Steiner systems

Bachelor End Project Thesis

by

Guy Briejer

To obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Thursday July 18, 2024 at 01:00 PM.

Student number: 5356210
Project duration: May 8, 2024 – July 11, 2024
Thesis committee: Dr. J. G. Spandaw, TU Delft, supervisor
Prof. Dr. D. C. Gijswijt, TU Delft, Exam committee member

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Summary

Lay summary
One of the main focuses of this thesis is a special kind of symmetrical arrangement called a "Steiner system." Imagine trying to organize a group of objects in such a way that every possible subset of them fits together in a very specific and symmetrical pattern. These patterns are not just mathematical coincidences; they have real-world applications, such as in coding theory.

More precisely, among these Steiner systems, there are some particularly interesting ones known as "Witt designs." These designs are important because their symmetrical properties are used in constructing certain types of error-correcting codes. These codes can correct a certain amount of errors within information that is sent or stored, making it more reliable.

To better understand these Witt designs, the thesis explores two main ways to create them. One method uses something called the Golay code, which is a specific type of error-correcting code. The other method involves a mathematical group called the projective special linear group. Both of these constructions are explained in detail, with proofs provided to show that they indeed form the desired symmetrical patterns.

General summary
Steiner systems are exceptional combinatorial structures and exhibit extremely high degrees of symmetry. Especially, the Witt designs are the 2 quintuple Steiner systems $S(5,6,12)$ and $S(5,8,24)$. These Steiner systems play an important role in coding theory as they are used to construct the only perfect t-error correcting codes with $t \geq 2$. Also, the automorphism groups of the Witt designs form the Mathieu groups M_{12} and M_{24}, which are sporadic simple groups. These simple groups form the building blocks of all finite groups and are therefore fundamental concepts in group theory.

This thesis aims to give different constructions of these Witt designs. Before being able to construct these Steiner systems, first the needed group theory and the definition of a projective space over a finite field is provided. Furthermore, it aims to give more insight into the Witt designs and their constructions. Particularly, for each construction one or several proofs are given confirming their Steiner properties. First, the Witt designs are constructed using the extended binary Golay code and the Ternary Golay code, which are perfect t-error correcting codes for $t = 3$ and $t = 2$, respectively. Then the Witt design $S(5,6,12)$ is constructed using the projective special linear group $PSL(2,11)$. Thereafter 2 proofs of this construction being a Steiner system are covered. Lastly the construction of the $S(5,8,24)$ using the projective special linear group $PSL(2,11)$ is presented.

List of Symbols

Symbol	Description
e	Identity element of a group
g	Element of a group
G	Group
H	Subgroup of a group G
G/H	Set of left cosets of H in G
$[G:H]$	Index of the subgroup H in G
Gx	Orbit of x under the group G
G_x	Stabilizer subgroup of x in G
$\sim_G$	Equivalence relation under group G
$\|G\|$	Order of the group G
$GL(n,q)$	General linear group of degree n over the finite field q
$SL(n,q)$	Special linear group of degree n over the finite field q
$PSL(2,p)$	Projective special linear group
$PSL(2,11)$	Projective special linear group for $p = 11$
$PSL(2,23)$	Projective special linear group for $p = 23$
A_n	Alternating group on n elements
S_n	Symmetric group on n elements
$\mathbb{Z}/p\mathbb{Z}$	Integers modulo p
$\mathbb{R}^n$	n-dimensional real vector space
$\mathbb{C}^n$	n-dimensional complex vector space
$V_d(\kappa)$	Vector space of polynomials of degree $\leq d$ over the field κ
$\mathbb{F}_q$	Finite field with q elements
$\mathbb{P}^n(K)$	n-dimensional projective space over field K
$\mathbb{P}^1(\mathbb{F}_q)$	Projective line over the finite field $\mathbb{F}_q$
$S(t,k,v)$	Steiner system with parameters (t,k,v)
$S(5,6,12)$	A specific Steiner system also known as Witt design
$S(5,8,24)$	Another specific Steiner system also known as Witt design
β	The set of blocks
M_{12}	Mathieu group 12
M_{24}	Mathieu group 24
G_{23}	Binary Golay code
$\tilde{G}_{24}$	Extended binary Golay code
G_{11}	Ternary Golay code
$\tilde{G}_{12}$	Extended ternary Golay code

Contents

Introduction

Symmetry is a key concept in both mathematics and physics, helping us understand various patterns and structures. Symmetries can be found all throughout these disciplines and are often studied using group theory, a branch of mathematics that deals with symmetries in a systematic way. In mathematics for example, the symmetric group S_n includes all possible ways to rearrange n items and is important for solving polynomial equations. In physics, the diffeomorphism group, which includes smooth changes in spacetime, is central to Einstein's theory of general relativity. Even in chemistry, the symmetrical arrangements of atoms within molecules are described using group theory.

In group theory, all finite groups can be broken down into smaller and smaller groups with the smallest being called finite "simple groups". This makes the simple groups the building blocks of all finite groups, much like the primes being the building blocks of all integers. Most of these groups belong to a few infinite families, such as the alternating groups A_n for $n \geq 5$. However, there are 26 exceptions called the *sporadic simple group*. These sporadic simple groups often exhibit extremely high degrees of symmetry, which can be very complex in structure and are therefore widely used. The smallest and one of the earliest discovered sporadic simple groups are the Mathieu groups. These groups arise as automorphism groups of certain exceptional combinatorial structures, called *Steiner systems*. Moreover, Steiner systems are used in the construction of error-correcting codes, which are essential for reliable data transmission and storage. Two particularly interesting Steiner systems are the Witt designs, whose automophishm groups form the Mathieu groups with the highest level of symmetry: M_{24} and M_{12}.

The primary aim of this thesis is to delve further into these Witt designs. In particular, two ways of constructing both Witt designs will be discussed. Also, the theory needed to do so will be treated to make the constructions understandable for undergraduates in applied mathematics. The first construction will make use of the Golay code and the second of the projective special linear group $PSL(2, p)$, for $p = 11$ and $p = 23$. After each construction, one or several proofs of its Steiner property will be given. On one hand, this is proven using a Maple code, and on the other hand, a formal proof will be provided.

Finally, a brief overview of this thesis: In chapter 1 all the needed knowledge on Group theory, certain groups and the definition of a projective space over a finite field is given. Section 2.1 will provide the definition of a Steiner system together with some relatively easy examples to give a better understanding of these systems. The first way of constructing the Witt designs, which is via the Golay codes, will be treated in section 2.2. Finally, sections 2.3 and 2.4 will provide the construction off the Witt designs using the projective special linear groups $PSL(2, 11)$ and $PSL(2, 23)$, respectively.

Note that this thesis is written for peers, i.e. third year students in applied mathematics. It is assumed that the reader of this thesis has prior knowledge in algebra on an undergraduate level. Furthermore, this thesis is largely based on on personal communication with dr. Jeroen Spandaw and several books, which are [1][11][2].

1

Group theory

Although already mentioned that a certain amount of prior knowledge on group theory is needed to understand this thesis, some definitions and theorems that will be used frequently in later chapters or are not part of the first year course algebra 1 will be presented in this chapter. Most of the definitions given in this chapter are from the book Algebra 1, written by Dion Gijswijt, or from documents created by Jeroen Spandaw.

1.1. Projective space over a finite field

Before treating the definitions and theorems from group theory first a projective space over a finite field needs to be defined, which will be used to construct Steiner systems in chapter 2 of this thesis. A projective space is a geometric structure that extends the concept of a plane in such a way that parallel lines intersect at a unique point at infinity. More formally,

Definition 1.1.1 (Projective Space)**.** An n-dimensional projective space $\mathbb{P}^n(K)$ over a field K is defined as the set of equivalence classes of the set $K^{n+1} \setminus \{0\}$ under the equivalence relation $\sim$ defined by:

$$(x_0, x_1, \ldots, x_n) \sim (y_0, y_1, \ldots, y_n) \iff \text{there exists a } \lambda \in K^* \text{ such that } (x_0, x_1, \ldots, x_n) = \lambda(y_0, y_1, \ldots, y_n).$$

Here, K^* denotes the set of non-zero elements of K. Each equivalence class, denoted $[x_0 : x_1 : \ldots : x_n]$, is called a point in the projective space.

For example, if K is the field of the real numbers, a projective space is called a real projective space $\mathbb{P}^n(\mathbb{R})$, which is the topological space of lines passing through the origin 0 in the real space $\mathbb{R}^{n+1}$. For instance, $\mathbb{P}^1(\mathbb{R})$ is called the real projective line, which is topologically equivalent to a circle[14].

