 Optimal indecomposable $\mathbb{F}_5$ codes (cont.)

$n\backslash k$	9	10	11	12	13	14	15	16	17	18	19	20	21	22
12	3^{476}													
13	4^{7258}	3^{669}												
14		4^{4678}	3^{832}											
15			4^{1810}	3^{948}										
16				4^{572}	3^{948}									
17					4^{183}	3^{832}								
18						4^{88}	3^{669}							
19							4^{36}	3^{476}						
20								4^{21}	3^{296}					
21									4^{7}	3^{174}				
22										4^{4}	3^{92}			
23											4	3^{42}		
24												4	3^{22}	
25													4	3^{12}

Table 9.17 Optimal indecomposable $\mathbb{F}_5$ codes (cont.)

$n\backslash k$	22	23	24	25	26	27	28
26	4	3^5					
27			3^3				
28				3^2			
29					3		
30						3	
31							3

Table 9.18 Optimal indecomposable $\mathbb{F}_8$ codes

$n\backslash k$	1	2	3	4	5	6	7	8	9
3	3								
4	4	$\underline{3}$							
5	5	$\underline{4}\,3^2$	$\underline{3}$						
6	6	$\underline{5}\,4^4$	$\underline{4}^3 3^{10}$	$\underline{3}$					
7	7	$\underline{6}$	$\underline{5}^2 4^{49}$	$\underline{4}^2 3^{54}$	$\underline{3}$				
8		$\underline{7}$	$\underline{6}^2$	$\underline{5}\,4^{1700}$	$\underline{4}^2 3^{323}$	$\underline{3}$			
9			$\underline{7}^2$	$\underline{6}$	$\underline{5}\,4^{68877}$	$\underline{4}^2 3^{2097}$	$\underline{3}$		
10							$\underline{4}\,3^{12868}$		
11								3^{72638}	
12									3^{373366}

Table 9.19 Optimal indecomposable $\mathbb{F}_9$ codes

$n\backslash k$	1	2	3	4	5	6	7	8
3	3							
4	4	$\underline{3^{2}}$						
5	5	$\underline{4^{2}}$	$\underline{3^{2}}$					
6	6	$\underline{5^{2}}$	$\underline{4^{6}}$	$\underline{3^{2}}$				
7	7	$\underline{6}$	$\underline{5^{3}}$	$\underline{4^{3}}$	$\underline{3}$			
8		$\underline{7}$	$\underline{6^{2}}$	$\underline{5^{5}}$	$\underline{4^{2}}$	$\underline{3}$		
9			$\underline{7}$	$\underline{6^{2}}$	$\underline{5^{2}}$	$\underline{4}$	$\underline{3}$	
10				$\underline{7}$	$\underline{6^{2}}$	$\underline{5}$	$\underline{4}$	$\underline{3}$

Table 9.20 Optimal indecomposable $\mathbb{F}_{16}$ codes

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	3														
4	4	$\underline{3^{2}}$													
5	5	$\underline{4^{3}}$	$\underline{3^{3}}$												
6		$\underline{5^{4}}$	$\underline{4^{22}}$	$\underline{3^{4}}$											
7			$\underline{5^{125}}$	$\underline{4^{125}}$	$\underline{3^{5}}$										
8				$\underline{5^{2981}}$	$\underline{4^{685}}$	$\underline{3^{6}}$									
9					$\underline{5^{6888}}$	$\underline{4^{1534}}$	$\underline{3^{6}}$								
10						$\underline{5^{356}}$	$\underline{4^{1262}}$	$\underline{3^{5}}$							
11							$\underline{5^{10}}$	$\underline{4^{300}}$	$\underline{3^{4}}$						
12								$\underline{5^{4}}$	$\underline{4^{159}}$	$\underline{3^{3}}$					
13									$\underline{5^{2}}$	$\underline{4^{70}}$	$\underline{3^{2}}$				
14										$\underline{5}$	$\underline{4^{30}}$	$\underline{3}$			
15											$\underline{5}$	$\underline{4^{9}}$	$\underline{3}$		
16												$\underline{5}$	$\underline{4^{5}}$	$\underline{3}$	
17													$\underline{5}$	$\underline{4^{3}}$	$\underline{3}$
18															$\underline{4^{2}}$

Table 9.21 Optimal indecomposable $\mathbb{F}_{25}$ codes

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3													
4	4	$\underline{3^{4}}$												
5	5	$\underline{4^{7}}$	$\underline{3^{7}}$											
6		$\underline{5^{19}}$	$\underline{4^{205}}$	$\underline{3^{19}}$										
7				$\underline{4^{7163}}$	$\underline{3^{34}}$									
8						$\underline{3^{79}}$								
9							$\underline{3^{132}}$							
10								$\underline{3^{223}}$						
11									$\underline{3^{293}}$					
12										$\underline{3^{379}}$				
13											$\underline{3^{391}}$			
14												$\underline{3^{379}}$		
15													$\underline{3^{293}}$	
16														$\underline{3^{223}}$

Table 9.22 Optimal indecomposable $\mathbb{F}_{25}$ codes (cont.)

$n\backslash k$	15	16	17	18	19	20	21	22	23	24
17	$\underline{3^{132}}$									
18		$\underline{3^{79}}$								
19			$\underline{3^{34}}$							
20				$\underline{3^{19}}$						
21					$\underline{3^{7}}$					
22						$\underline{3^{4}}$				
23							$\underline{3}$			
24								$\underline{3}$		
25									$\underline{3}$	
26										$\underline{3}$

Table 9.23 Optimal indecomposable $\mathbb{F}_{27}$ codes

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	3													
4	4	$\underline{3^{3}}$												
5	5	$\underline{4^{4}}$	$\underline{3^{4}}$											
6		$\underline{5^{14}}$	$\underline{4^{174}}$	$\underline{3^{14}}$										
7			$\underline{5^{8261}}$	$\underline{4^{8261}}$	$\underline{3^{29}}$									
8						$\underline{3^{72}}$								
9							$\underline{3^{134}}$							
10								$\underline{3^{257}}$						
11									$\underline{3^{390}}$					
12										$\underline{3^{565}}$				
13											$\underline{3^{670}}$			
14												$\underline{3^{738}}$		
15													$\underline{3^{670}}$	
16														$\underline{3^{565}}$

Table 9.24 Optimal indecomposable $\mathbb{F}_{27}$ codes (cont.)

$n\backslash k$	15	16	17	18	19	20	21	22	23	24	25	26
17	$\underline{3^{390}}$											
18		$\underline{3^{257}}$										
19			$\underline{3^{134}}$									
20				$\underline{3^{72}}$								
21					$\underline{3^{29}}$							
22						$\underline{3^{14}}$						
23							$\underline{3^{4}}$					
24								$\underline{3^{3}}$				
25									$\underline{3}$			
26										$\underline{3}$		
27											$\underline{3}$	
28												$\underline{3}$