New Proofs of the Kochen–Specker Theorem via Hadamard Matrices

by

Liam Salt

BSc. (Hons.), Queen's University, 2021

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2023

Declaration of Committee

Name:	**Liam Salt**
Degree:	**Master of Science**
Thesis title:	**New Proofs of the Kochen–Specker Theorem via Hadamard Matrices**
Committee:	**Chair:** Michael Monagan Professor, Mathematics
	Petr Lisoněk Supervisor Professor, Mathematics
	Jake Levinson Committee Member Assistant Professor, Mathematics
	Jonathan Jedwab Examiner Professor, Mathematics

Abstract

We demonstrate that Hadamard matrices and their generalizations can be used to prove the Kochen–Specker (KS) theorem. In particular, we prove that large classes of classical and generalized Hadamard matrices can be used to construct so-called KS pairs, which provide such proofs. We construct an infinite family of KS pairs by showing that for any odd prime p, there exists a KS pair in $\mathbb{C}^{2p}$ using p^4 vectors and p^3 orthogonal bases. Each of these pairs is shown to correspond to a 1-factorization of the complete graph K_{2p}. We explore various computational approaches to the search for new KS pairs. We develop an integer linear programming approach for testing whether an arbitrary set of vectors and bases forms a KS pair, which we use to simplify some existing KS pairs.

Keywords: Kochen–Specker theorem, Hadamard matrix, difference matrix, Butson Hadamard matrix, generalized Hadamard matrix, quantum information, contextuality, quantum computing, integer linear programming

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Petr Lisoněk for his mentorship, advice, attention to detail, and patience throughout this process. I also would like to give thanks to Dr. Jake Levinson for serving on my committee, to Dr. Jonathan Jedwab for agreeing to examine this thesis, and to Dr. Michael Monagan for serving as chair.

I would like to acknowledge all the sources of financial support I have received throughout the course of this program. Specifically, I am grateful to NSERC for providing the funding for my research assistantships, to SFU for funding my graduate fellowship, and to the Department of Mathematics for providing various sources of funding. I also thank the Centre for Experimental and Constructive Mathematics (CECM) for its computational support, especially for providing access to its machines and MAGMA licence.

I am grateful to all of the friends I have made along the way, and especially to Sam and Jingzhou for organizing the Grad Support Session, which I have found immensely helpful in my time here.

I would like to thank all of my family, without whom I would have never made it this far. I want to express special appreciation to my mother, my grandma, and of course, all of my siblings: Mackenzie, Molly, Quinn, Bailey, Keigan, and Benji.

Finally, I want to acknowledge my partner Damara, whose unending love and support, mathematical and otherwise, make everything I do possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The main objective of this thesis is to expand upon and develop new techniques for using Hadamard matrices and their generalizations to construct Kochen–Specker (KS) pairs. These pairs are important to the fields of quantum mechanical foundations and quantum information where they are used to prove the famous Kochen–Specker theorem of 1967 [20], which establishes the impossibility of a non-contextual hidden variable model of quantum mechanics. The existence of such a pair was shown in the original paper by Simon Kochen and Ernst Specker, and subsequent efforts have focused in two overlapping directions.

The first direction focuses on finding the minimum size of a KS pair, where size typically refers to either or both of the number of vectors and number of bases, and depends on the vector space in general. Minimization of KS pairs has been well-explored in previous decades; while the original KS paper used 127 vectors in $\mathbb{R}^3$, considerable simplifications came in the 1990s from authors such as Mermin [29], Peres [31], Kernaghan [18], and Cabello [9]. Research along these lines has continued such that the smallest known 3-dimensional real KS pair consists of 31 vectors, a 2006 result of John Conway and Kochen reported in [32]. There are lower bounds on the possible sizes of KS pairs in $\mathbb{R}^3$; namely, [1] demonstrates there are no such KS pairs of size smaller than 18, and [24] establishes via an exhaustive search that there are none using fewer than 23 vectors. The second direction of research focuses on developing new connections and techniques related to the KS theorem from other areas of mathematics. For example, a connection between specific Hadamard matrices and KS pairs was explored by Petr Lisoněk in 2019 [25].

In line with this second direction, the goal of the research presented in this thesis is to study this connection to Hadamard matrices to better understand a method for generating KS pairs which was published in a 2022 paper of Waegell and Aravind [38]. It was hypothesized by Lisoněk that the fundamental mechanism of their method could be explained in more generality by the previously known connection between the KS theorem and Hadamard matrices [25], and therefore that these matrices could be a bountiful source of KS pairs. The computer-free aspects of this thesis serve to confirm this proposed ap-

proach, expand upon it, find significant simplifications and generalizations, and establish connections to other areas such as graph theory.

1.1 Outline

We begin with Section 1.2, where we provide some historical exposition to help motivate and build context for contemporary research in the study of KS pairs and quantum contextuality. The remainder of Chapter 1 comprises an introduction to some necessary mathematical background, including the definitions for the various generalizations of Hadamard matrices we make use of, and an introduction to the specific mathematical formulation of the KS theorem used in this thesis.

In Chapter 2, we present the computer-free results of the research. Starting with Section 2.2, we prove a new result that any generalized Hadamard matrix, subject to some restrictions on the order, can be used to construct a KS pair. This construction uses the entire row space of a generalized Hadamard matrix; similar constructions using subspaces of the row space are also possible, which is explored in Section 2.3. In Section 2.4, we produce an infinite family of KS pairs in $\mathbb{C}^{2p}$ for any odd prime p, using a family of generalized Hadamard matrices given by Dieter Jungnickel in 1979 [17]. We finish Section 2.4 by proving that each KS pair in the infinite family corresponds to a 1-factorization of the complete graph K_{2p} in a natural way.

In Chapter 3, we detail the significant computational element that was involved in this thesis, and explore different computational lenses through which one might search for KS pairs. We begin with Section 3.1.1, where we detail the computational machinery we developed to investigate the creation of new KS pairs. In particular, we develop an integer linear program (ILP) with which we are able to test whether any set of vectors can be used to prove the KS theorem. We present the data for several families of KS pairs we produce via this method. In Section 3.2, we present an alternative computational approach using graph theory. In Section 3.3, we use our ILP to computationally explore small integer-valued KS pairs in $\mathbb{R}^3$. In Section 3.3.1, we explore a method for creating such KS pairs as outlined by Cortez and Reyes [11]. We provide a slight simplification of the explicit KS pair they constructed, and apply our ILP broadly to attempt to construct similar pairs. In Section 3.3.2, we implement a computational approach which searches for integer-valued KS pairs in $\mathbb{R}^3$ by considering their potential automorphism groups.

We finish in Chapter 4 with a brief summary of some remaining questions and future research directions, as well as give some additional computational approaches that could be further developed.

1.2 Historical Background

With the introduction of quantum mechanics in the last century came much scientific debate surrounding its correct interpretation and its broader philosophical implications. In 1927, Max Born and Werner Heisenberg of the Niels Bohr Institute in Copenhagen were noted to have said, "We consider quantum mechanics to be a closed theory, whose fundamental physical and mathematical assumptions are no longer susceptible of any modification" [3]. Eight years later, Albert Einstein, Boris Podolsky, and Nathan Rosen introduced the so-called EPR paradox, a thought experiment which argued that Born and Heisenberg were incorrect about the completeness of quantum mechanical theory [15]. Their argument concluded that quantum theory, as stated by Born and Heisenberg, must be non-local; that is, they claimed that it must be a feature of Born and Heisenberg's theory that particles in some circumstances may be influenced instantaneously at a distance. The EPR paper posited that this necessarily violated the Heisenberg uncertainty principle, but suggested that any concerns could be remedied by considering a hidden variable (deterministic) model of quantum mechanics. A hidden variable model of quantum mechanics is a framework in which all quantum observables are regarded as possessing their values intrinsically; that is, it is a framework where we remove any requirement of indeterminacy in the measurement of quantum entities. This was a viewpoint expressed by Einstein and others in part because of the philosophical implications of its negation, famously summarized in a letter from Einstein to Born by "I am convinced [God] does not play dice" [14].

Over the following decades, some clarity was provided by various results, namely Bell's theorem. In 1964, John Bell proved that any hidden variable model of quantum mechanics is incompatible with one that is local [4]. Bell argued that if a hidden variable is assumed to be local, then there are certain mathematical constraints on the correlations among measurements of entangled particles, known as the Bell inequality. Then, Bell proved that there are valid physical systems which violate this inequality, showing that any hidden variable model most be non-local.

Following this, Bell in 1966 [5] and Simon Kochen and Ernst Specker in 1967 [20] independently showed that any hidden variable model must also be contextual. In this setting, contextuality refers to the notion that, in general, the results of experiments depend on the specific setting in which they are performed. This result is known as the (Bell)–Kochen–Specker theorem. Kochen and Specker understood that due to the Heisenberg uncertainty principle, there are restrictions on which quantum observables may be simultaneously measured. Each distinct possibility for simultaneous measurement is known as a context. Kochen and Specker proved that there exist valid physical configurations of quantum observables, those which are ostensibly independent of context, such that the measurement outcomes of the observables must actually differ according to the specific context. This result proved that any hidden variable model must be contextual.

There is still some debate regarding whether or not a hidden variable model should be discarded given these results. Notably, while Bell's theorem shows that a local hidden variable model is impossible, it does not rule out the possibility of a non-local hidden variable model. For many, the purpose of Bell's theorem is to demonstrate non-locality, rather than to contradict determinacy. Such non-local models include the Bohm interpretation, also known as de Broglie's pilot wave theory. Similarly, with regard to contextuality, a contextual hidden variable model remains possible. Contextual models are of interest for many reasons, among them that it has been noted that contextuality could be a source of advantage in quantum computing.

1.3 Mathematical Background

In this section, we outline some standards for notation and relevant mathematical background used throughout this thesis. In particular, we present notational standards for the mathematical objects originating from linear algebra, give a brief introduction to projective space, and introduce some terminology from graph theory. Apart from Section 1.3.1, unless stated otherwise, all definitions, propositions, and theorems in Section 1.3 are adapted from outside sources. The particular sources will be mentioned at the beginning of each subsection.

1.3.1 Linear Algebra, Groups, and Rings

i) All vectors are assumed to be row vectors. For example, if $v \in \mathbb{R}^3$, then we write:

$$v = (1, 2, 3) = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}^\top,$$

where $v^\top$ denotes the transpose of v. Hence, the inner product of two real vectors $u, v \in \mathbb{R}^n$ is given by $\langle u, v \rangle = uv^\top$, and the inner product of two complex vectors $u, v \in \mathbb{C}^n$ is given by $\langle u, v \rangle = uv^*$, where v^* denotes the conjugate transpose of v.

ii) We denote the set of $n \times n$ matrices with entries in a ring R by $\mathrm{Mat}_n(R)$. We denote the row space of a matrix A by $\mathrm{Row}(A)$, and assume throughout that this space is defined over the same ring as A; for example, if $A \in \mathrm{Mat}_n(\mathbb{F}_q)$, then $\mathrm{Row}(A)$ is a subspace of an n-dimensional $\mathbb{F}_q$-vector space.

iii) For a positive integer n, we denote the group of integers under addition modulo n by $\mathbb{Z}_n$.

iv) We denote the vector in $\mathbb{Z}_n^k$ with all entries equal to 1, the *all ones vector* of length k, by $\mathbf{1}$.

v) We write $\mathbb{F}_p$ to refer to the finite field with a prime number of elements p. Similarly, we write $\mathbb{F}_q$ to refer to the finite field with a prime power number of elements q.

vi) When $z \in \mathbb{C}$, we denote the complex conjugate of z by $\overline{z}$ and the modulus of z by $|z|$.

vii) We write ω_n to denote the complex primitive nth root of unity given by $\exp(2\pi i/n)$.

viii) If G is an additive group and K is a normal subgroup of G, we denote the *quotient map* by $Q\colon\ G \to G/K$, where $Q(g) = g + K$.

1.3.2 Projective Space

In this section, we briefly introduce projective space and the relevant notation used in this thesis. If not stated otherwise, all definitions come from Cox, Little, O'Shea [12].

Definition 1.3.1. *Given an n-dimensional vector space V over a field K, the* projective space of V*, denoted $\mathbb{P}(V)$ or $K\mathbb{P}^{n-1}$, is the set of equivalence classes of $(V \setminus \{\mathbf{0}\})/\sim$, where for any two $v, w \in V \setminus \{\mathbf{0}\}$, the equivalence is given by $v \sim w$ if and only if $v = \lambda w$ for some $\lambda \in K \setminus \{0\}$.*

Given a basis for V, if $v = (v_1, v_2, \ldots, v_n) \in V$, then we write the equivalence class of v in $\mathbb{P}(V)$ as $[v_1 : v_2 : \cdots : v_n]$, which can equivalently be represented by $[\lambda v_1 : \lambda v_2 : \cdots : \lambda v_n]$ for any $\lambda \in K \setminus \{0\}$. An element of $\mathbb{P}(V)$ is called a projective point*.*

Given this definition, it is natural to think of $K\mathbb{P}^{n-1}$ geometrically as the space of lines through the origin in K^n, and indeed these are in one-to-one correspondence.

Example 1.3.2. *The space $\mathbb{RP}^1$ is given by equivalence classes $[x_1 : x_2]$. For any element of $\mathbb{RP}^1$ with x_1 nonzero, we may rewrite the class as $[1 : x_2/x_1]$; if $x_1 = 0$, then x_2 must be nonzero, so we may rewrite this class as $[0 : 1]$. Therefore, the elements of $\mathbb{RP}^1$ are given exactly by $[0 : 1]$ and $[1 : m]$, for each $m \in K$. These correspond to the vertical line though the origin and the line of slope m through the origin, respectively.*

1.3.3 Graph Theory

In this section, we introduce some graph theoretical notation used in this thesis, mainly in Sections 2.4 and 3.2. Unless stated otherwise, all definitions in this section come from van Lint and Wilson [37]. We assume throughout that all graphs are simple and undirected.

Definition 1.3.3. *Let $G = (V, E)$ be a graph, and let $S \subseteq V$. If for any pair of distinct vertices $u, v \in S$ we have $\{u, v\} \notin E$, then S is an* independent set *of G. If S is not contained in any larger independent set of G, we say S is* maximal*. The size of the largest independent set of G is called the* independence number *of G, denoted $\alpha(G)$.*

Definition 1.3.4. *Let $G = (V, E)$ be a graph, and let k be a positive integer. A* k-factor of *G is a spanning k-regular subgraph of G (not necessarily connected). A k-factorization of G is an edge partition of G into disjoint k-factors.*

Example 1.3.5. *The possible* 1*-factors of* K_4 *are given by*

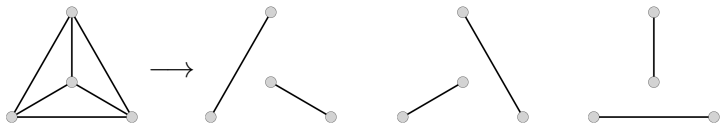

Figure 1.1: All possible 1-factors of K_4

Observe that these 1*-factors give the parts of a* 1*-factorization for* K_4*.*

1.4 Background on Hadamard Matrices and their Generalizations

In this section, we provide background information regarding Hadamard matrices and some of their generalizations. These matrices are of particular importance to this thesis, because they are starting objects of the novel approach for generating KS pairs presented in Chapters 2 and 3.

If not stated otherwise, any definitions, theorems, and proofs in this section come from van Lint and Wilson [37], Jungnickel [17] or Lampio [21].

1.4.1 Classical Hadamard Matrices

A *(classical) Hadamard matrix* of order n, named after French mathematician Jacques Hadamard, is an $n \times n$ matrix whose entries consist of $+1$ and -1, satisfying $HH^\top = nI$, or equivalently whose distinct rows are pairwise orthogonal. Matrices of this type were first considered by James Sylvester in 1867 [34], 26 years before Hadamard in 1893 [16]. An interesting note is that Sylvester is also credited with having coined the term "matrix", among several other terms such as "graph" and "discriminant".

Hadamard matrices satisfy several other properties, among them that for any Hadamard matrix H of order n, we have $H^\top H = nI$ and $\det(H) = \pm n^{n/2}$. This first property can be used to equivalently define Hadamard matrices as matrices whose entries consist of ± 1 and whose distinct columns are pairwise orthogonal. Another property is that any two distinct rows of a Hadamard matrix have matching entries in exactly half of their columns, and likewise, and two distinct columns have matching entries in exactly half of their rows.

Example 1.4.1. *Some small Hadamard matrices with* $n = 1, 2, 4$ *are given by*

$$H_1 = \begin{bmatrix} 1 \end{bmatrix}, \; H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \; H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

We also define a notion of equivalence for Hadamard matrices:

Definition 1.4.2. *Let H, H' be Hadamard matrices of the same order. We call H* monomial equivalent *to H' if H' can be obtained from H by applying a sequence of row permutations, column permutations, and multiplications of a row or of a column by ± 1.*

Definition 1.4.3. *A Hadamard matrix H is* normalized *if the first row and first column of H contain only $+1$.*

Note 1.4.4. *Specifying that the first row and column contain only $+1$ is only a convention. If the rows and columns of H are not indexed by positive integers, we say H is normalized if there exists a row and column of H containing only $+1$.*

Given this definition, it follows that through a sequence of multiplying rows and columns by ± 1, we have the following proposition:

Proposition 1.4.5. *Every Hadamard matrix is monomial equivalent to a normalized Hadamard matrix.*

Proposition 1.4.6 (van Lint, Wilson [37, Theorem 18.4])**.** *If H and H' are Hadamard matrices of respective order n and m, then the matrix given by the Kronecker product $H \otimes H'$ is a Hadamard matrix of order nm.*

In particular, if we take H_2 to be as in Example 1.4.1, then for any other Hadamard matrix H, we have

$$H_2 \otimes H = \begin{bmatrix} H & H \\ H & -H \end{bmatrix},$$

which by Proposition 1.4.6 gives a Hadamard matrix of order $2n$. Therefore, for $n \in \mathbb{Z}_{>0}$, we can recursively construct a Hadamard matrix of order 2^n via $H_{2^n} = H_2 \otimes H_{2^{n-1}}$, where H_2 and H_1 are as in Example 1.4.1. This is referred to as the *Sylvester construction* for order 2^n Hadamard matrices.

The Sylvester matrices do not constitute all Hadamard matrices, however. A necessary condition on the order n of a Hadamard matrix is that n must be 1, 2, or a multiple of 4:

Theorem 1.4.7 (van Lint, Wilson [37, Theorem 18.1])**.** *If H is a Hadamard matrix of order n, then $n = 1, 2$ or $n \equiv 0 \pmod 4$.*

That the converse of the above holds is known as *the Hadamard conjecture*, which is as of yet unresolved. As of the writing of this thesis, the smallest number n for which the existence of a Hadamard matrix of order n is unknown is $n = 668$.

There exist several constructions for Hadamard matrices of order not equal to a power of two. The first such examples came from Jacques Hadamard in his original discussion of these matrices [16], where he constructed Hadamard matrices of order 12 and 20. Decades later, in 1933, came the Hadamard matrix construction by Raymond Paley [30], from which comes the following theorem.

Theorem 1.4.8 (van Lint, Wilson [37, Theorem 18.5])**.** *Let q be a power of an odd prime. A Hadamard matrix of order $q+1$ exists if $q \equiv 3 \pmod 4$, and a Hadamard matrix of order $2(q+1)$ exists if $q \equiv 1 \pmod 4$.*

The proof of this theorem proceeds by giving explicit constructions of Hadamard matrices which meet the outlined criteria. We do not present the construction in this thesis, but code to generate the matrices is found in Appendix B.6.

1.4.2 Generalizations of Hadamard Matrices

The definition of a classical Hadamard matrix can be generalized in several ways. It is important to introduce these generalizations, because they allow us to extend the class of matrices with which we may generate KS pairs by the methods which are introduced in Chapter 2. One way to generalize classical Hadamard matrices is to ease the requirement that all entries be ± 1, by allowing the entries to take any values along the unit circle in the complex plane. This is defined in [21] by the following definition.

Definition 1.4.9. *A matrix $H \in \mathrm{Mat}_n(\mathbb{C})$ is a* complex Hadamard matrix *of order n if all entries h of H are such that $|h| = 1$ and $HH^* = nI$, where H^* denotes the conjugate transpose of H.*

That is, in analogy to the classical case, a complex Hadamard matrix is a matrix H with entries belonging to the unit circle whose rows are pairwise orthogonal with respect to the inner product for complex vector spaces; i.e. if v, w are distinct rows of a complex Hadamard matrix, then $vw^* = 0$.

Complex Hadamard matrices retain many other of the properties of classical Hadamard matrices as well, which are detailed in [21]. The property that the distinct rows of these matrices are pairwise orthogonal is most important to this thesis, because it is this property that guarantees the rows of an order n complex Hadamard matrix comprise a basis for $\mathbb{C}^n$.

In contrast to the classical case, it is true that a complex Hadamard matrix of order n exists for each $n \in \mathbb{Z}_{>0}$; for example, we may take the matrix of the n-point discrete Fourier transform [35].

We also define the complex Hadamard matrices where entries are restricted to powers of a fixed complex primitive root of unity:

Definition 1.4.10. *Let $n, q \in \mathbb{Z}_{>0}$. A complex Hadamard matrix $H \in \mathrm{Mat}_n(\mathbb{C})$ is of* Butson type $\mathrm{BH}(n, q)$ *if its entries consist only of integer powers of $\omega_q = \exp(2\pi i/q)$, i.e. powers of the same primitive qth root of unity. We also denote the set of all such matrices by $\mathrm{BH}(n, q)$.*

Note 1.4.11. *It is a competing standard to use the term complex Hadamard matrix to refer to $\mathrm{BH}(n, 4)$ matrices specifically. This is in part because Butson Hadamard matrices predate complex Hadamard matrices as in Definition 1.4.9.*

Note that classical Hadamard matrices are Butson Hadamard matrices of type $\mathrm{BH}(n, 2)$.

Example 1.4.12. *The following is an example of a Butson Hadamard matrix of type* $\mathrm{BH}(n = 6, q = 3)$*:*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \omega_3 & \omega_3 & \omega_3^2 & \omega_3^2 \\ 1 & \omega_3 & 1 & \omega_3^2 & \omega_3^2 & \omega_3 \\ 1 & \omega_3 & \omega_3^2 & 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3^2 & \omega_3 & 1 & \omega_3 \\ 1 & \omega_3^2 & \omega_3 & \omega_3^2 & \omega_3 & 1 \end{bmatrix}.$$

A separate generalization of Hadamard matrices comes in the form of what are commonly referred to simply as generalized Hadamard matrices, which were defined by David Drake in 1979 [13]. These matrices generalize the property of classical Hadamard matrices that the entries of any two distinct rows are equal in exactly half of their entries. They generalize this property by allowing the entries to belong to any finite group, and requiring that the difference between any two distinct rows contains each element of the group a uniform number of times.

These matrices are important to this thesis, because with Definition 1.4.18, we define a map which takes each generalized Hadamard matrix to a Butson Hadamard matrix, and hence an orthogonal basis for $\mathbb{C}^n$.

Definition 1.4.13. *Let G be a finite group of order g, and let λ be a positive integer. A* generalized Hadamard matrix *is a $g\lambda \times g\lambda$ matrix H with entries in G such that, for any two rows H_i and H_j with $i \neq j$, the multiset $\{H_{jk}H_{ik}^{-1} \colon 1 \leq k \leq g\lambda\}$ contains each element of G exactly λ times. If H is such a matrix, we say that it is* of type $\mathrm{GH}(G, \lambda)$, or $H \in \mathrm{GH}(G, \lambda)$.

If G is abelian, we usually write $H_{jk}H_{ik}^{-1}$ as $H_{jk} - H_{ik}$.