However, in this thesis we are interested in projective spaces over a finite field, also called a Galois field. A finite field, denote $\mathbb{F}_q$ or $GF(q)$, is a field containing exactly $q = p^n$ elements for some prime p and positive integer n. The field $\mathbb{F}_q$ has the following properties:

- **Addition and Multiplication:** The operations of addition and multiplication are defined and satisfy the field axioms (associativity, commutativity, distributivity, existence of additive and multiplicative identities, and existence of additive and multiplicative inverses).
- **Finite Number of Elements:** The field contains exactly q elements.
- **Existence and Uniqueness:** For any given $q = p^n$, there exists a unique (up to isomorphism) Galois field $\mathbb{F}_q$. [12]

The simplest examples of finite fields are the fields with $q = p$ elements, which are simultaneously the ones we're most interested in. In this particular case, $\mathbb{F}_p^*$ is equal to the familiar multiplicative group of integers modulo p: $(\mathbb{Z}/p\mathbb{Z})^*$, whose elements consist of $\{[1], [2], [3], \ldots, [p-1]\}$ where $[a] = a + p\mathbb{Z}$ represents the set of all integers congruent to a modulo p.

To get the definition of the projective space over a finite field $\mathbb{P}^n(\mathbb{F}_p)$, where p is prime, we simply combine the definitions above. The points in $\mathbb{P}^n(\mathbb{F}_p)$ are thus the set of equivalence classes of $\mathbb{F}_p^{n+1} \setminus \{0\}$ under the equivalence relation defined in definition 1.1.1, but now for $\lambda \in \mathbb{F}_p^*$. The number of points in $\mathbb{P}^n(\mathbb{F}_p)$ is

$$\frac{p^{n+1}-1}{p-1} = p^n + p^{n-1} + \cdots + p + 1. \tag{1.1}$$

This formula counts the non-zero vectors in $\mathbb{F}_p^{n+1}$ and divides by $p-1$ to account for the equivalence classes under scalar multiplication by $\lambda \in \mathbb{F}_p^*$.

1.2. Projective special linear group

A group that will be used frequently later on in this thesis is the projective special linear group. In this section we will give a formal definition of this special linear group and the groups needed to do so, together with a short explanation of its order.

1.2.1. Definition of the projective special linear group

Before being able to give a definition of the projective special linear group, first the definition of the general linear group and the special linear group should be revisited.

Definition 1.2.1. Let $n \in \mathbb{N}$. The general linear group $GL(n,\mathbb{F}_q)$ is the group of $n \times n$ invertible matrices with real coefficients under matrix multiplication. This group can be defined as

$$GL(n,\mathbb{F}_q) = \{A \in M_{n\times n}(\mathbb{F}_q) \mid \det(A) \neq 0\},$$

where $M_{n\times n}(\mathbb{R})$ denotes the set of all $n \times n$ matrices with real coefficients and $\det(A)$ is the determinant of A.

the special linear group is a subgroup of $GL(n,\mathbb{F}_q)$ and is defined as follows:

$$SL(n,\mathbb{F}_q) = \{A \in M_{n\times n}(\mathbb{F}_q) \mid \det(A) = 1\}.$$

Definition 1.2.2 (Projective Special Linear Group)**.** The projective special linear group $\mathrm{PSL}(n,\mathbb{F}_q)$ is the group of $n \times n$ matrices with real coefficients and determinant equal to 1, modulo scalar matrices. This group can be defined as the quotient group:

$$\mathrm{PSL}(n,\mathbb{F}_q) = SL(n,\mathbb{F}_q)/Z,$$

where Z is the center of $SL(n,\mathbb{F}_q)$, consisting of scalar matrices of the form λI_n where $\lambda \in \mathbb{F}_q$ and $\det(\lambda I_n) = 1$. The group operation in $\mathrm{PSL}(n,\mathbb{F}_q)$ is induced by the matrix multiplication in $SL(n,\mathbb{F}_q)$.[16]

A group is called simple if it does not have any normal subgroups other than the trivial subgroup and itself. The group $\mathrm{PSL}(n,p) = \mathrm{PSL}(n,\mathbb{F}_p)$ is known to be simple for most values of n and p. Specifically, $\mathrm{PSL}(n,p)$ is simple for $n \geq 2$ and $p > 3$ and $\mathrm{PSL}(2,p)$ is simple for $p \geq 5$. The simplicity of $\mathrm{PSL}(n,p)$ makes it a fundamental building block in the classification of finite simple groups. These groups appear in various areas of mathematics, including geometry, number theory, and the theory of algebraic groups, but more importantly their study provides insight into the symmetries of projective spaces.

1.2.2. Order of the groups

In this section, we explain the reasoning behind the formula for the order of the projective special linear group $PSL(n,q) = PSL(n,\mathbb{F}_q)$.[13] To do so we begin with the order of the general linear group $GL(n,q) = GL(n,\mathbb{F}_q)$. The order of $\mathrm{GL}(n,q)$, the group of invertible $n \times n$ matrices with entries from $\mathbb{F}_q$, can be understood by counting the number of possible ordered bases for the vector space $\mathbb{F}_q^n$.

Each invertible matrix corresponds to an ordered basis, with the columns of the matrix representing the vectors in the ordered basis. To determine the number of such ordered bases, we need to count the number of possible choices for each vector in the basis.

For the first vector in the ordered basis, any nonzero vector can be chosen, giving us $q^n - 1$ choices. For the second vector, we need a vector that is not in the span of the first, yielding $q^n - q$ choices. After selecting the first i vectors, the number of choices for the next vector is $q^n - q^i$, as q^i is the dimension of the subspace spanned by the first i vectors. Combining this, the total number of possible ordered bases is given by:

$$\prod_{i=0}^{n-1} (q^n - q^i) = (q^n - 1)(q^n - q)\cdots(q^n - q^{n-1}).$$

Thus, the order of the general linear group $\mathrm{GL}(n,q)$ is this product.

Now for $SL(n,p)$, we use that it is the kernel of the determinant map, which is a surjective homomorpism from $GL(n,p)$ to $\mathbb{F}_p^*$. The first isomorphism theorem states that the kernel of a group homomorpism $\phi : G \to H$ is a normal subgroup N and that the quotient group G/N is isomorphic with the image of the homomorpism $\phi(G)$[5]. So by this theorem, we get that $GL(n,p)/SL(n,p)$ is isomorphic to the image of $GL(n,p)$, which is equal to $\mathbb{F}_p^*$ since the homomorpism is surjective.

Lagrange's theorem tells us that the order of the group G is equal to the order of the subgroup times the index of the subgroup H in the group G : $|G| = |H|[G : H]$.[6] Here the index $[G : H]$ is the size of the set of left cosets of H in G and the size of the the quotient group if H is a normal group. So again applying this, it follows that:

$$|SL(n,q)| = \frac{|GL(n,q)|}{|\mathbb{F}_q^*|} = \frac{\prod_{i=0}^{n-1}(q^n - q^i)}{q-1}.$$

Where we used that $|\mathbb{F}^*| = q-1$.

Finally, recall from definition 1.2.2 that the center of $SL(n,q)$ is equal to the subgroup of scalar matrices with determinant 1. The number of elements in the center of $SL(n,q)$ is equal to the amount of matrices in this group whose scalar values are the n^{th} roots of unity, since the determinant of an $n \times n$ scalar matrix is the n^{th} power of the scalar value.

The center of $SL(n,q)$ is equal to the intersection of $SL(n,q)$ and the center of $GL(n,q)$. Furthermore, the center of $GL(n,q)$ is isomorphic to $\mathbb{F}_q^*$, which has $q-1$ elements as seen in section 2.1. Therefore, the kernel of the map from $SL(n,q)$ to $PSL(n,q)$ has order $gcd(n,q-1)$ and by Lagrange's theorem:

$$|PSL(n,q)| = \frac{|SL(n,q)|}{\gcd(n,q-1)} = \frac{\prod_{i=0}^{n-1}(q^n - q^i)}{(q-1)gcd(n,q-1)} \tag{1.2}$$

1.3. Group action

Lastly, the definition of an action of a group on a set and some related definitions and theorems are revisited. Most of the definitions, theorems and their proofs are obtained from the lecture notes of the Algebra 1 course.

Definition 1.3.1. Let X be a set and G be a group. We say that G *acts* on X if for every $g \in G$ and every $x \in X$ and element $g \circ x \in X$ is given such that

- $e \circ x = x$ for all $x \in X$, where e is the identity element of G.
- $(gh) \circ x = g \circ (h \circ x)$ for all $g, h \in G$ and $x \in X$.

