UNIVERSITY OF
CAMBRIDGE
Department of Computer
Science and Technology

Approaching the Conway-99 problem using SAT solvers

Ali Keramatipour

Churchill College

June 2023

Submitted in partial fulfillment of the requirements for the
Master of Philosophy in Advanced Computer Science

Total page count: 51

Main chapters (excluding front-matter, references and appendix): 44 pages (pp 7–50)

Main chapters word count: 13675

Methodology used to generate that word count: Overleaf

Declaration

I, Ali Keramatipour of Churchill College, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed: Ali Keramatipour

Date: 2023/06/18

Abstract

The Conway-99 problem questions the existence of a strongly regular graph with 99 vertices and specific parameters. A *strongly* regular graph is a regular graph that exhibits two additional properties: vertices must share a fixed number of neighbours, depending on whether they are adjacent or not, given by two parameters. Despite the search space for this graph being finite, the computational power needed to traverse it is substantial. Therefore, better strategies are required in order to find this graph or prove its non-existence. SAT solvers, designed to solve instances of boolean satisfiability formulas, have been developed and optimised significantly due to the simplicity of SAT problems. Based on Cook-Levin's theorem, computer scientists have been focusing on developing efficient SAT solvers as many problems can be reduced to a SAT problem instance. Hence, we decided to approach the Conway-99 problem using SAT solvers. To do this, we study strongly regular graphs' properties and SAT solvers' capabilities. By encoding the problem of finding strongly regular graphs into SAT instances and running experimental tests, we shall see the incapability of SAT solvers facing this problem in a reasonable time. We will then explore the underlying mathematical reasons for these limitations.

Contents

Chapter 1

Introduction

The Conway-99 problem [14], an open math problem in the field of graph theory, asks about the existence of a strongly regular graph with 99 vertices and specific parameters. Despite being named after John Horton Conway, J. Seidel was the one who initially showed interest in discovering this graph, as mentioned in a dedication to him [29]. In 1969, Norman L. Biggs suggested the possible existence of this graph with the said parameters [5].

1.1 Strongly regular graphs

A regular graph is a graph where each vertex has the same degree (number of neighbours). A strongly regular graph is a regular graph with the property that the number of common neighbours of two vertices can be determined specifically by their adjacency. The notion of strongly regular graphs was first introduced by Raj Chandra Bose [6], a mathematician known for his work in the theory of error-correcting codes.

Mathematicians have conducted extensive research on the existence of various strongly regular graphs. Their symmetrical properties have proved to be valuable and applicable in numerous fields, including statistics, Euclidean geometry, group theory, coding theory, cryptography, and more. For example, these symmetrical structures meet coding theory when building efficient and easy-to-decode codes. Crnković et al. [17] have developed various ternary codes using the structure of strongly regular graphs. Alternatively, in statistics, one can use these ideally structured graphs for experiments [9]. One distinctive example is [23], where Janmark et al. made use of strongly regular graphs' local symmetrical structure in quantum search.

These highly structured graphs are so interesting because they also show random behaviours [11]. In fact, this randomness makes it hard to prove the existence or non-existence of such graphs. Based on both random and structured properties of strongly regular graphs, many applications can be discovered.

One famous and small strongly regular graph, with ten vertices, is the Petersen graph shown in Figure 1.2. In [24], Donald Knuth describes the Petersen graph as *"a remarkable configuration that serves as a counterexample to many optimistic predictions about what might be true for graphs in general"*. This graph is a perfect example of strongly regular graphs' strange and valuable nature. An entire book, authored by Holton and Sheehan [22], has been dedicated to it because of its usefulness.

It can be proven that at most five strongly regular graphs exist, such that every edge be-

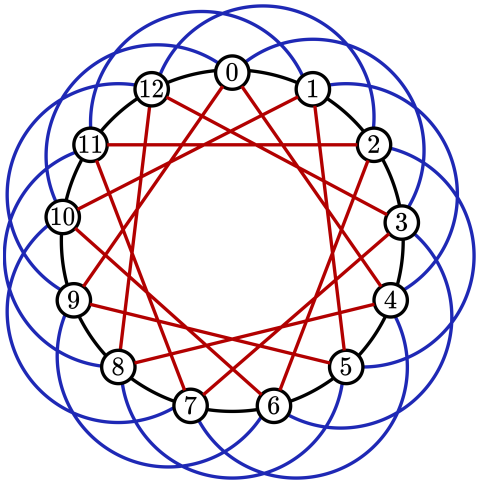

Figure 1.1: A strongly regular graph, Paley(13).

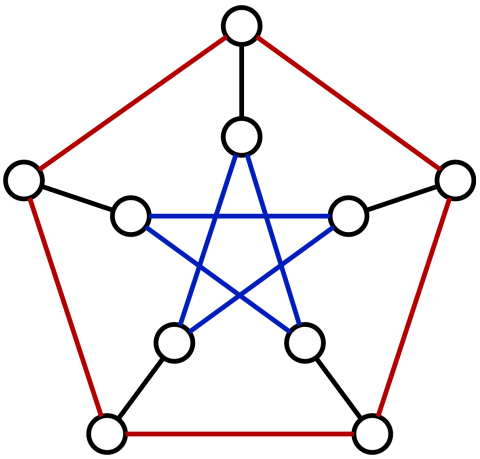

Figure 1.2: Petersen graph

longs to a unique triangle and every non-edge belongs to a unique quadrilateral. In other words, this means that when two vertices are adjacent, they share only one common neighbour (forming a triangle), and when two vertices are not adjacent, they share two common neighbours (forming a quadrilateral); therefore, the number of their common neighbours is determined by their adjacency. The Conway-99 problem asks for the existence of one such graph with 99 vertices and 14 regularity.

Brouwer and Haemers [8] provide an equivalent definition of strongly regular graphs based on the field of spectral graph theory and linear algebra. A graph is strongly regular, if its adjacency matrix has exactly three eigenvalues. We will go through these in more detail in Sections 2 and 3.

1.2 Boolean satisfiability problem

The boolean satisfiability problem (SAT) is one fundamental problem in mathematics and computer science. It involves determining the possibility of assigning truth values to boolean variables (0/`False`) or (1/`True`) so that the formula evaluates to `True`. The Cook-Levin [15][25] theorem concludes that SAT problem is NP-Complete, which means any NP problem can be reduced to an instance of the SAT problem. The simplicity of the SAT problem's form has motivated many to develop efficient algorithms to overcome this problem.

This is where SAT solver models come into play. They are designed to solve instances of the SAT problem. As an input, they receive a boolean formula and determine the existence of an assignment of truth values to the variables, such that the whole formula evaluates to `True`. SAT solvers use a variety of algorithms and heuristics to reach and converge on a solution.

It is trivial that the search space to find the Conway-99 graph is limited. However, an exhaustive search attack to this problem is beyond the capabilities of computing power. This led us to consider using SAT solvers to tackle the problem. SAT search is a widely used approach to go through the search space more efficiently.

A commonly used method in SAT solvers is *symmetry-breaking*, which is particularly relevant when dealing with strongly regular graphs that have *highly symmetrical* structures. This technique allows SAT solvers to avoid searching for equivalent cases, making them

valuable tools in these scenarios. Even though strongly regular graphs exhibit symmetry, it is their distinctive asymmetrical characteristics that make them difficult to identify. More techniques and algorithms are commonly used, which we will discuss later in Section 4. Then, in Section 5, we will examine the experiments we had and analyse them mathematically.

1.3 Work completed

Our research involves converting the Conway-99 problem into a SAT problem to determine if it can be solved within a reasonable timeframe using SAT solvers. We generalised this using different SAT methods to search for any strongly regular graph with its specific properties. On this path, we studied the theoretical aspects of strongly regular graphs, their symmetries and patterns. We proved some of these patterns must not exist in any possible answers to the Conway-99.

Overall, the research combined two significant and intriguing areas of study that have the ability to provide meaningful insights into both fields.

Chapter 2

Preliminaries

2.1 Basics

In this work, for a graph $G = (V, E)$, the vertex set used is $\{1, 2, \cdots, n\}$, while n represents the number of vertices in a graph, or $n = |V|$. For edges, notations $\{u, v\}$ or uv are used, while $u, v \in V$. It is worth noting that all graphs under consideration are connected and simple, that is, undirected and without any double-edges or self-loops. Thus, when we refer to a graph, we specifically mean a simple graph. The neighbourhood of a vertex v is denoted as $N(v)$, and its closed neighbourhood, that is, $N(v) \cup \{v\}$, is denoted as $N[v]$.

We begin by mathematically defining our main concept, strongly regular graphs:

Definition 1 (Strongly regular graphs)**.** *A graph $G = (V, E)$ is said to be strongly regular with parameters (n, k, λ, μ) if it satisfies the following conditions:*

1. *G is regular of degree k and is neither complete nor empty.*
2. *Every pair of adjacent vertices in G has exactly λ common neighbors.*
3. *Every pair of non-adjacent vertices in G has exactly μ common neighbors.*

In our context, when we state that a graph has a λ (μ) parameter, we mean every pair of adjacent (non-adjacent) vertices share λ (μ) neighbours. This is independent of the underlying graph being strongly regular or not.

While searching for strongly regular graphs, it can be readily proved that for a (n, k, λ, μ) strongly regular graph G, its complement $\overline{G}$ is a strongly regular graph with parameters $(n, \overline{k}, \overline{\lambda}, \overline{\mu})$:

$$\overline{k} = n - k - 1$$
$$\overline{\lambda} = n - 2 - 2k + \mu$$
$$\overline{\mu} = n - 2k - \lambda$$

A graph G is called a *trivial* strongly regular, and is **not** considered as a strongly regular graph, if either itself or its complement $\overline{G}$ is disconnected. These graphs satisfy strong regularity conditions trivially. As an example, the complete K_n and the empty graphs trivially satisfy these conditions.

Our focus in this section will be to limit the possible parameters further and examine the specific requirements and conditions that must be met by a set of parameters (n, k, λ, μ) so that there could exist an (n, k, λ, μ) strongly regular graph.