Under this definition, for p prime, Butson Hadamard matrices of type $\mathrm{BH}(n, p)$ are the same as those of type $\mathrm{GH}(C_p, n/p)$, where

$$C_p = \{\omega_p^k \colon 0 \leq k \leq p-1\}$$

under multiplication. In particular, classical Hadamard matrices of order $n > 1$ are those of type $\mathrm{GH}(C_2, n/2)$. However, there exist Butson Hadamard matrices that are not generalized Hadamard matrices. See the following example:

Example 1.4.14. *For q composite, there exist Hadamard matrices of type* $\mathrm{BH}(n, q)$ *that are not generalized Hadamard matrices. One such matrix is given in [21]:*

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & i & -i \\ 1 & -1 & -i & i \end{bmatrix}.$$

This cannot be a generalized Hadamard matrix: its entries belong to a group of order at least 4, but between the second and first rows, only 2 distinct elements appear in the difference, regardless of the group.

Proposition 1.4.15 (Jungnickel [17, Theorem 2.2])**.** *$H \in \mathrm{GH}(\mathrm{G}, \lambda)$ if and only if $H^T \in \mathrm{GH}(\mathrm{G}, \lambda)$.*

The above proposition gives us that generalized Hadamard matrices also exhibit balanced differences among their distinct columns.

Theorem 1.4.16 ([8, Theorem 3.5], or [13, Theorem 1.4])**.** *Let p be prime. If m and k are non-negative integers with $m \leq k$, then there exists a matrix of type $\mathrm{BH}(2^m p^k, p)$. If $k \neq 0$, then this matrix is of type $\mathrm{GH}(C_p, 2^m p^{k-1})$ as well.*

An explicit construction for the matrices guaranteed by Theorem 1.4.16 is provided in [8].

We introduce a map from matrices of type $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ to matrices of Butson type. First, we define a map between the vectors that form the rows:

Definition 1.4.17. *Let $\varphi_n \colon \mathbb{Z}_n^k \to \mathbb{C}^k$ be the map which sends $(v_1, \ldots, v_k) \in \mathbb{Z}_n^k$ to $(\omega_n^{v_1}, \ldots, \omega_n^{v_k})$. We likewise define ψ_n which maps $(\omega_n^{e_1}, \ldots, \omega_n^{e_k})$ to $(e_1, \ldots, e_k) \in \mathbb{Z}_n^k$.*

We define a map between generalized Hadamard matrices and Hadamard matrices of Butson type by applying φ_n to each row individually.

Definition 1.4.18. *We define a map $\Phi_n : \mathrm{GH}(\mathbb{Z}_n, \lambda) \to \mathrm{BH}(\lambda n, n)$, where for $H \in \mathrm{GH}(\mathbb{Z}_n, \lambda)$, the ith row of $\Phi_n(H)$ is given by applying φ_n to the ith row of H. Similarly, we define $\Psi_n \colon \mathrm{BH}(\lambda n, n) \to \mathrm{Mat}_{\lambda n}(\mathbb{Z}_n)$ by applying ψ_n to each row of a Butson Hadamard matrix.*

Example 1.4.19. *An example of Φ_3 applied to a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_3, 1)$, resulting in a matrix in $\mathrm{BH}(3, 3)$:*

$$\Phi_3\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3^2 & \omega_3 \\ 1 & \omega_3 & \omega_3^2 \end{bmatrix}.$$

Note 1.4.20. *The map Ψ_n cannot be considered an inverse of Φ_n for all values of n. The reason is that for composite n, Butson Hadamard matrices using powers of nth roots of unity, can achieve pairwise orthogonality among the distinct rows by different means than uniformity in each element; for example, if $n = 6$, then because $(x^6 - 1)/(x - 1)$ has non-*

trivial divisors over $\mathbb{Q}$*, we can have a* $\mathrm{BH}(4,6)$ *matrix*

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & \omega_6^3 & \omega_6^3 \\ 1 & \omega_6^3 & \omega_6 & \omega_6^4 \\ 1 & \omega_6^3 & \omega_6^4 & \omega_6 \end{bmatrix},$$

because, for example,

$$1 + 1 + \omega_6^3 + \omega_6^3 = 0, \quad 1 + \omega_6^3 + \omega_6 + \omega_6^4 = 0, \quad \textit{etc.}$$

Definition 1.4.21. *Let* $k, n \in \mathbb{Z}_{>0}$*, and let* $H \in \mathrm{BH}(k,n)$*. We call* $\Psi_n(H) \in \mathrm{Mat}_k(\mathbb{Z}_n)$ *the* logarithmic form of H*.*

We also extend the notion of monomial equivalence (Definition 1.4.2) to complex Hadamard matrices, and do the same for generalized Hadamard matrices with $G = \mathbb{Z}_n$.

Definition 1.4.22. *Let* H *and* H' *be complex Hadamard matrices of the same order. We say that* H *is* monomial equivalent *to* H' *if* H' *can be obtained from* H *by a finite sequence of row permutations, column permutations, and multiplications of a row or of a column by some* $z \in \mathbb{C}$ *such that* $|z| = 1$*.*

Definition 1.4.23. *A complex Hadamard matrix* H *is* normalized *if there exists a row and a column of* H *that contain only* 1*.*

Using the map from $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ to $\mathrm{BH}(n\lambda, n)$, we induce a notion of monomial equivalence on generalized Hadamard matrices over $\mathbb{Z}_n$. The following definition was created for the purposes of this thesis and is not found elsewhere:

Definition 1.4.24. *Let* H *and* H' *be generalized Hadamard matrices of type* $\mathrm{GH}(\mathbb{Z}_n, \lambda)$*. We say* H *is* monomial equivalent *to* H' *if* $\Phi_n(H)$ *is monomial equivalent to* $\Phi_n(H')$*.*

Note 1.4.25. *Alternatively, we could have defined that two generalized Hadamard matrices of type* $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ *are monomial equivalent if they differ by row and column permutations, and adding to any row or column a multiple of the all ones vector.*

We also extend the notion of normalization to generalized Hadamard matrices.

Definition 1.4.26. *A generalized Hadamard matrix* H *of type* $\mathrm{GH}(G, \lambda)$ *is called* normalized *if the first row and the first column of* H *contain only the identity element of* G*.*

Note 1.4.27. *As in the case of classical Hadamard matrices, if the rows and columns are not indexed by positive integers, we say* H *is normalized if there exists a row and column which contain only the identity element of* G*.*

Proposition 1.4.28 (Lampio [21]). *Every generalized Hadamard matrix is monomial equivalent to a normalized generalized Hadamard matrix of the same type.*

The definition of generalized Hadamard matrices (Definition 1.4.13) can be even further loosened by removing the requirement that the matrices be square. Matrices of this type are called *difference matrices*. Of relevance to this thesis, they were studied in a 1979 paper of Dieter Jungnickel [17] who provided a construction for a family of generalized Hadamard matrices of order $2q$ over $\mathbb{F}_q$ regarded as a group under addition, for any odd prime power q.

Theorem 1.4.29 (Jungnickel [17, Theorem 2.2]). *If q is an odd prime power and $n \in \mathbb{F}_q$ is a non-square, then there exists a matrix D of type $\mathrm{GH}(\mathbb{F}_q, 2)$. Specifically, define the matrices $d^i = (d^i_{xy})$, where $1 \leq i \leq 4$, and $x, y \in \mathbb{F}_q$, by*

$$d^1_{xy} = xy - \frac{x^2}{4} \qquad d^2_{xy} = xy + n\frac{x^2}{4}$$
$$d^3_{xy} = xy - y^2 - \frac{x^2}{4} \qquad d^4_{xy} = \left(xy - y^2 - \frac{x^2}{4}\right)/n.$$

Then,

$$D = \begin{bmatrix} d^1 & d^2 \\ d^3 & d^4 \end{bmatrix}.$$

To summarize the various generalizations of Hadamard matrices mentioned in this section, we include an Euler diagram in Appendix A.3, which is a reproduction of a diagram in [21], where the author also describes the details of the intersections among these collections of matrices.

1.5 Formulation of the Kochen–Specker Theorem

In physics, the Kochen–Specker theorem states that, in a Hilbert space of dimension 3 or higher, it is impossible to preassign a value to a collection of $\{0, 1\}$-valued quantum-mechanical observables such that this value does not depend on the *context* in which they are jointly measured.

Mathematically, we denote a collection of quantum-mechanical observables by projection operators on a complex inner product space, or equivalently by points in complex projective space. For readability and simplicity, it is common to denote a collection of these points by a collection of representative complex vectors in the equivalence class of each projective point. For example, in [11], the authors define a notion of *well-signedness* of a vector to make the choice of representative. A preassignment of values of the vectors in $\mathbb{C}^n$ to either 0 or 1 is given by a function from $\mathbb{C}^n$ to $\{0, 1\}$. In this setting, a context is an orthogonal basis for $\mathbb{C}^n$, and for each of these bases we may perform a measurement which reveals the values of each basis element.

With this translation to mathematical language in mind, the following is the statement of the KS theorem, as formulated in [25] and [26]:

Theorem 1.5.1 (Kochen–Specker)**.** *Let $n \geq 3$. There does not exist a function $f : \mathbb{C}^n \to \{0, 1\}$, such that for each orthogonal basis B of $\mathbb{C}^n$, we have $f(v) = 1$ for exactly one vector $v \in B$.*

This theorem was first proved by Kochen and Specker in 1967, but we are nonetheless interested in finding new proofs for several reasons: mathematically, we are interested in finding new approaches which either simplify existing proofs, or establish or demonstrate deeper connections with various areas of mathematics. In physics, new proofs of this theorem are desirable for the purpose of using them as blueprints for quantum mechanical experiments demonstrating quantum contextual phenomenology; for example, in [26], many potential physical applications motivating the search for new proofs are given.

The condition that each orthogonal basis should possess exactly one vector v such that $f(v) = 1$ is ultimately due to the Heisenberg uncertainty principle and other physical considerations, which require that for any orthogonal basis $\{v_1, \ldots, v_n\}$ for $\mathbb{C}^n$, we have that $\sum_{i=1}^{n} f(v_i) = 1$, and therefore that exactly one of the vectors evaluates to 1.

A method of proof for the KS theorem is to provide a (finite) collection of nonzero non-collinear vectors $\mathcal{V} \subset \mathbb{C}^n$ and assume for the sake of contradiction that a function satisfying the properties of Theorem 1.5.1 does exist. We proceed by considering all of the orthogonality relations among the vectors of $\mathcal{V}$; that is, we consider:

Definition 1.5.2. *Let $\mathcal{V} \subset \mathbb{C}^n$ be a collection of non-collinear nonzero vectors. We define $\mathcal{O}(\mathcal{V})$ to be the set of all* maximal *subsets of $\mathcal{V}$ whose members are pairwise orthogonal; that is, $\mathcal{O}(\mathcal{V})$ is the set of subsets of $\mathcal{V}$ whose members are pairwise orthogonal and are not themselves subsets of any larger subsets of $\mathcal{V}$ whose members are pairwise orthogonal. Also, denote by $\mathcal{B}(\mathcal{V}) \subseteq \mathcal{O}(\mathcal{V})$ the subsets in $\mathcal{O}(\mathcal{V})$ which are of size n.*

With the aim of proving Theorem 1.5.1 by contradiction, we note that any collection of n pairwise orthogonal vectors in $\mathbb{C}^n$ forms a basis, and therefore if a function f did exist, then exactly one of them needs to be mapped to 1. Additionally, any smaller collection of $m < n$ pairwise orthogonal vectors still forms a subset of an orthogonal basis for $\mathbb{C}^n$, so in this case as well, we require that *at most* one of them be mapped to 1, as we can still make deductions using implicitly the bases these pairwise orthogonal vectors are guaranteed to be part of.

Definition 1.5.3. *Let $\mathcal{V} \subseteq \mathbb{C}^n$ be a collection of non-collinear nonzero vectors. If there exists a function $f : \mathcal{V} \to \{0, 1\}$ such that $f(v) = 1$ for exactly one element $v \in B$ of each $B \in \mathcal{B}(\mathcal{V})$, and* at most *one element $v \in S$ for each set $S \in \mathcal{O}(\mathcal{V}) \setminus \mathcal{B}(\mathcal{V})$, then we say $\mathcal{V}$* admits a KS colouring*, or is* KS colourable*.*

Note 1.5.4. *We can now rephrase Theorem 1.5.1 as the non-existence of a KS colouring of* $\mathbb{C}^n$ *for* $n \geq 3$*. The existence of a KS colouring on a finite set of vectors does not constitute a disproof of the KS theorem. A set of vectors being KS colourable is only enough to say that they are insufficient to comprise a proof of the KS theorem. This terminology is useful because when showing the impossibility of a KS colouring, we often need to suppose the existence of a KS colouring.*

Example 1.5.5. *Take* $\mathcal{V} = \{(1,0,0),(0,1,0),(0,0,1)\}$ *and* $S = \{\mathcal{V}\} = \mathcal{O}(\mathcal{V})$*. Then,* $(\mathcal{V}, S)$ *is not a KS pair, because if* $f((1,0,0)) = 1$*,* $f((0,1,0)) = 0$*, and* $f((0,0,1)) = 0$*, then* f *is a KS colouring of* $\mathcal{V}$ *because exactly one element of each basis in* $\mathcal{O}(\mathcal{V})$ *is mapped to* 1*.*

Example 1.5.6. *To show that the KS theorem does not apply when* $n = 2$*, below we provide a KS colouring for* $\mathbb{R}^2$*. This colouring* $f : \mathbb{R}^2 \to \{0,1\}$ *is defined to be* 0 *for vectors in quadrants* 1 *and* 3 *(including* $(0,1)$ *and* $(0,-1)$*), and defined to be* 1 *for vectors in quadrants* 2 *and* 4 *(including* $(1,0)$ *and* $(-1,0)$*). We illustrate this colouring with Figure 1.2.*

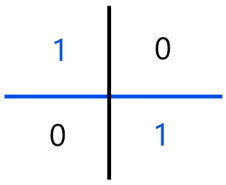

Figure 1.2: KS colouring of $\mathbb{R}^2$

This is a KS colouring because each vector $(a,b) \in \mathbb{R}^2$ *belongs to exactly one orthogonal basis (up to scaling), namely* $\{(a,b),(-b,a)\}$*. These vectors are orthogonal, so each basis contains one vector coloured* 1 *and one coloured* 0*.*

Definition 1.5.7. *Let* $\mathcal{V} \subset \mathbb{C}^n$ *be a collection of non-collinear nonzero vectors. If it can be proved that* $\mathcal{V}$ *does not admit a KS colouring by considering the orthogonality relations in a subset* $S \subseteq \mathcal{O}(\mathcal{V})$*, then we say* $(\mathcal{V}, S)$ *is a* Kochen–Specker (KS) pair*.*

The logical structure of many historical proofs of the KS theorem, and of the new proofs provided in this thesis take the following form: let $n \geq 3$, and assume f is a KS colouring for $\mathbb{C}^n$. Then, for all $\mathcal{V} \subset \mathbb{C}^n$, we must have that $f(v) = 1$ for exactly one vector $v \in B$, for each orthogonal basis $B \in \mathcal{O}(\mathcal{V})$, and $f(v) = 1$ for at most one vector $v \in C$, for each non-basis orthogonal set $C \in \mathcal{O}(\mathcal{V})$. If there exists a subset $S \subseteq \mathcal{O}(\mathcal{V})$ for which such a map is impossible to define, then we have a contradiction, and we call $(\mathcal{V}, S)$ a KS pair.

The impossibility usually comes from the fact that each basis must contain exactly one vector which f maps to 1, and therefore the total number of bases must equal to the sum of the number of bases each vector mapped to 1 appears in. Therefore, if for example, each

vector appears in an odd number of bases, but there are an even number of bases, then there cannot exist a KS colouring on those vectors.

The purpose of using a subset $S \subseteq \mathcal{O}(\mathcal{V})$ is that the full collection $\mathcal{O}(\mathcal{V})$ may not be necessary. Likewise, $\mathcal{O}(\mathcal{V})$ is only a subset of the set of all subsets of $\mathbb{C}^n$ whose members are pairwise orthogonal. In either case, finding a collection of vectors for which a colouring is impossible is enough to prove the KS theorem, as we surely cannot colour all vectors if we cannot colour a subset of them.

This is the method that is employed in the original KS paper [20], as well as in numerous works since, including [29], [31], [9], [10], [25], [38], and [11]. In these papers, the size of the KS pair is usually reported as the size of the set $\mathcal{V}$, but some disagreement exists regarding whether the reported size ought to include any other vectors used implicitly in arguments which involve $\mathcal{O}(\mathcal{V}) \setminus \mathcal{B}(\mathcal{V})$, which in some cases might result in a significantly larger number of vectors than those of $\mathcal{V}$. Another consequence of emphasizing only the size of $\mathcal{V}$ is that much of the overall difficulty of a physical realization of these results arises ultimately not from the number of vectors, but from the number of bases used in the argument. As an example, the original KS paper used an argument which explicitly involved 117 vectors, but required 132 contexts.

It is also possible to formulate KS pairs using a basis-first approach, which in some cases can help to assuage these concerns of counting. In this approach, a contradiction is shown by providing a collection of orthogonal bases whose constituent vectors are not colourable. That is, we start with a collection of orthogonal sets, then take the set of all the vectors which appear in them. To this end, we define:

Definition 1.5.8. *Let $\mathcal{O}$ be a collection of sets of pairwise orthogonal vectors. We denote by $\mathcal{V}(\mathcal{O})$ the collection of all vectors that appear in at least one set in $\mathcal{O}$.*

Often, results following this basis-first approach rely on taking advantage of contradictions which arise from properties of uniformity between the vectors and bases; for example, if the bases are constructed such that every vector appears in an even number of bases, while the total number of bases is odd, then a colouring is surely impossible by double counting the incidences between vectors coloured 1 and the bases. This is often called a parity proof of the theorem. However, it is not strictly necessary that the contradiction arises as a result of some even–odd incompatibility, a result demonstrated in Section 2.3. This approach via bases is the approach used in the paper by Waegell and Aravind [38], as well as in [26] where a proof is constructed using only 7 contexts; it is also the primary mechanism of the results in this thesis.

Chapter 2

Kochen–Specker Pairs via Generalized Hadamard Matrices

This chapter begins the presentation of the new computer-free results proved in this thesis.

We start with Section 2.1, where we introduce some background motivation for this new approach, and define some new notation which will be used throughout the remainder of the thesis.

In Section 2.2, we prove Theorem 2.2.4, with which we demonstrate that a large class of generalized Hadamard matrices may be used to form KS pairs. In Section 2.3, we formulate a sufficient condition for when subspaces of the row spaces of generalized Hadamard matrices can also be used to form KS pairs, and conjecture that this condition might be necessary as well.

Finally, in Section 2.4, we prove some new results on the rank of the generalized Hadamard matrices given by Jungnickel's construction, and use these results to produce an infinite family of KS pairs of known size.

2.1 Introduction to the Approach

The mechanism of this section was initiated as a generalization of how several KS pairs were produced in a 2022 paper of Waegell and Aravind [38]. In that paper, it was shown that some particular objects of classical coding theory (binary and ternary Golay codes) can be used to construct KS pairs. The key observation leading to the research presented in this thesis is that the coding theoretical aspect of [38] is not crucial to how its construction functions. Indeed, it is only the existence of a set of d vectors in $\mathbb{Z}_2^d$ which pairwise differ in half of their entries that underpins the construction. This condition is equivalent to the balancedness condition found in generalized Hadamard matrices over $\mathbb{Z}_2$ of order d, where the images under φ_2 of each of the distinct rows of such a matrix are orthogonal in $\mathbb{C}^d$. This more fundamental aspect was noted earlier in 2019 by Petr Lisoněk [25], where the connection to generalized Hadamard matrices first appeared.

This simpler connection motivated the study of whether any Hadamard matrix might be a candidate for providing a KS pair in a similar manner as the construction in [38].

Prompted by the role of divisibility in that construction, we began by focusing on Hadamard matrices of order not a power of 2, starting with order 12 and 20. Initial tests using these matrices were successful, and generated KS pairs making use of only half the vectors in [38].

2.1.1 Notation and Outline for the Construction

We begin by introducing some notation for the objects used in all of the new constructions of KS pairs in this thesis. Recall that the rows of an order d complex Hadamard matrix are pairwise orthogonal vectors, and therefore form a basis for $\mathbb{C}^d$. Also recall the map Φ_n (Definition 1.4.18), with which we may map any generalized Hadamard matrix over $\mathbb{Z}_n$ to a Butson Hadamard matrix.

The method we follow in this section is outlined as follows: start with an initial (seed) generalized Hadamard matrix, and apply to it a sequence of modifications, such that each of them remains a generalized Hadamard matrix of the same type. This was used, for example in [38] and [25]. In our method, we apply the same modifications as in [38], that is, shifting the seed matrix by a set of suitable vectors. To the resulting collection of generalized Hadamard matrices, we apply the map Φ_n to obtain a collection of Butson type Hadamard matrices, all corresponding to orthogonal bases of $\mathbb{C}^d$. After noting symmetries in the structure of these bases guaranteed by their construction, we prove that they (and their constituent vectors) form a KS pair.

Definition 2.1.1. *Let H be a generalized Hadamard matrix of type* $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ *for $n \in \mathbb{Z}_{>0}$ with rows $\{h_1, \ldots, h_{n\lambda}\}$, and let S be a subset of $\mathbb{Z}_n^{n\lambda}$. For each $s \in S$, the* s-prebasis of H, *denoted by* $\mathrm{PB}(H, s)$, *is given by the sequence*

$$\mathrm{PB}(H, s) = (h_i + s)_{i=1}^{n\lambda}.$$

The set of all s-prebases of H, *denoted* $\mathrm{PB}(H, S)$, *is given by*

$$\mathrm{PB}(H, S) = \{\mathrm{PB}(H, s) \colon\ s \in S\}.$$

Additionally, denoting the additive identity element of $\mathbb{Z}_n^{n\lambda}$ by 0, we call $\mathrm{PB}(H, 0)$ *the* seed prebasis of H.

Note that the vectors in $\mathrm{PB}(H, 0)$ are simply the rows of the matrix H. We call these sets "prebases" because their images under φ_n always form ordered orthogonal bases for $\mathbb{C}^{n\lambda}$, which we prove in Lemma 2.2.1. First, we define some additional notation:

Definition 2.1.2. *Let* $\mathrm{PB}(H,S)$ *be the* S*-prebases of a generalized Hadamard matrix* H *over* $\mathbb{Z}_p$*. For each prebasis* $\mathrm{PB}(H,s)$*, denote the* ordered basis generated by $\mathrm{PB}(H,s)$ *by*

$$\mathrm{B}(H,s) = (\varphi_p(u_i))_{i=1}^{n\lambda}, \text{ where } u_i = \mathrm{PB}(H,s)_i,$$

and denote the set of all ordered bases generated by $\mathrm{PB}(H,S)$ *by*

$$\mathrm{B}(H,S) = \{\mathrm{B}(H,s)\colon\ s \in S\}.$$

Note 2.1.3. *We define the elements of* $\mathrm{PB(H,S)}$ *and* $\mathrm{B(H,S)}$ *to be sequences and ordered bases, respectively, only for the purposes of the proofs of the theorems presented in the remaining sections of this chapter. The specific ordering given by the rows of the matrix is not critical, and if the rows of the given matrix are unordered, one should be assigned arbitrarily for the purpose of the above definitions.*

Note 2.1.4. *Especially in the case of* $H \in \mathrm{GH}(\mathbb{Z}_2,\lambda)$*, the resulting complex orthogonal bases could also be viewed as real orthogonal bases, and indeed, the complex KS pairs which result from these matrices in Sections 2.2 and 2.3 could also be understood as real KS pairs.*

2.2 Constructions using the Entire Row Space

In this section, we prove Theorem 2.2.4, which is a new result of this thesis, showing that each generalized Hadamard matrix H over $\mathbb{Z}_p$ with parameter λ which does not divide p^{r-1}, where r is the rank of H, may be used to construct a KS pair by considering the prebases of the matrix given by its row space. First, we prove some supporting lemmas:

Lemma 2.2.1. *Let* p *be a prime, and let* H *be a generalized Hadamard matrix of type* $\mathrm{GH}(\mathbb{Z}_p,\lambda)$*. Then, for each* $v \in \mathbb{Z}_p^{p\lambda}$*,* $\mathrm{B}(H,v)$ *is an orthogonal basis for* $\mathbb{C}^{p\lambda}$*.*

Proof. Since $\mathrm{PB}(H,0)$ is the set containing the rows of H, and the rows of $\Phi_p(H) \in \mathrm{BH}(p\lambda,p)$ are pairwise orthogonal, we have that $\mathrm{B}(H,0)$ is an orthogonal basis for $\mathbb{C}^{p\lambda}$.