If G acts on X, then the map $f : G \times X \to X$, given by $(g, x) \mapsto g \circ x$, is called an *action* of G on X. [2, p. 93]

In particular, the action we're most interested in is the Möbius transformation of 2×2-matrices of $\mathrm{PSL}(2, p)$ on the projective line $\mathbb{P}^1(\mathbb{F}_q)$, which is of the form

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z = \frac{az+b}{cz+d},$$

where $z \in \mathbb{P}^1(\mathbb{F}_q)$ and $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{PSL}(2, p)$. Here, the determinant condition $ad - bc \neq 0$ must be satisfied to ensure that the transformation is well-defined. This transformation maps each point on the projective line to another point on the projective line. Note that,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z = \infty \quad \text{if} \quad z = -\frac{d}{c}, \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \infty = \frac{a}{c}, \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z = 0 \quad \text{if} \quad z = -\frac{b}{a}.$$

Now for the following definition, let the group G act on the set X. Two elements $x, y \in X$ are called equivalent under G, denoted $x \sim_G y$, if there is some $g \in G$ such that $g \circ x = y$. The relation $\sim_G$ is symmetric:

$$\begin{aligned} x \sim_G y &\implies g \circ x = y \text{ for some } g \in G \\ &\implies x = g^{-1} \circ y \text{ for some } g \in G \\ &\implies y \sim_G x. \end{aligned}$$

The relation is also reflexive, since $x = e \circ x$, and also transitive:

$$\begin{aligned} x \sim_G y \text{ and } y \sim_G z &\implies \text{there exist } g, h \in G \text{ such that } g \circ x = y \text{ and } h \circ y = z \\ &\implies \text{there exist } g, h \in G \text{ such that } (hg) \circ x = z \\ &\implies x \sim_G z. \end{aligned}$$

We conclude that $\sim_G$ is indeed an equivalence relation on X. [2, p. 96]

Definition 1.3.2 (Orbit)**.** Let the group G act on the set X and let $x \in X$. The orbit of x under G, denoted by Gx, is the subset

$$Gx = \{g \circ x : g \in G\}.$$

So the orbits are precisely the equivalence classes of $\sim_G$. Since two different equivalence classes are always disjoint, we have that for all $x, y \in X$,

$$\text{either } Gx = Gy, \text{ or } Gx \cap Gy = \emptyset.$$

We say that the action of G on X is transitive if there is exactly one orbit of X under G. [2, p. 96]

The transivity of an action of a group on a set can also be defined in another way. In particular, a 'higher' transitivity has the following definition.

Definition 1.3.3 (t-transitivity)**.** A group G acting on a set X is said to be *t-transitive* if for any two t-tuples of distinct elements $(x_1, x_2, \ldots, x_t)$ and $(y_1, y_2, \ldots, y_t)$ in X, there exists a group element $g \in G$ such that

$$g \circ (x_1, x_2, \ldots, x_t) = (y_1, y_2, \ldots, y_t).$$

In other words, the group G is t-transitive on X if it can map any ordered t-tuple of distinct elements to any other ordered t-tuple of distinct elements via its action.

The last definition given in this section is that of the Stabilizer subgroup:

Definition 1.3.4 (Stabilizer subgroup)**.** Let the group G act on the set X and let $x \in X$. If $g \circ x = x$, then we say that x is a *fixed point* of g. The *stabilizer subgroup* of x in G, notation G_x, is the subset

$$G_x = \{g \in G : g \circ x = x\}.$$

Note that we can also speak of the a *stabilizer subgroup of a (sub)set*, say B, in G. In this case we get that:

$$G_B = \{g \in G : g \circ x \in B, \forall x \in B\}.$$

Lastly we need to recall the following three important theorems:

Theorem 1.3.1 (Lagrange's Theorem)**.** Let G be a group and let $H \subseteq G$ be a subgroup. Then $|G| = [G : H] \cdot |H|$.

Proof. [2, p. 62] □

Theorem 1.3.2 (Cauchy's Theorem)**.** Let G be a finite group and let p be a prime number that divides the order of G. Then there exists an $x \in G$ with $ord(x) = p$.

Proof. [2, p. 68] □

Theorem 1.3.3 (Orbit-stabilizer Theorem)**.** Let G act on X, and let $x \in X$. Then the map $f : G/G_x \to Gx$ given by $f(aG_x) = a \circ x$ is a well-defined bijection. Hence,

$$\#Gx = [G : G_x] = \frac{|G|}{|G_x|}.$$

Proof. [2, p. 97] □

2

The Witt Designs

This chapter aims to state and explain the definition of Steiner systems and how these can be constructed together with common examples. After understanding what Steiner systems entail, we will look at different constructions of the special Steiner quintuple systems S(5,6,12) and S(5,8,24), which are also known as the Witt designs.

The Witt designs are very special and unique combinatorial structures due to several of their properties. Namely, the automorphism group of $S(5,6,12)$ is the Mathieu group M_{12}, while the automorphism group of $S(5,8,24)$ is the Mathieu group M_{24}, both of which are among the smallest sporadic simple groups (Mathieu groups) and possess many symmetry properties. The Witt designs are also related to other mathematical structures, such as the perfect error-correcting Golay codes which we will also encounter in this chapter as a way of constructing our Witt designs. After these codes we will look into a construction using the projective special linear group $PSL(2,p)$ for $p = 11$ and $p = 23$. For both constructions we prove several properties and finally we will prove their Steiner property.

2.1. What is a Steiner system?

2.1.1. Definition of a Steiner system

Before defining a Steiner system, first the definition of a t-design is given.

Definition 2.1.1 (t-design)**.** Let $t < k < v$ be positive integers, and let λ be a positive integer. We say that a collection $\beta = B_1, B_2, \ldots, B_N$ of distinct subsets of $\{1,2,\ldots,v\}$, called *blocks*, is a (t,k,v)-design with parameter λ (denoted $S_\lambda(t,k,v)$) if it satisfies the following two properties:

- For each i, $|B_i| = k$.
- For every subset $T \subset \{1,2,\ldots,v\}$ with $|T| = t$, there are exactly λ blocks B_i such that $B_i \supset T$.[7][8]

A Steiner system is a t-design. More specifically, it's a t-design with parameter $\lambda = 1$: $S_1(t,k,v)$. Thus now every subset of length t is contained in *exactly one* block. A more formal definition is given.

Definition 2.1.2 (Steiner system)**.** Let $t < k < v$ be positive integers. We say that a collection $\beta = B_1, B_2, \ldots, B_N$ of distinct subsets of $\{1,2,\ldots,v\}$, called *blocks*, is an (t,k,v)-Steiner system (denoted $S(t,k,v)$) if it satisfies the following two properties:

- For each i, $|B_i| = k$.
- For every subset $T \subset \{1,2,\ldots,v\}$ with $|T| = t$, there is exactly one i so that $B_i \supset T$.[7] [8]

2.1.2. Examples

Example 2.1.1. There are several examples of Steiner systems, one being the family of the projective spaces over a finite field: $\mathbb{P}^n(\mathbb{F}_q)$. The most straightforward example is the Fano plane $S := \mathbb{P}^2(\mathbb{F}_2)$, which is a finite projective space with the minimal possible number of points and lines. It has 7 lines and 7 points, with 3 points on every line and 3 lines going through every point[15], like in figure 2.1 below.

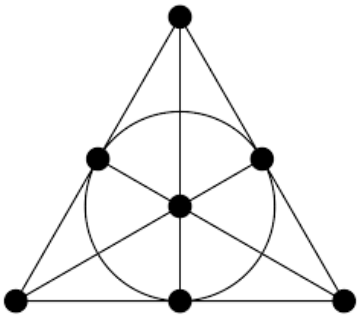

Figure 2.1: The Fano plane is a (2,3,7)-Steiner system.

If we now let the set of blocks β be the set of projective lines, then (S,β) is a Steiner system $S(2,3,7)$. That is, any subset of S consisting of 2 points is contained in exactly one block.

Example 2.1.2. Like mentioned before, this can be generalized. Specifically, Let $S := \mathbb{P}^n(\mathbb{F}_q)$ be the projective n-space over the finite field $\mathbb{F}_q$. Let β be the set of projective m-planes in S. Then (S,β) is a Steiner system $S(m+1, q^m+\cdots+1, q^n+\cdots+1)$, where the parameters are derived from equation 1.1.

Example 2.1.3. Another example that will be used later on in this thesis, is the *affine plane*. This plane is obtained from a projective plane by removing a line with all the points on it. Take for example the affine plane of order 3, that is $\mathbb{A}_2(\mathbb{F}_3)$. As a set this is simply $S := \mathbb{F}_3 \times \mathbb{F}_3$. This plane looks like the figure below. If we

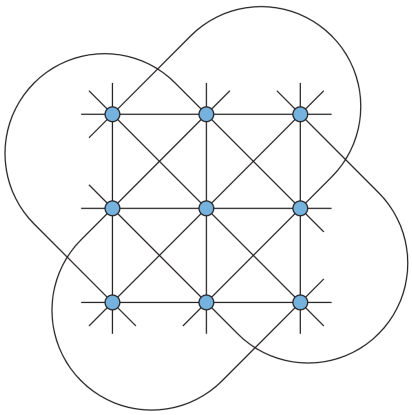

Figure 2.2: The affine plane $\mathbb{A}_2(\mathbb{F}_3)$ is a Steiner system $S(2,3,9)$.

now let the set of blocks β be the affine lines, then S,β is a Steiner system S(2,3,9).