Theorem 2.1.1. *In a strongly regular graph with parameters (n, k, λ, μ), the following equation holds:*

$$(n - k - 1)\mu = k(k - \lambda - 1).$$

Proof. To prove this, we will employ a double-counting approach and count the number of edges between two sets: $N(v)$ and $V - N[v]$. The size of the set $V - N[v]$ equals $n - k - 1$. Each vertex $u \in V - N[v]$ is connected to exactly μ vertices in $N(v)$. As a result, there are $(n - k - 1)\mu$ edges between these two sets.

On the other hand, there exist k vertices in $N(v)$, such that each of them has λ neighbours within $N(v)$ and one neighbour v itself. Therefore, $k - \lambda - 1$ edges remain for each vertex, connecting it to v's non-neighbours. This gives us a total of $k(k - \lambda - 1)$ edges between the sets.

By counting the same thing with two different expressions, we can infer that the equation

$$(n - k - 1)\mu = k(k - \lambda - 1) \tag{2.1}$$

holds. □

2.2 Algebraic graph theory

The algebraic implications of strongly regular graph parameters are of significant interest to mathematicians and researchers in the field. The study of such parameters can provide valuable insights into the structural properties of graphs and their relationships with other mathematical concepts.

To study these properties, let A be the graph G's adjacency matrix. The identity matrix is denoted by I, while the all-one matrix is J.

Theorem 2.2.1. *A graph G is an (n, k, λ, μ) strongly regular graph if, and only if, its adjacency matrix satisfies the equation*

$$A^2 = kI + \lambda A + \mu(J - I - A). \tag{2.2}$$

There is a more generalised result for regular graphs that states a graph is connected and regular if and only if the matrix J is a linear combination of powers of A. The proof of it is similar to the proof of Theorem 2.2.1. Their proofs rely on the following lemma regarding the powers of an adjacency matrix:

Lemma 2.2.2. *Let G be a graph with an adjacency matrix A. For a pair of vertices $\{i, j\}$, and for any non-negative integer p, the ij-th entry of the matrix A^p is the number of walks of length p from vertex i to vertex j in G.*

Proof. An induction on p will be used.

Base case: For $p = 0$, the ij-th entry of $A^0 = I$, which is equal to the identity matrix, is 1 on its diagonal. This is in line with the fact that there is one walk of length 0 from any vertex to itself and no walks of length 0 between different vertices.

Induction step: We assume the lemma holds for k. We prove it for $k + 1$.

For the ij-th entry of matrix A^{k+1} the equation

$$a_{ij}^{k+1} = \sum_{v=1}^{n} a_{iv}^{k} a_{vj}$$

holds. This is the sum over the products of the iv-th entry of A^k and the vj-th entry of A.

By induction's hypothesis, the iv-th entry of A^k is the number of walks of length k between i and v. Therefore, if vertex v is adjacent to j, extending all these walks with the edge $\{v, j\}$ would result in new walks that are of length $k+1$. As a result, all walks are counted. Also, since walks of length $k-1$ are counted only once, and the newly counted walks are different as they all end in different vertices v, each walk gets counted exactly once, and the lemma holds. □

Proof of theorem 2.2.1. The number of walks of length two between a vertex and itself equals its degree; hence, in a k-regular graph, matrix A^2's diagonal equals k. On the other hand, if Equation 2.2 holds, the graph must be k-regular.

A pair of adjacent vertices u, v in a strongly connected graph share λ neighbours. We can infer that there are λ walks of length two between u and v, if, and only if, the equality $a^2_{u,v} = \lambda$ holds for all uv such that $a_{u,v} = 1$.

For non-adjacent vertices, the number of shared neighbours, and thus, the walks of length two in an (n, k, λ, μ) strongly regular graph, are μ. The matrix $J - I - A$ contains 1 only on disconnected pairs of vertices. Thus, the matrix A^2 contains μ where on indices where $J - I - A$ is 1, if and only if, each two non-neighbours share μ neighbours.

The summation of the multiplications of matrix I by the regularity parameter k, matrix A by λ parameter, and matrix $J - I - A$ by μ parameter will generate the matrix A^2 if and only if G is an (n, k, λ, μ) strongly regular graph. □

We examined the algebraic representation of the Definition 1 through the equation 2.2.

Definition 2 (Eigenvalues and eigenvectors)**.** *For an adjacency matrix A, a scalar τ is called an eigenvalue of A if there exists a non-zero vector x, such that $Ax = \tau x$. The vector x is called an eigenvector of the corresponding matrix. The multiplicity of eigenvalue τ is the number of linearly independent eigenvectors corresponding to it.*

Consider vector $\mathbf{1} = (1, 1, \cdots, 1)^T$ of length n. If G is a k-regular graph, the sum of each row and column of its adjacency matrix A would be k. Consequently, the equation

$$A\mathbf{1} = k\mathbf{1} \tag{2.3}$$

holds. This implies that k is an eigenvalue and $\mathbf{1}$ is an eigenvector of the adjacency matrix A.

Theorem 2.2.3. *The maximum absolute eigenvalue of a connected graph G's adjacency matrix A is $\Delta(G)$. This maximum value is attained if, and only if, the graph is $\Delta(G)$-regular. The multiplicity of $\Delta(G)$ as an eigenvalue is one.*

Proof. Let x be the eigenvector of matrix A with the largest absolute value at coordinate i, and let τ be its eigenvalue. The inequality

$$|\tau||x_i| = |(Ax)_i| = \left|\sum a_{ij}x_j\right| \leq deg(v_i)|x_i| \leq \Delta(G)|x_i| \tag{2.4}$$

holds, as there are $deg(v_i)$ non-zero entries on the i-th row of matrix A, with an absolute value less than $|x_i|$. Therefore, for each eigenvalue τ, inequality $|\tau| \leq \Delta(G)$ holds.

For all inequalities in Equation 2.4 to be equalities, all neighbours v_j of vertex v_i, that is, $v_j \in N(v_i)$ and $a_{ij} = 1$, need $x_i = |x_j|$. The degree of v_i must also be $\Delta(G)$.

Since the graph is connected, and the neighbours of vertex v_i all have the largest value in their respective coordinate in vector x, by iterating these equalities for more vertices, we infer that x is a constant vector, and the graph is $\Delta(G)$-regular. Since vector x is a constant vector, that is, a multiplication of vector $\mathbf{1}$, eigenvalue $\Delta(G)$ has multiplicity one. □

The following lemma is useful, as an adjacency matrix is symmetric and has real values. Note that for a vector v or a matrix A, its transpose is denoted as v^T and A^T, respectively.

Lemma 2.2.4. *Let A be a real symmetric matrix. If vectors u and v are eigenvectors of A, each corresponding to a distinct eigenvalue, then u and v are orthogonal.*

Proof. The two eigenvectors must satisfy the following equations: $Au = \tau u$ and $Av = \theta v$. Since A is symmetric, we can infer:

$$(Au)^T = u^T A^T = u^T A.$$

Thus, by multiplying this with v, we can see that:

$$(u^T \tau)v = (u^T A)v = u^T(Av) = u^T(\theta v)$$

holds; since $u^T v(\tau - \theta) = 0$, and $\theta \neq \tau$, we can infer that u and v are orthogonal. □

Theorem 2.2.5. *The adjacency matrix A of an (n, k, λ, μ) strongly regular graph has three distinct eigenvalues.*

Proof. Based on Theorem 2.2.3, we know that k is an eigenvalue of multiplicity one with eigenvector $\mathbf{1}$.

Consider Equation 2.2, and suppose θ is an eigenvalue of A, and its eigenvector is $x \neq \mathbf{1}$. Rewrite the equation by multiplying it with the eigenvector x, as follows:

$$A^2x = kx + \lambda Ax + \mu(J - I - A)x.$$

By considering $Ax = \theta x$,

$$\theta^2 x = kx + \lambda\theta x + \mu Jx - \mu x - \mu Ax.$$

According to Lemma 2.2.4, columns of matrix J are orthogonal with x; thus, $\mu Jx = 0$ and the equation can be rewritten as:

$$\theta^2 x = kx + \lambda\theta x - \mu x - \mu\theta x.$$

Ergo, the eigenvalues of matrix A must satisfy the equation

$$\theta^2 + (\mu - \lambda)\theta + (\mu - k) = 0.$$

Two more eigenvalues can be obtained by this quadratic equation:

$$r = \frac{1}{2}\left[(\lambda - \mu) + \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right], \tag{2.5}$$

and

$$s = \frac{1}{2}\left[(\lambda - \mu) - \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}\right]; \tag{2.6}$$

therefore, we have exactly three eigenvalues for the adjacency matrix of a strongly regular graph. Considering values r and s, since

$$\sqrt{(\lambda-\mu)^2+4(k-\mu)} > \lambda-\mu,$$

eigenvalue r is positive, and s is negative.

□

Definition 3 (Characteristic polynomial)**.** *The characteristic polynomial of a matrix A is the polynomial*

$$P(\tau) = det(\tau I - A).$$

The eigenvalues of a matrix are the roots of the characteristic polynomial, and this polynomial is of degree n.

Consider an eigenvalue τ and an eigenvector x for matrix A: $Ax = \tau x$. Rewriting the equation as $x(\tau I - A) = 0$, we can infer that $(\tau I - A)$ has no inverse. Consequently, $\det(\tau I - A) = 0$, which provides a polynomial equation whose roots correspond to the eigenvalues of A. The number of times an eigenvalue appears as a root is equal to its multiplicity.