For each $v \in \mathbb{Z}_p^{p\lambda}$, and every distinct pair u,w in $\mathrm{PB}(H,0)$ we have that

$$(u+v)-(w+v) = u-w+v-v = u-w.$$

Then, since u,w are rows of $H \in \mathrm{GH}(\mathbb{Z}_p,\lambda)$, we must have that $(u+v)-(w+v) = u-w$ contains each element of $\mathbb{Z}_p$ exactly λ times, and therefore that $\varphi_p(u+v)\varphi_p(w+v)^* = 0$. Since u and w were chosen arbitrarily, we must have that $\mathrm{B}(H,v)$ is an orthogonal basis for $\mathbb{C}^{p\lambda}$. □

We will also need the following in order to prove Theorem 2.2.4.

Lemma 2.2.2. *Let* $H \in \mathrm{GH}(\mathbb{Z}_p,\lambda)$*, then (1)–(3) are equivalent, and imply (4):*

1. *There exist distinct* $v_1, v_2 \in \mathrm{Row}(H)$ *such that their images under* φ_p *represent the same projective equivalence class.*

2. $\mathbf{1} \in \mathrm{Row}(H)$.

3. *There exist distinct* $v_1, v_2, \ldots, v_p \in \mathrm{Row}(H)$ *such that their images under* φ_p *represent the same projective equivalence class.*

4. H *is not normalized.*

Proof. $(3 \Rightarrow 1)$ This is immediate, as $p \geq 2$.
$(1 \Rightarrow 2 \Rightarrow 3)$ Let $n = p\lambda$. If v_1 and v_2 correspond to the same point in $\mathbb{CP}^{n-1}$, then we must have that for some $\lambda \in \mathbb{C}$

$$\varphi_p((v_{11}, v_{12}, \ldots, v_{1n})) = \lambda \varphi_p((v_{21}, v_{22}, \ldots, v_{2n})),$$

and hence that for all $1 \leq i \leq n$, $\varphi_p(v_{1i}) = \lambda \varphi_p(v_{2i})$, so $\omega_p^{v_{1i}} = \lambda \omega_p^{v_{2i}}$, which implies $\lambda = \omega_p^a$ for some $0 \leq a \leq p$. Therefore, we must have that $v_1 = v_2 + a\mathbf{1}$ for some $a \in \mathbb{Z}_p$. The previous implications all work in the opposite direction as well, so two vectors of $\mathrm{Row}(H)$ correspond to the same point in $\mathbb{CP}^{n-1}$ if and only if the vectors differ by a multiple of the all-ones vector. If v and $v + a\mathbf{1}$ are both in $\mathrm{Row}(H)$, then their difference $v + a\mathbf{1} - v = a\mathbf{1}$ is also in $\mathrm{Row}(H)$, and therefore $a^{-1}a\mathbf{1} = \mathbf{1} \in \mathrm{Row}(H)$ as well. If the all ones vector is in $\mathrm{Row}(H)$, then for all $v \in \mathrm{Row}(H)$, we must have that $v + a\mathbf{1} \in \mathrm{Row}(H)$ for all $a \in \mathbb{Z}_p$, and since this was shown to be equivalent to the existence of two rows corresponding to the same point, we have these all correspond to the same point, and thus there is a p-to-1 relationship between the vectors of $\mathrm{Row}(H)$ and the set of corresponding points in $\mathbb{CP}^{n-1}$.
$(\neg 4 \Rightarrow \neg 2)$ If H is normalized, then its first column and first row contain only 0. Thus, the row span of H can never contain any vector with a nonzero first entry, and in particular cannot contain the all ones vector. Thus, we have shown (1) $\iff$ (2) $\iff$ (3) $\Rightarrow$ (4). $\square$

Example 2.2.3. *A non-normalized Hadamard matrix need not contain the all ones vector in its row space. For example, the below matrix of type* $\mathrm{GH}(\mathbb{Z}_2, 2)$ *does not contain the all ones vector in its row space.*

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

We now prove the main result of this section:

Theorem 2.2.4. *Let* p *be a prime, let* $H \in \mathrm{GH}(\mathbb{Z}_p, \lambda)$ *be a generalized Hadamard matrix over* $\mathbb{Z}_p$ *for a positive integer* λ *such that* λ *does not divide* p^{r-1}*, where* $r = \mathrm{rank}(H)$*. There exists a KS pair in* $\mathbb{C}^{\lambda p}$ *with at most* p^r *vectors and* p^r *bases. If* H *is normalized, then the resulting KS pair comprises exactly* p^r *vectors and* p^r *bases.*

Proof of Theorem 2.2.4. Let $V = \mathrm{Row}(H)$ be the r-dimensional row space of H. The space V contains p^r vectors which correspond to p^r vectors in $\mathbb{C}^{p\lambda}$ under φ_p. Now, consider $\mathrm{PB}(H, V)$, the set of V-prebases of H, which we know is of size p^r. We will show that a KS colouring cannot exist, because $|\mathrm{PB}(H, V)|$ is not divisible by the number of prebases each vector of the row space appears in.

For each $v \in V$ and for each vector u appearing in $\mathrm{PB}(H, 0)$, we may write v as $v = u + (v - u)$. We know that both v and u are elements of V, and therefore v appears in $\mathrm{PB}(H, v-u)$, for each vector appearing in $\mathrm{PB}(H, 0)$. Each of the prebases must be distinct, as each vector in $\mathrm{PB}(H, 0)$ must be distinct. We know that $|\mathrm{PB}(H, 0)| = p\lambda$, so we have shown that each v appears in at least λp distinct prebases. It cannot occur in any other prebasis. If it did, we would necessarily have that $v = u + w$ and $v = u + t$ for some u appearing in $\mathrm{PB}(H, 0)$ and some pair of distinct elements $w, t \in V$, but this immediately implies $w = t$.

We have found that every vector of V appears in exactly λp prebases, and since we assumed λ does not divide p^{r-1}, it cannot be that λp divides p^r. This is enough to say that it is impossible to produce a KS colouring of the vectors, as any choice of vectors to be coloured 1 necessarily involves shared prebases.

Finally, by Lemma 2.2.1, we know that under the map φ_p, all of these prebases correspond to orthogonal bases of $\mathbb{C}^{p\lambda}$, $(\mathcal{V}(\mathrm{B}(H, V)), \mathrm{B}(H, V))$ is a KS pair with at most p^r vectors. If H is normalized, then by Lemma 2.2.2, no two vectors of its row space correspond to the same projective point, so the KS pair contains exactly p^r vectors. □

Example 2.2.5. *If H is any classical Hadamard matrix of order n not a power of 2, then $\Psi_2(H) \in \mathrm{GH}(\mathbb{Z}_2, n/2)$, and since $n/2$ is not a power of 2, we must have that H gives a KS pair as in Theorem 2.2.4.*

Then, because we have existence results and explicit constructions for generalized Hadamard matrices of various types due to Theorem 1.4.16, we have the following corollary:

Corollary 2.2.6. *If $m \leq k$ are positive integers and p is an odd prime, then there exists a KS pair with vectors in $\mathbb{C}^{2^m p^k}$ with at most $p^{2^m p^k}$ vectors and $p^{2^m p^k}$ bases.*

Proof. By Theorem 1.4.16, there exists a GH matrix H over $\mathbb{Z}_p$ with $\lambda = 2^m p^{k-1}$. Since $p \neq 2$, it follows that λ does not divide p. Therefore, by our theorem, if r is the rank of H, we know that there is a KS pair in $\mathbb{C}^{2^m p^k}$ with at most p^r vectors and bases, and since $r \leq 2^m p^k$, we may write this bound as at most $p^{2^m p^k}$ vectors and bases. □

While an explicit construction exists for the matrices of Theorem 1.4.16, the size of the KS pair each matrix provides depends on the rank of the matrix. It is for this reason that we may only give an upper bound on the size of the KS pairs.

In Section 2.3, we explore how proper subspaces of the row space of H can be used to construct KS pairs similarly to Theorem 2.2.4.

2.3 Constructions using Subspaces of the Row Space

In this section, we explore modifying Theorem 2.2.4 by considering proper subspaces of the row space of a Hadamard matrix rather than necessarily using the whole row space; that is, for a generalized Hadamard matrix H of type $\mathrm{GH}(\mathbb{Z}_p, \lambda)$ for a prime λ, and a subspace K of $\mathrm{Row}(H)$, we consider whether we can construct a KS pair using $\mathrm{B}(H, K)$. We show in this section that it is possible to construct additional KS pairs in this way. In particular, in Theorem 2.3.9, we give a sufficient condition for when we may form a KS pair with $\mathrm{B}(H, K)$. We also conjecture that this is a necessary condition.

We begin by defining a particular feature of the K-prebases of a generalized Hadamard matrix H, for a subspace K of the row space of H. In the computational study of these objects, it has become clear that the following seems to predict whether each subspace leads to a KS pair:

Definition 2.3.1. *Let H be a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ and K be a subspace of $\mathrm{Row}(H)$ viewed as a group under addition. Denote the elements of $\mathrm{PB}(H, 0)$ by v_i, for $1 \leq i \leq n\lambda$. We define the sequence $S = (S_i)_{i=1}^{n\lambda}$ of* slices *of the K-prebases of H, where S_i is given by the image of v_i under the quotient map $Q\colon \mathrm{Row}(H) \to \mathrm{Row}(H)/K$. Note that $Q(v_i)$ is the coset of K containing v_i, that is, $Q(v_i) = v_i + K$.*

Example 2.3.2. *Let $H \in \mathrm{GH}(\mathbb{Z}_3, 2)$, and label the rows of H by $\{v_1, v_2, \ldots, v_6\}$. If $K = \{k_1, k_2, \ldots, k_9\}$ is a 2-dimensional subspace of $\mathrm{Row}(H)$, then the K-prebases of H are given by*

$$\begin{array}{rcccccc}
\mathrm{PB}(H, k_1) = & \{v_1 + k_1, & \boxed{v_2 + k_1,} & v_3 + k_1, & v_4 + k_1, & v_5 + k_1, & v_6 + k_1\}, \\
\mathrm{PB}(H, k_2) = & \{v_1 + k_2, & \boxed{v_2 + k_2,} & v_3 + k_2, & v_4 + k_2, & v_5 + k_2, & v_6 + k_2\}, \\
\vdots & \vdots & \boxed{\vdots} & \vdots & \vdots & \vdots & \vdots \\
\mathrm{PB}(H, k_9) = & \{v_1 + k_9, & \boxed{v_2 + k_9,} & v_3 + k_9, & v_4 + k_9, & v_5 + k_9, & v_6 + k_9\},
\end{array}$$

where the entries in the box correspond to the slice S_2.

These slices may be equal for several elements of the seed prebasis. Therefore, we encode the degree of overlap between them by the following definition:

Definition 2.3.3. *Let $S = (S_i)_{i=1}^{n\lambda}$ be a sequence of slices of $\mathrm{PB}(H, K)$, where H is a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_n, \lambda)$, and K a subspace of $\mathrm{Row}(H)$, and let $Q : \mathrm{Row}(H) \to \mathrm{Row}(H)/K$ denote the quotient map. Then, we define the* weight *of slice S_i, denoted w_i, by*

$$w_i = |Q^{-1}(S_i) \cap \mathrm{PB}(H, 0)|,$$

which is the number of rows of H in the coset $v_i + K$.

Example 2.3.6 demonstrates the above definition for a specific Hadamard matrix and subspace.

Note 2.3.4. *The sum of the weights of the distinct slices of* $\mathrm{PB}(H, K)$ *is equal to the number of rows of* H*.*

Lemma 2.3.5. *Let* S *be a slice of weight* w *of* $\mathrm{PB}(H, K)$ *for a generalized Hadamard matrix* H *of type* $\mathrm{GH}(\mathbb{Z}_n, \lambda)$ *and subspace* K *of the row space of* H*. For each* $v \in S$*, we have that* v *appears in exactly* w *distinct* K*-prebases of* H*.*

Proof. Since S is of weight w, by definition there must exist exactly w distinct slices S_j such that $v \in S_j = v_j + K$. Label these slices $S_{j_1}, \ldots, S_{j_w}$, for some set of distinct indices $\{j_1, \ldots, j_w\} \subseteq \{1, \ldots, n\lambda\}$. Then, v appears in $\mathrm{PB}(H, v - v_{j_i})$ for all $i \in \{j_1, \ldots, j_w\}$, as

$$v_{j_i} + (v - v_{j_i}) = v.$$

These must be distinct as well, because all vectors in a prebasis are distinct. □

Example 2.3.6. *If* H *is the* $\mathrm{GH}(\mathbb{Z}_3, 2)$ *Hadamard matrix given by*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 0 & 2 & 1 & 2 \\ 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 2 & 1 & 2 & 0 & 1 \\ 0 & 2 & 2 & 1 & 1 & 0 \end{bmatrix},$$

where we denote the rows of H *by* $\{v_1, v_2, \ldots, v_6\}$*, and* K *is the subspace of the row space of* H *given by* $\mathrm{span}\{v_2, v_3 + v_4, v_3 + v_5\}$*, then it can be computed that there exist three distinct slices each of weight* 2*. In particular,* $S_1 = S_2$*,* $S_3 = S_6$*, and* $S_4 = S_5$*. Therefore by Lemma 2.3.5, we know that, for example,* v_2 *appears in exactly two* K*-prebases, which are given by* $\mathrm{PB}(H, v_2 - v_2) = \mathrm{PB}(H, 0)$ *and* $\mathrm{PB}(H, v_2 - v_1) = \mathrm{PB}(H, v_2)$*.*

We present the following definition and then Theorem 2.3.9, which gives a sufficient condition for when a subspace leads to a KS pair.

Definition 2.3.7. *Let* H *be a generalized Hadamard matrix of type* $\mathrm{GH}(\mathbb{Z}_p, \lambda)$*,* $n = p\lambda$*,* K *be a subspace of the row space of* H*, and* w_i *be the weight of slice* S_i *of* $\mathrm{PB}(H, K)$ *for each* $1 \leq i \leq n$*. We define the following linear form:*

$$\ell(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i x_i.$$

Note 2.3.8. *For computational purposes, one should modify the linear form to only incorporate distinct weights.*

Theorem 2.3.9. *Let* p *be a prime,* H *be a generalized Hadamard matrix of type* $\mathrm{GH}(\mathbb{Z}_p, \lambda)$*,* $n = p\lambda$*,* K *be a subspace of the row space of* H*, and* w_i *be the weight of slice* S_i *of* $\mathrm{PB}(H, K)$

for each $1 \leq i \leq n$. If the linear equation $\ell(x_1, \ldots, x_n) = |K|$ has no non-negative integer solutions, then $(\mathcal{V}(\mathcal{B}), \mathcal{B})$ is a KS pair, where $\mathcal{B} = \mathrm{B}(H, K)$.

Proof. We show the result via the contrapositive; that is, if a KS colouring of $(\mathcal{V}(\mathrm{B}(H, K)), \mathrm{B}(H, K))$ exists, then the linear equation $\ell(x_1, \ldots, x_n) = |K|$ subject to the above constraints has a solution.

To this end, suppose there exists a map $f : \mathbb{C}^n \to \{0, 1\}$ such that for each $k \in K$, there exists exactly one element v_k of $\mathrm{PB}(H, k)$ such that $f(\varphi_p(v_k)) = 1$. Denote a slice of $\mathrm{PB}(H, K)$ to which v_k belongs by S_k, and denote the weight of S_k by d_k. By Lemma 2.3.5, we know that v_k appears in d_k prebases, so it is the unique vector coloured 1 in exactly d_k bases.

Denote by $\{v_{k_1}, \ldots, v_{k_r}\}$ for some $r \in \mathbb{Z}_{>0}$ the set of all vectors in $\mathcal{V}(\mathrm{B}(H, K))$ such that $f(\varphi_p(v_k)) = 1$. Then, each v_{k_j} appears in exactly d_{k_j} bases, and therefore $\sum_{j=1}^{r} d_{k_j} = |\mathrm{PB}(H, K)| = |K|$. Some of these may appear in the same slice, so denote by a_i the number of distinct $v_{k_j} \in S_i$, for each $1 \leq n$, where $0 \leq a_i \leq |K|$. Then, we may write the above sum as $\sum_{i=1}^{n} w_i a_i$. Therefore, we may satisfy the linear form as follows:

$$\ell(a_1, \ldots, a_n) = \sum_{j=1}^{r} d_{k_j} = |K|.$$

□

Therefore, we have shown that if the slices given by a subspace of the row space of a generalized Hadamard matrix are such that their weights do not give a satisfiable linear equation, then we can always construct a KS pair.

Note 2.3.10. *We may now view Theorem 2.2.4 as a corollary of Theorem 2.3.9, which generalizes the mechanism of proof.*

Example 2.3.11. *Considering Example 2.3.6, since each of the three slices is of weight 2, we are looking for solutions to $\ell(x, y, z) = 2(x + y + z) = 27$, which is impossible to satisfy since 2 does not divide 27. Therefore, this subspace gives a KS pair in $\mathbb{C}^6$ with $|\mathcal{B}| = 27$ and $|\mathcal{V}| = 81$.*

Corollary 2.3.12. *Let p be prime, H be a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_p, \lambda)$, $n = p\lambda$, and K be a subspace of the row space of H. If the slices of $\mathrm{PB}(H, K)$ are of uniform weight d where d does not divide $|K|$, then $(\mathcal{V}(\mathrm{B}(H, K)), \mathrm{B}(H, K))$ is a KS pair.*

Proof. The linear equation in this case may be written as $\ell(x_1, \ldots, x_n) = d\,(\sum x_i) = |K|$, which has no solutions as d does not divide $|K|$. □

We have shown that the slices of the K-prebases of a generalized Hadamard matrix for a subspace K of the row space of H satisfying the linear equation $\ell(x) = |K|$ is a sufficient condition for constructing a KS pair.

It is natural to ask whether this condition is also necessary. In all examples studied throughout the course of the research (Appendix A.1), whether $\ell(x) = |K|$ has a solution exactly predicts whether $(\mathcal{V}(\mathrm{B}(H,K)), \mathrm{B}(H,K))$ is a KS pair. With this computational evidence in mind, we present the following conjecture:

Conjecture 2.3.13. *Let H be a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_p, \lambda)$, $n = p\lambda$, K be a subspace of the row space of H, and w_i be the weight of slice S_i of $\mathrm{PB}(H,K)$ for each $1 \leq i \leq n$. If the linear equation $\ell(x_1, \ldots, x_n) = |K|$ has at least one solution subject to $0 \leq x_i \leq |K|$, then $(\mathcal{V}(\mathcal{B}), \mathcal{B})$ is not a KS pair, where $\mathcal{B} = \mathrm{B}(H,K)$.*

Some results helping to support this conjecture are given in Appendix A.1. This table additionally compiles whether each subspace S leads to a KS pair after considering some number of additional orthogonality relations in addition to those given by $\mathrm{B}(H,S)$.

2.4 Building an Infinite Family of KS Pairs using the Jungnickel Construction

In this section, we present several new results using a construction for generalized Hadamard matrices by Dieter Jungnickel found in [17], which were previously defined in this thesis in the statement of Theorem 1.4.29.

In Section 2.2, we proved that for generalized Hadamard matrices of type $\mathrm{GH}(\mathbb{Z}_p, \lambda)$, if λ does not divide p^{r-1}, where r is the rank of H, then we are able to able to construct a KS pair. As seen in the statement of Theorem 2.2.4 and Theorem 2.3.9, the number of vectors and bases used in each pair depends on the rank of the matrix used in the construction. Therefore, to fully understand the sizes of KS pairs we are generating, we need to understand the ranks of the generalized Hadamard matrices we use in their construction.

In Theorem 2.4.3, we prove that if the generalized Hadamard matrices resulting from the construction given in Theorem 1.4.29 are normalized, then they are all of rank 4. Hence, we may use Theorem 2.2.4 to infer that they generate KS pairs with p^4 vectors and p^4 bases of $\mathbb{C}^{2p}$. Additionally, we show that the number of bases may always be reduced to p^3, and that the structure of the KS pairs in these cases has a natural correspondence to a 1-factorization of K_{2p}.

To begin, we first introduce some additional notation to help unambiguously refer to the rows and columns of the matrices given by the construction.

Definition 2.4.1. *Let D be a generalized Hadamard matrix over $\mathbb{F}_q$ as an additive group as constructed in Theorem 1.4.29. For $1 \leq i \leq 4$, denote the rows and columns of each block d^i of D by $r^i = (r^i_x)$ and $c^i = (c^i_y)$, respectively, for each $x, y \in \mathbb{F}_q$, where*

$$r^i_x = \{d^i_{xy} \colon\ y \in \mathbb{F}_q\},\ \text{and}\ c^i_y = \{d^i_{xy} \colon\ x \in \mathbb{F}_q\}.$$

Additionally, we denote each combined column of blocks 1 *and* 3*, and each combined column of blocks* 2 *and* 4 *by*

$$c^{13} = \begin{bmatrix} c^1 \\ c^3 \end{bmatrix}, \text{ and } c^{24} = \begin{bmatrix} c^2 \\ c^4 \end{bmatrix}$$

and do the same for each combined row of blocks 1 *and* 2*, and each combined row of blocks* 3 *and* 4 *by*

$$r^{12} = \begin{bmatrix} r^1 & r^2 \end{bmatrix}, \text{ and } r^{34} = \begin{bmatrix} r^3 & r^4 \end{bmatrix}.$$

Now, we normalize the matrix D, by subtracting r_0^{12} from each row, and c_0^{13} from each column.

Construction 2.4.2. *After subtracting off* r_0^{12} *and* c_0^{13} *from each row and column, respectively, we are left with the following definition for* d^i*, where* $1 \leq i \leq 4$*. For some non-square* $n \in \mathbb{F}_q$*,*

$$d^1_{xy} = xy \qquad d^2_{xy} = xy + (n-1)\frac{x^2}{4}$$
$$d^3_{xy} = xy - y^2 \qquad d^4_{xy} = \left(xy - y^2 + (n-1)\frac{x^2}{4}\right)/n.$$

Then, we build D *in the same way as the original theorem:*

$$D = \begin{bmatrix} d^1 & d^2 \\ d^3 & d^4 \end{bmatrix}.$$

From this point forward in this section, the matrix D and the notation given in Definition 2.4.1 are in reference to the normalized Hadamard matrix given by Construction 2.4.2. We now proceed to prove the main theorems of this section.

Theorem 2.4.3. *If* p *is an odd prime and* D *is a* $\mathrm{GH}(\mathbb{Z}_p, 2)$ *Hadamard matrix given by Construction 2.4.2 using any non-square parameter* $n \in \mathbb{Z}_p$*, then* $\mathrm{rank}(D) = 4$*.*

Proof. We recall that the assumptions that p be odd and n be non-square are required by Construction 2.4.2. We begin by showing that rows $r^{12}_{-2}, r^{12}_{-1}, r^{34}_0$, and r^{34}_1 are linearly independent. Suppose there exist $a_1, a_2, a_3, a_4 \in \mathbb{Z}_p$ such that

$$a_1 r^{12}_{-2} + a_2 r^{12}_{-1} + a_3 r^{34}_0 + a_4 r^{34}_1 = 0.$$

This also gives an equation for the corresponding entries of the four rows:

$$a_1(-2y) + a_2(-y) + a_3(-y^2) + a_4(y - y^2) = 0 \tag{2.1}$$
$$a_1(-2y + (n-1)) + a_2\left(-y + \frac{n-1}{4}\right) + a_3\left(-\frac{y^2}{n}\right) + a_4\left(\frac{y-y^2}{n} + \frac{n-1}{4n}\right) = 0, \tag{2.2}$$

where (2.1) and (2.2) correspond to the equations given by columns c_y^{13} and c_y^{24}, respectively. The equations must be satisfied for all $y \in \mathbb{Z}_p$.