2.2. Golay codes

One way to construct the Witt designs $S(5,8,24)$ and $S(5,6,12)$ is to use the extended binary Golay code and the more complex ternary Golay code, respectively. These Golay codes are the only perfect t-error correcting Golay codes with $t \geq 2$ and are widely used because of this. it will become clear what is meant with this later on in this section. Both of the constructions are obtained from J.G. Spandaw[8][9][4][10].

2.2.1. The extended binary Golay code

In this section we will write $V_d(\kappa)$ for the vector space of polynomials of degree $\leq d$ over the field κ. Using the basis $1, x, \ldots, x^d$ for this vector space, we get an isomorphism:

$$V_d(\kappa) \cong \kappa^{d+1}.$$

Here $\kappa = \mathbb{F}_2$ since every the code is binary, meaning that the bit can only be equal to 0 or 1. Thus every polynomial in $V_d(\mathbb{F}_2)$ can be seen a series of 12 bits. Now to get to the Golay code consider the factorization over $\mathbb{F}_2$:

$$\frac{x^{23}-1}{x-1} = x^{22} + \cdots + 1 = g(x)h(x),$$

where the polynomials $g(x) := x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$ and $h(x) := x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + x + 1$ are irreducible over $\mathbb{F}_2$. Now by multiplying the polynomials in $V_{11}(\mathbb{F}_2)$ with $g(x)$ we get a map:

$$G : V_{11}(\mathbb{F}_2) \to V_{22}(\mathbb{F}_2),$$

This map is linear and injective and by looking at the polynomials as binary codes again this becomes:

$$G : \mathbb{F}_2^{12} \to \mathbb{F}_2^{23}.$$

Finally, this is our linear *code* called the *binary Golay code:* G_{23}. The elements of the image of G in $\mathbb{F}_2^{23}$are called *codewords*. The weight of a codeword in $\mathbb{F}_2^{23}$ is the number of non-zero coefficients(bits). Since we have that

$$G(0) = 0, \quad G(1) = g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

Using maple code A.1 it is found that the minimal weight of a non-zero codeword is 7. Thus Each non-zero codeword is 23 bits long, encodes 12 bits of information and has a minimum weight of 7, making the binary Golay code a [23, 12, 7]-code.

This linear code is an error-correcting code, which is used in digital communication and data storage to detect and correct errors that may occur during data transmission or storage. This code adds redundant bits to the original codewords so that errors can be detected and corrected. The Golay binary code is 3-error correcting code, that is it can correct up to 3 error-bits within the original codeword.

The binary Golay code is also called *perfect*, which is explained as follows. In a space with 2^{23} points, each of the 2^{12} codewords can be seen as the center of a sphere of radius $t = 3$. A perfect code is one where these spheres completely cover the space. This means that every possible codeword in $\mathbb{F}_2^{23}$is within 3 errors of exactly one original codeword in $\mathbb{F}_2^{12}$, and thus can be corrected.

Now to construct the Steiner system, Maple code from appendix A.1 is used. This code constructs all these $2^{12} = 4096$ codewords and determines their weights. It was found that the codewords can have weights 0, 7, 8, 11, 12, 15, 16, or 23 only. The number of codewords with these weights are 1, 253, 506, 1288, 1288, 506, 253, and 1, respectively.

Finally, the extended binary Golay code

$$\tilde{G} : \mathbb{F}_2^{12} \to \mathbb{F}_2^{24}$$

is obtained from the binary Golay code G_{23} by adding a parity bit. This parity bit is 0 for the codewords with even weight and 1 for the codewords with odd weight. We denote this extended binary Golay code as $\tilde{G}_{24}$. Obviously from the results of the ordinary binary Golay code it follows that the weights occurring in this extended code are 0, 8, 12, 16, and 24 and the number of codewords with these weights are 1, 759, 2576, 759, and 1, respectively. This gives us our $S(5,8,24)$ Steiner system:

Theorem 2.2.1. Let $S = \{1, \ldots, 24\}$ and let B be the set of 8-subsets of S determined by the codewords in $\tilde{G}_{24}$ of weight 8. A subset B of S of size 8 is in B if and only if the word

$$f_B(x) := \sum_{\beta \in B} x_\beta$$

of length 24 and weight 8 is in $\tilde{G}_{24}$. Then (S, B) is a Steiner system $S(t, k, v)$ with $(t, k, v) = (5, 8, 24)$.[8]

Proof. We give a proof by contradiction. Suppose that a 5-subset A of S is contained in two distinct blocks $B_1, B_2 \in B$ of size 8. Then obviously $|B_1 \cap B_2| \in \{5, 6, 7\}$. If $|B_1 \cap B_2| = 5$, then the difference of the corresponding codewords of B_1 and B_2 in $\tilde{G}_{24}$ has weight $2 \times (8-5) = 6$. If $|B_1 \cap B_2| = 6$, then this weight is $2 \times (8-6) = 4$, and if $|B_1 \cap B_2| = 7$, then this weight is $2 \times (8-7) = 2$.

These differences of codewords are also codewords since $\tilde{G}_{24}$ is a linear code. However, we know that $\tilde{G}_{24}$ does not contain words of weight 6, 4, or 2, so we have a contradiction. We conclude that each 5-subset is contained in at most one block $B \in B$.

The number of 5-subsets of S is $\binom{24}{5} = 42,504$. The number of blocks is the number of codewords in $\tilde{G}_{24}$ of weight 8, which is 759. The number of 5-subsets of each block is $\binom{8}{5} = 56$. Since

$$759 \times 56 = 42,504,$$

we conclude that each 5-subset occurs in exactly one block. [8] □

2.2.2. The extended ternary Golay code

The same can be done when constructing the Steiner system $S(5, 6, 12)$. However, now the ternary Golay code is used meaning that $\kappa = \mathbb{F}_3$. Consider the factorization over $\mathbb{F}_3$:

$$\frac{x^{23} - 1}{x - 1} = x^{22} + \cdots + 1 = g(x)h(x),$$

where $f(x) = x^5 + 2x^3 + x^2 + 2x + 2$ and $g(x) = x^5 + x^4 + 2x^3 + x^2 + 2$ are irreducible. Again we get our ternary Golay code by multiplying all the polynomials in $V_6(\mathbb{F}_3)$ with $g(x)$:

$$G : V_6(\mathbb{F}_3) \to V_{11}(\mathbb{F}_3),$$

If we also look at the coefficients of the polynomials as being the trigits in our codewords, we get the following map:

$$G : \mathbb{F}_3^6 \to \mathbb{F}_3^{11}.$$

This ternary Golay code G_{11} is a perfect 2-error correcting code. There are $3^6 = 729$ codewords, where each non-zero codeword is 11 bits long, encodes 5 bits of information and has a minimum weight of 5, making it a $[11, 6, 5]_3$-code.

With the Maple code in appendix A.2 it is found that the codewords can have weights 0, 5, 6, 8, 9 or 11 only with the number of codewords with these weights being 1, 132, 132, 330, 110 and 24, respectively. Adding a zero-sum check digit to every codeword gives the extended ternary Golay code:

$$\tilde{G} : \mathbb{F}_3^6 \to \mathbb{F}_3^{12}.$$

This check digit adds the digit that is needed to make the sum of the digits of the extended codeword equal to zero modulo 3. Using maple, the codewords of $\tilde{G}_{12}$ have weight 0, 6, 9 or 12 and the number of codewords with these weights are 1, 264, 440 and 12. So there are 264 words with exactly 6 zero digits and 6 non-zero digits.

If we consider the positions of the non-zero digits in these 264 words, we get 132 blocks of length 6 in $\{0, 1, 2, \cdots, 11\}$. Namely, take for example the extended codeword $f(x) = (2, 2, 1, 2, 0, 1, 0, 0, 0, 0, 0, 1)$. This codeword corresponds with the block $B_1 = \{0, 1, 2, 3, 5, \infty\}$. However, another extended code word is $-f(x) = (1, 1, 2, 1, 0, 2, 0, 0, 0, 0, 0, 2)$, Which corresponds to the same block. Thus the 264 codewords form 132 pairs and each of these pairs form a block. Finally,these 132 blocks then form a Steiner system $S(5, 6, 12)$.

Theorem 2.2.2. Let $S = \{1, \ldots, 12\}$ and let B be the set of 6-subsets of S determined by the positions of the non-zero digits of the codewords in $\tilde{G}_{12}$ of weight 6. A subset B of S of size 6 is in B if and only if the word

$$f_B(x) := \sum_{\beta \in B} x_\beta$$

of length 12 and weight 6 is in $\tilde{G}_{12}$. Then (S, B) is a Steiner system $S(t, k, v)$ with $(t, k, v) = (5, 6, 12)$.[8]

Proof. We give a proof by contradiction. Suppose that a 5-subset A of S is contained in two distinct blocks $B_1, B_2 \in B$ of size 6. Then $|B_1 \cap B_2| = 5$ and the difference of the corresponding codewords of B_1 and B_2 in $\tilde{G}_{12}$ has weight $2 \times (6-5) = 2$.