Definition 4. *The trace of an $n \times n$ square matrix A, denoted as $tr(A)$, is defined as:*

$$tr(A) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + a_{33} + \ldots + a_{nn}.$$

Lemma 2.2.6. *The sum of diagonal entries of a matrix A, i.e. its trace $tr(A)$, is equal to the summation of A's eigenvalues:*

$$tr(A) = \sum_{i=1}^{n} \tau_i.$$

Proof. The characteristic polynomial of A is given by:

$$p(\tau) = \det(\tau I - A),$$

which can be rewritten using eigenvalues of A as

$$p(\tau) = (\tau - \tau_1)(\tau - \tau_2)\ldots(\tau - \tau_n).$$

Expanding this product

$$p(\tau) = \tau^n - (\tau_1 + \tau_2 + \ldots + \tau_n)\tau^{n-1} + \ldots$$

shows us that the coefficient of $-\tau^{n-1}$ is the negation of the summation of eigenvalues.

The Leibniz formula is another way to calculate the determinants of matrices. For a matrix Z, the formula is defined as follows:

$$det(Z) = \sum_{p \in S_n} sgn(p) \prod_{i=1}^{n} z_{i,p(i)},$$

where $z_{i,p(i)}$ is Z's i-th row and $p(i)$-th column element; the group S_n is the symmetric group defined over a finite set of n symbols. The order of the group S_n is $n!$ as it contains all permutations of n elements. Ergo, each p is a permutation. The sign function sgn is

defined to be +1 if the parity of a permutation is even, and -1 for odd parity. The parity of a permutation p is the parity of its number of inversions, that is, two values a and b such that $a < b$ and $p(a) > p(b)$.

Calculating the determinant of a matrix with this approach requires $O(n!)$ operations, which makes it impractical. However, it provides us with more insights into understanding the determinants of matrices.

Considering the matrix $\tau I - A$, the only permutation capable of producing τ^{n-1} is the identity, as otherwise, at least two non-diagonal elements would be chosen, and the maximum order of τ can be $n-2$. For the identity permutation, the polynomial generated is

$$\prod_{i=1}^{n}(\tau - a_{ii}).$$

Hence, the coefficient of τ^{n-1} is $-\sum_{i=1}^{n} a_{ii}$. Therefore, the equation

$$tr(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \tau_i$$

holds. □

Lemma 2.2.6 comes in handy when working with graphs and adjacency matrices, as the trace of simple graphs are 0. In Theorem 2.2.5, we proved that exactly three distinct eigenvalues exist. Let m_r and m_s each represent the multiplicity (number of times the eigenvalue appears as a root of the characteristic polynomial) of their respective eigenvalue. According to Lemma 2.2.6, we have

$$m_r r + m_s s + k = tr(A) = 0,$$

and given that we are dealing with a polynomial of degree n, it follows that:

$$m_r + m_s = n - 1.$$

Hence, we can calculate the multiplicities based on their eigenvalues:

$$m_s = \frac{(n-1)r + k}{r - s}, \qquad m_r = \frac{(n-1)s + k}{s - r}. \tag{2.7}$$

In Equations 2.5 and 2.6, the eigenvalues were expressed in terms of the strongly regular graph's parameters, (n, k, λ, μ). Substituting the values based on it yields

$$m_r = \frac{1}{2}\left[(v-1) - \frac{2k + (v-1)(\lambda - \mu)}{\sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}\right], \tag{2.8}$$

and

$$m_s = \frac{1}{2}\left[(v-1) + \frac{2k + (v-1)(\lambda - \mu)}{\sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}\right]. \tag{2.9}$$

A powerful condition, the *integrality condition*, can be derived from the multiplicities, as these numbers must be natural numbers. Therefore, many strongly regular graph parameter sets can be ruled out, as they will not yield natural numbers for the multiplicities. We will use this further in Section 3 to narrow down the parameter set possibilities based on our needs.

Lemma 2.2.7. *Let A denote a simple and connected graph's adjacency matrix with n vertices. If A^2 is a linear combination of three matrices A, I, and J, then the graph it represents is strongly regular.*

Proof. Let a, b, and c denote the coefficients in the linear representation. Then,

$$A^2 = aI + bA + cJ,$$

so

$$A^2 = (a + c)I + (b + c)A + c(J - I - A).$$

According to Theorem 2.2.1, it can be inferred that this represents a strongly regular graph with parameters $(n, a + c, b + c, c)$. □

Theorem 2.2.8. *A k-regular connected graph G with an adjacency matrix A that has exactly three distinct eigenvalues is strongly regular.*

Proof. Since the graph is k-regular, one of these eigenvalues is k. Name the other two eigenvalues as r and s.

Let τ denote an eigenvalue of matrix $(A - rI)(A - sI)$, and x be an associated eigenvector:

$$(A - rI)(A - sI) \cdot x = \tau x.$$

Since matrix A has only three eigenvalues, and the sum of r and s's multiplicities is $n - 1$, matrix $(A - rI)(A - sI)$ has 0 as an eigenvalue of multiplicity $n - 1$.

The eigenvalues of matrix $A - rI$ are obtained by the equation

$$(A - rI)x = \tau x.$$

By rewriting the equation as follows:

$$Ax = (r + \tau)x,$$

we can determine that $r+\tau$ is an eigenvalue of the initial matrix A. Moreover, we can deduce that $(k - r)$ is an eigenvalue of matrix $(A - rI)$. With the same approach, we can see $(k - s)$ is an eigenvalue for $(A - sI)$. In Equation 2.3, we saw that the eigenvector $\mathbf{1}$ belongs to the eigenvalue k. Therefore,

$$(A - rI)(A - sI)\mathbf{1} = (A - rI)(k - s)\mathbf{1} = (k - r)(k - s)\mathbf{1}$$

holds, and the only non-zero eigenvalue is $(k-r)(k-s)$ with multiplicity 1 and eigenvector $\mathbf{1} = (1, 1, \ldots, 1)^T$. Because there is only one linearly independent eigenvector, $\mathbf{1}$, all the entries in the matrix have the same value. Thus,

$$(A - rI)(A - sI) = \frac{(k - r)(k - s)}{n} J.$$

By expanding the left side of the equation, we obtain

$$A^2 - (r + s)A + rsI = \frac{(k - r)(k - s)}{n} J.$$

Based on Lemma 2.2.7, it is concluded that A is the adjacency matrix of a strongly regular graph. □

2.3 Automorphism groups

We begin with two graph-related definitions:

Definition 5. *An **isomorphism** between two graphs G and H is defined by a bijective function*

$$f : V(G) \to V(H)$$

such that $\{u, v\} \in E(G)$ if and only if $\{f(u), f(v)\} \in E(H)$.

Definition 6. *An **automorphism** refers to an isomorphism that maps a graph G to itself. It is defined using a permutation φ, such that $\{u, v\} \in E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(G)$.*

In group theory, a group is defined as follows:

Definition 7. *A **group** G is a non-empty set with a binary operation "$\cdot$"*

$$\cdot : G \times G \to G$$

such that it satisfies three requirements:

1. *Associativity: for all a,b,c in G, equality $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ holds.*
2. *Identity: there exists $i \in G$, such that for all $a \in G$, equalities $a = a \cdot i = i \cdot a$ hold.*
3. *Inversion: for all $a \in G$, there exists an element $a^{-1} \in G$, such that $a \cdot a^{-1} = a^{-1} \cdot a = i$.*

Consider all the permutations that form automorphisms in a graph G. The composition of any two automorphisms φ_1 and φ_2 yields an automorphism $\varphi_1\varphi_2$. All three group requirements are satisfied in this set:

1. Associativity: for three automorphisms φ_1, φ_2, and φ_3, the equality $\varphi_1 \cdot (\varphi_2 \cdot \varphi_3(x)) = (\varphi_1 \cdot \varphi_2) \cdot \varphi_3(x)$ holds.
2. Identity: the identity permutation, denoted by φ_I is an automorphism.
3. Inversion: If φ is an automorphism, then its inverse permutation φ^{-1} is also an automorphism.

Hence, the set of all automorphisms is a group, which is named the automorphism group and is denoted with $Aut(G)$.

Automorphism groups are meaningful in studying strongly regular graphs because their order indicates the level of *symmetry* in graphs. The order of $Aut(K_n)$ for a complete graph K_n is $n!$, making it highly symmetrical. On the other hand, a graph G is considered asymmetric if its automorphism order, denoted as $|Aut(G)|$, is one. It is noteworthy that numerous strongly regular graphs do not possess high symmetries. We will review this later in Section 5.

Definition 8 (Spectrum)**.** *The spectrum of a matrix is the set of its eigenvalues and their multiplicity. In algebraic graph theory, a **graph's spectrum** is the graph's adjacency matrix's spectrum.*

Consider the following as an example of a strongly regular graph's spectrum:

Example 2.3.1. *The Petersen graph, a $(10, 3, 0, 1)$ strongly regular graph, has three eigenvalues: $k = 3$, $r = 1$, and $s = -2$. The multiplicity of the eigenvalues are $m_k = 1$, $m_r = 5$, and $m_s = 4$. To denote the spectrum, we write*

$$\tau(A) = \{3^1, 1^5, (-2)^4\}.$$

The Petersen graph is an ***integral*** graph. A graph is integral if its spectrum consists of integers. When two graphs have the same spectrum, they are referred to as *cospectral*. Although isomorphic graphs are cospectral, the reverse is not always true. This is evident even among the highly structured, strongly regular graphs, which we will explore in Section 5.

n	k	λ	μ	r	m_r	s	m_s
5	2	0	1	$(-1+\sqrt{5})/2$	2	$(-1-\sqrt{5})/2$	2
9	4	1	2	1	4	-2	4
10	3	0	1	1	5	-2	4
13	6	2	3	$(-1+\sqrt{13})/2$	6	$(-1-\sqrt{13})/2$	6
15	6	1	3	1	9	−3	5
16	5	0	2	1	10	-3	5
16	6	2	2	2	6	-2	9
17	8	3	4	$(-1+\sqrt{17})/2$	8	$(-1+\sqrt{17})/2$	8
21	10	5	4	3	6	-2	14

Table 2.1: Parameters of multiple strongly regular graphs and their spectrums. The eigenvalue k and its multiplicity 1 is omitted.

Chapter 3

Strongly regular graphs

In this chapter, we will be studying the theoretical aspects of several classes of strongly regular graphs and how they are constructed. Moreover, we apply the conditions we have learned in the Preliminaries chapter 2.