We proceed by evaluating equation (2.1) at $y = 1$ and $y = 2$, and equation (2.2) at $y = 0$ and $y = 1$. This leads to the following system:

$$\begin{aligned} 2a_1 + a_2 + 4a_3 &= 0 \\ 4a_1 + 2a_2 + 4a_3 + 2a_4 &= 0 \\ 4n(n-1)a_1 + n(n-1)a_2 + (n-1)a_4 &= 0 \\ (-8n + 4n(n-1))a_1 + (-4n + n(n-1))a_2 - 4a_3 + (n-1)a_4 &= 0. \end{aligned}$$

We encode this system into a matrix A, and using Maple [28] (Appendix B.1.2), it is shown that

$$\det(A) = -64n^3 + 80n^2 - 16n.$$

Therefore, showing linear independence of these rows reduces to showing that $\det(A) = 0$ has no solutions for any valid choice of n, which we recall is a non-square. Since $p \neq 2$ and $n \neq 0$, we reduce this to looking for solutions of:

$$4n^2 - 5n + 1 = (n-1)(4n-1) = 0$$

over $\mathbb{Z}_p$. This has solutions $n = 1$, $n = 4^{-1}$; however, n is assumed to be non-square and $p \neq 2$, so n may be neither of these values, as $1 = 1^2$ and $4^{-1} = (2^{-1})^2$. Therefore, we have shown that A is always invertible, and that $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$, and r_1^{34} are linearly independent. Hence, D is of rank at least 4.

We proceed to show that D is of rank exactly 4, by proving we may write each of its rows as a linear combination of $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$, and r_1^{34}. First, we find necessary conditions for writing r_x^{12} as a linear combination of r_{-2}^{12} and r_{-1}^{12}, and for writing r_x^{34} as a linear combination of all four. Then we show that these conditions are sufficient. If r_x^{12} is given by a linear combination of r_{-2}^{12} and r_{-1}^{12}, then for some $a_1, a_2 \in \mathbb{Z}_p$, any entry d_{xy}^1 and d_{xy}^2 satisfies:

$$a_1(-2y) + a_2(-y) = xy,$$

$$a_1(-8ny + 4n(n-1)) + a_2(-4ny + n(n-1)) = 4nxy + n(n-1)x^2,$$

respectively. We know that these hold for $y = 1$, and 0, giving the equations

$$-2a_1 - a_2 = x, \text{ and}$$

$$4n(n-1)a_1 + n(n-1)a_2 = n(n-1)x^2 \iff 4a_1 + a_2 = x^2.$$

Solving this system gives $a_1 = \frac{x^2+x}{2},\ a_2 = -(x^2+2x)$ (Appendix B.1.3). Then, we verify that these conditions are sufficient for any column:

$$\left(\frac{x^2+x}{2}\right)(-2y) - (x^2+2x)(-y) = -x^2y - xy + x^2y + 2xy$$
$$= xy$$

and

$$\left(\frac{x^2+x}{2}\right)(-8ny + 4n(n-1)) - (x^2+2x)(-4ny + n(n-1))$$
$$= (x^2+x)(-4ny + 2n(n-1)) + 4nyx^2 + 8nyx - x^2n(n-1) + 2nx(n-1)$$
$$= 4nxy + n(n-1)x^2.$$

Indeed, they are. We have shown that r_x^{12} can always be written as a linear combination of $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$, and r_1^{34}. It remains to show that we can do the same for any r_x^{34}. The method for these remaining rows is similar to that of r_x^{12}, but lengthier, so the details of the calculations have been moved to Appendix B.1.4, where we present the Maple code which was used to solve for the following coefficients:

$$a_1 = \frac{x^2-x}{2n},\quad a_2 = \frac{x-x^2}{n},\quad a_3 = 1-x,\quad a_4 = x.$$

The above coefficients give a general expression for r_x^{34} in terms of $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$, and r_1^{34}. Hence, we have found that each row of the matrix can be expressed as a linear combination of four fixed linearly independent rows, and therefore, we have shown that D is rank 4. □

Having proved the above theorem, we now know the exact rank of an infinite family of generalized Hadamard matrices. We combine this fact with Theorem 2.3.9 to show that these give us KS pairs of known size.

Corollary 2.4.4. *If p is an odd prime and D is a $\mathrm{GH}(\mathbb{Z}_p, 2)$ Hadamard matrix given by Construction 2.4.2 using any non-square parameter $n \in \mathbb{Z}_p$, then writing $V = \mathrm{Row}(D)$, $(\mathcal{V}(\mathrm{B}(D,V)), \mathrm{B}(D,V))$ is a KS pair in $\mathbb{C}^{2p}$ with p^4 vectors and p^4 bases.*

Proof. Let V be the row space of D. As proved in Theorem 2.4.3, rows $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$ and r_1^{34} always span V, which hence consists of p^4 vectors. Then, since each row of D is contained in V, there is only one slice of $\mathrm{PB}(D,V)$which is of weight $2p$. Then, since $2px = p^4$ has no solution in non-negative integers, by Theorem 2.3.9, $(\mathcal{V}(\mathrm{B}(D,V)), \mathrm{B}(D,V))$ is a KS pair in $\mathbb{C}^{2p}$ with exactly p^4 vectors and p^4 bases. □

Therefore, we have shown that we can generate an infinite family of KS pairs in complex dimension $2p$ for any odd prime p, with p^4 vectors and p^4 bases. This being said, we also

prove the following theorem, which says that we are always able to reduce this to a KS pair involving only p^3 bases.

Theorem 2.4.5. *Let p be an odd prime and D be a $\mathrm{GH}(\mathbb{Z}_p, 2)$ Hadamard matrix given by Construction 2.4.2 using any non-square parameter $n \in \mathbb{Z}_p$. There exist at least $2p-1$ distinct 3-dimensional subspaces S of $\mathrm{Row}(D)$ such that $(\mathcal{V}(\mathrm{B}(D,S)), \mathrm{B}(D,S))$ is a KS pair in $\mathbb{C}^{2p}$ with p^4 vectors and p^3 bases.*

Proof. It is proved in Theorem 2.4.3 that the row space of D is always 4-dimensional and is given by the span of $r_{-2}^{12}, r_{-1}^{12}, r_0^{34}$, and r_1^{34}. For each nonzero column of D, i.e. every column except c_0^{13}, we define a subspace of the row space by considering the elements of the row space with 0 in the position corresponding to the column. That is, for each nonzero column c_y^{13} and c_y^{24}, define the spaces S_y^{13} and S_y^{24} by

$$S_y^{13} = \{a_1 r_{-2}^{12} + a_2 r_{-1}^{12} + a_3 r_0^{34} + a_4 r_1^{34} \colon\ a_1 d_{-2,y}^1 + a_2 d_{-1,y}^1 + a_3 d_{0,y}^3 + a_4 d_{1,y}^3 = 0\},$$

$$S_y^{24} = \{a_1 r_{-2}^{12} + a_2 r_{-1}^{12} + a_3 r_0^{34} + a_4 r_1^{34} \colon\ a_1 d_{-2,y}^2 + a_2 d_{-1,y}^2 + a_3 d_{0,y}^4 + a_4 d_{1,y}^4 = 0\}.$$

If we write the row space in terms of the $\{r_{-2}^{12}, r_{-1}^{12}, r_0^{34}, r_1^{34}\}$ basis, then

$$S_y^{12} = \{(a_1,a_2,a_3,a_4) \in \mathbb{Z}_p^4 \colon\ (a_1,a_2,a_3,a_4) \cdot (d_{-2,y}^1, d_{-1,y}^1, d_{0,y}^3, d_{1,y}^3) = 0\},\ \text{and}$$
$$S_y^{34} = \{(a_1,a_2,a_3,a_4) \in \mathbb{Z}_p^4 \colon\ (a_1,a_2,a_3,a_4) \cdot (d_{-2,y}^2, d_{-1,y}^2, d_{0,y}^4, d_{1,y}^4) = 0\};$$

that is, these spaces are given by the orthogonal complements of the subspaces given by the span of each of the following vectors:

$$\{(d_{-2,y}^1, d_{-1,y}^1, d_{0,y}^3, d_{1,y}^3) \colon\ y \in \mathbb{Z}_p - \{0\}\} \cup \{(d_{-2,y}^2, d_{-1,y}^2, d_{0,y}^4, d_{1,y}^4) \colon\ y \in \mathbb{Z}_p\}. \tag{2.3}$$

Each of these vectors is nonzero, because if for some $y \in \mathbb{Z}_p$, $(d_{-2,y}^1, d_{-1,y}^1, d_{0,y}^3, d_{1,y}^3) = (0,0,0,0)$, then there exists a column of D wherein 0 occurs at least 4 times, which is impossible because D is a normalized generalized Hadamard matrix with $\lambda = 2$.

Therefore each of these spaces, S_y^{12} and S_y^{34}, is a 3-dimensional subspace of the 4-dimensional row space of D, as they are given by the orthogonal complement of a nonzero vector. We now show that for each of these subspaces S every slice of $\mathrm{PB}(D,S)$ is of weight 2.

Since $D \in \mathrm{GH}(\mathbb{Z}_p, 2)$, and D is normalized, each value $k \in \mathbb{Z}_p$ appears exactly twice in each column except the zero column.

We fix a column c_y^{13} or c_y^{24}, and fix some $k \in \mathbb{Z}_p$. If the column is c_y^{13} for $y \neq 0$, then there are exactly two (possibly indistinct) values x_1,$x_2 \in \mathbb{Z}_p$ such that $d_{x_i,y}^1 = k$ or $d_{x_i,y}^3 = k$. Similarly if the column is c_y^{24}, then there are exactly two values x_1,$x_2 \in \mathbb{Z}_p$ such that $d_{x_i,y}^2 = k$ or $d_{x_i,y}^4 = k$.

From this point on, we omit the superscripts on the columns and subspaces given by columns. We write S_y instead of S_y^{13}, and c_y instead of c_y^{13}. The arguments involving c_y^{24} and S_y^{24} are the same up to replacing symbols.

Since $r_{x_1,y} = r_{x_2,y} = k$, we know that $r_{x_1,y} - r_{x_2,y} = k - k = 0$, and therefore that $r_{x_1} - r_{x_2} \in S_y$. Hence, $r_{x_1} \in \mathrm{PB}(D, r_{x_1} - r_{x_2})$, so r_{x_1} and r_{x_2} belong to the same slice of $\mathrm{PB}(D, S_y)$. Because there are exactly two such rows for any value of k, we know that every slice of $\mathrm{PB}(D, S_y)$ is of weight 2.

Then, using Corollary 2.3.12, because each slice is of uniform weight 2, and 2 does not divide $|S_y| = p^3$, we know that $(\mathcal{V}(\mathrm{B}(D, S_y)), \mathrm{B}(D, S_y)$ is a KS pair in $\mathbb{C}^{2p}$ with p^4 vectors and p^3 bases.

□

In the computational exploration around the KS pairs produced by Theorem 2.4.5, it became clear that their slices seemed to always correspond to a 1-factorization of K_{2p} in a natural way. By the following corollary, we prove that this always happens.

Corollary 2.4.6. *Let p be an odd prime and D be a $\mathrm{GH}(\mathbb{Z}_p, 2)$ Hadamard matrix given by Construction 2.4.2 using any non-square parameter $n \in \mathbb{Z}_p$. Then, each of the $2p-1$ KS pairs produced by Theorem 2.4.5 corresponds to a 1-factor of K_{2p}, and moreover the collection of these gives a 1-factorization of K_{2p}.*

Proof. Begin by labelling the vertices of K_{2p} by the $2p$ rows of D. It is clear that each of the subspaces S as in Theorem 2.4.5 gives a 1-factor on these vertices, as we may pair those vertices which correspond to a pair of rows which share the same weight 2 slice of $\mathrm{PB}(D, S)$. We will show that these $2p-1$ 1-factors of K_{2p} give us a 1-factorization of K_{2p} as well.

Suppose for a contradiction that the pair $\{r_1, r_2\}$ of rows of D appears in more than one of the 1-factors. Then, for two distinct subspaces S_1 and S_2 we have that:

$$r_1 + S_1 = r_2 + S_1 \text{ and } r_1 + S_2 = r_2 + S_2.$$

As S_1 and S_2 are defined to be the subspaces of the row space with 0 in entries corresponding to two distinct nonzero columns y_1, y_2 of D, it must be that $(r_1)_{y_1} = k_1 = (r_2)_{y_1}$ and $(r_1)_{y_2} = k_2 = (r_2)_{y_2}$ for possibly indistinct values $k_1, k_2 \in \mathbb{Z}_p$. Recall that D is a generalized Hadamard matrix of type $\mathrm{GH}(\mathbb{Z}_p, 2)$, and therefore each entry of D appears exactly twice in the multiset of differences between the entries of any two distinct rows. Therefore, as the difference 0 must appear exactly twice, any two rows will possess the same value in exactly two of their entries. However, we have shown that the multiset of differences between the entries of r_1 and r_2 contains 0 at least three times, as in addition to the normalized column of D wherein each row contains a zero, we have shown $(r_1)_{y_1} - (r_2)_{y_1} = k_1 - k_1 = 0$ and $(r_1)_{y_2} - (r_2)_{y_2} = k_2 - k_2 = 0$. Thus, we have shown that each pair of vertices appears in at most one 1-factor of K_{2p}. We know each pair of vertices appears in at least one 1-factor

as well, because each row has the same value as any other row in exactly one position apart from the normalized column. Therefore, we have shown that the 1-factors given by the subspaces of Theorem 2.4.5 give a 1-factorization of K_{2p}. □

Example 2.4.7. *Figure 2.1 is an example of a* 1*-factorization of* K_6 *corresponding to the subspaces of the row space of the normalized* $\mathrm{GH}(\mathbb{Z}_3, 2)$ *given by Construction 2.4.2.*

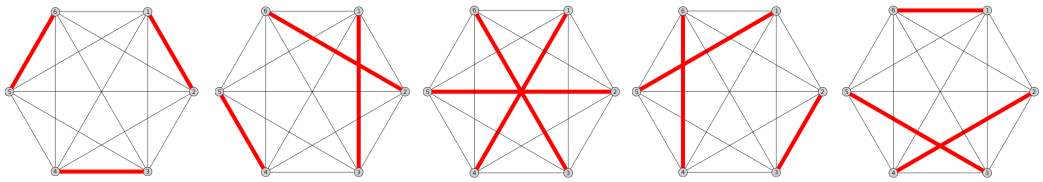

Figure 2.1: 1-factorization corresponding to subspaces

To summarize, for any odd prime, the Jungnickel difference matrix construction found in [17] yields a generalized Hadamard matrix which can always be used to construct a KS pair for $\mathbb{C}^{2p}$ using p^4 vectors and p^4 bases. This can always be reduced to p^3 bases by considering particular subspaces of the row space, and particular cosets of these subspaces correspond to a 1-factorization of K_{2p} in a natural way.

Chapter 3

Computational Results

In this chapter, we detail the large computational component of the research. In Section 3.1, we formulate an integer linear program (ILP) which tests whether any given set of vectors and orthogonal sets form a KS pair. In Section 3.2, we formulate a graph theoretical algorithm which tests whether any given set of vectors and sets of pairwise orthogonal vectors form a KS pair. Finally, in Section 3.3, we use our ILP to computationally explore looking for small integer-valued KS pairs in $\mathbb{R}^3$. In Section 3.3.1, we explore a method for creating such KS pairs which was detailed by Cortez and Reyes [11]. We provide a slight simplification of their KS pair, and apply our ILP to attempt to construct similar pairs. In Section 3.3.2, we implement a computational approach which searches for integer-valued KS pairs in $\mathbb{R}^3$ by considering their automorphism groups.

3.1 Using Integer Linear Programming to find KS Pairs

In Section 3.1.1, we formulate an integer linear program (ILP) which tests whether any set of vectors $\mathcal{V}$ and subset of $\mathcal{O}(\mathcal{V})$ form a KS pair. We implement this program in MAGMA [7] and use it to study the sizes of KS pairs generated by the methods of Chapter 2. In particular, we focus on KS pairs given by Paley Hadamard matrices [30], which are provided in Appendix B.6.

3.1.1 ILP Formulation

In many proofs of the KS theorem, the contradiction arises from some numerical incompatibility; for example, each vector might be observed to occur an even number of times in an odd number of bases. These are the so-called parity proofs of the KS theorem. We can rephrase this as a situation where it is impossible satisfy

$$ax = b;\ a, b \in \mathbb{Z}_{>0};\ x \in \mathbb{Z}_{\geq 0}.$$

Example 3.1.1. *In the paper of Waegell and Aravind [38], the set of vectors and bases is proved to be a KS pair, fundamentally by the unsatisfiability of* $24x = 2048$ *with any non-negative integer value of* x*.*

We proceed by posing the question, "What other types of contradiction are possible?" Waegell and Aravind [38] give an example showing that KS pairs can come from the (more general) unsatisfiability of certain Diophantine equations. A variation of this idea appears in Section 2.3, where in Theorem 2.3.9, we show that the unsatisfiability of a particular linear equation is a sufficient condition for generating a KS pair.

In this section, we study whether a given set of vectors and orthogonal sets form a KS pair in more generality. We do so by posing the question as a satisfiability problem, which we cast as an integer linear programming (ILP) instance. The related decision problem is as follows:

Definition 3.1.2. *Let* $V \subset \mathbb{C}^n$ *and let* $O \subseteq \mathcal{O}(V)$ *(see Definition 1.5.2). The* KS decision problem *asks whether* (V, O) *is a KS pair.*

Recall from Definition 1.5.7 and Theorem 1.5.1, that V and O form a KS pair if there is not a function which takes exactly one element of each basis in O to 1, and *at most* one element of all other non-basis pairwise orthogonal sets in O to 1. Given this, we may decide whether a set of vectors V and set of orthogonal sets O form a KS pair by asking whether the following ILP is feasible:

Construction 3.1.3. *Let* V *be a set of vectors in* $\mathbb{C}^n$ *and* $O \subseteq \mathcal{O}(V)$*. We denote the bases contained in* O *by* $B = O \cap \mathcal{B}(V)$*, and the non-basis pairwise orthogonal sets contained in* O *by* $C = O \cap (\mathcal{O}(V) \setminus \mathcal{B}(V))$*.*

$$\begin{aligned}
&\textit{Objective: } 1 \\
&\textit{Constraints: } x_i \in \{0,1\} \\
&\qquad \sum_{j=1}^{|V|} b_{ij} x_j = 1, \ \textit{for all } B_i \in B, \ \textit{where } b_{ij} = \begin{cases} 0 & v_j \notin B_i \\ 1 & v_j \in B_i \end{cases} \\
&\qquad \sum_{j=1}^{|V|} c_{ij} x_j \leq 1, \ \textit{for all } C_i \in C, \ \textit{where } c_{ij} = \begin{cases} 0 & v_j \notin C_i \\ 1 & v_j \in C_i \end{cases}
\end{aligned}$$

Note 3.1.4. *The objective in this case is of no consequence, as we only require that such an assignment exists. In fact, this could more concisely be formulated as a satisfiability (SAT) instance; however, we choose to formulate it as an ILP to more easily make use of the optimization tools in MAGMA and Maple.*

An implementation of the above in MAGMA is provided in Appendix B.2.

Example 3.1.5. *We applied this ILP to the* 31*-vector KS pair found by Conway and Kochen [32]. In Appendix B.2, using this ILP, we were able to show that this set of vectors does indeed form a KS pair, and moreover that it is minimal in the sense that no vector can be removed from the set without making it colourable.*

3.1.2 Applying the ILP to Paley Hadamard Matrices

The Paley construction for Hadamard matrices [37] [30] is a good source of classical Hadamard matrices whose order is not a power of 2. This property allows for many of these matrices to provide KS pairs using Theorem 2.2.4, but this theorem only gives us an expression in terms of rank for the size of these pairs. In this section, we both confirm that these matrices do indeed produce KS pairs, using the ILP from Section 3.1.1, and present a conjecture on the particular sizes of the KS pairs produced by Paley Hadamard matrices using Theorem 2.2.4.

The Paley construction for classical Hadamard matrices takes as an input an odd prime power q, and yields a classical Hadamard matrix of order $q-1$ if $q \equiv 3 \pmod 4$, and a classical Hadamard matrix of order $2(q+1)$, if $q \equiv 1 \pmod 4$. In Appendix B.6, we provide explicit code for the construction for Paley Hadamard matrices for each odd prime power q.

For each odd prime power $q < 10^3$, we took the corresponding Paley Hadamard matrix and generated by the method of Theorem 2.2.4 a set of vectors and bases. We confirmed that the related ILP instances were not feasible, and then we recorded the sizes of the KS pairs. The patterns in our extensive computational experimentation lead us to present the following conjecture on the sizes of KS pairs generated using the Paley Hadamard matrices:

Conjecture 3.1.6. *Let* $H \in \mathrm{GH}(\mathbb{Z}_2, d/2)$ *be the normalized logarithmic form of the order* d *classical Hadamard matrix given by the Paley construction for* q *an odd prime power not one less than a power of 2. Then,* $(\mathcal{V}, \mathcal{B}) = (\mathcal{V}(\mathrm{B}(H, \mathrm{Row}(H))), \mathrm{B}(H, \mathrm{Row}(H)))$ *is a KS pair in* $\mathbb{R}^d$ *with the following data:*

$q \mod 8$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{B}\rvert$	d
1, or 5	$2^{2(q+1)}$	$2^{2(q+1)}$	$2(q+1)$
3	2^q	2^q	$q+1$
7	$2^{\frac{q+1}{2}}$	$2^{\frac{q+1}{2}}$	$q+1$

Table 3.1: Sizes of KS pairs for Paley Hadamard matrices

The purpose of mentioning that q not be one less than a power of 2 is to ensure that $(n+1)/2$ is not a power of 2, so as to meet the premises of Theorem 2.2.4.

Appendix A.2 presents the computational results which motivate Conjecture 3.1.6. As can be noted in Figure A.1, KS pairs generated with these matrices by the method of

Theorem 2.2.4 grow quite large very quickly. In Figure A.2, the number of vectors already begins to exceed 2^{20} for $q > 9$.

From this we conclude that, while the Paley construction is a bountiful source of Hadamard matrices which lead to KS pairs using Theorem 2.2.4, the produced KS pairs are rather large even in the smallest cases.

3.2 Graph Theoretical Formulation

In this section, we outline a graph theoretical formulation for deciding whether a set of vectors and orthogonal sets form a KS pair. This formulation allows us to transfer the KS decision problem (Definition 3.1.2) to the problem of finding all *maximal independent sets* of a graph. In our computational exploration of this topic, the formulation presented in this section was secondary to the approach outlined in Section 3.1.1, but gave an additional perspective which was useful in understanding why particular KS pairs are uncolourable in a more fundamental way.

We begin by defining the notion of an orthogonality graph of a set of vectors, which is based on a definition in a paper of Lovász [27], but has been adapted for the purposes of this thesis.