These differences of codewords are also codewords since $\tilde{G}_{12}$ is a linear code. However, we know that $\tilde{G}_{12}$ does not contain words of weight 2, so we have a contradiction. We conclude that each 5-subset is contained in at most one block $B \in B$.

The number of 5-subsets of S is $\binom{12}{5} = 792$. The number of blocks is the number of codewords in $\tilde{G}_{12}$ of weight 6 divided by 2 , which is 132. The number of 5-subsets of each block is $\binom{6}{5} = 6$. Since

$$132 \times 6 = 792,$$

we conclude that each 5-subset occurs in exactly one block. [8] □

2.3. Construction using $PSL(2,11)$

In this section the Steiner system $S(5,6,12)$ will be constructed using the projective special linear group PSL(2,11) obtained from Jeroen Spandaw[8] [9][10]. After constructing the Steiner system a proof, confirming its Steiner property, will be given.

Consider the group $G := \mathrm{PSL}(2,p)$ that acts on the projective line $S := \mathbb{P}^1(\mathbb{F}_p) = \hat{\mathbb{F}}_p = \mathbb{F}_p \cup \{\infty\} = \mathbb{Z}/p\mathbb{Z} \cup \{\infty\}$, where the action is the Möbius transformation defined in section 1.3. From equation 1.2 we derive that this group has an order of:

$$\frac{(p^2-1)(p^2-p)}{2(p-1)} = \frac{1}{2}(p+1)p(p-1). \tag{2.1}$$

In this section we will take $p = 11$, then the group has an order of 660.

Now we let $B_1 = \{0,1,3,4,5,9\}$ be the first *starting block* of size $k = 6$ in the set of blocks β in our Steiner system (S,β). We will use this starting block B_1 to construct the other blocks. Note that B_1 consists of exactly the 5 squares of $\mathbb{F}_{11}$ plus zero, since $B_1 = \{0,1,3,4,5,9\} = \{0,1,4,9,16,25\} mod 11$.

Using the maple code in appendix A.3, we find that the elements of the group $PSLL(2,11)$ that stabilize the set B_1 are the following 5 matrices:

$$\left\{\begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \begin{bmatrix} 9 & 0 \\ 0 & 5 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix}\right\}$$

Thus the stabilizer subgroup $H = G_{B_1}$ of B_1 in G turns out to be the group of diagonal matrices in G, which indeed has an order of $(11-1)/2 = 5$. Therefore, since $|H| = 5$, according to the Orbit-stabilizer theorem 1.3.3 the orbit β of B_1 under the action in G has a size of

$$|\beta| = \frac{660}{5} = 132.$$

Each block B in this orbit contains 6 subsets of length 5, since you can delete an element of the block and be left with a different subset of length 5 in 6 different ways. Doing this for every block will result in an array that has a total of $132 \times 6 = 792$ 5-subsets of S. The array will look as follows:

B_1	$B_1 \setminus \{0\}$	$B_1 \setminus \{1\}$	$B_1 \setminus \{4\}$	$B_1 \setminus \{9\}$	$B_1 \setminus \{16\}$	$B_1 \setminus \{25\}$
B_2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$	$\vdots$					$\vdots$
B_{132}	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Note that the total number of possible subsets of length 5 of S is also equal to $\binom{12}{5} = 792$. This construction (S,β) then forms a Steiner system $S(t,k,v)$ with parameters $(t,k,v) = (5,6,12)$.

To actually prove that this construction is indeed a Steiner system, we need to answer the following two questions. Firstly, can we prove that the stabilizer subgroup has order 5? Because if the order of the stabilizer subgroup H wouldn't have been equal to 5 then the order of the orbit β would not have been equal to 132. This way, we wouldn't have had the coincidence that the number of 5-subsets in our array is the exact same as the possible number of 5-subsets in S: $\binom{12}{5} = 792$. And secondly, can we prove that all subsets of length 5 in our array are pairwise distinct? Since if they are not, then not all possible 5-subsets would be contained in exactly one block.

2.3.1. Stabilizer subgroup of B_1

We'll begin with the proof of the order of the stabilizer subgroup H being equal to the diagonal matrices, lets call this set $D :=$ {diagonal matrices in G}. Despite this already being confirmed by the maple code in appendix A.3 used in the previous section, a formal proof will now be presented. The proof is split into two parts: The first part proves that $|H| \geq |D| = 5$ and the second part proves that $|H| \leq |D| = 5$.

Proof. For the first proof it suffices to show that the 5 diagonal matrices do indeed stabilize the block B_1, which is rather easy. Namely, let $g = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} \in D$ over $\mathbb{F}_{11} = \mathbb{Z}/11\mathbb{Z}$ act on $S = \hat{\mathbb{F}}_{11}$ via the Möbius transformation. Since g is an element of $G = PSL(2,11)$ it must hold that $ad = 1 \mod 11$, equivalently $d = a^{-1} \mod 11$. This means that these matrices map $x \in S$ to

$$g \circ x = \frac{ax+0}{0x+a^{-1}} = a^2 x.$$

In particular in our starting block B_1, 0 gets mapped to 0 and the finite squares to finite squares (it also maps infinity to infinity). Hence these matrices stabilize B_1 and therefore $|D| \leq |H|$. □

We're left with the task of proving that the stabilizer subgroup of B_1 in PSL(2,11) can't be bigger than these 5 matrices, that is $|H| \leq |D|$. A proof of this was presented by Dion Gijswijt[3][9]:

Proof. Firstly, note that G acts transitively on S, which means that there is only one orbit and this orbit is the set S itself. Therefore the size of the orbit is $|Gx| = |S| = 12$ and by the Orbit-Stabilizer theorem 1.3.3 the size of the stabilizer subgroup is $|G_x| = \frac{|G|}{|Gx|} = \frac{660}{12} = 55$ for any $x \in S$. For the next steps recall that $H = G_{B_1}$, which is the stabilizer subgroup of the whole set B_1.

Now let $H_0 = H \cap G_0$ be the stabilizer subgroup in H of 0, in other words these are the matrices in H that map 0 to 0. It then follows by the previous part of the prove and by the fact that H_0 is a subgroup of G_0 that $|D| \leq |H_0| \leq |G_0|$. Then by Lagrange's theorem 1.3.1 and because D is also a subgroup of H_0 this implies that $|D| = 5$ divides $|H_0|$ and moreover $|H_0|$ divides $|G_0| = 55$. Therefore it can be concluded that either $H_0 = D$ or $H_0 = G_0$. It is easily shown that $H_0 \neq G_0$ as for example $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \in G_0$ maps $9 \in B_1$ to $\frac{9}{10} = -9 = 2 \mod 11$, which is not in B_1. Thus we've shown that $H \cap G_0 := H_0 = D$.

The problem is now reduced to showing that $H_0 = H$, i.e. that $H_0 H$ since H_0 is a subgroup of H. We will prove this by contradiction, so let's assume that $H_0 < H$. Note that by definition the group $H_0 = D$ acts transitively on $B_1 \setminus \{0\}$ as this group maps 0 itself and all the other elements of B_1 to the elements of B_1 again.

The assumption $H_0 < H$ implies that there is an $h \in H$ such that $h \notin H_0$. This means that h does not map 0 to itself but to an element of $B_1 \setminus \{0\}$. Thus because of our assumption and the fact that $H_0 = D$ acts transitively on $B_1 \setminus \{0\}$, we now know that H acts transitively on B_1. Then again by the Orbit-Stabilizer theorem: $\forall x \in B_1 : |Hx| = [H : H_x] = [H : H_0] = \frac{|H|}{|H_0|} = \frac{|H|}{5} = 6$, Hence $|H| = 6 \cdot 5 = 30$.

To come to a contradiction it will be shown that, because of our assumption, H consists of more than 30 elements. To do this we use the following: Since $H_0 = D$ acts transitively on $B_1 \setminus \{0\}$ and since H acts transitively on B_1, we conclude that H acts 2-transitively on B_1. We give a short proof for this statement:

Proof. Let $(a,b) \in B_1$ and $(c,d) \in B_1$. Since H is transitive, we know that there exists a $g_1 \in H$ that maps a to 0 and a $g_2 \in H$ that maps c to 0. We then get that acting on the tuples gives $g_1 \circ (a,b) = (0,b')$ and $g_2 \circ (c,d) = (0,d')$ for some $b', d' \in B_1 \setminus \{0\}$. Since H_0 acts transitively on $B_1 \setminus \{0\}$, we know that there exists a map $g_3 \in H_0$ that maps b' to d' and 0 to 0. Thus taking the following map: $g(x) = g_2^{-1} \circ g_3 \circ g_1(x) = g_2^{-1}(g_3(g_1(x)))$, we get that $g \circ (a,b) = (c,d)$. This holds for every $a,b,c,d \in B_1$ as they were taken arbitrarily. □

Also, if $x \neq y$ in B_1 then $H_x \neq H_y$ since the 5 matrices that map x to itself and map the other elements to each other will definitely not map y to itself. Hence, because $H_x \cap H_y$ is a subgroup of H_y, we must have that $|H_0 \cap H_y|$ is a divisor of 5 and $|H_0 \cap H_y| < 5$. Therefore it can only be that $|H_0 \cap H_y| = 1$ and from this it can be concluded that $H_x \cap H_y = \{e\}$ $\forall x \neq y \in B_1$.