We begin this chapter by mathematically defining the Conway-99 open problem. Next, we will examine several similar structures to the potential Conway-99 graph. Finally, we will analyse these structures and their connection to the Conway-99 problem.

3.1 The Conway-99 problem and similar graphs

The Conway-99 is an open graph problem that asks the following question:

Problem 3.1.1 (Conway-99)**.** *Does an instance of* $(99, 14, 1, 2)$ *strongly regular graph exist?*

The $\lambda = 1$ and $\mu = 2$ conditions state that every edge should be part of a *unique triangle* and every non-edge should be part of a *unique quadrilateral*. Only five possible parameter set exists for strongly regular Graphs with $\lambda = 1$ and $\mu = 2$. These parameters are shown in Table 3.1.

$\exists$	n	k	λ	μ
+	9	4	1	2
?	99	14	1	2
+	243	22	1	2
?	6273	112	1	2
?	494019	994	1	2

Table 3.1: Strongly regular graphs with $\lambda = 1$ and $\mu = 2$. Only the existence of two graphs has been verified.

An instance of $(9, 4, 1, 2)$ strongly regular graph is named the Paley(9) graph. Paley graphs form an infinite group of strongly regular graphs, which will be studied further in Section 3.2.

In Section 3.3, we delve into the construction process of the Berlekamp-Van Lint-Seidel graph, a $(243, 22, 1, 2)$ strongly regular graph, which was discovered in 1973.

To prove only five possible parameter sets exist, we use the multiplicity integrality condition and prove the following theorem:

Theorem 3.1.2. *There are no strongly regular graphs with $\lambda = 1$ and $\mu = 2$ out of the parameter sets of Table 3.1.*

Proof. Let $(n, k, 1, 2)$ be a possible parameter set. By using the Equation 2.1 we obtain

$$(n - k - 1)2 = k(k - 2),$$

so we can infer that

$$n = \frac{k^2}{2} + 1$$

must hold.

We can now rewrite the Equations 2.8 and 2.9, the eigenvalues' multiplicities, only based on one parameter. We then obtain

$$M = \frac{1}{2}\left[\frac{k^2}{2} + \frac{2k - \frac{k^2}{2}}{\sqrt{4k - 7}}\right].$$

Since these multiplicities must be positive integers, we can rule out many possibilities for strongly regular graphs with $\lambda = 1$ and $\mu = 2$.

To ensure the multiplicities are natural numbers, equality $4k - 7 = t^2$ must hold for a natural t. Replacing k with $\frac{t^2+7}{4}$ we obtain

$$M = \frac{1}{2}\left[\frac{\left(\frac{t^2+7}{4}\right)^2}{2} + \frac{2\left(\frac{t^2+7}{4}\right) - \frac{\left(\frac{t^2+7}{4}\right)^2}{2}}{\sqrt{4\left(\frac{t^2+7}{4}\right) - 7}}\right].$$

To simplify the equation, we take these steps

$$\begin{aligned}
M &= \frac{1}{2}\left[\frac{\left(t^2+7\right)^2}{32} + \frac{\left(t^2+7\right) - \left(\frac{t^2+7}{4}\right)^2}{2t}\right] \\
\Rightarrow M &= \frac{(7+t^2)(9-t^2+7t+t^3)}{64t} \\
\Rightarrow 64M &= \frac{(7+t^2)(9-t^2+7t+t^3)}{t} \\
\Rightarrow 64M &= \frac{63 - 7t^2 + 49t + 7t^3 + 9t^2 - t^4 + 7t^3 + t^5}{t} \\
\Rightarrow 64M &= 49 + \frac{63}{t} + 2t + 14t^2 - t^3 + t^4
\end{aligned}$$

The values that would make $\frac{63}{t}$ an integer would be $t \in \{\pm 1, \pm 3, \pm 7, \pm 9, \pm 21, \pm 63\}$. These would result in k values $\{2, 4, 14, 22, 112, 994\}$. Since for $k = 2$, the resulting graph would be a complete graph K_3, and complete graphs are not considered as strongly regular graphs in their nature, this case would be omitted. The five final possibilities have regularity $k \in \{4, 14, 22, 112, 994\}$ and number of vertices $n \in \{9, 99, 243, 6273, 494019\}$, as shown in Table 3.1. □

3.2 Paley graphs

Paley strongly regular graphs are constructed based on Galois fields. To begin, we must first define ***field***. A field is a set with four operations of arithmetic: addition, subtraction, multiplication, and division (excluding division by zero). For instance, the set of rational numbers $\mathbb{Q}$ is considered a field, while natural numbers do not form a field because not all divisions can be defined. For example, $\frac{3}{4} \notin \mathbb{N}$. The mathematical definition is as follows:

Definition 9 (Field)**.** *A field is a set F equipped with two binary operations, namely addition (+) and multiplication (+). These binary operations are mappings in the form $F \times F \to F$. The following axioms must hold in these operations:*

1. *Associativity: For all $a, b, c \in F$, we have $(a+b)+c = a+(b+c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.*
2. *Commutativity: For all $a, b \in F$, we have $a + b = b + a$ and $a \cdot b = b \cdot a$.*
3. *Identity Elements: There exist two distinct elements 0 and 1 in F such that for all $a \in F$, we have $a + 0 = a$ and $a \cdot 1 = a$.*
4. *Additive Inverses: For every $a \in F$, there exists an element $-a$ in F such that $a+(-a) = 0$.*
5. *Multiplicative Inverses: For every $a \neq 0$ in F, there exists an element a^{-1} in F such that $a \cdot a^{-1} = 1$.*
6. *Distributivity: For all $a, b, c \in F$, we have $a \cdot (b + c) = a \cdot b + a \cdot c$.*

To better understand the concept, we consider this simple example. For the set $\{0, 1, 2\}$, by defining the aforementioned axioms, modulo the prime number 3, we obtain a field. The first two conditions arbitrarily hold. The identity elements are 0 and 1. The additive inverse of 1 is 2 and vice-versa. The multiplicative inverse of 2 is 2 itself, as $4 \equiv 1 \pmod 3$. This field is named $GF(3)$. Galois fields are defined as follows:

Definition 10 (Galois field)**.** *A finite or Galois field is a field that contains a finite number of elements.*

The order of a finite field must be a prime power, i.e. p^n. For each power, there exists, up to isomorphism, exactly one finite field, which we denote as $GF(p^n)$.

A *quadratic residue* of a field $GF(p^n)$ is an element's square in $GF(p^n)$. For example, element 1 is the quadratic residue of 1 and 2 in $GF(3)$, since $1^2 \equiv 2^2 \equiv 1 \pmod 3$, while 2 is not.

The set of quadratic residues of a field $GF(p^n)$ is denoted by $QR(p^n)$. That is,

$$a \in QR(p^n) \Leftrightarrow \exists b \in GF(p^n) : b \cdot b = a.$$

The mathematical definition of Paley graphs' structure is described as follows:

Definition 11 (Paley graphs)**.** *Let q be a prime power of p, such that $q = p^n \equiv 1 \pmod 4$. The graph $Paley(q) = (V, E)$ is defined over the set of vertices $V = GF(q)$. The set of edges E is defined as*

$$E = \{\{u, v\} : u - v \in QR(q)\},$$

that is, two vertices form an edge if their values differ by a quadratic residue.

We first build a simple example of Paley graphs, specifically, Paley(13). The field $GF(13)$ is defined as the usual arithmetic operations modulo 13, over the set of $\{0, 1, 2, \ldots, 12\}$. To obtain the quadratic residue of $GF(13)$, consider the squares of $\pm1, \pm2, \pm3, \pm4, \pm5$, and ±6,

which respectively give us $+1, +4, -4, +3, -1$, and -3. In figure 1.1, an instance of Paley(13) is drawn. A vertex v is adjacent to 6 more vertices $v \pm 1$, $v \pm 3$, and $v \pm 4$.

To build the Paley(9) graph, which is a strongly regular graph with parameters $(9, 4, 1, 2)$, we first need to build the field $GF(3^2)$. The process of building Galois fields $GF(p^n)$ for $n > 1$ is more complex. The process requires polynomials and, more specifically, irreducible polynomials. These polynomials are chosen so that the field's arithmetic structure remains solid. To start off, we need to find a polynomial of degree 2 that has no solutions in the field $GF(3)$. The polynomial $x^2 + 1$ has no zeroes in this field, and thus, it can be used here. We suppose there is an x, not in this field but in larger fields, that satisfies the equality $x^2 + 1 = 0$ for now. The set built around x and $GF(3)$,

$$\{0, 1, 2, x, x + 1, x + 2, 2x, 2x + 1, 2x + 2\},$$

contains 9 elements; this set, based on the degree 2 polynomial $x^2 + 1$, satisfy the field's arithmetic conditions. For example, the additive inverse of $2x + 1$ is $x + 2$, as $3x + 3 = 0$; the multiplicative inverse of it is $2x + 2$ because

$$(2x + 1)(2x + 2) = 4x^2 + 6x + 2 = x^2 + 2 = 1.$$

The quadratic residue of $GF(9)$ is

$$\{1, 2, x, 2x\},$$

where 2 and $2x$ can be also written as -1 and $-x$. For an arbitrary vertex $v \in GF(9)$, by connecting it to four vertices $v \pm 1$ and $v \pm x$, modulo 3, we obtain the Paley(9) graph in Figure 3.1. Note that we can also consider vertices as a pair (i, j), where i represents variable x's coefficient.