Definition 3.2.1. *Let $\mathcal{V} \subseteq \mathbb{C}^d$ be a finite set of d-dimensional complex vectors. The* orthogonality graph *of $\mathcal{V}$, denoted $\Omega(\mathcal{V})$ is the labelled graph whose vertices correspond to the vectors in $\mathcal{V}$, and where two vertices are adjacent if and only if their corresponding vectors are orthogonal in $\mathbb{C}^d$.*

Example 3.2.2. *If $\mathcal{V} = \{(1,0,0),(0,1,0),(0,0,1),(2,0,1),(-1,2,2)\}$, then $\Omega(\mathcal{V})$ is given by*

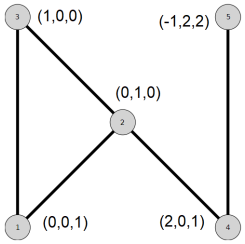

Figure 3.1: Example of orthogonality graph

Definition 3.2.3. *Let $\mathcal{V} \subseteq \mathbb{C}^d$ be a finite set of vectors, and let $S \in \mathcal{O}(\mathcal{V})$ be an orthogonal set. We define the subgraph $\Omega(\mathcal{V}, S)$ of $\Omega(\mathcal{V})$ to be the labelled graph whose vertices correspond the vectors in $\mathcal{V}$, and where two vertices are adjacent if and only if their corresponding vectors both belong to the set S. For a collection of sets $T \subseteq \mathcal{O}(\mathcal{V})$, we define the graph $\Omega(\mathcal{V}, T)$ to be the union of $\Omega(\mathcal{V}, S)$ for all $S \in T$.*

This allows for a graph theoretical formulation of the KS decision problem. Namely, given a finite set of vertices $\mathcal{V} \subset \mathbb{C}^d$ and a finite set of orthogonal sets $T \subseteq \mathcal{O}(\mathcal{V})$, we seek to show that there does not exist a maximal independent set I of $\Omega(\mathcal{V}, T)$ such that for all bases $B \in T \cap \mathcal{B}(\mathcal{V})$, the subgraph $\Omega(\mathcal{V}, B)$ contains a vertex in I. This formulation models the vectors mapped to 1 as elements of I.

Thus, we have transferred the KS decision problem into the problem of deciding whether there exists a maximal independent set on a graph with an additional covering property. Checking this additional property can be done in polynomial time, so the complexity of this approach is primarily determined by the complexity of the problem of finding all maximal independent sets of a graph (all maximal independent sets can be found with $\mathrm{O}\left(3^{n/3}\right)$ time complexity [23], where n is the number of vertices).

Example 3.2.4. *The following is an example of the orthogonality graph of the vectors found in* $\mathrm{B}(H, S)$ *for* $H \in \mathrm{GH}(\mathbb{Z}_3, 2)$ *as given by Construction 2.4.2 and* S *a particular* 2*-dimensional subspace of* $\mathrm{Row}(H)$ *for which* $\mathcal{V}(\mathrm{B}(H, S))$ *has a KS colouring. The vertices belonging to the associated maximal independent set are circled in red.*

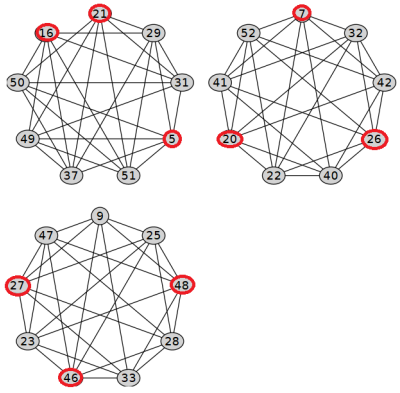

Figure 3.2: Orthogonality graph with indicated maximal independent set

An implementation of this formulation in MAGMA [7] is given in Appendix B.3.

3.3 Searching for Integer-Valued KS Pairs in $\mathbb{R}^3$

In this section we present two approaches to searching for KS pairs with integer-valued vectors in $\mathbb{R}^3$. In Section 3.3.1, we follow a construction for a KS pair introduced in a 2022 paper by Cortez and Reyes [11]. We use the ILP presented in Section 3.1.1 to find a simplification of their KS pair, and show how the computational methods we developed might be used to address one of the open problems they present. In Section 3.3.2, we search for integer-valued KS pairs in $\mathbb{R}^3$ by studying the finite order subgroups of $\mathrm{GL}_3(\mathbb{Z})$, the idea

being that they may be a subgroup of the automorphism group of a subset of vectors lying on the integer lattice in $\mathbb{R}^3$ that yields a KS pair.

3.3.1 Applying the ILP to the Cortez–Reyes KS Pair

This section starts by presenting an approach for finding KS pairs which was introduced in a 2022 paper by Cortez and Reyes [11]. This approach focuses on finding integer-valued KS pairs in $\mathbb{R}^3$. We use the ILP presented in Section 3.1.1 to find a simplification of their KS pair, and present some additional computational results.

The approach begins by introducing the quadratic form $q\colon \mathbb{Z}^3 \to \mathbb{Z}_{\geq 0}$, defined by

$$q(v) = vv^\top = v_1^2 + v_2^2 + v_3^2.$$

Additionally, for each $N \in \mathbb{Z}_{\geq 0}$ we define

$$S(N) = \{v \in \mathbb{R}^3 \colon\ q(v) \text{ divides a power of } N\}.$$

This set-up was motivated by an earlier result in a paper coauthored by Reyes [6], which (in different language) showed that if N is not divisible by 2 or not divisible by 3, then each $V \subset S(N)$ admits a KS colouring.

It can also be noted that if N divides M, then $S(N) \subseteq S(M)$, which gives both that it suffices to consider N square-free, and that $S(N)$ being KS uncolourable implies $S(M)$ is KS uncolourable.

The above give us that if $S(N)$ is uncolourable, then N is a square-free multiple of 6. The KS pair given by Conway and Kochen [32] consists of a set of 31 integer-valued vectors in $\mathbb{R}^3$, which is a subset of $S(30)$. This implies that $S(30)$ is uncolourable, and moreover, that $S(N)$ is uncolourable for any multiple of $30 = 2 \cdot 3 \cdot 5$.

The paper by Cortez and Reyes constructs an uncolourable subset of $S(462)$, where $462 = 6 \cdot 7 \cdot 11$, which establishes that N need not be a multiple of 30 for $S(N)$ to be uncolourable. Their construction uses 85 integer-valued vectors formed by considering unions of the sets of vectors in $S(462)$ whose images under q equal particular divisors of 462:

$$Q = \{v \in \mathbb{Z}^3 \colon\ q(v) \in \{1, 2, 3, 6, 21, 33, 77\} \text{ and } v \text{ is well-signed}\} \setminus \{(1, 4, 4), (4, 5, 6)\},$$

where *well-signedness* is a notion (defined in [11]) for choosing a unique representative of the projective equivalence class each vector belongs to.

In Appendix B.2 we provide MAGMA [7] code which uses the ILP of Section 3.1.1 to verify that this set of 85 vectors Q along with $\mathcal{O}(Q)$ comprise a KS pair. Additionally, we use this same code to show that their given set of vectors is not minimal, in the sense that there exists a subset of Q which is also uncolourable. We show explicitly in Appendix B.2 that there is a minimal uncolourable subset of Q which contains only 57 vectors.

We may also test systematically whether for any N the set $S(N)$ is uncolourable. As the set $S(N)$ is an infinite set, we must choose a maximum power of N, say e, such that we include only those well-signed vectors whose image under q divides N^e. The MAGMA code which performs this is given in Appendix B.5.

We were able to verify that there are no uncolourable subsets of well-signed vectors in $S(6)$ with quadratic form dividing up to 6^5. That is, all subsets of $S(6)$ consisting of vectors v with $q(v)$ dividing 6^5 are KS colourable. It could be that higher powers of 6 must be considered to prove uncolourability. We denote these subsets by

$$Q(N, e) = \{v \in S(N) \colon\ q(v) \text{ divides } N^e\} \subseteq S(N).$$

We were able to find several new KS pairs by considering these subsets. We summarize some results of this approach in Table 3.2. In all three cases, the size of the KS pair we found is strictly smaller than the pair presented in [11].

N	e	KS pair	#Vectors	#Bases	#Orth. Sets.	Minimal
6	5	No	—	—	—	—
$30 = 6 \cdot 5$	1	Yes	31	17	20	Yes
$42 = 6 \cdot 7$	2	No	—	—	—	—
$462 = 6 \cdot 7 \cdot 11$	1	Yes	57	28	32	Yes
$714 = 6 \cdot 7 \cdot 17$	1	Yes	65	36	32	Yes
$6p$ $p \in \{11, 13, \ldots, 167\}$	1	No	—	—	—	—
$42p$ $p \in \{13, 19, 23, 29, 31, 37\}$	1	No	—	—	—	—
$66p$ $p \in \{13, 17, 19\}$	1	No	—	—	—	—
$78p$ $p \in \{17, 19\}$	1	No	—	—	—	—

Table 3.2: New KS pairs found as subsets of $S(N)$

The code to generate the results of Table 3.2 is included in Appendix B.7. It should be noted that the given data for the listed KS pairs does not necessarily represent the *minimum* KS pair which is a subset of $S(N)$; it should also be emphasized that a "No" entry on the table does not imply that a KS pair is not possible for higher powers of N.

A key insight from the results in Table 3.2 is that, by showing that $S(714)$ is uncolourable, we have shown that $N = 462$ is not unique in being an integer not divisible by 30 for which $S(N)$ is uncolourable. Both 462 and 714 are divisible by 42, but from the data, it does not appear that $S(42)$ is uncolourable, as there exist several examples of multiples of 42 which are demonstrated to be colourable, at least up to maximum exponent 1.

Example 3.3.1. *We were also able to generate several distinct uncolourable minimal subsets of* $S(30)$ *of size* 31*, including the* 31*-vector KS pair by Conway and Kochen [32]. In addition to recreating the Conway–Kochen pair, we were able to generate* 5 *new pairs, each consisting of* 31 *vectors and* 17 *bases. All of these pairs were found by first considering*

$$Q(30,1) \setminus \{v \in \mathbb{R}^3 \colon\ q(v) \in \{10,15\}\},$$

which is an uncolourable subset consisting of 37 *vectors. We then reduce the number of vectors in the subset by removing them one at a time. We were able to compute that there are exactly* 6 *distinct ways to remove* 6 *vectors from the above set to get a KS pair of size* 31*.*

In all of these cases, the reduction of each uncolourable set proceeded initially by removing vectors one at a time, verifying each time that the set remained uncolourable, until no removable vectors remained. However, in performing this procedure, it became clear that there were patterns in which vectors were removable; for example, in the case of the 85 vector set of Cortez and Reyes, it was the case that whenever (a, b, c) was removable, so too were $(-a, b, c)$, $(a, -b, c)$, and $(a, b, -c)$. This is partially what motivates Section 3.3.2, where we study the potential symmetries of integer-valued vectors in $\mathbb{R}^3$ forming KS pairs.

All of the KS pairs presented in this section can be found in Appendix B.7.

3.3.2 Potential Symmetries of Integer-Valued KS Pairs in $\mathbb{R}^3$

In this section, we describe a method for searching for small integer-valued KS pairs in $\mathbb{R}^3$ by considering their potential symmetries, rather than looking for the vectors themselves. In particular, we aim to study to what extent there are any small subsets of the 3-dimensional integer lattice in $\mathbb{R}^3$ that are invariant under the action of finite-order subgroups of $\mathrm{GL}_3(\mathbb{Z})$. In this section, by *small* we mean containing strictly fewer than 31 vectors; that is, we are looking for KS pairs that are smaller than the record Conway–Kochen pair [32]. It has previously been shown that there are no such subsets of integer-valued vectors in $\mathbb{R}^3$ containing fewer than 23 vectors [24], so we aim for subsets containing a number of vectors in the range of 23 to 30.

The finite order subgroups of $\mathrm{GL}_3(\mathbb{Z})$ are enumerated in a 1971 paper of Ken-Ichi Tahara [36], where it is shown that any finite order element of $\mathrm{GL}_3(\mathbb{Z})$ is of order $1, 2, 3, 4$, or 6, and therefore that any finite subgroup of $\mathrm{GL}_3(\mathbb{Z})$ is of order $2^i \cdot 3^j$ for some $i, j \in \mathbb{Z}_{>0}$. Tahara uses these facts to exhaustively list all finite-order subgroups of $\mathrm{GL}_3(\mathbb{Z})$ up to conjugation. In Appendix B.4, we catalog these groups using MAGMA.

Example 3.3.2. *One such group is given by the following three generators:*

$$W_1 = \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right\},$$

which is isomorphic to $A_4 \times \mathbb{Z}_2$.

This paper [36] contains some errors; namely, the group W_5 is presented as being of order 24, but is actually order 12, and there is a typo in the second generator matrix of the order 24 group W_{11}, which should be

$$W_{11} = \left\{ -\begin{bmatrix} 1 & 1 & 0 \\ -2 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix}, -\begin{bmatrix} -\mathbf{1} & -1 & -1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \right\}.$$

Apart from these issues, we were able to verify (Appendix B.4) that all groups presented in the paper are of the stated orders and are indeed isomorphic to the groups that the paper claims.

The vectors of the 31-vector KS pair found by Conway and Kochen [32] are all integer-valued, lying on the $5 \times 5 \times 5$ integer lattice centred at the origin. For this reason, we limit our search to vectors in this set; that is we consider potential KS sets with vectors arising from the set

$$A = \{(a, b, c) \in \mathbb{R}^3 \colon\ a, b, c \in \{0, \pm 1, \pm 2\}\}.$$

Adopting the notion of *well-signedness* from [11], the set A consists of 49 distinct nonzero non-collinear vectors. We already know that $(A, \mathcal{B}(A))$ is a KS pair, as it contains the Conway–Kochen set as a subset. The remainder of this section will focus on searching for a construction of a subset of the 49 vectors that remains uncolourable.

We proceed by considering the action of each of the groups $G \in \mathrm{GL}_3(\mathbb{Z})$ found in [36] on A. That is, we fix a group G, and we consider the set of orbits given by

$$O_G = \{G \cdot v \colon\ v \in A, G \cdot v \subseteq A\},$$

where we note that we are restricting to the orbits which are contained in the set A, as there is the possibility that $G \cdot v \not\subseteq A$, for some $v \in A$.

Then, we look for unions of the orbits in O_G such that the total number of vectors falls approximately into the range of 23 to 31, i.e. a number of vectors larger than the proven minimum and less than the current minimum KS pair in $\mathbb{R}^3$. In other words, we are looking for a subset of orbits $S \subseteq O_G$ such that

$$\left| \bigcup_{s \in S} s \right| \in [23, 31].$$

After such subsets are found, if we write $\mathcal{V} = \bigcup_{s \in S} s$, then the ILP of Section 3.1 can be used to check whether $(\mathcal{V}, \mathcal{O}(\mathcal{V}))$ is a KS pair.

The motivation for constructing the vectors of a KS pair as a union of orbits is twofold. One reason is that combinatorial objects with exceptional properties typically do have large

automorphism groups. It is possible that the exceptional properties of such objects can be better understood by first examining their symmetries. The second reason is that, in terms of computational searches for these objects, prescribing the potential symmetries of a set significantly reduces the computation time in that the size of the search space is profoundly cut down. For example, if we want to construct a putative KS set V of vectors as a subset of some finite set A containing n vectors, then the size of the search space is 2^n, but if we prescribe an order 2 symmetry on V, then this cuts the size of the search space to approximately $2^{n/2}$.

Using the implementation in Appendix B.5, we were able to verify that for several subgroups G, there does not exist an $S \subseteq O_G$ such that the union of sets in S gives a KS pair. In particular, we have verified that for any $S \subseteq O_G$, for any subgroup $G \in \mathrm{GL}_3(\mathbb{Z})$ listed in [36] of order larger than 8, there are no KS pairs of size in the target range produced using via the method outlined above.

Note 3.3.3. *The groups listed in [36] and given in Appendix B.4 comprise the complete collection of the finite subgroups of* $\mathrm{GL}_3(\mathbb{Z})$*, up to conjugation. We have verified that none of the subgroups of order* 8 *or higher yield KS pairs via the above method, but only as they are given in the paper. It is possible that subgroups conjugate to those listed could yield KS pairs, as the action of two subgroups of* $\mathrm{GL}_3(\mathbb{Z})$ *in the same conjugacy class can produce significantly different sets of orbits.*

The decision to define A as the size 5 integer lattice is based primarily on the precedent set by Conway and Kochen, but it remains possible a KS pair could exist with the above symmetries by expanding to a larger subset of the lattice. This is likely unnecessary, as the size 5 lattice is already uncolourable.

It is also useful to consider the symmetry groups of existing KS pairs; for example, knowing the symmetry group of the 31-vector KS pair found by Conway and Kochen would help illuminate what symmetries are possible for small KS pairs, and ultimately help in the search for KS pairs using fewer than 31 vectors. We have found that the symmetry group of the Conway–Kochen pair is non-trivial; in Appendix B.5 we provide code which demonstrates that the set vectors is invariant (up to scaling) under the action of the subgroup of $\mathrm{GL}_3(\mathbb{Z})$ generated by

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

which is isomorphic to $C_4 \times C_2$, and conjugate to the order 8 group W_1 in [36].

From this point forward, we consider the groups acting on a set of lines (projective points) rather than representative vectors. We are therefore concerned with finding transformations that fix our set of lines, that is we are looking for projectivities (homographies)

under which our set of lines is invariant. Thus, we will look for a symmetry group that is a finite subgroup of $\mathrm{PGL}_3(\mathbb{Q})$.

As a subgroup of $\mathrm{PGL}_3(\mathbb{Q})$, we were able to compute that the symmetry group of the Conway–Kochen KS pair is generated by A and B, but is only of order 4, as

$$A^2B = -I,$$

which is the identity in $\mathrm{PGL}_3(\mathbb{Q})$.

A projective frame in $\mathbb{RP}^{d-1}$ is a set of $d+1$ points such that no subset of d points is linearly dependent. For the remainder of this section, let $d = 3$.

To calculate the symmetry group of a finite set of lines T (points in $\mathbb{RP}^2$), we use the fact that any projectivity will map any projective frame to another projective frame. We may find the projective frames by finding all sets of four points with the property that no subset of size 3 is linearly dependent. Once all projective frames are found, the procedure to calculate all projectivities fixing a given set of lines is as follows. Fix a frame $P = \{v_1, v_2, v_3, v_4\} \subset T$, and compute a_1, a_2, a_3 such that $v_4 = a_1v_1 + a_2v_2 + a_3v_3$ (note that all a_i are nonzero by the assumption that P is a frame). For every other frame $Q = \{u_1, u_2, u_3, u_4\} \subset T$ where $u_4 = b_1u_1 + b_2u_2 + b_3u_3$, let L be the linear transformation on $\mathbb{R}^3$ taking v_i to $\frac{b_i}{a_i}u_i$ for $1 \leq i \leq 3$ so that

$$L(v_4) = L\left(\sum_{i=1}^{3} a_iv_i\right) = \sum_{i=1}^{3} b_iu_i = u_4,$$

and check whether L fixes T set-wise. If this is the case, then L is a symmetry of T. In this way, we can find all elements of the symmetry group.

Code which implements this procedure for the Conway–Kochen KS pair is found in Appendix B.8. Using this code, we were able to verify that as a subgroup of $\mathrm{PGL}_3(\mathbb{Q})$, the symmetry group of the Conway–Kochen KS pair is order 4 isomorphic to $C_2 \times C_2$, and is generated by the matrices A and B as above.

Chapter 4

Outlook

In this chapter, we begin in Section 4.1 with a presentation of a selection of some questions which remain to be answered, and suggest some directions of future research. In Section 4.2, we present some additional theoretical and computational perspectives which were not ultimately pursued during the course of writing this thesis, but could still be interesting to consider.

4.1 Remaining Questions and Future Directions of Research

In this section, we pose some relevant questions which remain to be answered and propose future research directions.

- What is the smallest KS pair that can be generated by the methods of Chapter 2?
- Could a smaller KS pair be found by generating the prebases of a Hadamard matrix with a set that is not a subspace of the row space of the matrix?
- Does Conjecture 2.3.13 hold? If so, it would be interesting to explore whether there is a more natural characterization of which subspaces of the row space give KS pairs, as it could be that slice weights are simply an indicator of a more fundamental explanation.
- How might a characterization for which subspaces lead to KS pairs change if we include non-basis pairwise orthogonal sets? Could there exist a modification of Definition 2.3.9 for this case?
- Can all KS pairs be understood as having originated from shifts of a Hadamard matrix? If so, is it possible to find new Hadamard matrices by considering existing KS pairs?

We now present some future research directions:

In Section 2.3, we finish by presenting Conjecture 2.3.13, which posits that the existence of a solution to $\ell(x) = |K|$ is enough to guarantee colourability of the bases given by a subspace.

There are numerous examples of Butson Hadamard matrices of type $\mathrm{BH}(n, q)$ for q non-prime, a large number of which are collected in [22]. As in the prime case, many of these can be shown to correspond to KS pairs in the same way as in the prime case, the key difference being that these matrices do not necessarily correspond to generalized Hadamard matrices. The row spaces of the logarithmic forms of these matrices are also no longer necessarily vector spaces, as $\mathbb{Z}_q$ is not a field, so there is a need to generalize the theorems of Chapter 2 to accommodate $\mathbb{Z}_q$-(sub)modules.

In Section 3.3.2, we presented a method for searching for integer-valued KS pairs in $\mathbb{R}^3$ by studying symmetries of the lattice $\mathbb{Z}^3$. It would be most interesting to fully establish that there are no KS pairs in the target size up to symmetries given by the groups mentioned in the section. It would also be of interest to consider applying this method to groups which belong the conjugacy classes of the groups in [36]. This pursuit would benefit from both additional computational resources and development of a more efficient method.

4.2 Alternative Computational Perspectives

In this section, we briefly outline some additional computational perspectives that were considered during the course of the research, but were never fully developed into any significant theoretical or computational results. It is possible that these perspectives could be useful in future research.

4.2.1 Exact Hitting Set Formulation

The KS decision problem is perhaps best phrased as an instance of the exact hitting set problem:

Given a collection S of subsets of a set X, an *exact hitting set* is a set M such that every subset of S contains exactly one element of M. Each instance of a hitting set problem can be encoded into an $|V| \times |S|$ matrix A, where $A_{i,j}$ is 1 if $v_i \in S_i$ and 0 otherwise.

Given this encoding, the KS decision problem is shifted to finding a subset of rows of A such that each column contains exactly one 1. We can look for solutions to an instance of an exact hitting set problem using Knuth's algorithm X [19].

We managed to find and slightly modify a simple implementation of this algorithm in Python [2]. This algorithm has been useful for independently confirming the results shown with the purpose-built ILP in Magma [7], but has not thus far provided any speedup or novel results.

4.2.2 Frobenius Coin Problem

When considering solutions to the linear equation (Definition 2.3.9), presented in Section 2.3, it became clear that establishing existence of solutions to this equation was essentially an instance of the Frobenius coin problem [33]. This problem asks, given coprime pos-

itive integers $a_1, \ldots, a_k$, what is the largest integer which cannot be expressed as $\sum_{i=1}^{k} a_i x_i$, where x_i are non-negative integers?

If $k = 1$, then only multiples of a_1 can be expressed; in particular, every integer can be expressed if $a_1 = 1$. If $k = 2$, then the largest integer which cannot be expressed is given by $a_1 a_2 - a_1 - a_2$.

For $k > 2$, even if a closed form solution exists, it may be rather complicated, so instead we use an approach with generating functions. For each a_i consider the following functions where the coefficient of x^t is either 0 or 1, depending on whether t is a multiple of a_i:

$$p_{a_i}(x) = 1 + x^{a_i} + x^{2a_i} + \cdots = \frac{1}{1 - x^{a_i}}$$

Then, we consider the product of these:

$$\prod_{i=1}^{k} p_{a_i}(x) = \prod_{i=1}^{k} \frac{1}{1 - x^{a_i}},$$

whose coefficients could be found by considering its Taylor expansion.

Taking advantage of this, denote k coprime slice weights $w_1, w_2, \ldots, w_k$ and the number of prebases by n. Then, a solution exists to $\ell(x) = n$ if and only if the coefficient of x^n in the above function is greater than zero.

This approach would become quite useful in the event a proof of Conjecture 2.3.13 can be found. That is, if we can show that the existence of a solution to $\ell(x)$ implies the colourability of a set of bases, we could use this formulation to quickly determine whether a subspace will generate a KS pair.

Bibliography

[1] F. Arends, J. Ouaknine, and C. W. Wampler. On searching for small Kochen-Specker vector systems. In P. Kolman and J. Kratochvíl, editors, *Graph-Theoretic Concepts in Computer Science*, pages 23–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[2] A. Assaf. Algorithm X in 30 lines! `https://www.cs.mcgill.ca/~aassaf9/python/algorithm_x.html`. Accessed 19 Jan. 2023.