Combining the two arguments above, $\forall x, y \in B_1$: if $x \neq y$ in B_1, then there is an $h \in H$ such that $h(x) = y$ and $h(y) = x$. In fact, h^2 fixes both x and y, so $h^2 \in H_x \cap H_y = \{e\}$, which is equivalent to saying that h has order 2. We will now begin counting the number of elements we can find in H.

Firtsly, there are $\binom{6}{2} = 15$ possible pairs in B_1 and an $h \in H$ with order 2 swapping a pair, can swap at most 3 pairs of those 6 elements at ones. So there are at least $\frac{15}{3} = 5$ elements of order 2 in H. Secondly, the union of the 6 stabilizer subgroups H_x with $x \in B_1$ contains $6 \times (5-1) = 24$ elements of order 5. Finally, By Cauchy's theorem 1.3.2, $|H| = 30 = 2 \cdot 3 \cdot 5$ implies that H has an element of order 3.

So together with the identity element, we have found $1+5+24+1 = 31$ distinct elements in H, contradicting $|H| = 30$. This shows that the assumption $H_0 < H$ was wrong and therefore $H_0 = H$.

□

2.3.2. Proof that it is a Steiner system

The second question is also already confirmed to be true by the maple code in appendix A.4. Here the orbit is constructed where, after the whole array is constructed in the way described in section 2.3., for every two 5-subsets it is verified if they are pairwise distinct. If 2 are found that are not distinct then the code will give back false, otherwise it will have true as output. It turns out that the array indeed only consist of pairwise distinct 5-subsets. However, again a formal prove of this fact is desired and given. Before doing this we need the following theorem.

Theorem 2.3.1. Given a $S_\lambda(t,k,v)$ design (S,β) and a subset A_j of S with $|A_j| = j \in \{0,\dots,t\}$. Let λ_j denote the number of blocks $B \in \beta$ containing A_j. Then

$$\frac{\lambda_j}{\lambda} = \frac{\binom{v-j}{t-j}}{\binom{k-j}{t-j}}.$$

In particular, this number does not depend on the choice of the subset A_j of S, but only on the size j of the subset A_j.

Proof. [1, p. 56] □

Note that λ_0 is equal to the number of blocks $|\beta|$, λ_1 is equal to the number of blocks containing a given point of S and lastly λ_t is equal to λ by definition.

Example 2.3.1. Recall that the Fano plane $\mathbb{P}^2(\mathbb{F}_2)$ defines a Steiner system $S(2,3,7)$. We have $|\beta| = \lambda_0 = \binom{7}{2}/\binom{3}{2} = 21/3 = 7$, $\lambda_1 = \binom{6}{1}/\binom{2}{1} = 6/2 = 3$ and $\lambda_2 = \lambda = 1$.

The following proof does not only confirm that all these 5-subsets are distinct, but it also again proves that the stabilizer subgroup H in G of B_1 has order 5. Hence it is a independent proof of the construction being a Steiner system $S(5,6,12)$. This proof is mostly obtained from the book on design theory from T. Beth, D. Jungnickel and H. Lenz[11].

Proof. What will be used in this proof is that $G = PSL(2,11)$ is 3-transitive on unordened triples. This means that in our construction any 3 elements in a block can simultaneously be mapped to any other 3 elements. From this it can be concluded that every possible subset of length 3 must be contained in a certain number of blocks, which is denoted as λ_3 as seen in theorem 2.3.1.

Recall that $|G_B| \geq 5$, as we've found that the 5 diagonal matrices $D \in G$ stabilize the starting block B_1. Hence $|G_B|$ is a multiple of 5, say $|G_B| = 5m$ as D is a subgroup of G. Thus, by the Orbit-Stabilizer theorem 1.3.3 the number of blocks equals $|\beta| = \lambda_0 = \frac{|G|}{|G_B|} = \frac{660}{5m}$. Then by theorem 2.3.1 $\lambda_0 = \lambda \cdot \frac{\binom{12}{5}}{\binom{6}{5}} = 132 \cdot \lambda = \frac{660}{5m}$, which gives $\lambda = \frac{1}{m}$. Again applying theorem 2.3.1 finally gives $\lambda_3 = \lambda \cdot \frac{\binom{9}{2}}{\binom{3}{2}} = \frac{1}{m} \cdot 12 = \frac{12}{m}$.

So we now know that the number of blocks containing 3 given points equals $\frac{12}{m}$. Let's say those 3 given points are $\infty, 0$ and 1. We know that the blocks $B_1 - 3 = \{\infty, 0, 1, 2, 6, 9\}$ and $B_1 - 4 = \{\infty, 0, 1, 5, 8, 10\}$ are contained in the orbit β of B_1 under G, because the corresponding matrices are in the group G. The same argument can be repeated, together with the fact that $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \in PSL(2,11)$, to find that the following blocks are also

contained in the orbit β:

$$A \circ (B-3) + 2 = \{\infty, 0, 1, 2, 7, 8\}$$
$$A \circ (B-3) + 6 = \{\infty, 0, 1, 4, 5, 6\}$$
$$A \circ (B-3) + 1 = \{\infty, 0, 1, 6, 7, 10\}$$
$$A \circ (B-1) + 7 = \{\infty, 0, 1, 3, 4, 7\}$$
$$A \circ (B-1) + 4 = \{\infty, 0, 1, 4, 8, 9\}$$

So we have now found seven distinct blocks containing $\{\infty, 0, 1\}$, which tells us that $\lambda_3 \geq 7$. Hence, since $\lambda_3 = \frac{12}{m}$ we know that $m = 1$. Using the maple code in appendix A.5, it's found that the 12 blocks containing $\{\infty, 0, 1\}$ are the following:

$$\left\{ \begin{array}{l} \{0,1,2,3,9,\infty\}, \{0,1,2,4,7,\infty\}, \{0,1,2,5,6,\infty\}, \{0,1,2,8,10,\infty\}, \\ \{0,1,3,4,8,\infty\}, \{0,1,3,5,7,\infty\}, \{0,1,3,6,10,\infty\}, \{0,1,4,5,10,\infty\}, \\ \{0,1,4,6,9,\infty\}, \{0,1,5,8,9,\infty\}, \{0,1,6,7,8,\infty\}, \{0,1,7,9,10,\infty\} \end{array} \right\}$$

These are the blocks $\{\infty, 0, 1, a, b, c\} \in \beta$, where $\{a, b, c\}$ are the rows, columns and transversals of the following 3×3-matrix:

$$\begin{bmatrix} 2 & 3 & 5 \\ 6 & 7 & 10 \\ 9 & 4 & 8 \end{bmatrix}$$

We can look at the entries of the matrix as being the points and the rows, colums and transversals as being the lines of the affine plane in figure 2.2. Hence, Example 2.1.3 tells us that the rows, colums and transversals of this matrix form a Steiner system $S(2,3,9)$ on the set $S \setminus \{0, 1, \infty\}$.

Finally, since G is 3-transitive on unordered triple this holds for all $\binom{12}{3} = 220$ subsets of length 3. Hence, (S, β) is a Steiner system $S(5,6,12)$.

□

2.4. Construction using $PSL(2,23)$

The Steiner system $S(5,8,24)$ can be constructed using the projective special linear group $PSL(2,23)$ similarly to the construction in section 2.3.

Consider the group $G := \mathrm{PSL}(2,23)$ that acts on the projective line $S := \mathbb{P}^1(\mathbb{F}_{23}) = \hat{\mathbb{F}}_{23} = \mathbb{F}_{23} \cup \{\infty\} = \mathbb{Z}/23\mathbb{Z} \cup \{\infty\}$, where the action is still the Möbius transformation defined in section 1.3. From equation 2.1 we derive that this group has an order of

$$|G| = \frac{1}{2} \cdot 24 \cdot 23 \cdot 22 = 6072.$$

This time we take $B_1 = \{\infty, 0, 1, 12, 15, 21, 22\}$ as our starting block of size 8. The stabilizer subgroup G_{B_1} has an order of 8 and applying the Orbit-Stabilizer theorem 1.3.3 again tells us that the orbit β of B_1 under the action in G has a size of

$$|\beta| = \frac{6072}{8} = 759.$$

The number of possible subsets A of size $|A| = 5$ is $\binom{24}{5} = 42504$ and each block $B \in \beta$ contains $\binom{8}{5} = 56$ subsets A. Note that $56 \times 759 = 42504$, so the numbers indeed match again and this (S, β) is a Steiner system $S(5,8,24)$.