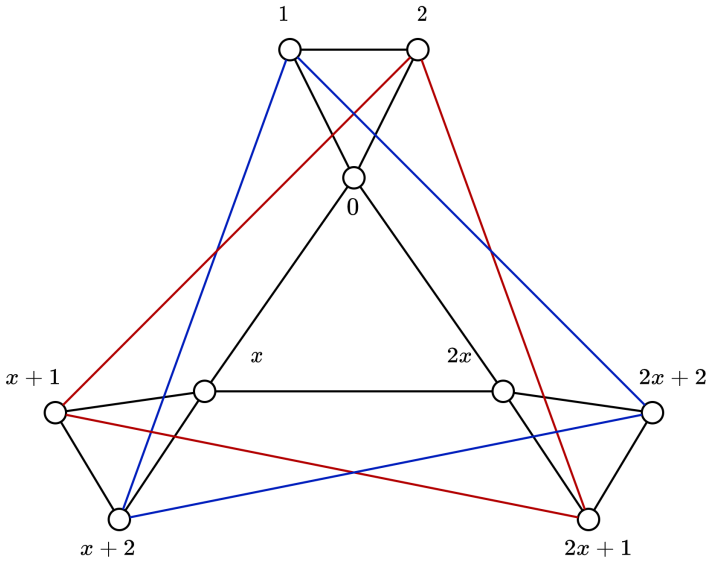

Figure 3.1: Paley(9) with $GF(9)$ vertices. This representation highlights the triangles.

In a Galois field of order q, the size of the quadratic residues' set $QR(q)$ is $\frac{1}{2}(q - 1)$. Every two elements of $GF(q)$ share $\frac{1}{4}(q - 1)$ neighbours unless the two elements differ by one quadratic residue. In that case, they share $\frac{1}{4}(q - 5)$ neighbours, which is one less because of the quadratic residue connecting them. Therefore, for every prime power $q \equiv 1 \pmod 4$, there exists a Paley(q) graph with the parameter set:

$$(q, \frac{1}{2}(q - 1), \frac{1}{4}(q - 5), \frac{1}{4}(q - 1)).$$

In the next section, we will use Paley(9) graph to build a larger, strongly regular graph with $\lambda = 1$ and $\mu = 2$.

3.3 Berlekamp–Van Lint–Seidel graph

The Berlekamp-Van Lint–Seidel is the largest strongly regular graph discovered with $\lambda = 1$ and $\mu = 2$. This graph was constructed using perfect ternary Golay codes [2], which have their roots in the field of data transmission and coding theory. This example highlights the overlap between the strongly regular graphs and coding theory and shows that graphs can be used to generate better coding schemes, depending on the need to transmit data. To gain a better understanding of how this graph was created, we must first complete some preliminary steps.

3.3.1 Preliminaries

While transmitting data, usually binary data containing bits or ternary data containing *trits*, over a noisy channel susceptible to interference, the transmitted bit may experience alteration. The data X being transmitted over a channel can be affected by a noise N (note that the signal being sent is not a discrete value), and the result would be $Y = X + N$, which might be different than expected. Some coding schemes in coding theory are used to reduce the chance of data loss.

As an example, consider a 3-bit message code (c_1, c_2, c_3) and a parity check matrix which we can use to generate

$$\begin{bmatrix} c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

three more parity check bits. After receiving the data, if the bits do not satisfy the parity conditions, we can infer an error has occurred. The word $(c_1, c_2, c_3, c_4, c_5, c_6)$ is considered as a codeword. A *codeword* is a word that is generated by the transmitter; thus, it has not been affected by any noise and is correct.

Within the field of information theory, error-correcting codes serve a dual purpose: not only do they help the detection of errors, but they also enable their correction. To do this, they encode the data into a larger space, and by increasing the distance between the codewords, they offer a high chance of correcting the data 3.2. The distance measure used is the Hamming distance between codewords: the number of positions (indices) in which they differ. The number of elements in a codeword that are not zero determines its weight.

3.3.2 Ternary Golay codes

The ternary Golay code [20], discovered by Marcel J. E. Golay, is one of the remarkable examples of *perfect* codes. Perfect codes are the codes with maximum efficiency, that is, by utilising their redundancy efficiently, these codes enable maximum error correction. In mathematical terms, a code can be considered perfect if the spheres - consisting of words with a maximum Hamming distance that are centred around the codewords - cover the entire space, and every word belongs to the sphere of a unique codeword.

With the ternary Golay code, 6 trits of information can be encoded into 11 trits of data. This coding scheme allows for the correction of up to 2 errors. This can also mean that the

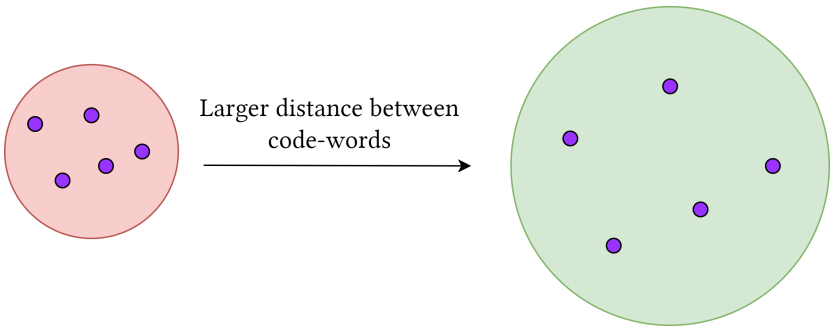

Figure 3.2: Correcting errors by having larger distances between code words

distance between each pair of codewords is at least 5; otherwise, there would be a word that is at a distance of 2 or less of multiple codewords.

This code is defined over the finite field of $GF(3)$, which consists of the elements {0,1,2} or {-1,0,1}. The ternary Golay code can be generated using a generator matrix, denoted by the symbol G, which is equal to:

$$G = [I_6 \mid P] = \left[\begin{array}{cccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 & 1 & 1 \end{array}\right].$$

Notice that the structure of G allows a distance of 5 between codewords. A word w can be represented as a vector with 6 dimensions, where each coordinate is 0,1 or 2.

A *vector space* over a field (in this case, the Galois field GF(3)) is a set of vectors that must satisfy multiple axioms. These axioms include, for vectors u and v and scalar a, associativity ($(u+v)+w = u+(v+w)$), commutativity ($u+v = v+u$), identity element ($u+0 = u$), inverse elements ($u + -u = 0$), and distributivity of scalar ($a(u + v) = au + av$)[1]. We denote the vector space of dimension l on $GF(q)$ with $V(l, q)$. As an example, consider vector $(0, 0, 2)$ of length three in $GF(3)$, so $(0, 0, 2) \in V(3, 3)$.

It is worth noting that this code is classified as linear because of its generation process. That is, for a word $w \in V(6, 3)$, the codeword $w \cdot G \in V(11, 3)$ is generated by multiplication.

3.3.3 Construction of the Berlekamp–Van Lint–Seidel graph

We show two ways of constructing this graph: partitioning and parity.

The partitioning approach

The perfect ternary Golay code can be used to partition the vector space $V(11, 3)$ into 3^5 *cosets* by adding 3^5 different fixed words to all codewords. A subset of vectors H from a vector space V is a subspace if itself is a vector space; that is, it satisfies the axioms we defined previously under the same addition and scalar multiplications operators. Considering a vector $v \in V$, the set of vectors $H_v = \{v + h : h \in H\}$ is a *coset*, with v as its *coset leader*. The coset leader refers to the word with the lowest weight within a coset.

[1]We have omitted some axioms because of their simplicity, but we still believe that the remaining ones convey the general idea.

Let C represent the set of codewords. Considering how the codewords were linearly generated, it is not hard to see that this set forms a subspace of the vector space $V(11, 3)$. Since the ternary Golay code encodes vectors of length 6, we have 3^6 codewords.

Let W_2 be the set of vectors whose weight does not exceed 2. The size of W_2 is

$$1 + 2 \times 11 + 4 \times \binom{11}{2} = 243.$$

This count represents the exact number of vertices we seek for.

We can use the vectors in W_2 to generate the 3^5 cosets we need. Add each vector in W_2 to all codewords in C. The cosets' leaders are the W_2 vectors that generated them. Name cosets based on their leader's non-zero indices. That is, the cosets S_0, $S_{(i,b)}$, and $S_{\{(i,b),(j,b')\}}$, where b and b' are 1 or 2, and i and j are the non-zero trits' indices. These cosets also cover the whole vector space $V(11, 3)$ because each word is at Hamming distance 0, 1, or 2 of a unique codeword, and all possibilities of a 1 or 2 trit errors have been considered.

Since the Golay code is a linear code, by considering a linear combination between two vectors $l_1 = w_1 \cdot G$ and $l_2 = w_2 \cdot G$, we can obtain

$$a \cdot l_1 + b \cdot l_2 = (a \cdot w_1 + b \cdot w_2) \cdot G,$$

which itself is a codeword.

In this partitioning, two vectors are classified as being in the same coset if a codeword is obtained by taking their difference. This fact is useful because it means that any two vectors in the same coset must have a Hamming distance of at least 5. This is because the codewords used to create them met this requirement, and they were both added to the same vector - which happens to be the leader vector of the coset in W_2.

Our graph-building strategy involves assigning each coset a vertex. We connect two vertices if their corresponding subsets (S_x and S_y) have two vectors ($v_x \in S_x$ and $v_y \in S_y$) with a Hamming distance of 1 ($d_H(v_x, v_y) = 1$).

To prove that this generates a $(243, 14, 1, 2)$ strongly regular graph, we take a few steps to show this graph satisfies the k, λ, and μ parameters' conditions.

First, without loss of generality, consider the all-zero vector $z = (0, 0, \cdots, 0)$. Vector z connects S_0 to cosets containing vectors

$$\begin{aligned} N_z = \{&(x, 0, 0, \cdots, 0, 0), \\ &(0, x, 0, \cdots, 0, 0), \\ \cdots &,(0, 0, 0, \cdots, 0, x)\}, \end{aligned}$$

for all $x \in \mathrm{GF}(3) - 0$, because these vectors are at a Hamming distance of one from z; This shows that the degree $deg(S_0)$ is at least 22. To prove no more neighbours exist, consider a vector y' in coset $S_{\{(i,b),(j,b')\}}$. By reducing the leader from y', we obtain a codeword y, which is at least at Hamming distance 5 of codeword $x \in S_0$ where $x \neq y$. Therefore, the Hamming distance $d_H(x, y') \geq 3$, and the two cosets have no vectors of distance one with each other. By generalising this approach for all other W_2 vectors, we can see that all cosets (vertices) have 22 neighbours. Hence, the graph is 22-regular.