[3] G. Bacciagaluppi and A. Valentini. *Quantum Theory at the Crossroads: Reconsidering the 1927 Solvay Conference*. Cambridge University Press, 2009.

[4] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1:195–200, 1964.

[5] J. S. Bell. On the problem of hidden variables in quantum mechanics. *Rev. Mod. Phys.*, 38:447–452, 1966.

[6] M. Ben-Zvi, A. Ma, and M. Reyes. A Kochen-Specker theorem for integer matrices and noncommutative spectrum functors. *J. Algebra*, 491:280–313, 2017. With an appendix by A. Chirvasitu.

[7] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[8] A. T. Butson. Generalized Hadamard matrices. *Proc. Amer. Math. Soc.*, 13:894–898, 1962.

[9] A. Cabello. A proof with 18 vectors of the Bell-Kochen-Specker theorem. *New developments on fundamental problems in quantum physics*, pages 59–62, 1997.

[10] A. Cabello. State-independent quantum contextuality and maximum nonlocality. arXiv:1112.5149, 2012.

[11] I. Cortez and M. L. Reyes. A set of integer vectors with no Kochen-Specker coloring. arXiv:2211.13216, 2023.

[12] D.A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer New York, 2008.

[13] D. A. Drake. Partial λ-geometries and generalized Hadamard matrices over groups. *Canadian J. Math.*, 31(3):617–627, 1979.

[14] A. Einstein. *The Collected Papers of Albert Einstein, Volume 15 (Translation Supplement): The Berlin Years: Writings & Correspondence, June 1925–May 1927*. Princeton University Press, 2018.

[15] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, 1935.

[16] J. S. Hadamard. Résolution d'une question relative aux déterminants. *Bulletin des Sciences Mathématiques*, 17:240–246, 1893.

[17] D. Jungnickel. On difference matrices, resolvable transversal designs and generalized Hadamard matrices. *Math. Z.*, 167(1):49–60, 1979.

[18] M. Kernaghan. Bell-Kochen-Specker theorem for 20 vectors. *Journal of Physics A: Mathematical and General*, 27(21):L829, 1994.

[19] D. E. Knuth. Dancing links. arXiv:cs/0011047, 2000.

[20] S. Kochen and E. P. Specker. The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics*, 17(1):59–87, 1967.

[21] P. H. J. Lampio. *Classification of difference matrices and complex Hadamard matrices*. Doctoral thesis, School of Electrical Engineering, Aalto University, 2015.

[22] P. H. J. Lampio, P. R. J. Östergård, and F. Szöllősi. Orderly generation of Butson Hadamard matrices. *Math. Comp.*, 89(321):313–331, 2020.

[23] E. L. Lawler. A note on the complexity of the chromatic number problem. *Information Processing Lett.*, 5(3):66–67, 1976.

[24] Z. Li, C. Bright, and V. Ganesh. A SAT solver and computer algebra attack on the minimum Kochen-Specker problem. arXiv:2306.13319, 2023.

[25] P. Lisoněk. Kochen-Specker sets and Hadamard matrices. *Theoretical Computer Science*, 800:142–145, 2019. Special issue on Refereed papers from the CAI 2017 conference.

[26] P. Lisoněk, P. Badziąg, J. R. Portillo, and A. Cabello. Kochen-Specker set with seven contexts. *Physical Review A*, 89(4), 2014.

[27] L. Lovász, M. Saks, and A. Schrijver. Orthogonal representations and connectivity of graphs. *Linear Algebra Appl.*, 114/115:439–454, 1989.

[28] Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. Maple 2021.1. `https://www.maplesoft.com/`.

[29] D. N. Mermin. Simple unified form for the major no-hidden-variables theorems. *Phys. Rev. Lett.*, 65:3373–3376, 1990.

[30] R. E. A. C. Paley. On orthogonal matrices. *Journal of Mathematics and Physics*, 12(1-4):311–320, 1933.

[31] A. Peres. Two simple proofs of the Kochen-Secker theorem. *Journal of Physics A: Mathematical and General*, 24(4):L175, 1991.

[32] A. Peres. *Quantum Theory: Concepts and Methods*. Fundamental Theories of Physics. Springer Netherlands, 2006.

[33] J. L. Ramírez Alfonsín. *The Diophantine Frobenius Problem*, volume 30 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2005.

[34] J. J. Sylvester. LX. Thoughts on inverse orthogonal matrices, simultaneous sign-successions, and tessellated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):461–475, 1867.

[35] W. Tadej and K. Życzkowski. A concise guide to complex Hadamard matrices. *Open Syst. Inf. Dyn.*, 13(2):133–177, 2006.

[36] K. Tahara. On the finite subgroups of GL(3, Z). *Nagoya Math. J.*, 41:169–209, 1971.

[37] J.H. van Lint and R.M. Wilson. *A Course in Combinatorics*. Cambridge University Press, 2001.

[38] M. Waegell and P. K. Aravind. Golay codes and quantum contextuality. *Phys. Rev. A*, 106(6):Paper No. 062421, 6, 2022.

Appendix A

Data for Constructions of KS Pairs

A.1 KS pairs Generated by Subspaces of the Row Spaces of Hadamard matrices

In this section, we present a selection of data related to which subspaces of the row spaces of generalized Hadamard matrices provide KS pairs. This data was collected using the MAGMA [7] code in Appendix B.2. Several of the listed subspaces do not provide KS pairs via the methods of Section 2.3, but are nonetheless shown to be uncolourable by considering smaller non-basis pairwise orthogonal sets of vectors.

For a generalized Hadamard matrix, the table below catalogs whether various proper subspaces of its row space yield KS pairs. In all computed examples, whether a subspace has led to a KS pair is entirely dependent on the weights of its slices. Therefore, we list the subspaces only up to their slice weights, writing the total number of such subspaces in the column labelled "Count". The final column lists whether each type of space gives a KS pair via the ILP. The format displaying the size of each KS pair is

$$(\#\text{Vectors}, \#\text{Bases}, \#\text{Non-basis Orthogonal Sets}),$$

and if the subspace does not generate a KS pair by any method outlined in this thesis, we write "No". We notate the slice weights for a subspace by $w_1^{e_1} w_2^{e_2} \cdots w_n^{e_n}$, where each w_i is a distinct weight, and e_i is the number of slices of weight w_i.

The matrices H_3, H_5, and H_7 are monomial equivalent to those produced by Construction B.1.1, for $p = 3, 5, 7$, and were originally sourced from [22]. The matrices P_q are those given by the Paley construction for classical Hadamard matrices for prime power q.

Hadamard Matrix	Vector Space	R.Space Dim.	Subspace Dim.	Weights	Count	KS Pair
H_3	$\mathbb{C}^6$	4	3	2^3	15	$(81, 27, 0)$
				$1^2 4$	15	$(81, 27, 810)$
				3^2	10	No
			2	$1^4 2$	45	$(45, 9, 252)$
				$1^2 2^2$	45	No
				$1^3 3$	20	No
				1^6	20	No
			1	1^6	25	No
				$1^4 2$	15	No
H_5	$\mathbb{C}^{10}$	4	3	2^5	45	$(625, 125, 0)$
				$1^1 2^1 3^1 4^1$	101	No
				$1^4 6$	10	No
			2	1^{10}	204	$(250, 25, 8250)$
				$1^8 2^1$	125	$(225, 25, 6475)$
				$1^6 2^2$	300	$(200, 25, 4825)$
				$1^4 2^3$	50	$(175, 25, 3525)$
				$1^5 2^1 3^1$	100	No
				$1^5 5^1$	2	No
				$1^2 2^4$	25	No
			1	1^{10}	121	No
				$1^8 2^1$	25	No
				$1^6 2^2$	10	No
H_7	$\mathbb{C}^{14}$	4	3	2^7	91	$(2401, 343, 0)$
				$2^4 3^2 4^1$		No
			2	1^{14}	888	No
				$1^{12} 2^1$	343	No
				$1^2 2^6$	49	No
				$1^{10} 2^2$	882	No
				$1^6 2^4$	98	No
				$1^8 2^3$	294	No
				$1^7 2^2 3^1$	294	No
				$1^7 7$	2	No
			1	1^{14}	337	No
				$1^{12} 2^1$	49	No
				$1^8 2^3$	14	No
P_5	$\mathbb{R}^{12}$	12	4	$1^{10} 2^1$	1	$(176, 16, 6120)$
P_{11}	$\mathbb{R}^{12}$	11	8	3^4	1	$(1024, 256, 0)$
			9	6^2	1	$(1024, 512, 0)$
P_{23}	$\mathbb{R}^{24}$	12	8	3^8	1	$(2048, 256, 0)$
			9	6^4	1	$(2048, 512, 0)$
			10	12^2	1	$(2048, 1024, 0)$

Table A.1: Data for KS pairs given by subspaces of the row spaces of various Hadamard matrices

A.2 Paley Construction

The following graphs present the sizes of KS pairs given by the Paley Hadamard matrices for each odd prime power $q < 10^3$. The second graph presents a subset of the first, focusing only on odd prime powers less than 50. In both graphs, the number of vectors is presented logarithmically.

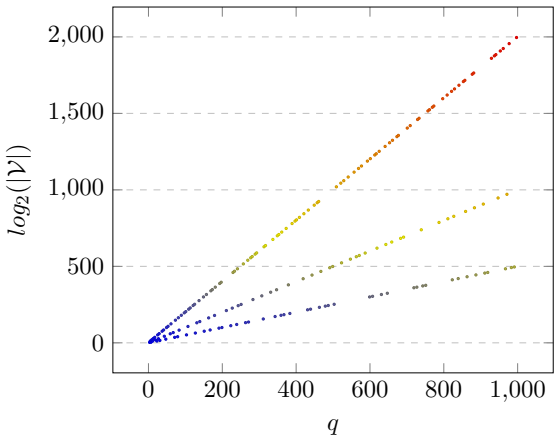

Figure A.1: Size of KS pair ($|\mathcal{V}|$) over various prime powers $q < 10^3$

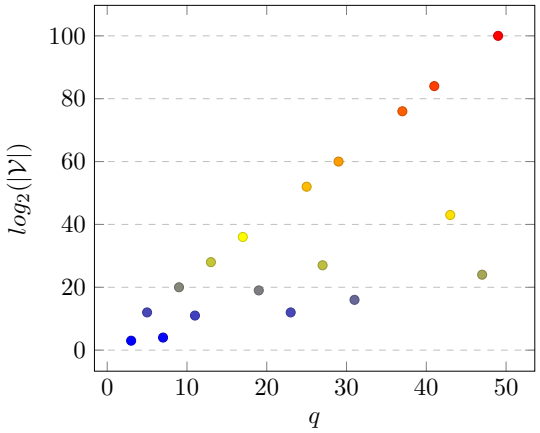

Figure A.2: Size of KS pair ($|\mathcal{V}|$) over various prime powers $q < 50$

A.3 Euler Diagram for Generalizations of Hadamard Matrices

The following is a reproduction of an Euler diagram found in [21], which summarizes the various generalizations of Hadamard matrices mentioned in Chapter 1.

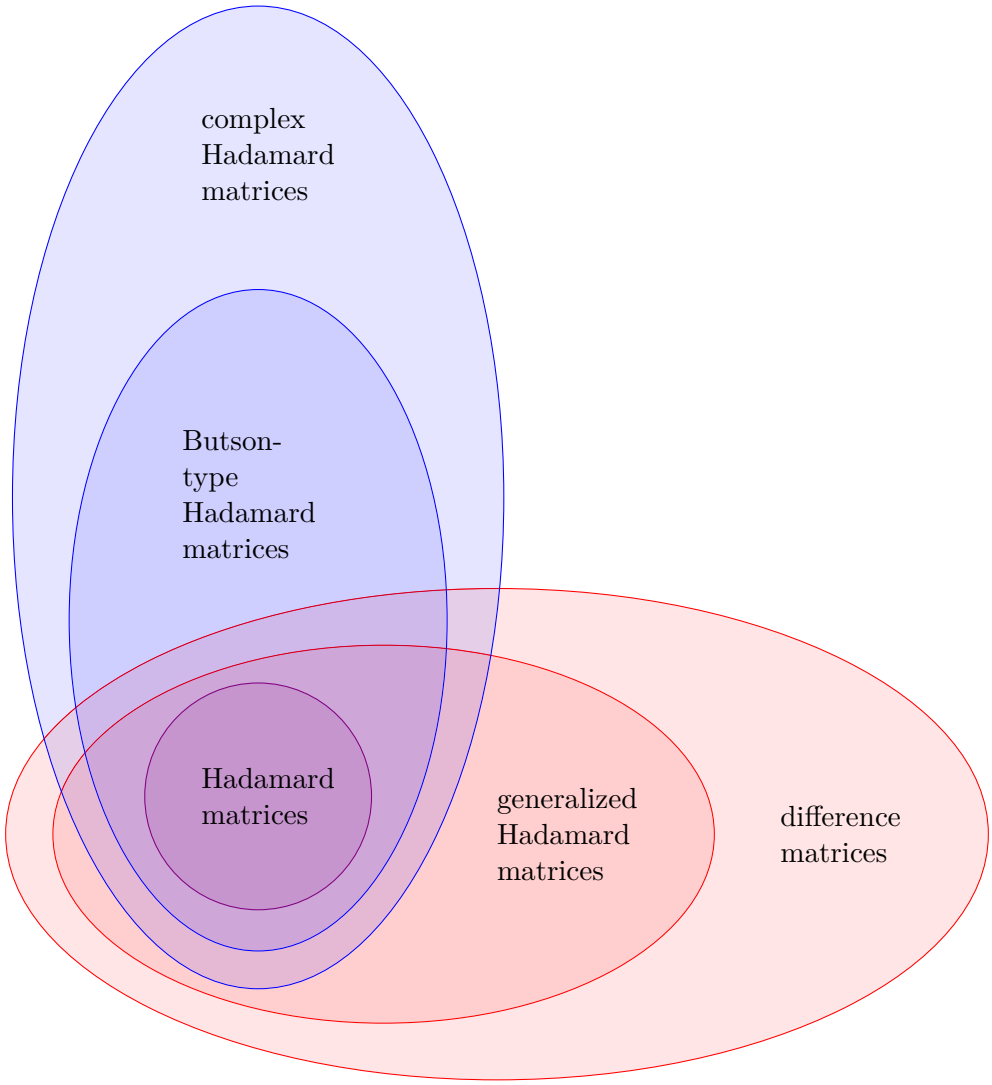

Figure A.3: Euler diagram presenting various generalizations of Hadamard matrices. Reproduced from the cover of [21] with author's permission.

Appendix B

Computer Code

This chapter presents much of the relevant computer code which was developed throughout the course of this thesis research. The majority of the code was written using either MAGMA [7] or Maple [28], making use of built-in packages only.

B.1 Normalized Jungnickel Construction and Related Calculations

This section presents code relevant to the infinite family of KS pairs produced in Section 2.4.

B.1.1 Normalized Jungnickel Construction (MAGMA)

The following is a MAGMA function which produces the normalized generalized Hadamard matrix defined in Theorem 1.4.29. The function has required argument `q` which is the odd prime-power required by the construction, and optional argument for the desired non-square parameter `NS` which by default assumes the smallest non-square in the field.

```
Jungnickel := function(q:NS:=0)
  //q is some odd prime power, matrix will be order 2q over F_q
  //NS non-square parameter, default is to take either -1, or
     smallest one, depending on case

  if IsPrimePower(q) then //input verification

    if IsPrime(q) then
      F:=[GF(q)!i:i in [0..q-1]];
    else
      F := SetToSequence(Set(GF(q)));
      copy:=F[1]; //force 0 to be the first element
      idx:=Index(F,0);
      F[1]:=F[idx];
```

```
  F[idx]:=copy;

end if;

N:=Setseq(Set(GF(q)) diff {x^2:x in Set(GF(q))}); //set of
    all non-squares

if NS in N then //if inputted NS is actually a non-square
  n:=NS;
elif q mod 4 eq 3 then //else if q is 3 mod 4
  n:=q-1;
else //else just take the smallest non-square
  n:=N[1];
end if;

//printf "\nq=%o\ZZ_{>0}=%o\n",q,n;

D:=Matrix(GF(q),2*q,2*q, []);

for i:=1 to q do
  for j:=1 to q do

    x:=F[i];
    y:=F[j];

    D[i,j]:=x*y+(x^2/4);
    D[i,j+q]:=x*y+(n*x^2/4);
    D[i+q,j]:=x*y-y^2-(x^2/4);
    D[i+q,j+q]:=(x*y-y^2-(x^2/4))/n;

  end for;
end for;

//now make the first row all 0 (F[1]=0, so it shouldn't
    matter anyway)
for j:=1 to 2*q do
  t:=D[1,j];
  for i:=1 to 2*q do
    D[i,j]:=D[i,j]-t;
  end for;
end for;

//now make first column all 0
for i:=1 to 2*q do
  t:=D[i,1];
  for j:=1 to 2*q do
    D[i,j]:=D[i,j]-t;
```

```
        end for;
      end for;
      printf "q=%o  n=%o  rank=%o\n rank of inner 4 rows:%o\n", q
          ,n,Rank(D),Dimension(sub<RowSpace(D) | {D[i]: i in [q-1..q
          +2]}>);
      return D;
    end if;
  end function;
```

B.1.2 Linear Independence of Rows (Maple)

This section presents a portion of the proof of Theorem 2.4.3 written in Maple [28]. The following code computes the determinant of the matrix `A` and computes its zeros.

```
with(LinearAlgebra):
A := Matrix(4, 4, [[2, 1, 4, 0], [4, 2, 4, 2], [4*n*(n - 1), n*(n
    - 1), 0, n - 1], [-8*n + 4*n*(n - 1), -4*n + n*(n - 1), -4, n
    - 1]]):
d := Determinant(A);
                    > d := -64*n^3 + 80*n^2 - 16*n
solve(d);
                    > Vector[column](3, [0, 1, 1/4])
```

B.1.3 Finding Coefficients for r^{12} (Maple)

This section presents a portion of the proof of Theorem 2.4.3 written in Maple. The following code shows that the upper rows of the matrix D can be written as a linear combination of r^{12}_{-2} and r^{12}_{-1} by finding a necessary condition on the coefficients, and verifying that they work in general.

```
A := Matrix([[-2, -1], [4, 1]]):
b := Vector([x, x^2]):
Determinant(A);

                    > 2

X := LinearSolve(A, b);

                    > X := Vector[column](2, [1/2*x^2 + 1/2*x, -x^2 -
                        2*x])

verify(X[1]*(-2*y) - X[2]*y, x*y, equal);

                    > true
```

```
verify(X[1]*(-8*n*y + 4*n*(n - 1)) + X[2]*(-4*n*y + n*(n - 1)),
    4*n*x*y + n*(n - 1)*x^2, equal);
```

```
> true
```

B.1.4 Finding Coefficients for r^{34} (Maple)

This section presents a portion of the proof of Theorem 2.4.3 written in Maple. The following code shows that the lower rows of the matrix D can be written as a linear combination of r^{12}_{-2}, r^{12}_{-1}, r^{34}_{0}, and r^{34}_{1} by finding a necessary condition on the coefficients, and verifying that they work in general.

```
A := Matrix(4, 5, [[n - 1, 1/4*n - 1/4, 0, 1/4*(n - 1)/n, 1/4*(n
    - 1)*x^2/n], [n - 3, 1/4*n - 5/4, -1/n, 1/4*(n - 1)/n, (x - 1
    + 1/4*(n - 1)*x^2)/n], [-2, -1, -1, 0, x - 1], [-4, -2, -4,
    -2, 2*x - 4]]):
B := rref(A):
X := Column(B, 5);
```

```
> Vector[column](4, [1/2*x*(x - 1)/n, -x*(x - 1)/
    n, -x + 1, x])
```

```
verify(X[1]*(-2*y) - X[2]*y - X[3]*y^2 + X[4]*(-y^2 + y), x*y - y
    ^2, equal);
```

```
> true
```

```
verify(X[1]*(-2*y + n - 1) + X[2]*(-y + (n - 1)/4) - X[3]*y^2/n +
    X[4]*(y - y^2 + (n - 1)/4)/n, (x*y - y^2 + ((n - 1)*x^2)/4)/n
    , equal);
```

```
> true
```

B.2 ILP Implementation (MAGMA)

This section presents an implementation in MAGMA of the ILP as described in Section 3.1, as well as code which tests whether any subspaces of the row spaces of any generalized Hadamard matrix can be used to form KS pairs via the methods outlined in Chapter 2.

B.2.1 Arbitrary Vectors (MAGMA)

The following is a MAGMA procedure which takes in any finite set of vectors belonging to the same vector space, computes all maximal orthogonal sets which contain the vectors,

implements the ILP as described in Section 3.1, and outputs whether or not they form a KS pair.

```
ILP := procedure(V:B:=[],C:=[])
  //V is any finite set of vectors belonging to the same vector
     space
  //B is any set of bases consisting of vectors in V, default it
     generate all possible
  //C is any non-basis set of sets of pairwise orthogonal
     elements of V, default it to generate all possible

  d:=Degree(Random(V)); //gets dimension of vectors

  if B eq [] then
  B:=Setseq({b: b in Subsets(V,d)| forall{{u,v}:u,v in b|Round
     (1000*Modulus(InnerProduct(u,v))) eq 0 or u eq v}}); //bases
  end if;
  if C eq [] then
  C:=Setseq(&join{{c: c in Subsets(V,k)|forall{{u,v}:u,v in c|
     Round(1000*Modulus(InnerProduct(u,v))) eq 0 or u eq v}}: k
     in [2..(d-1)]}); //non-basis orthsets
  end if;

  n:=#V; //number of vectors
  m:=#B; //number bases
  l:=#C; //number of non-basis orthsets

  V:=Setseq(V);
  R:=RealField();

  lhs:=Matrix(R,m+l,n,[]);
  for i:=1 to m+l do
    for j:=1 to n do
      if i lt m+1 then
        if V[j] in B[i] then
          lhs[i][j]:=1;
        else
          lhs[i][j]:=0;
        end if;
      else
        if V[j] in C[i-m] then
          lhs[i][j]:=1;
        else
          lhs[i][j]:=0;
        end if;
      end if;
    end for;
```

```
  end for;


  rhs:=Matrix(R,m+l,1,[1^^(m+l)]);
  rel:=Matrix(R,m+l,1,[0^^(m),(-1)^^(l)]);
  obj:=Matrix(R,1,n,[0^^(n)]);

  a,b:=MaximalZeroOneSolution(lhs,rel,rhs,obj);

  if b eq 0 then //output the result of the ILP
    printf "Solution Exists\n\n";//"Optimal Solution:\n%o",a;
  elif b eq 1 or b eq 4 then
    printf "Failure";
  elif b eq 2 then
    printf "**Infeasible problem**\n";
    printf "Vectors: %o\n\n",n;
    printf "Bases: %o\n\n%o\n\n Non-Basis Orthogonal Sets: %o\n\n
       %o\n\n",m,B,l,C;

  elif b eq 3 then
    printf "Unbounded problem";
  end if;
end procedure;
```

We can demonstrate the above using the 31 vectors of Conway and Kochen [31], and also show that this set is minimal in the sense that no vector can be removed from the set without losing uncolourability.

```
Q:= [[2,2,2], [1,1,2], [1,0,2], [1,-1,2], [0,-1,2], [0,0,2],
    [0,1,2], [0,2,2] , [-1,-1,2], [-1,0,2], [-1,1,2], [-2,0,2],
    [-2,2,2], [-2,-1,1], [-2,-1,0], [-2,-1,-1], [-2,0,1],
    [-2,0,0], [-2,0,-1], [-2,0,-2], [-2,2,0], [-2,2,-2], [-1,2,1],
     [0,2,1], [-1,2,0], [0,2,0], [2,2,0], [-1,2,-1], [0,2,-1],
    [0,2,-2], [2,2,-2]];
V:={VectorSpace(Rationals(),3)!v:v in Q};
ILP(V);
        >> **Infeasible problem**
        >> Vectors: 31
        >> Bases: 17
           ...
        >> Non-Basis Orthogonal Sets: 20
           ...
for u in Q do ILP(V diff {V!u});end for; //tests whether any
   vector can be removed
        >> Solution Exists
           ...
        >> Solution Exists
```