Conclusion

In this thesis we have explored the highly symmetrical combinatorial structures known as Steiner systems, with a particular focus on Witt designs. These structures play a significant role in various applications, including coding theory, where they help in the construction of error-correcting codes. The thesis delves into two main methods for constructing Witt designs: using the Golay codes and the projective special linear groups.

Firstly, however, the necessary group theory was presented. After giving the definition of a projective space over a finite field, the definition of the projective special linear group, needed to construct our Witt designs, was provided. Lastly, the definition of the group action, orbit and satbilizer subgroup were revisited together with a new definition about t-transitivity and three important theorems.

we looked into the constructions via the Golay codes. We found that these codes are t-error correcting codes, where $t = 3$ for the binary Golay code and $t = 2$ for the ternary Golay code. It was also found, using Maple code, that the codewords of weight 8 of the extended binary Golay code form the blocks of the Steiner system $S(5,8,24)$. Similarly, for the extended ternary Golay code we found that the set of blocks is determined by the positions of the non-zero trigits in the codewords of weight 6. Finally, both these statements were backed with a proof.

Then we we explored the construction of the Witt designs using the projective special linear groups, specifically $PSL(2,11)$ and $PSL(2,23)$. Here we chose a starting block that had a stabilizer subgroup of the right size. Because of this size of the stabilizer subgroup in our construction we found that the number of 5-subsets of the blocks exactly matched with the total number of possible 5-subsets, which was was a first sign of it being a Witt design. Lastly, we proved that the starting block did in fact have a stabilizer subgroup of this size and that all the 5-subsets in the blocks were pairwise distinct, which is equivalent to proving its Steiner property.

Bibliography

[1] Wolfgang Ebeling. *Lattices and Codes*. Springer Spektrum Wiesbaden, 3rd edition edition, 2012.

[2] D.C. Gijswijt. *Algebra 1 Academic Year 2020/2021*. Delft University of Technology, January 2021. URL https://brightspace.tudelft.nl/content/enforced/398362-AM1060%2b2021%2b3/algebra1.pdf.

[3] D.C. Gijswijt. personal communication. 2024.

[4] Eindhoven University of Technology. The witt designs, golay codes and mathieu groups. URL https://www.win.tue.nl/~aeb/2WF02/Witt.pdf.

[5] Wikipedia Group properies. First isomorphism theorem. 2011. URL https://groupprops.subwiki.org/wiki/First_isomorphism_theorem.

[6] Wikipedia Group properties. Lagrange's theorem. 2012. URL https://groupprops.subwiki.org/wiki/Lagrange%27s_theorem.

[7] Simon Rubinstein-Salzedo. Mathieu groups. *Standford University*, October 2011.

[8] J.G. Spandaw. Designs and steiner systems and sporadic simple groups. *Unpublished notes*, December 2023.

[9] J.G. Spandaw. A construction of the steiner system s(5; 6; 12) with proofs. *Unpublished notes*, June 2024.

[10] J.G. Spandaw. personal communication. 2024.

[11] H. Lenz T. Beth, D. Jungnickel. *Design Theory*, volume 1. Cambridge University Press, second edition edition, 1999.

[12] Wikipedia. Finite field. URL https://en.wikipedia.org/wiki/Finite_field.

[13] Wikipedia. Order formulas for linear groups. 2019. URL https://groupprops.subwiki.org/wiki/Order_formulas_for_linear_groups#Explanation_for_order_of_special_linear_group_over_a_finite_field.

[14] Wikipedia. Real projective space. 2023. URL https://en.wikipedia.org/wiki/Real_projective_space.

[15] Wikipedia. Fano plane. 2024. URL https://en.wikipedia.org/wiki/Fano_plane.

[16] Wikipedia. Projective linear group. 2024. URL https://en.wikipedia.org/wiki/Projective_linear_group.

A

Maple code

A.1. Maple code 1

This code was used to firstly find all the different codewords created by the binary Golay code and secondly find the different weights of these codewords and lastly find how many times these weights occur.

```
with(GroupTheory):
with(LinearAlgebra):

# Define the polynomials g(x) and h(x) over GF(2)
g := x^11 + x^9 + x^7 + x^6 + x^5 + x + 1:
h := x^11 + x^10 + x^6 + x^5 + x^4 + x^3 + x + 1:

# Define the binary Golay code polynomial
G := f -> expand(f*g) mod 2:

# Generate the base polynomials for V_11
V11 := [seq(x^i, i = 0 .. 11)]:

# Initialize the list with the zero polynomial
allPolynomials := [0]:

# Helper function to add new combinations to the list
addCombinations := proc(baseList, term)
    local newList, i, newPoly:
    newList := []:
    for i in baseList do
        newPoly := i + term:
        newList := [op(newList), newPoly]:
    end do:
    return newList:
end proc:

# Generate all combinations
for i from 0 to 11 do
    newTerms := addCombinations(allPolynomials, x^i):
    allPolynomials := [op(allPolynomials), op(newTerms)]:
end do:

# Generate the codewords by multiplying each polynomial by g(x)
codewords := [seq(G(f), f in allPolynomials)]:
numelems(codewords):

# Function to find the weight of a codeword
weight := proc(codeword)
    local coeffList, w, coeff:
    coeffList := convert(codeword, list):
    w := 0:
    for coeff in coeffList do
        if coeff <> 0 then
            w := w + 1:
```

```
        end if:
    end do:
    return w:
end proc:

# Calculate the weights of the codewords
weights := [seq(weight(c), c in codewords)]:

# Find the unique weights and their counts
unique_weights := sort([op(map(op, {op(weights)}))]):
weight_counts := map(w -> nops(select(x -> x = w, weights)), unique_weights):

# Output the codewords, their unique weights, and counts of each weight
codewords, unique_weights, weight_counts;
```

A.2. Maple code 2

This code was used to firstly find all the different codewords created by the ternary Golay code and secondly find the different weights of these codewords and lastly find how many times these weights occur.

```
with(GroupTheory):
with(LinearAlgebra):

# Define the polynomials g(x) and h(x) over GF(2)
f := x^5 + 2*x^3 + x^2 + 2*x + 2:
g := x^5 + x^4 + 2*x^3 + x^2 + 2:

# Define the ternary Golay code polynomial
G := poly -> expand(poly * g) mod 3:

# Generate the base polynomials for V_5
V5 := [seq(x^i, i = 0 .. 5)]:

# Initialize the list with the zero polynomial
allPolynomials := [0]:

# Helper function to add new combinations to the list
addCombinations := proc(baseList, term)
    local newList, i, newPoly:
    newList := []:
    for i in baseList do
        for coeff in [0, 1, 2] do
            newPoly := i + coeff * term:
            newList := [op(newList), newPoly]:
        end do:
    end do:
    return newList:
end proc:

# Generate all combinations
for i from 0 to 5 do
    newTerms := addCombinations(allPolynomials, x^i):
    allPolynomials := [op(allPolynomials), op(newTerms)]:
end do:

# Remove the zero polynomial if it was duplicated
allPolynomials := convert({op(allPolynomials)}, list):

# Generate the codewords by multiplying each polynomial by g(x)
codewords := [seq(G(poly), poly in allPolynomials)]:
numelems(codewords):

# Function to find the weight of a codeword
weight := proc(codeword)
    local coeffList, w, coeff:
    coeffList := convert(codeword, list):
    w := 0:
    for coeff in coeffList do
```

```
        if coeff <> 0 then
            w := w + 1:
        end if:
    end do:
    return w:
end proc:

# Calculate the weights of the codewords
weights := [seq(weight(c), c in codewords)]:

# Find the unique weights and their counts
unique_weights := sort([op(map(op, {op(weights)}))]):
weight_counts := map(w -> nops(select(x -> x = w, weights)), unique_weights):

# Output the codewords, their unique weights, and counts of each weight
unique_weights, weight_counts;
```