Next, without loss of generality, consider vector $x_+ = (+x, 0, 0, \cdots, 0, 0)$, one vector of distance 1 from z. We can observe that since both z and x_+ vectors are at distance 1 of vector

$x_- = (-x, 0, 0, \cdots, 0, 0)$, they cause cosets S_0, $S_{(1,x)}$, and $S_{(1,-x)}$ to form a triangle. All other vectors in set N_z have a Hamming distance of two from x_+ and x_-. Thus, their respective cosets are not adjacent, and consequently, a complete matching with 11 edges exists in S_0's neighbourhood. As a result, the $\lambda = 1$ condition is satisfied.

Also, for a coset that's not S_0's neighbour, its leader, vector xy_+, must have weight 2. Without loss of generality, consider these two non-zero elements at indices 1 and 2, which means $xy_+ = (x, y, \cdots, 0) \in W_2$. This vector xy_+ is at distance 1 of vectors x_+ and y_+. Hence, the coset $S_{(1,x),(2,y)}$ is adjacent to $S_{(1,x)}$ and $S_{(2,y)}$. That being the case, the $\mu = 2$ condition is satisfied.

The parity approach

A code's parity check matrix describes the conditions a word must satisfy to be a valid codeword. The parity matrix of the Golay code is

$$H = \left[\begin{array}{cccccc|ccccc} 2 & 2 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 2 & 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 2 & 1 & 2 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

For a word w, to be a valid codeword, it needs to satisfy the equation $H \cdot w = (0, 0, 0, 0, 0)$.

Each column of matrix H gives us a ternary vector of length 5 in vector space $V(5, 3)$. Naming these vectors x_1, x_2, $\cdots$, x_{11}, we can obtain 11 more vectors by the negation of these vectors. Also, 220 more vectors can be obtained by $\pm x_i \pm x_j$ for all $i, j \in \{1, 2, \cdots, 11\}$ such that $i < j$. By proving that these 242 vectors are distinct and by adding z to this set (name the set θ), we can obtain the whole set of vectors available in vector space $V(5, 3)$.

Since every codeword (but word z) has at least weight 5 (due to the distance two condition), it can be inferred every 4 columns of the H matrix must be independent. To prove this assume that there exist four vectors x_i, x_j, x_k, x_l (where $i, j, k, l \in \{1, \cdots 11\}$ are distinct) such that $ax_i + bx_j + cx_k + dx_l = 0$ for some $a, b, c, d \in GF(3)$, but not all zero. Based on this, we can build a word w, which is a, b, c, d at indices i, j, k, l, respectively. Now, the equation $H \cdot w = 0$ is satisfied, and w becomes a codeword. However, we know each codeword has a weight of 5, leading to a contradiction. As a result, no two vectors in θ are the same, and $\theta = V(5, 3)$.

Now consider vectors $V(5, 3)$ as the vertices of a graph. Note that every two neighbours on this graph will have corresponding vectors that differ by a vector $\pm x_1, \pm x_2, \cdots, \pm x_{11}$.

We can simply infer each vertex has 22 neighbours, as we showed all these summations lead to different values. For any v, and vector x_i, three vertices v, $v + x_i$, and $v - x_i$ form a triangle (note that $v - x_i - x_i = v + x_i$ in $GF(3)$). These 7 edges that form a perfect matching are the only available edges in $N(v)$, and thus, the $\lambda = 1$ condition holds. By adding or subtracting two vectors from vertex v, $v \pm x_i \pm x_j$, we can reach any other vertex of the graph, and also the vertex $v \pm x_i \pm x_j$ share two neighbours with v, which are $v \pm x_i$ and $v \pm x_j$. Hence, the $\mu = 2$ condition is satisfied as well.

We have seen how beautifully a strongly regular graph was derived from a perfect code. More ways of building this graph can be found in [2].

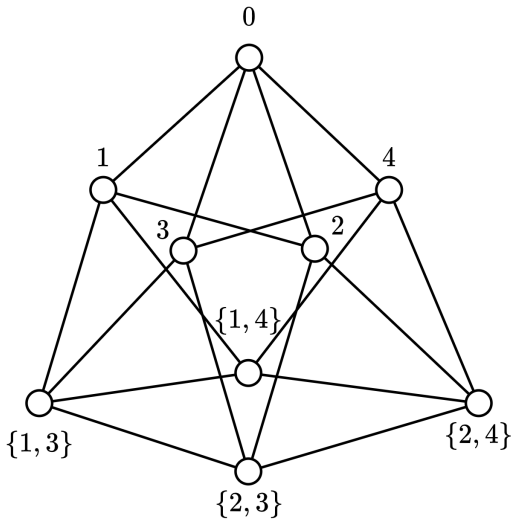

Figure 3.3: Paley(9) graph: A representation of the graph emphasising the quadrilateral formations in the structure of the graph.

3.4 Subgraphs and patterns

This section aims to explore the possibilities of different subgraphs and patterns, in a possible Conway-99 graph. The goal is to reduce the search space we have to explore.

3.4.1 Paley(9) pattern

There is a special property associated with the Paley(9) graph. This can be the only possible subgraph that satisfies the $\lambda = 1$ and $\mu = 2$ conditions within the subgraph itself. This is because there are no other strongly regular graphs with $\lambda = 1$ and $\mu = 2$ with less than 99 vertices.

To make the writing simpler, in strongly regular graphs with $\lambda = 1$ and $\mu = 2$, for a vertex v, we sometimes refer to v's non-neighbours as a pair of its neighbours since each non-neighbour is adjacent to two specific neighbours. For example, in figure 3.3, vertex $\{1, 3\}_0$ is 0's non-neighbour which is adjacent to 1 and 3 in $N(0)$.

Definition 12 (Paley(9) pattern)**.** *The Paley(9) pattern is established in an* $(n, k, 1, 2)$ *strongly regular graph* G *when for all vertices* $v \in V$ *and two edges* $\{v_1, v_2\}, \{v_3, v_4\} \in E(N(v))$*, the induced subgraph by vertices*

$$P_v = \{v, v_1, v_2, v_3, v_4, (v_1, v_3)_v, (v_1, v_4)_v, (v_2, v_3)_v, (v_2, v_4)_v\}$$

would be a Paley(9) graph.

Lemma 3.4.1. *The Paley(9) pattern is present in the Berlekamp-Van Lint-Seidel graph.*

Proof. In the parity approach 3.3.3, we noted that a vector v has 22 neighbouring vectors, which are $v \pm x_i$ for $1 \leq i \leq 11$. By considering a vector v and two parity matrix columns, x_i and x_j, it is clear that the subgraph shown in 3.4 will be created. □

We now prove this pattern cannot appear in any possible $(99, 14, 1, 2)$ strongly regular graphs.

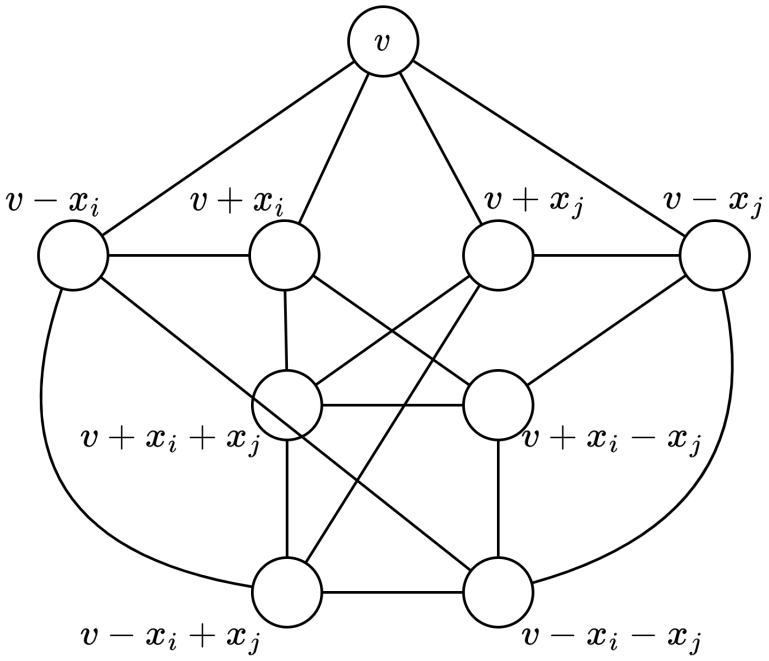

Figure 3.4: Paley(9) graph: A representation obtained from the Golay ternary code parity matrix.

Theorem 3.4.2. *If a* $(99, 14, 1, 2)$ *strongly regular graph exists, it cannot follow the Paley(9) pattern.*

Proof. We proceed by assuming the pattern holds and then derive a contradiction out of it. We start off by considering a Paley(9) subgraph, as in Figure 3.5. Name vertex 0's neighbours with numbers 1 to 14, such that vertices $2i - 1$ for $1 \leq i \leq 7$ are adjacent to vertices $2i$.

Consider vertex 5 in the graph. According to the $\mu = 2$ requirement, each of the sets $N(i)$ for $1 \leq i \leq 4$ must contain one of 5's neighbours, named (1,5), $\cdots$, (4,5). Considering two triangles $\{0, 1, 2\}$ and $\{0, 5, 6\}$, based on the Paley(9) pattern, we can infer edges

$$\{\{(1, 5), (1, 6)\}, \{(2, 5), (2, 6)\}, \{(1, 5), (2, 5)\}, \{(1, 6), (2, 6)\}\}$$

exist, which form a C_4. By applying the same reasoning to triangles $\{0, 3, 4\}$ and $\{0, 5, 6\}$ we can obtain more edges. We call this the Paley(9) parallelism approach, as the two triangles $\{0, 1, 2\}$ and $\{5, (1, 5), (2, 5)\}$ are adjacent with each other.