We also use the above to simplify the 85 vector KS pair created by Cortez and Reyes.

```
Q:= [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 1, 0], [1, 0, 1], [0,
    1, 1], [1, -1, 0], [1, 0, -1], [0, 1, -1], [1, 1, 1], [1, 1,
    -1], [1, -1, 1], [-1, 1, 1], [1, 1, 2], [1, 2, 1], [2, 1, 1],
    [-1, 1, 2], [-1, 2, 1], [-2, 1, 1], [1, -1, 2], [1, -2, 1],
    [2, -1, 1], [1, 1, -2], [1, 2, -1], [2, 1, -1], [1, 2, 4], [1,
     4, 2], [2, 1, 4], [2, 4, 1], [4, 1, 2], [4, 2, 1], [-1, 2,
    4], [-1, 4, 2], [-2, 1, 4], [-2, 4, 1], [-4, 1, 2], [-4, 2,
    1], [1, -2, 4], [1, -4, 2], [2, -1, 4], [2, -4, 1], [4, -1,
    2], [4, -2, 1], [1, 2, -4], [1, 4, -2], [2, 1, -4], [2, 4,
    -1], [4, 1, -2], [4, 2, -1], [2, 2, 5], [2, 5, 2], [5, 2, 2],
    [-2, 2, 5], [-2, 5, 2], [-5, 2, 2], [2, -2, 5], [2, -5, 2],
    [5, -2, 2], [2, 2, -5], [2, 5, -2], [5, 2, -2], [2, 3, 8], [2,
     8, 3], [3, 2, 8], [3, 8, 2], [8, 2, 3], [8, 3, 2], [-2, 3,
    8], [-2, 8, 3], [-3, 2, 8], [-3, 8, 2], [-8, 2, 3], [-8, 3,
    2], [2, -3, 8], [2, -8, 3], [3, -2, 8], [3, -8, 2], [8, -2,
    3], [8, -3, 2], [2, 3, -8], [2, 8, -3], [3, 2, -8], [3, 8,
    -2], [8, 2, -3], [8, 3, -2]];
V:={VectorSpace(Rationals(),3)!v:v in Q};
ILP(V);
        >> **Infeasible problem**
        >> Vectors: 85
        >> Bases: 40
            ...
        >> Non-Basis Orthogonal Sets: 180
            ...
ILP(V diff {V!u: u in {[2, 5, 2], [2, 5, -2], [2, -5, 2], [-2, 5,
    2], [4,2,1], [-4,2,1], [4,-2,1], [4,2,-1], [2,4,1], [-2,4,1],
    [2,-4,1], [2,4,-1], [1,2,4], [-1,2,4], [1,-2,4], [1,2,-4],
   [3,2,8], [-3,2,8], [3,2,-8], [3,-2,8], [8,2,3], [-8,2,3],
   [8,-2,3], [8,2,-3], [2,3,8],[-2,3,8],[2,-3,8],[2,3,-8]}});
        >> **Infeasible problem**
        >> Vectors: 57
        >> Bases: 28
            ...
        >> Non-Basis Orthogonal Sets: 116
            ...
```

B.2.2 Using Generalized Hadamard Matrices (MAGMA)

```
// Example Hadamard matrices

H3:=Matrix(GF(3),6,6,[0,0,0,0,0,0,0,0,1,1,2,2,0,1
    ,0,2,2,1,0,1,2,0,1,2,0,2,2,1,0,1,0,2,1,2,1
     ,0]);
```

```
H5:=Matrix(GF(5),10 ,10 ,[0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1
    ,1 ,2 ,2 ,3 ,3 ,4 ,4 ,0 ,1 ,0 ,3 ,2 ,4 ,1 ,4 ,2 ,3 ,0 ,1 ,3 ,4
     ,3 ,1 ,0 ,2 ,4 ,2 ,0 ,2 ,3 ,0 ,1 ,3 ,4 ,1 ,2 ,4 ,0 ,2 ,4 ,2
    ,0 ,1 ,3 ,4 ,3 ,1 ,0 ,3 ,1 ,2 ,4 ,0 ,4 ,2 ,1 ,3 ,0 ,3 ,2 ,4 ,1
     ,4 ,2 ,3 ,0 ,1 ,0 ,4 ,2 ,1 ,4 ,3 ,1 ,0 ,3 ,2 ,0 ,4 ,4 ,3 ,3
    ,2 ,2 ,1 ,1 ,0]);

//Subspaces code
M:=function(k,q) return &*[ q^i-1 : i in [ 1 .. k ] ];end
    function;

G:=function(n,k,q) return M(n,q)/ ( M(k,q) * M(n-k,q) );end
    function;

Subspaces := function(V,k,q) //returns all the subspaces of V of
    dim k
  n:=Dimension(V);
  S:=Set(V);
  subsp := { sub< V | b > : b in Subsets(S,k) | Dimension( sub<V|
     b> ) eq k };
  //printf "check number of subspaces ... %o\n", G(n,k,q) eq #
     subsp;
  return subsp;
end function;


// main bit

diophantineSol := function(P,m)
  R:=RealField();
  N:=Setseq({#p:p in P});
  N;
  lhs:=Matrix(R,2*#N+1 ,#N,[]);
  for j:= 1 to #N do
    lhs[1][j]:= N[j];
    lhs[j+1][j]:= 1;
    lhs[j+#N+1][j]:=1;
  end for;

  rhs:=Matrix(R,2*#N+1 ,1 ,[m,m^^#N,0^^#N]);
  rel:=Matrix(R,2*#N+1 ,1 ,[0 ,(-1)^^#N,1^^#N]);
  obj:=Matrix(R,1 ,#N,[0^^#N]);

  a,b:=MaximalIntegerSolution(lhs,rel,rhs,obj);

  if b eq 2 then
    return false;
```

```
  else
    print "Diophantine Equation has solution";
    return true;
  end if;
end function;


toReal := function(B,p,d)
// function used to check if KS bases are orthogonal in C
// takes as input: set of vectors, characteristic, and length of
    vectors
// returns multiset of inner products of the vectors

  C<i>:= ComplexField();
  V:=VectorSpace(C,d);
  w:=RootOfUnity(p);

  Br:=SetToSequence({[C!w^(Integers()!k):k in Eltseq(B[i])]:i in
      [1..#B]});

  A:={* Round(1000*Modulus(InnerProduct(V!Br[i],V!Br[j]))) : i,j
      in [1..#Br] | i lt j *};
  return A;

end function;


ssLP:= procedure(H: L:=[],OrthSets:=0)
// Main function for generating vectors/bases, and formulating/
    solving the ILP using magma
// H is any Hadamard matrix written in Z_p
// L parameter is a either list of subspaces to consider, or
    which sizes of subspaces to consider, defualt is to use only
    the full space
// OrthSets determines the maximum size of non-basis orthogonal
    sets the ILP will consider

  r:=NumberOfRows(H);
  c:=NumberOfColumns(H);
  B0:=[H[i]:i in [1..r]];
  V:=sub<RowSpace(H)|B0>;
  q:=Characteristic(BaseRing(H));

  if Type(L) eq SeqEnum and L eq [] then
  SS:={{V}};

  else
```

```
  if ExtendedType(L) eq SeqEnum[ModTupFld] or ExtendedType(L)
     eq SeqEnum[ModTupRng] then
    SS:={L};
  else
    SS := {Subspaces(V,k,q):k in L}; //generate all subspaces
  end if;
end if;

SSl := SetToSequence(SS);

for SSk in SSl do //outer loop is over dim of subspace
  for s in SSk do
    Sl:=SetToSequence(Set(s)); //set of all vectors in subspace

    n:=#Sl;

    BB:={[Sl[i]+b:b in B0]:i in [1..n]}; //creates bases

    BBl:=SetToSequence(BB);

    assert forall{b: b in BB| toReal(b,q,c) eq {* 0^^(Binomial(
       r,2)) *}}; //checks all bases are orthogonal

    T:={v: v in V |exists{b:b in BB | v in b}}; //finds set of
       vectors actually used in the bases

    Tl:=SetToSequence(T);

    if OrthSets gt 0 then
      O:={U:U in &join{Subsets(T,i):i in [OrthSets..OrthSets]}|
          toReal(Setseq(U),q,r) eq {*0^^Binomial(#U,2)*} and not
           exists{v:v in T diff U|#U+1 le OrthSets and toReal(
          Setseq(U join {v}),q,r) eq {*0^^Binomial(#U+1 ,2)*}}
          and not exists{b:b in BB| U subset b}};

    else
      O:={};
    end if;

    Ol:=SetToSequence(O);

    m:=#BBl; //number bases
    n:=#Tl; //number of vectors
    l:=#Ol;

    ONl:={{Index(Tl,v):v in Ol[j]}:j in [1..l]};
```

```
BN:={[Index(Tl,v):v in BBl[j]]:j in [1..m]}; //converts
    vectors to distinct integers
BNl:=SetToSequence(BN);

printf "Number of bases: %o\ZZ_{>0}umber of vectors/
    codewords: %o\ZZ_{>0}umber of Orth Pairs: %o\n",m,n,l;
print s;
L:=[{H[i]+u:u in s}:i in [1..c]];

P:={{i:i in [1..c]| L[i] eq L[j]}: j in [1..c]};
printf "slices: %o\n\n",P;

if OrthSets eq 0 and exists{i:i in [1..#P]|#Setseq(P)[i] in
     Divisors(m)} then //checks easy condition for
    colourability
  runILP:=false;
  b:=0;
elif OrthSets eq 0 and not diophantineSol(P,m) then
  runILP:=false;
  b:=2;
else
  runILP:=true;
end if;

//ILP
if runILP then
  R:=RealField();

  lhs:=Matrix(R,m+l,n,[]);
  for i:=1 to m+l do
    for j:=1 to n do
      if i lt m+1 then
        if Tl[j] in BBl[i] then
          lhs[i][j]:=1;
        else
          lhs[i][j]:=0;
        end if;
      else
        if Tl[j] in Ol[i-m] then
          lhs[i][j]:=1;
        else
          lhs[i][j]:=0;
        end if;
      end if;
    end for;

  end for;
```

```
        rhs:=Matrix(R,m+l,1  ,[1^^(m+l)]);
        rel:=Matrix(R,m+l,1  ,[0^^(m),(-1)^^(l)]);
        obj:=Matrix(R,1  ,n,[0^^(n)]);

        a,b:=MaximalZeroOneSolution(lhs,rel,rhs,obj);
      end if;

      if b eq 0 then //output the result of the ILP
        printf "Solution Exists\n\n";//"Optimal Solution:\n%o",a;
      //printf "Bases: %o\n\n Orthogonal Pairs: %o\n\n",BNl,ONl;
      elif b eq 1 or b eq 4 then
        printf "Failure\n\n";
        printf "Bases: %o\n\n Orthogonal Pairs: %o\n\n",BNl,ONl;
      elif b eq 2 then
        printf "**Infeasible problem**\n";
        printf "Bases: %o\n\n Orthogonal Pairs: %o\n\n",BNl,ONl;

      if #{#p:p in P} gt 1 then print "flag";end if; //flags if
         any interesting cases arise

      elif b eq 3 then
        printf "Unbounded problem";
      end if;

    end for;
  end for;
end procedure;

f:= map<GF(p)->ComplexField()|x:->RootOfUnity(p)^(Integers()!x)>;
    //code to change FF matrix to Complex
g:= map<ComplexField()-> GF(p)|x:-> GF(p)!Truncate((p/(2*Pi(
   RealField()))))*Imaginary(Log(x)))>;
```

B.3 Graph Theoretical Implementation (MAGMA)

The following code is an implementation of the graph theoretical procedure for solving instances of the KS decision problem, which was presented in Section 3.2.

```
GraphKS:= procedure(V:B:=[],C:=[])
  //V is any finite set of vectors belonging to the same vector
     space
  //B is any set of bases consisting of vectors in V, default it
     generate all possible
  //C is any non-basis set of sets of pairwise orthogonal
     elements of V, default it to generate all possible
```

```
  d:=Degree(Random(V)); //gets dimension of vectors

  if B ne [] then
    B:=Setseq({b: b in Subsets(V,d)| forall{{u,v}:u,v in b|Round
        (1000*Modulus(InnerProduct(u,v))) eq 0 or u eq v}}); //
        bases
  end if;

  if C ne [] then
    C:=Setseq(&join{{c: c in Subsets(V,k)|forall{{u,v}:u,v in c|
        Round(1000*Modulus(InnerProduct(u,v))) eq 0 or u eq v}}: k
         in [2..(d-1)]}); //non-basis orthsets
  end if;

  n:=#V; //number of vectors
  m:=#B; //number bases
  l:=#C; //number of non-basis orthsets

  V:=Setseq(V);

  //convert vectors to their indices in V
  Bn := {{Index(V,u):u in b}: b in B};
  Cn := {{Index(V,u):u in c}: c in C};

  E:= {{u,v}:u,v in [1..n]| exists{s:s in Bn join Cn| {u,v}
     subset s and u ne v}}; //Defines Edge set of Orthogonality
     graph

  G:=Graph<n|E>; //Orthogonality Graph

  IS := AllCliques(Complement(G)); //Set of all Maximal
     Independent Sets

  if exists{M: M in IS| forall{{v,b,c}:v in M,b in Bn,c in Cn|{
     Index(G,v)} meet b eq 1 and {Index(G,v)} meet c le 1}} then
    print "Solution Exists";
  else
    printf "**KS pair**\n\nVectors: %o\n\n%o\n\nBases: %o\n\n%o\n
       \nOrthsets: %o\n\n%o\n\n",n,V,m,B,l,C;
  end if;
end procedure;
```

B.4 Matrix Groups in $\mathrm{GL}_3(\mathbb{Z})$ (MAGMA)

This section catalogs the small matrix groups presented in the paper by [36]. It also includes a procedure for verifying that these groups are as purported in that paper. The group `O24W11` is presented without the typo that was identified in Section 3.3.2.

```
Q:=Rationals();

//Order 2
O2W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O2W2:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,1 ,0 ,0 ,0 ,1]>;
O2W3:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0]>;
O2W4:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,-1 ,0]>;
O2W5:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O2:=[[O2W1 ,O2W2,O2W3,O2W4,O2W5]];
G2:=[<2,1>];

//Order 3
O3W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1]>;
O3W2:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0]>;

O3:=[[O3W1 ,O3W2]];
G3:=[<3,1>];

//Order 4
O4W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,0]>;
O4W2:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,-1 ,0]>;
O4W3:=MatrixGroup<3,Q|[1 ,0 ,1 ,0 ,0 ,-1 ,0 ,1 ,0]>;
O4W4:=MatrixGroup<3,Q|[-1 ,0 ,-1 ,0 ,0 ,1 ,0 ,-1 ,0]>;

O4W5:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,-1]>;
O4W6:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,1]>;
O4W7:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[1 ,0 ,0 ,0
    ,1 ,0 ,0 ,0 ,-1]>;
O4W8:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0]>;
O4W9:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O4W10:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,1 ,0 ,0 ,0 ,1],[1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O4W11:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,-1]>;
O4W12:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,1
    ,0 ,-1 ,-1 ,-1 ,0]>;
```

```
O4W13:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[1 ,0 ,0 ,-1
    ,0 ,1 ,1 ,1 ,0]>;
O4W14:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,1 ,-1 ,0
     ,0 ,-1 ,0 ,-1 ,0]>;
O4W15:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[1 ,-1 ,1 ,0
    ,0 ,1 ,0 ,1 ,0]>;

O4:=[[O4W1 ,O4W2,O4W3,O4W4],[O4W5,O4W6,O4W7,O4W8,O4W9,O4W10 ,
   O4W11 ,O4W12,O4W13,O4W14,O4W15]];
G4:=[<4,1>,<4,2>];


//Order 6
O6W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,1]>;
O6W2:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,-1 ,-1]>;
O6W3:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,-1 ,1]>;
O6W4:=MatrixGroup<3,Q|[0 ,-1 ,0 ,0 ,0 ,-1 ,-1 ,0 ,0]>;

O6W5:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0]>;
O6W6:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1],[1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O6W7:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1],[-1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O6W8:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1],[1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0]>;
O6W9:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[0 ,0 ,-1 ,0
    ,-1 ,0 ,-1 ,0 ,0]>;
O6W10:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[0 ,0 ,1 ,0 ,1
     ,0 ,1 ,0 ,0]>;

O6:=[[O6W1 ,O6W2,O6W3,O6W4],[O6W5,O6W6,O6W7,O6W8,O6W9,O6W10]];
G6:=[<6,2>,<6,1>];

//Order 8
O8W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,0],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,-1]>;
O8W2:=MatrixGroup<3,Q|[1 ,0 ,1 ,0 ,0 ,-1 ,0 ,1 ,0],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,-1]>;
O8W3:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0
    ,-1 ,0 ,0 ,0 ,1],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O8W4:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O8W5:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,1
    ,0 ,-1 ,-1 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O8W6:=MatrixGroup<3,Q|[-1 ,0 ,0 ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,1 ,-1 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
```

```
O8W7:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,−1 ,0 ,1 ,0],[−1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O8W8:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,−1 ,0 ,1 ,0],[1 ,0 ,0 ,0 ,0
     ,−1 ,0 ,−1 ,0]>;
O8W9:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,0],[−1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O8W10:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,0],[1 ,0 ,0 ,0
    ,0 ,−1 ,0 ,−1 ,0]>;
O8W11:=MatrixGroup<3,Q|[1 ,0 ,1 ,0 ,0 ,−1 ,0 ,1 ,0],[−1 ,0 ,0 ,0
    ,0 ,−1 ,0 ,−1 ,0]>;
O8W12:=MatrixGroup<3,Q|[1 ,0 ,1 ,0 ,0 ,−1 ,0 ,1 ,0],[1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O8W13:=MatrixGroup<3,Q|[−1 ,0 ,−1 ,0 ,0 ,1 ,0 ,−1 ,0],[−1 ,0 ,0
    ,0 ,0 ,−1 ,0 ,−1 ,0]>;
O8W14:=MatrixGroup<3,Q|[−1 ,0 ,−1 ,0 ,0 ,1 ,0 ,−1 ,0],[1 ,0 ,0 ,0
     ,0 ,1 ,0 ,1 ,0]>;

O8:=[[O8W1 ,O8W2],[O8W3,O8W4,O8W5,O8W6],[O8W7,O8W8,O8W9,O8W10 ,
   O8W11 ,O8W12,O8W13,O8W14]];
G8:=[<8,2>,<8,5>,<8,3>];

//Order 12
O12W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,−1 ,0 ,1 ,1],[−1 ,0 ,0 ,0
    ,−1 ,0 ,0 ,0 ,−1]>;

O12W2:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,−1 ,0 ,1 ,1],[−1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0]>;
O12W3:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,−1 ,0 ,1 ,1],[1 ,0 ,0 ,0
    ,0 ,−1 ,0 ,−1 ,0]>;
O12W4:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,−1],[−1 ,0 ,0
    ,0 ,0 ,1 ,0 ,1 ,0]>;
O12W5:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,−1],[1 ,0 ,0 ,0
     ,0 ,−1 ,0 ,−1 ,0]>;
O12W6:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,1],[−1 ,0 ,0 ,0
     ,0 ,−1 ,0 ,−1 ,0]>;
O12W7:=MatrixGroup<3,Q|[−1 ,0 ,0 ,0 ,0 ,1 ,0 ,−1 ,1],[−1 ,0 ,0 ,0
     ,0 ,1 ,0 ,1 ,0]>;
O12W8:=MatrixGroup<3,Q|[0 ,−1 ,0 ,0 ,0 ,−1 ,−1 ,0 ,0],[0 ,0 ,−1
    ,0 ,−1 ,0 ,−1 ,0 ,0]>;

O12W9:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[−1 ,0 ,0 ,0
    ,1 ,0 ,0 ,0 ,−1]>;
O12W10:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[0 ,−1 ,1 ,0
    ,−1 ,0 ,1 ,−1 ,0]>;
O12W11:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[−1 ,−1 ,−1
    ,0 ,0 ,1 ,0 ,1 ,0]>;
```

```
O12:=[[O12W1],[O12W2,O12W3,O12W4,O12W5,O12W6,O12W7,O12W8],[O12W9,
    O12W10 ,O12W11]];
G12:=[<12,5>,<12,4>,<12,3>];

//Order 16
O16W1:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,0],[-1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O16W2:=MatrixGroup<3,Q|[1 ,0 ,1 ,0 ,0 ,-1 ,0 ,1 ,0],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O16:=[[O16W1 ,O16W2]];
G16:=[<16,11>];

//Order 24
O24W1:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[-1 ,0 ,0 ,0
    ,1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W2:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[0 ,-1 ,1 ,0
    ,-1 ,0 ,1 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W3:=MatrixGroup<3,Q|[0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0],[-1 ,-1 ,-1 ,0
    ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O24W4:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,1],[-1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W5:=MatrixGroup<3,Q|[1 ,0 ,0 ,0 ,0 ,-1 ,0 ,1 ,-1],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O24W6:=MatrixGroup<3,Q|[0 ,0 ,1 ,0 ,1 ,0 ,-1 ,0 ,0],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W7:=MatrixGroup<3,Q|[0 ,0 ,-1 ,0 ,-1 ,0 ,1 ,0 ,0],[1 ,0 ,0 ,0
    ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W8:=MatrixGroup<3,Q|[0 ,-1 ,0 ,1 ,1 ,1 ,-1 ,0 ,0],[-1 ,-1 ,0
    ,0 ,1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W9:=MatrixGroup<3,Q|[0 ,1 ,0 ,-1 ,-1 ,-1 ,1 ,0 ,0],[1 ,1 ,0 ,0
    ,-1 ,0 ,0 ,0 ,1],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W10:=MatrixGroup<3,Q|[1 ,1 ,0 ,-2,-1 ,-1 ,0 ,0 ,1],[-1 ,-1 ,-1
    ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O24W11:=MatrixGroup<3,Q|[-1 ,-1 ,0 ,2,1 ,1 ,0 ,0 ,-1],[1 ,1 ,1 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O24:=[[O24W1 ,O24W2,O24W3],[O24W4,O24W5],[O24W6,O24W7,O24W8,O24W9
    ,O24W10 ,O24W11]];
G24:=[<24,13>,<24,14>,<24,12>];

//Order 48
O48W1:=MatrixGroup<3,Q|[0 ,0 ,1 ,0 ,1 ,0 ,-1 ,0 ,0],[-1 ,0 ,0 ,0
    ,0 ,-1 ,0 ,-1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
```

```
O48W2:=MatrixGroup<3,Q|[0 ,-1 ,0 ,1 ,1 ,1 ,-1 ,0 ,0],[-1 ,-1 ,0
    ,0 ,1 ,0 ,0 ,0 ,-1],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;
O48W3:=MatrixGroup<3,Q|[1 ,1 ,0 ,-2,-1 ,-1 ,0 ,0 ,1],[-1 ,-1 ,-1
    ,0 ,0 ,1 ,0 ,1 ,0],[-1 ,0 ,0 ,0 ,-1 ,0 ,0 ,0 ,-1]>;

O48:=[[O48W1 ,O48W2,O48W3]];

checkGroups:= procedure()
  //Checks that the groups are all as the paper claims
  for i:=1 to #O2 do for G in O2[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G2[i],G2[i];end for;end
      for;
  for i:=1 to #O3 do for G in O3[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G3[i],G3[i];end for;end
      for;
  for i:=1 to #O4 do for G in O4[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G4[i],G4[i];end for;end
      for;
  for i:=1 to #O6 do for G in O6[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G6[i],G6[i];end for;end
      for;
  for i:=1 to #O8 do for G in O8[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G8[i],G8[i];end for;end
      for;
  for i:=1 to #O12 do for G in O12[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G12[i],G12[i];end for;
     end for;
  for i:=1 to #O16 do for G in O16[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G16[i],G16[i];end for;
     end for;
  for i:=1 to #O24 do for G in O24[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: %o, %o\n\n"),G,
     IdentifyGroup(G),IdentifyGroup(G) eq G24[i],G24[i];end for;
     end for;
  for i:=1 to #O48 do for G in O48[i] do printf("Group: %o\n\
     nIdentity: %o\n\nSame as Paper Claims: N/A\n\n"),G,
     IdentifyGroup(G);end for;end for;
end procedure;
```