A.3. Maple code 3

This code was used for finding the stabilizer subgroup in SPL(2,11) of the starting block $B_1 = \{0,1,3,4,5,9\}$.

```
with(GroupTheory);
with(LinearAlgebra);

#Define determinant function for 2x2 matrices modulo 11
det := A -> (A[1, 1]*A[2, 2] - A[1, 2]*A[2, 1]) mod 11;

#Generate all matrices with determinant 1 mod 11 to define SL(2,11)
P := {};
for a from 0 to 10 do
    for b from 0 to 10 do
        for c from 0 to 10 do
            for d from 0 to 10 do
                A := Matrix([[a, b], [c, d]]);
                if det(A) = 1 then
                    P := P union {A};
                end if;
            end do;
        end do;
    end do;
end do;

#Initialize the set P for SL(2,11)
Pnew := {};

#Function to check if two matrices are equal element-wise‘
isEqual := proc(A, B)
    return A[1, 1] = B[1, 1] and A[1, 2] = B[1, 2] and A[2, 1] = B[2, 1] and A[2, 2] =
        B[2, 2];
end proc;

#Function to check if a matrix or its negative is in the set‘
matrixOrNegativeInSet := proc(matrix, set)
    local m;
    for m in set do
        if isEqual(matrix, m) or isEqual((-matrix) mod 11, m) then
            return true;
        end if;
    end do;
    return false;
end proc;

#Filter the set to remove equivalent matrices‘
for A in P do
    if not matrixOrNegativeInSet(A, Pnew) then
        Pnew := Pnew union {A};
    end if;
end do;
```

```
#Define the set B
B := {0, 1, 3, 4, 5, 9};

#Mobius transformation function with check for invertibility of the denominator
mobiusTransform := proc(A, x)
    local denom;
    denom := (A[2, 1]*x + A[2, 2]) mod 11;
    if denom = 0 or igcd(denom, 11) <> 1 then
        return NULL;
    else
        return (A[1, 1]*x + A[1, 2])/denom mod 11;
    end if;
end proc;

#Check if a matrix stabilizes the set B
isStabilizer := proc(A, S)
    local x, transformed;
    for x in S do
        transformed := mobiusTransform(A, x);
        if transformed = NULL or not member(transformed, S) then
            return false;
        end if;
    end do;
    return true;
end proc;

#Find all stabilizers in the group for the set B
stabilizers := proc(G, S)
    local g, result;
    result := {};
    for g in G do
        if isStabilizer(g, S) then
            result := result union {g};
        end if;
    end do;
    return result;
end proc;

#Calculate the stabilizer subgroup
stabilizerSubgroup := stabilizers(Pnew, B);
```

A.4. Maple code 4

This code was used to help prove that the construction using PSL(2,11) does indeed produce a Steiner system S(5,6,12), that is to prove that every 5-subset in the array of 792 5-subsets are pairwise distinct.

```
with(GroupTheory):
with(LinearAlgebra);

# Define determinant function for 2x2 matrices modulo 11
det := A -> (A[1, 1]*A[2, 2] - A[1, 2]*A[2, 1]) mod 11:

# Initialize the set P for SL(2,11)
P := {}:

# Generate all matrices with determinant 1 mod 11
for a from 0 to 10 do
    for b from 0 to 10 do
        for c from 0 to 10 do
            for d from 0 to 10 do
                A := Matrix([[a, b], [c, d]]):
                if det(A) = 1 then
                    P := P union {A}:
                end if:
            end do:
        end do:
    end do:
```

```
end do:

# Initialize the set for PSL(2,11) and name it P_new
P_new := {}:

# Function to check if two matrices are equal element-wise
isEqual := proc(A, B)
    return A[1, 1] = B[1, 1] and A[1, 2] = B[1, 2] and A[2, 1] = B[2, 1] and A[2, 2] =
        B[2, 2]
end proc:

# Function to check if a matrix or its negative is in the set
matrixOrNegativeInSet := proc(matrix, set)
    local m:
    for m in set do
        if isEqual(matrix, m) or isEqual((-matrix) mod 11, m) then
            return true:
        end if:
    end do:
    return false:
end proc:

# Filter the set to remove equivalent matrices for PSL(2,11)
for A in P do
    if not matrixOrNegativeInSet(A, P_new) then
        P_new := P_new union {A}:
    end if:
end do:

# Define the set B_1
B_1 := {0, 1, 3, 4, 5, 9}:

# Mobius transformation function with check for zero denominator
mobiusTransform := proc(A, x)
    local denom, denom_inv, numer:
    denom := (A[2, 1]*x + A[2, 2]) mod 11:
    if denom = 0 then
        return infinity: # Handle the infinity case
    else
        denom_inv := 1/denom mod 11:
        numer := (A[1, 1]*x + A[1, 2]) mod 11:
        return numer*denom_inv mod 11:
    end if:
end proc:

# Apply Mobius transformation to the entire set
applyMobiusToSet := proc(A, S)
    local x, result:
    result := {}:
    for x in S do
        result := result union {mobiusTransform(A, x)}:
    end do:
    return result:
end proc:

# Function to check if two sets are equal
setsEqual := proc(S1, S2)
    return S1 = S2:
end proc:

# Function to check if a set is already in the list of sets
setInList := proc(S, L)
    local existingSet:
    for existingSet in L do
        if setsEqual(S, existingSet) then
            return true:
        end if:
    end do:
    return false:
end proc:
```

```
# Calculate the orbit of B_1 under PSL(2,11)
orbit := {}:
for g in P_new do
    newBlock := applyMobiusToSet(g, B_1):
    if not setInList(newBlock, orbit) then
        orbit := orbit union {newBlock}:
    end if:
end do:

# Function to generate all subsets of length 5 from a set of length 6
generateSubsetsOfLength5 := proc(block)
    local subsSet, elem, subs:
    subsSet := {}:
    for elem in block do
        subs := block minus {elem}:
        subsSet := subsSet union {subs}:
    end do:
    return subsSet:
end proc:

# Generate all subsets of length 5 from all blocks in the orbit
allSubsetsOfLength5 := {}:
for block in orbit do
    allSubsetsOfLength5 := allSubsetsOfLength5 union generateSubsetsOfLength5(block):
end do:

# Check if all subsets are distinct
allDistinct := true:
allSubsetsList := convert(allSubsetsOfLength5, list):

for i from 1 to numelems(allSubsetsList)-1 do
    for j from i+1 to numelems(allSubsetsList) do
        if setsEqual(allSubsetsList[i], allSubsetsList[j]) then
            allDistinct := false:
            break:
        end if:
    end do:
    if not allDistinct then
        break:
    end if:
end do:

# Output the results
allDistinct;
```

A.5. Maple code 5

This code was used to find the blocks containing $\{0,1,\infty\}$ in the orbit β of B_1 under $G = SPL(2,11)$.

```
with(GroupTheory);
with(LinearAlgebra);

# Define a determinant function for 2x2 matrices modulo 11
det := A -> (A[1, 1]*A[2, 2] - A[1, 2]*A[2, 1]) mod 11;

# Initialize an empty set P to store matrices with determinant 1
P := {};
for a from 0 to 10 do
    for b from 0 to 10 do
        for c from 0 to 10 do
            for d from 0 to 10 do
                A := Matrix([[a, b], [c, d]]);
                if det(A) = 1 then
                    P := P union {A};
                end if;
            end do;
        end do;
    end do;
```

```
end do;

# Initialize an empty set P_new to store unique matrices up to negation
P_new := {};

# Define a function to check if two matrices are equal
isEqual := proc(A, B)
    return A[1, 1] = B[1, 1] and A[1, 2] = B[1, 2] and A[2, 1] = B[2, 1] and A[2, 2] =
        B[2, 2];
end proc;

# Define a function to check if a matrix or its negative is in a given set
matrixOrNegativeInSet := proc(matrix, set)
    local m;
    for m in set do
        if isEqual(matrix, m) or isEqual((-matrix) mod 11, m) then
            return true;
        end if;
    end do;
    return false;
end proc;

# Populate P_new with matrices from P that are unique up to negation
for A in P do
    if not matrixOrNegativeInSet(A, P_new) then
        P_new := P_new union {A};
    end if;
end do;

# Define the set B_1
B_1 := {0, 1, 3, 4, 5, 9};

# Define a function to perform the M bius transformation
mobiusTransform := proc(A, x)
    local denom, denom_inv, numer;
    denom := (A[2, 1]*x + A[2, 2]) mod 11;
    if denom = 0 then
        return infinity;
    else
        denom_inv := 1/denom mod 11;
        numer := (A[1, 1]*x + A[1, 2]) mod 11;
        return numer*denom_inv mod 11;
    end if;
end proc;

# Apply M bius transformation to a set S using matrix A
applyMobiusToSet := proc(A, S)
    local x, result;
    result := {};
    for x in S do
        result := result union {mobiusTransform(A, x)};
    end do;
    return result;
end proc;

# Check if two sets are equal
setsEqual := proc(S1, S2)
    return S1 = S2;
end proc;

# Check if a set S is in a list L
setInList := proc(S, L)
    local existingSet;
    for existingSet in L do
        if setsEqual(S, existingSet) then
            return true;
        end if;
    end do;
    return false;
end proc;

```

```
# Generate the orbit of set B_1 under the action of matrices in P_new
orbit := {};
for g in P_new do
    newBlock := applyMobiusToSet(g, B_1);
    if not setInList(newBlock, orbit) then
        orbit := orbit union {newBlock};
    end if;
end do;

# Check if a block contains all elements in a given set of elements
containsElements := proc(block, elements)
    local x;
    for x in elements do
        if not member(x, block) then
            return false;
        end if;
    end do;
    return true;
end proc;

# Find blocks in orbit that contain all elements in a given set
findBlocksContainingElements := proc(orbit, elements)
    local block, result;
    result := {};
    for block in orbit do
        if containsElements(block, elements) then
            result := result union {block};
        end if;
    end do;
    return result;
end proc;

# Find blocks in orbit that contain 0, 1, and infinity
blocksContaining_0_1_infinity := findBlocksContainingElements(orbit, {0, 1, infinity});
```