Next, we name the vertex adjacent to $(1, 5)$ in the sets $N_{(1,3)}$ and $N_{(1,4)}$ as $(1, 3, 5)$ and $(1, 4, 5)$. We can now show these two vertices must be adjacent. To do this, consider two triangles $\{1, (1, 3), (1, 4)\}$ and $\{1, (1, 5), (1, 6)\}$. We can infer that the edges

$$\{\{(1, 3, 5), (1, 4, 5)\}, \{(1, 3, 5), (1, 3, 6)\}, \{(1, 4, 5), (1, 4, 6)\}, \{(1, 3, 6), (1, 4, 6)\}\}$$

exist.

If we apply the parallelism approach used for vertex 0 to other vertices and generalise it, we can deduce that the edge present among the sets $N_{(1,3)}$ and $N_{(2,3)}$ must create a triangle with vertices in N_3.

Vertex 5 needs to have more neighbours and, therefore, more triangles. Thus far, we have inferred the existence of three triangles in which vertex 5 appears, which are

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

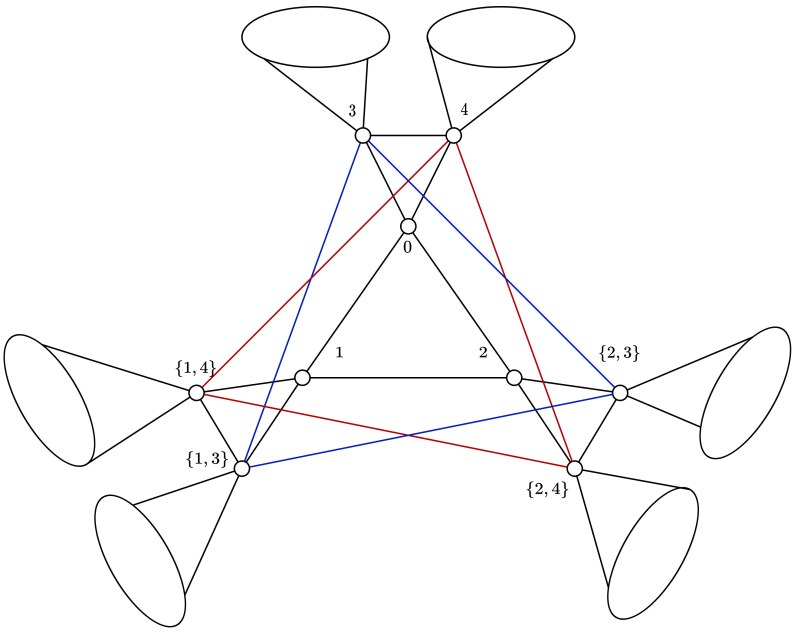

Figure 3.5: Building a $(99, 14, 1, 2)$ strongly regular graph based on the Paley(9) pattern.

Vertex 5 must have two neighbours in $N_{1,3}$. Name one vertex as $(1, 3, x)$. This neighbour must form a triangle with 5. Let us consider the possibilities for the third vertex v_3 of this triangle:

1. Initially, it is not possible for vertex v_3 to be a part of $N_{1,3}$ because it would break the condition $\lambda = 1$, as the vertices 5 and $(1, 3)$ would become common neighbors of v_3 and $(1, 3, x)$.
2. The edges connecting $N_{(1,3)}$ and $N_{(1,4)}$ create a triangle with vertices in N_1 (parallelism). Thus, the third vertex cannot belong to the set $N_{1,4}$.
3. The edges among $N_{(1,3)}$ and $N_{(2,3)}$ form a triangle with vertex 3 (parallelism), and cannot be form a triangle with vertex 5.

Therefore, the only possibility is to have a triangle between vertices $\{5, (1, 3, x), (2, 4, y)\}$, where $(2, 4, y) \in N_{(2,4)}$.

Consider two triangles $\{5, (1, 3, x), (2, 4, y)\}$ and $\{5, 0, 6\}$. Without the loss of generality, suppose $(1, 3, x)$ is adjacent to vertex 7. This can simply be generalised for all other vertices, but 6, which will break the $\lambda = 1$ condition. These two triangles must also form a Paley(9) graph as shown in figure 3.6, such that vertex 6 appears in triangle $\{6, u, v\}$. We now examine the possible sets to which vertices v and u can belong.

1. As a result of the $\lambda = 1$ condition, $v \notin N_{2,4}$ and $u \notin N_{1,3}$.
2. As a result of the $\lambda = 1$ condition, $v, u \notin N_0$.
3. As a result of the $\mu = 2$ condition, $v, u \notin N_1, N_2, N_3, N_4$.
4. As a result of parallelism, $v \notin N_{2,3} \cup N_{1,4}$ and $u \notin N_{1,4} \cup N_{2,3}$
5. As a result of the $\lambda = 1$ condition, $u \notin N_{1,3}$ and $v \notin N_{2,4}$.

Therefore, the only possibility that remains is that u must belong to $N_{2,4}$ and v must belong to $N_{1,3}$. This leads to a contradiction as $v = (1, 3, x')$ and $(1, 3, x)$ share three neighbours:

$$\{(1, 3), (2, 4, y), (2, 4, y')\}.$$

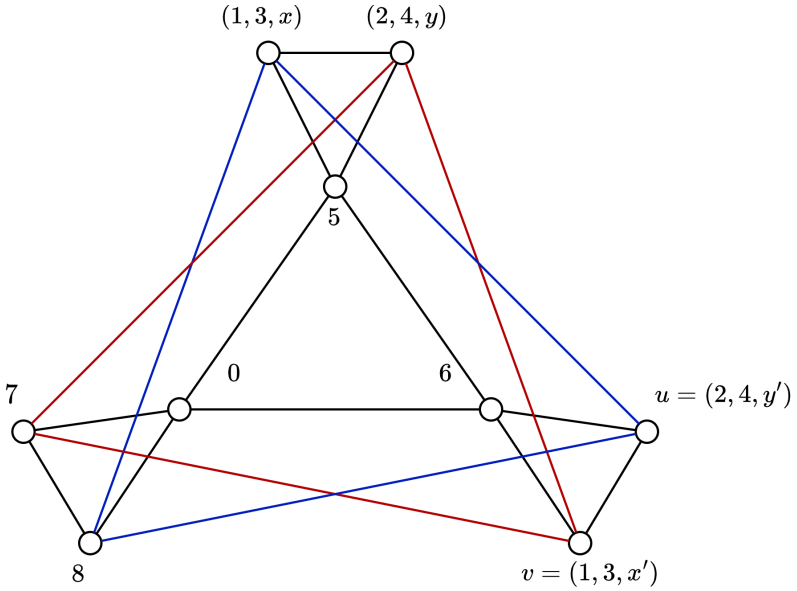

Figure 3.6: Examining potential neighbouring sets for v and u in a Paley(9) subgraph. This figure also depicts a triangular view of the Paley(9) graph.

We can now infer that the Paley(9) pattern cannot hold in a $(99, 14, 1, 2)$ strongly regular graph. □

3.4.2 Paley(9) subgraph

In Theorem 3.4.2, we proved possible $(99, 14, 1, 2)$ strongly regular graphs cannot follow the Paley(9) pattern. With a similar approach, we can prove a $(99, 14, 1, 2)$ strongly regular graph cannot contain eleven independent Paley(9) subgraphs. We omit the proof here:

Theorem 3.4.3. *If a* $(99, 14, 1, 2)$ *strongly regular graph exists, it cannot contain eleven independent Paley(9) subgraphs.*

By observing these, we conjecture that no Paley(9) subgraphs can exist in a possible $(99, 14, 1, 2)$ strongly regular graph:

Conjecture 3.4.4. *In a* $(99, 14, 1, 2)$ *strongly regular graph instance, there cannot be any Paley(9) subgraph.*

By demonstrating this, we establish the infeasibility of a subgraph that meets the neighbouring conditions *within itself.*

3.4.3 Triangular view

An interesting and possible way to approach the Conway-99 problem is to focus on its triangles. As we studied, the $\lambda = 1$ property makes triangles one of the fundamental units of this graph. In this part, we want to study how the triangles interact with each other and affect the overall graph's structure.

As can be seen in Figure 3.6, the Paley(9) graph consists of six triangles. To construct the triangular graph, we will follow a two-step process. Firstly, we will assign a vertex to each triangle, and secondly, we connect the triangles that share a vertex.

Lemma 3.4.5. *The triangular graph of an* $(n, k, 1, 2)$ *strongly regular graph, consists of* $\frac{nk}{6}$ *vertices and is* $\frac{3k-6}{2}$*-regular. Every two neighbours share* $\frac{k}{2} - 2$ *neighbours. Also, every two non-neighbour can share at most three neighbours.*

Proof. In the initial graph with $\lambda = 1$ and $\mu = 2$ parameters, each vertex v, has a neighbourhood of $\frac{k}{2}$ edge perfect matching, which means it is inside $\frac{k}{2}$ triangles. By adding up the number of triangles adjacent to each vertex, we get $n\frac{k}{2}$. Since each triangle is counted three times, the actual number of triangles is $\frac{nk}{6}$.

Consider one triangle. Each vertex of it has $\frac{k}{2} - 1$ more triangles adjacent to it. Also, because of the $\mu = 2$ condition, no triangle is counted twice. Therefore, each triangle is adjacent to exactly $\frac{3k-6}{2}$ triangles.

Consider two adjacent triangles, that is, they share a vertex v. It becomes straightforward to infer that no edge can exist among the four other vertices as such a scenario would violate the $\lambda = 1$ condition. Therefore, the two adjacent triangles are only adjacent to the other $\frac{k}{2} - 2$ triangles of vertex v.