B.5 Testing the Method of Section 3.3.2 (MAGMA)

The following code is an implementation of the method described in Section 3.3.2.

```
Q:=Rationals();
V:=VectorSpace(Q,3);

f:=function(u) return Basis(sub<V|u>)[1];end function;

A:={f(V![a,b,c]):a,b,c in [-2..2]|[a,b,c] ne [0,0,0]}; // nonzero
    non-collinear lattice points

Action:= procedure(Groups:size:=[23..31],subsets:=0,rev:=false)
  // Groups is either a single matrix group, or a sequence of
     matrix groups
  // size refers to the target size of KS set to be generated,
     default is [23..31] (also accepts single integer)
  // rev decides how the unions of orbits are iterated over,
     default is to generate all of them, set to true for
     revolving door
  // subsets determines max number of orbits to combine when
     looking for solution, default (0) means check all subsets

  if Type(Groups) ne SeqEnum then Groups:=[Groups];end if;
  if Type(size) ne SeqEnum then size:=[size]; end if;
  for G in Groups do
    print G,IdentifyGroup(G);

    O:={f(a*Transpose(M)):M in G}:a in A};
    Orbits:=[b:b in O| b subset A];
    printf "Number of Orbits: %o\n\n", #Orbits;
    N:=1;

    while N in [1..2^#Orbits-1] do
      if not rev then
        N:=0;//turn off while loop
        if subsets eq 0 then
          ssOrbits:=Subsets({i: i in [1..#Orbits]});
        else
          ssOrbits:=&join{Subsets({i: i in [1..#Orbits]},i): i in
              [1..subsets]};
        end if;

        U:={&join{Orbits[i]:i in T}:T in ssOrbits|#(&join{Orbits[
            i]:i in T}) gt 22}; //look for all unions of orbits of
             specifized size(s)
      else
```

```
    I:=Intseq(N,2);
    T:=&join{Orbits[j]: j in [1..#I]|I[j] eq 1};
    if #T gt 22 then U:={T}; else U:={};end if;
    N:=N+1;
  end if;
  for u in U do
    O3:={{v,w,t}:v,w,t in u diff {V![0,0,0]}|{DotProduct(v,w)
        ,DotProduct(v,t),DotProduct(w,t)} eq {0}}; //
        Orthogonal Triples
    O2:={{v,w}:v,w in u diff {V![0,0,0]} |DotProduct(v,w) eq
        0 and not exists{C:C in O3| {v,w} subset C}}; //
        Orthogonal pairs

    m:=#O3;
    l:=#O2;

    if m ne 0 then //if there are no bases, no vector needs
        to be coloured

      Tl:=[v:v in u|exists{b:b in O3 join O2|v in b}]; //find
          vectors actually used in the orth sets
      n:=#Tl;

      if n in size then //check that number of vectors is at
          least that of the proved lower bound
        R:=RealField();

        lhs:=Matrix(R,m+l,n,[]);
        for i:=1 to m+l do
          for j:=1 to n do
            if i lt m+1 then
              if Tl[j] in Setseq(O3)[i] then
                lhs[i][j]:=1;
              else
                lhs[i][j]:=0;
              end if;
            else
              if Tl[j] in Setseq(O2)[i-m] then
                lhs[i][j]:=1;
              else
                lhs[i][j]:=0;
              end if;
            end if;
          end for;

        end for;
```

```
            rhs:=Matrix(R,m+l,1,[1^^(m+l)]);
            rel:=Matrix(R,m+l,1,[0^^(m),(-1)^^(l)]);
            obj:=Matrix(R,1,n,[0^^(n)]);

            a,b:=MaximalZeroOneSolution(lhs,rel,rhs,obj);
          else
            b:=0;
          end if;

          if b eq 0 then //output the result of the ILP
            printf "Solution Exists\n\n";
          elif b eq 1 or b eq 4 then
            printf "Failure";
          elif b eq 2 then
            printf "**Infeasible problem**\n";
            printf("Number of Vectors: %o\n\n"),n;
            printf "Vectors: %o\n\nBases: %o\n\n Orthogonal Pairs
                : %o\n\n", u, Setseq(O3), Setseq(O2);

          elif b eq 3 then
            printf "Unbounded problem";
          end if;

        end if;
      end for;
    end while;
  end for;
end procedure;
```

We also test that the KS pair found by Conway and Kochen has a nontrivial symmetry group.

```
Q:=Rationals();
V:=VectorSpace(Q,3);
Q37 := {V!u: u in [[1,1,1], [1,1,2], [1,0,2], [1,-1,2], [0,1,-2],
     [0,0,1], [0,1,2], [0,1,1], [1,1,-2], [1,0,-2], [-1,1,2],
    [1,0,-1], [-1,1,1], [2,1,-1], [2,1,0], [2,1,1], [2,0,-1],
    [1,0,0], [2,0,1], [1,0,1], [1,-1,0], [1,-1,1], [-1,2,1],
    [0,2,1], [1,-2,0], [0,1,0], [1,1,0], [1,-2,1], [0,2,-1],
    [0,1,-1], [1,1,-1], [2,-1,0], [1,2,0], [-2,1,1], [1,2,-1],
    [1,2,1], [2,-1,1]]};

//Conway-Kochen Set
CK:= Q37 diff {V!u: u in [[2,-1,0], [1,2,0], [-2,1,1], [1,2,-1],
    [1,2,1], [2,-1,1]]};

G:=MatrixGroup<3,Q|[0,-1,0,1,0,0,0,0,-1],[1,0,0,0,1,0,0,0,-1]>;
```

```
for M in G do {sub<V|v*Transpose(M)>:v in CK} eq {sub<V|v>:v in
    CK};end for;
```

B.6 Paley Hadamard Matrices (MAGMA)

The following MAGMA code gives an explicit construction for the Paley Hadamard matrices (Theorem 1.4.8).

```
X := function(K,q)
     sq:={x^2:x in K} diff {K!0};
     if q eq K!0 then
         return 0;
     elif q in sq then
         return 1;
     else
         return -1;
     end if;
end function;
Paley := function(q);
     if IsOdd(q) and IsPrimePower(q) then
         printf "\nq=%o␣␣␣␣␣␣␣q␣mod␣8=%o\n",q, q mod 8;
             Fq:=GF(q);
         Fql:=SetToSequence(Set(Fq));

         Q:=Matrix(Rationals(),q,q,[]);
         for i:=1 to q do
             for j:=1 to q do
                 Q[i,j]:=X(Fq,Fql[j]-Fql[i]);
             end for;
         end for;
         C:= VerticalJoin(Matrix(Rationals(),1,q+1,[0,1^^(q)]),
            HorizontalJoin(Matrix(Rationals(),q,1,[1^^q]),Q));
         I := Matrix(Rationals(),q+1,q+1,[]);
         for i:=1 to q+1 do
             for j:=1 to q+1 do
                 if i eq j then
                     I[i,j]:=1;
                 else
                     I[i,j]:=0;
                 end if;
             end for;
         end for;
         if (q mod 4) eq 3 then
             for i:=2 to q+1 do
                 C[i,1]:=-1;
             end for;
```

```
            H:= I+C;
        elif (q mod 4) eq 1 then
                H1:=I+C;
            H2:=-I+C;
            H3:=H2;
            H4:=-I-C;

            H:= VerticalJoin(HorizontalJoin(H1,H2),HorizontalJoin
                (H3,H4));
        end if;
        assert IsHadamard(H);
        return H;
    end if;
    return 0;
end function;
```

B.7 KS Pairs from Cortez–Reyes Construction (MAGMA)

The following is MAGMA [7] code which computes the well-signed vectors within $Q(N, e)$, as defined in Section 3.3.1, then determines whether they may be used to form a KS pair.

```
q:=function(v)
  return &+[k^2:k in Eltseq(v)];
end function;

wsVecs:= function(k)
  //returns the list of well-signed integer vectors with q(v)=k
  V:=VectorSpace(Rationals(),3);

  return {V![a,b,c]:a,b,c in [-Floor(Sqrt(k))..Floor(Sqrt(k))]|q
     ([a,b,c]) eq k and (((a eq 0 and b eq 0 and c gt 0) or (a eq
      0 and b gt 0 and c eq 0) or (a gt 0 and b eq 0 and c eq 0))
      or ((a eq 0 and b gt 0 and c ne 0) or (b eq 0 and a gt 0
     and c ne 0) or (c eq 0 and a gt 0 and b ne 0)) or ((a*b*c ne
      0) and ((a gt 0 and b gt 0 ) or (a gt 0  and c gt 0) or ( b
      gt 0 and c gt 0))))};
end function;

vecLP:= procedure(z,p)
  //N is a square-free integer
  //p is the maximum exponent
  printf "N=%o\nExp=%o\n",z,p;
  Q:= [wsVecs(d):d in Divisors(z^p)];
  Sl:= SetToSequence(&join Q);
```

```
O3 := SetToSequence({{v,w,u}:v,w,u in Sl|DotProduct(v,w) eq 0
    and DotProduct(w,u) eq 0 and DotProduct(v,u) eq 0}); //
    orthogonal triples
O2 := SetToSequence({{v,w}:v,w in Sl| DotProduct(v,w) eq 0  and
     not exists{b:b in O3|v in b and w in b}}); //orthogonal
    pairs
n:=#Sl;
m:=#O3; //number bases
l:=#O2;
O2Nl:=SetToSequence({{Index(Sl,v):v in O2[j]}:j in [1..l]});
O3N:={{Index(Sl,v):v in O3[j]}:j in [1..m]}; //converts vectors
     to distinct integers
O3Nl:=SetToSequence(O3N);
printf "Number of bases: %o\nNumber of vectors/codewords: %o\
    nNumber of Orth Pairs: %o\n",m,n,l;

//ILP
R:=RealField();
lhs:=Matrix(R,m+l,n,[]);
  for i:=1 to m+l do
    for j:=1 to n do
      if i lt m+1 then
        if Sl[j] in O3[i] then
          lhs[i][j]:=1;
        else
          lhs[i][j]:=0;
        end if;
      else
        if Sl[j] in O2[i-m] then
          lhs[i][j]:=1;
        else
  lhs[i][j]:=0;
   end if;
end if;
    end for;
  end for;
rhs:=Matrix(R,m+l,1,[1^^(m+l)]);
rel:=Matrix(R,m+l,1,[0^^(m),(-1)^^(l)]);
obj:=Matrix(R,1,n,[0^^(n)]);
a,b:=MaximalZeroOneSolution(lhs,rel,rhs,obj);

if b eq 0 then //output the result of the ILP
  printf "Solution Exists\n\n";//"Optimal Solution:\n%o",a;
elif b eq 1 or b eq 4 then
  printf "Failure";
elif b eq 2 then
  printf "**Infeasible problem**\n";
```

```
    printf "Bases: %o\n\n Orthogonal Pairs: %o\n\n",O3Nl,O2Nl;
  elif b eq 3 then
    printf "Unbounded problem";
  end if;
end procedure;

main:=procedure(e)
  for i:=1 to 10^3 do
    if IsSquarefree(i) and i mod 6 eq 0 then
      vecLP(i,e);
    end if;
  end for;
end procedure;
```

The following code gives explicitly the vectors forming the new KS pairs presented in Section 3.3.1, as well as the existing pairs of Cortez–Reyes and Conway–Kochen.

```
V:=VectorSpace(Rationals(),3);

Q37 := {V!u: u in [[1,1,1], [1,1,2], [1,0,2], [1,-1,2], [0,1,-2],
    [0,0,1], [0,1,2], [0,1,1], [1,1,-2], [1,0,-2], [-1,1,2],
   [1,0,-1], [-1,1,1], [2,1,-1], [2,1,0], [2,1,1], [2,0,-1],
   [1,0,0], [2,0,1], [1,0,1], [1,-1,0], [1,-1,1], [-1,2,1],
   [0,2,1], [1,-2,0], [0,1,0], [1,1,0], [1,-2,1], [0,2,-1],
   [0,1,-1], [1,1,-1], [2,-1,0], [1,2,0], [-2,1,1], [1,2,-1],
   [1,2,1], [2,-1,1]]};

//Conway-Kochen Set
CK:= Q37 diff {V!u: u in [[2,-1,0], [1,2,0], [-2,1,1], [1,2,-1],
   [1,2,1], [2,-1,1]]};


//Other size 31 sets
Q31a:=Q37 diff {V!u: u in [[1,0,2], [2,0,-1], [1,-1,2], [-2,1,1],
    [2,1,-1], [1,1,2]]};
Q31b:=Q37 diff {V!u: u in [[1,0,-2], [2,0,1], [-1,1,2], [2,-1,1],
    [2,1,1], [1,1,-2]]};
Q31c:=Q37 diff {V!u: u in [[0,1,2], [0,2,-1], [1,-2,1], [1,2,-1],
    [-1,1,2], [1,1,2]]};
Q31d:=Q37 diff {V!u: u in [[2,1,0], [1,-2,0], [2,1,-1], [-1,2,1],
    [1,-2,1], [2,1,1]]};
Q31e:=Q37 diff {V!u: u in [[0,1,-2], [0,2,1], [1,-1,2], [-1,2,1],
    [1,1,-2], [1,2,1]]};

//Cortez-Reyes set
Q85:={V!u:u in [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 1, 0], [1,
   0, 1], [0, 1, 1], [1, -1, 0], [1, 0, -1], [0, 1, -1], [1, 1,
   1], [1, 1, -1], [1, -1, 1], [-1, 1, 1], [1, 1, 2], [1, 2, 1],
```

```
[2, 1, 1], [-1, 1, 2], [-1, 2, 1], [-2, 1, 1], [1, -1, 2], [1,
 -2, 1], [2, -1, 1], [1, 1, -2], [1, 2, -1], [2, 1, -1], [1,
2, 4], [1, 4, 2], [2, 1, 4], [2, 4, 1], [4, 1, 2], [4, 2, 1],
[-1, 2, 4], [-1, 4, 2], [-2, 1, 4], [-2, 4, 1], [-4, 1, 2],
[-4, 2, 1], [1, -2, 4], [1, -4, 2], [2, -1, 4], [2, -4, 1],
[4, -1, 2], [4, -2, 1], [1, 2, -4], [1, 4, -2], [2, 1, -4],
[2, 4, -1], [4, 1, -2], [4, 2, -1], [2, 2, 5], [2, 5, 2], [5,
2, 2], [-2, 2, 5], [-2, 5, 2], [-5, 2, 2], [2, -2, 5], [2, -5,
 2], [5, -2, 2], [2, 2, -5], [2, 5, -2], [5, 2, -2], [2, 3,
8], [2, 8, 3], [3, 2, 8], [3, 8, 2], [8, 2, 3], [8, 3, 2],
[-2, 3, 8], [-2, 8, 3], [-3, 2, 8], [-3, 8, 2], [-8, 2, 3],
[-8, 3, 2], [2, -3, 8], [2, -8, 3], [3, -2, 8], [3, -8, 2],
[8, -2, 3], [8, -3, 2], [2, 3, -8], [2, 8, -3], [3, 2, -8],
[3, 8, -2], [8, 2, -3], [8, 3, -2]]};

//minimal subset of S(462)
Q57:={V!u:u in [[-8,3,2], [8,3,2], [-1,4,2], [1,4,2], [1,4,-2],
    [8,3,-2], [5,2,-2], [1,2,-1], [3,-8,2], [2,2,-5], [2,-8,3],
    [4,1,-2], [2,1,-1], [1,1,-1], [0,1,-1], [1,-1,0], [2,-1,1],
    [4,-1,2], [1,1,-2], [-2,8,3], [1,-1,1], [2,1,-4], [1,-1,2],
    [1,0,0], [0,0,1], [2,-1,4], [-3,8,2], [1,0,1], [3,8,2],
    [2,8,3], [1,0,-1], [5,-2,2], [3,8,-2], [2,8,-3], [-2,1,4],
    [1,-2,1], [-1,1,2], [0,1,0], [1,1,0], [-1,1,1], [0,1,1],
    [-2,1,1], [-4,1,2], [1,1,1], [2,1,1], [2,-2,5], [1,1,2],
    [4,1,2], [8,-3,2], [2,1,4], [-2,2,5], [-1,2,1], [1,2,1],
    [-5,2,2], [5,2,2], [2,2,5], [1,-4,2]]};

//minimal subset of S(714)
Q65 := {V!u: u in [[1,-4,5], [4,5,-1], [-1,4,5], [4,-5,1],
    [1,5,-4], [1,4,0], [1,-5,4], [1,4,5], [-1,5,4], [1,4,-5],
    [-4,5,1], [4,5,1], [1,5,4], [2,3,-1], [3,2,-1], [1,2,-1],
    [4,-1,0], [2,1,-1], [3,1,-2], [1,1,-1], [0,1,-1], [1,-1,0],
    [2,-1,1], [1,1,-2], [3,-1,2], [2,1,-3], [1,-1,1], [1,-1,2],
    [4,1,-5], [1,0,0], [2,-1,3], [0,0,1], [4,-1,5], [1,0,1],
    [4,0,1], [4,0,-1], [1,0,4], [1,0,-1], [3,-2,1], [-4,1,5],
    [1,-2,1], [-2,1,3], [-1,1,2], [0,1,0], [1,0,-4], [1,1,0],
    [-1,1,1], [-3,1,2], [0,1,1], [-2,1,1], [1,1,1], [2,1,1],
    [4,1,0], [1,1,2], [3,1,2], [2,1,3], [2,-3,1], [4,1,5],
    [-1,2,1], [1,2,1], [-3,2,1], [3,2,1], [1,-4,0], [-2,3,1],
    [2,3,1]]};
```

B.8 Computing Projectivies

The following is a function which computes the projective symmetries (projectivies) of a set of rational points.

```
Q3:=VectorSpace(Rationals(),3);

Q37 := {Q3!u: u in [[1,1,1], [1,1,2], [1,0,2], [1,-1,2],
    [0,1,-2], [0,0,1], [0,1,2], [0,1,1], [1,1,-2], [1,0,-2],
    [-1,1,2], [1,0,-1], [-1,1,1], [2,1,-1], [2,1,0], [2,1,1],
    [2,0,-1], [1,0,0], [2,0,1], [1,0,1], [1,-1,0], [1,-1,1],
    [-1,2,1], [0,2,1], [1,-2,0], [0,1,0], [1,1,0], [1,-2,1],
    [0,2,-1], [0,1,-1], [1,1,-1], [2,-1,0], [1,2,0], [-2,1,1],
    [1,2,-1], [1,2,1], [2,-1,1]]};

//Conway-Kochen Set
CK:= Q37 diff {Q3|[2,-1,0], [1,2,0], [-2,1,1], [1,2,-1], [1,2,1],
     [2,-1,1]};
Q85:={Q3!u:u in [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 1, 0], [1,
    0, 1], [0, 1, 1], [1, -1, 0], [1, 0, -1], [0, 1, -1], [1, 1,
    1], [1, 1, -1], [1, -1, 1], [-1, 1, 1], [1, 1, 2], [1, 2, 1],
    [2, 1, 1], [-1, 1, 2], [-1, 2, 1], [-2, 1, 1], [1, -1, 2], [1,
     -2, 1], [2, -1, 1], [1, 1, -2], [1, 2, -1], [2, 1, -1], [1,
    2, 4], [1, 4, 2], [2, 1, 4], [2, 4, 1], [4, 1, 2], [4, 2, 1],
    [-1, 2, 4], [-1, 4, 2], [-2, 1, 4], [-2, 4, 1], [-4, 1, 2],
    [-4, 2, 1], [1, -2, 4], [1, -4, 2], [2, -1, 4], [2, -4, 1],
    [4, -1, 2], [4, -2, 1], [1, 2, -4], [1, 4, -2], [2, 1, -4],
    [2, 4, -1], [4, 1, -2], [4, 2, -1], [2, 2, 5], [2, 5, 2], [5,
    2, 2], [-2, 2, 5], [-2, 5, 2], [-5, 2, 2], [2, -2, 5], [2, -5,
     2], [5, -2, 2], [2, 2, -5], [2, 5, -2], [5, 2, -2], [2, 3,
    8], [2, 8, 3], [3, 2, 8], [3, 8, 2], [8, 2, 3], [8, 3, 2],
    [-2, 3, 8], [-2, 8, 3], [-3, 2, 8], [-3, 8, 2], [-8, 2, 3],
    [-8, 3, 2], [2, -3, 8], [2, -8, 3], [3, -2, 8], [3, -8, 2],
    [8, -2, 3], [8, -3, 2], [2, 3, -8], [2, 8, -3], [3, 2, -8],
    [3, 8, -2], [8, 2, -3], [8, 3, -2]]};

L:=function(P,Q)
        a:= Coordinates(VectorSpaceWithBasis([P[i]:i in [1..3]]),
           P[4]);
        b:= Coordinates(VectorSpaceWithBasis([Q[i]:i in [1..3]]),
           Q[4]);

        A:= Matrix(Rationals(),3,3,&cat[Eltseq(P[i]):i in
           [1..3]]);
        Ainv:=A^(-1);
        B:= Matrix(Rationals(),3,3,&cat[Eltseq(b[i]/a[i]*Q[i]):i
           in [1..3]]);

        return Ainv*B;
end function;

AG := function(V:mem:=false)
```

```
if not mem then //option worse on memory
        //finds 4-seqs such that all its 3-subsets are L.
            I.
        M:={U: U in Permutations(V,4)|forall{T:T in
            Subsets(Set(U),3)|Rank(Matrix(Rationals()
            ,3,3,&cat[Eltseq(t):t in T]))eq 3}};
        print "Finding generators\n";
        //random fixed frame
        P:=Random(M);

        gen:={};

        for Q in M do
                T:=L(P,Q);
                if {sub<Q3|p*T>:p in V} eq {sub<Q3|p>: p
                    in V} then gen:= gen join {T};printf "
                    %o\n\n",T;end if;
        end for;

else

        repeat P:={Random(V):i in [1..4]}; until #P eq 4
            and forall{T:T in Subsets(P,3)|Rank(Matrix(
            Rationals(),3,3,&cat[Eltseq(t):t in T]))eq 3}
            ;
        print "Finding generators\n";

        P:=Setseq(P);
        gen:={};
        V:=Setseq(V);

        for a,b,c,d in V do
                if #{a,b,c,d} eq 4 and forall{T:T in
                    Subsets(Set([a,b,c,d]),3)|Rank(Matrix(
                    Rationals(),3,3,&cat[Eltseq(t):t in T
                    ]))eq 3} then
                        T:=L(P, [a,b,c,d]);
                        if {sub<Q3|p*T>:p in V} eq {sub<
                            Q3|p>: p in V} then
                                gen:= gen join {T};
                                printf "%o\n\n",T;
                        end if;
                end if;
        end for;
end if;
```

```
        G:=sub<GL(3,Rationals())|gen>;

        printf "small group type %o\n", IdentifyGroup(G);
        return G;
end function;

AG(CK);

//test
/* P:=Random(M);
Q:=Random(M diff {P});

printf "Set P:\n%o\n",P;
printf "Set Q:\n%o\n",Q;
printf "Linear Transformation:\n%o\n",L(P,Q);

P:=Setseq(P);
a:= Coordinates(VectorSpaceWithBasis([P[i]:i in [1..3]]),P[4]);
printf "P[4] eq sum P[i] %o\n",P[4] eq &+{a[i]*P[i]:i in [1..3]};
Q:=Setseq(Q);
b:= Coordinates(VectorSpaceWithBasis([Q[i]:i in [1..3]]),Q[4]);
printf "Q[4] eq sum Q[i] %o\n",Q[4] eq &+{b[i]*Q[i]:i in [1..3]};


for i in [1..3] do printf"L(P[%o]) eq b[%o]/a[%o]*Q[%o] %o\n",i,i
    ,i,i,P[i]*L(Set(P),Set(Q)) eq b[i]/a[i]*Q[i];end for; */
```