Among two non-adjacent triangles of the initial graph, each vertex is adjacent to at most one vertex of the other triangle. Therefore, a maximum of three edges can exist between two non-adjacent triangles. Hence, in the triangular graph, every non-neighbour pair has at most three shared neighbours. □

Although this is not the original description of the μ parameters, we use it somewhat differently here. In this graph, we have $\lambda = \frac{k}{2} - 2$, and $\mu \leq 3$. The notation $\mu \leq 3$ here shows that two non-neighbours have at most three common neighbours.

Therefore, for a $(99, 14, 1, 2)$ strongly regular graph to exist, we first must have a graph G with parameter set $(231, 18, 5, \leq 3)$.

Observation 3.4.6. *The triangular graph of Paley(9) is* $K_{3,3}$*. This graph has 6 vertices, is 3-regular, and is triangle-free.*

This graph is a trivial, strongly regular graph.

Chapter 4

Computer search using SAT solvers

Computer search has been historically used to find instances of strongly regular graphs or prove they do not exist. Through the usage of backtracking algorithms, Coolsaet et al. [16] successfully discovered all potential non-isomorphic strongly regular graphs with a parameter set of $(45, 12, 3, 3)$. Behbahani and Lam [1] used matrices and strongly regular graphs' algebraic structures to study automorphism groups. They stated that they had used multiple years of computation using hundreds of Intel-based Linux machines until they were able to produce their results.

In 1989, Bussemaker et al. [10] used computation power to prove the non-existence of a strongly regular graph with parameters $(49, 16, 3, 6)$. They started by eliminating various possibilities through theoretical analysis and then used a computer to search the remaining space. Searching through that space, although not computationally intensive, was not doable by hand.

A fairly recent result was obtained by Gritsenko [21], where a $(65, 32, 15, 16)$ strongly regular graph was discovered. The idea was to construct a matrix with blocks and then circulate through those blocks. More methods and results can be seen in [28] [27]. In [7], Brouwer has compiled a great amount of data on strongly regular graphs.

In this chapter, we will delve into the SAT problem and explore how to encode the search problem for strongly regular graphs as a SAT formula. We will cover multiple types of SAT formulas for different encodings.

4.1 SAT problem and SAT solvers

SAT solvers are considered black-box tools in this research project. Although sufficient in many cases, it does not offer the level of information and efficiency needed to solve the Conway-99 problem.

The following is a basic definition of the *Boolean satisfiability*, abbreviated as *SAT*, problem:

Definition 13 (SAT problem)**.** *Given a Boolean formula F with variables $x_1, x_2, \ldots, x_n$ and logical operators $\wedge$ (AND), $\vee$ (OR), and $\neg$ (NOT), is there an assignment of truth values $0, 1$ to the variables such that F evaluates to* `True`*? If so, the formula F is called satisfiable, and unsatisfiable otherwise.*

Boolean formulas can be represented in many forms; however, the most commonly used representation is the *conjunctive normal form* (CNF).

Definition 14 (CNF). *A CNF formula is a conjunction (logical AND) of multiple clauses, such that each clause is a disjunction (logical OR) of literals. Literals are either variables or their negation.*

The use of CNF in SAT is simple and straightforward, providing a clear representation of the formula. In fact, that is why many SAT solvers take their inputs in CNF; It should also be noted that any SAT formula can be transformed into CNF using boolean algebra.

One simple formula can be the following:

$$\phi = (x_1 \wedge x_2) \vee \neg x_3,$$

which can be transformed into a CNF version:

$$\phi = (x_1 \vee \neg x_3) \wedge (x_2 \vee \neg x_3).$$

SAT solvers use different algorithms to determine the satisfiability of a formula.

In complexity theory, the SAT problem plays a pivotal role. Since NP-Hard problems are all reducible to a SAT instance in polynomial time, SAT solvers have numerous applications. The development and advancements in SAT solver algorithms have led to more efficient and faster solutions over the years. SAT solvers are a powerful tool widely used in various domains, such as hardware and software verification, planning and scheduling, cryptography, and artificial intelligence. In recent years, the development of parallel and distributed SAT solvers has enabled the solving of even larger and more complex instances of the SAT problem. A specific competition is held each year, known as the international SAT competition, to test, benchmark, compare, and suggest new SAT algorithms.

The development of the Davis-Putnam-Logemann-Loveland (DPLL) algorithm, which is a backtracking algorithm, was a key moment in the history of SAT solvers. Since then, many variations and heuristics have been applied to explore the space of possible assignments more efficiently. These improvements include conflict-driven clause learning (CDCL), non-chronological backtracking, and clause deletion.

One of the most significant improvements to SAT algorithms has been the use of preprocessing techniques to simplify the input formula before solving it [4]. Preprocessing can involve various operations such as unit propagation (removing clauses that contain single literals), clause learning, variable elimination, and symmetry breaking. These techniques can reduce the size of the input formula and simplify the search space, leading to faster solving times and more efficient use of resources.

Conflict-Driven Clause Learning (CDCL) SAT solvers work by incrementally assigning values to boolean variables. These solvers maintain the constraints with another set of clauses derived from the previous set. When a conflict arises, meaning the algorithm can no longer continue assigning variables to satisfy the SAT formula, the algorithm backtracks and learns from the conflict. In the learning process, it adds a new clause that blocks the previous assignment causing the conflict. This continues until a solution is found or it is determined that no valid assignment exists. Algorithm 1 presents a simple pseudo-code of the process.

Algorithm 1: The Conflict-Driven Clause Learning (CDCL) Algorithm

```
Function CDCL(Formula F):
    while F variables are not entirely assigned do
        state ← save the state
        Select a variable in F and assign it to True or False
        Simply do a unit propagation
        conflict ← Find the conflicts in F
        if conflict = ∅ then
            continue
        clause ← derive a new clause from the conflict
        Add clause to F
        backtrack to state
```

One great example of a practical SAT solver is CaDiCaL [3], an open-source project also available on GitHub. It is a high-performance solver used in many applications.

The huge search space involving the Conway-99 problem is deemed impractical to search at the moment. With basic exhaustive approaches, there are a total of $2^{\binom{99}{2}}$ potential graphs; obtained by placing or not placing each edge out of the $\binom{99}{2}$ edges.

In this research, we have implemented tools and programs for encoding the Conway-99 problem and, more generally, encoding the search for all strongly regular graphs as SAT problems. Two approaches are explained in Sections 4.3 and 4.4. However, before getting into the problem's encoding, we explain in Section 4.2 why we are utilising SAT solvers and what makes them special.

4.2 Symmetry-breaking techniques

In simple words, symmetry-breaking tries to avoid considering multiple identical copies of the same object. It avoids redundant computation by ensuring the solver does not explore multiple solutions that are essentially the same due to their symmetry.

This approach is usually made by adding symmetry-breaking clauses to our SAT instance. The following example illustrates this and similar approaches:

Example 4.2.1. *Consider a SAT problem that involves three boolean variables, namely x, y, and z:*

$$(x \vee y) \wedge (x \vee \neg z) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y \vee z).$$

The symmetry between x and y is visible. By swapping these, the SAT problem remains unchanged. One way to break this symmetry is to add the clause $(\neg x \vee y)$. By considering this clause, the case $x =$ `True` and $y =$ `False` is not considered.

However, symmetry detection itself is not an easy task. This problem is essentially equivalent to the graph isomorphism problem. The graph isomorphism problem asks whether the two input graphs are isomorphic. This problem is classified as NP [26].

In the realm of graph theory, SAT solvers have been employed to solve many problems. One common usage is in graph colouring problems, which contain many symmetrical answers to it.

When dealing with strongly regular graphs, the application of SAT solvers can prove to be valuable in breaking symmetries. Many of these graphs possess a great amount of sym-

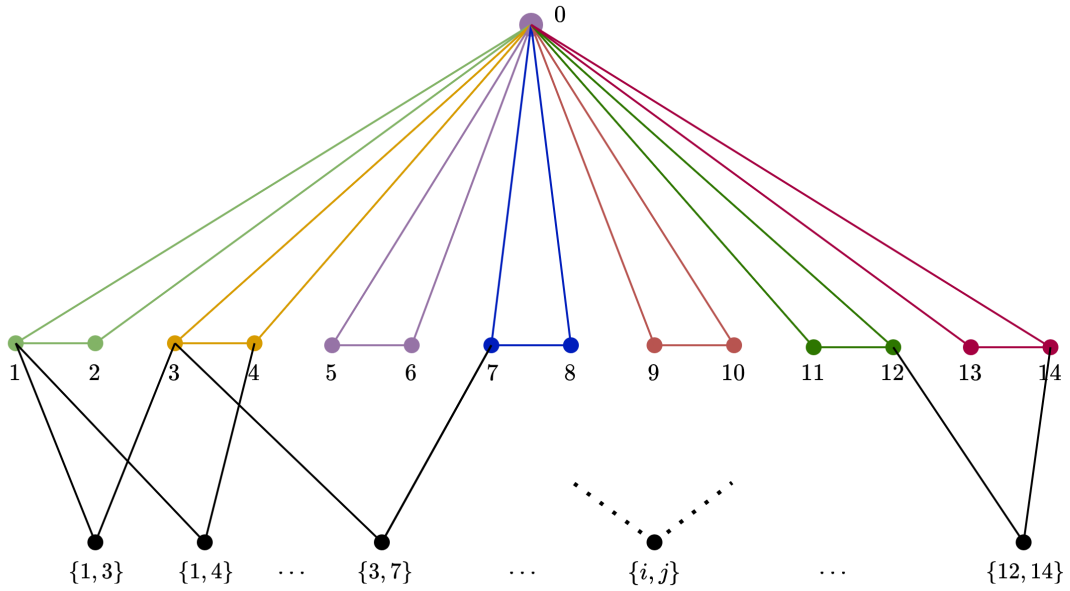

Figure 4.1: Conway99 graph's view from one vertex. Each non-neighbour can be represented as a set with cardinality two.

metry, which can be eliminated through the addition of extra conditions. This process can effectively decrease the search area.

A basic symmetry-breaking example for the Conway-99 problem can be the graph's view from one vertex, as shown in Figure 4.1. Consider a possible solution as graph $G = (V, E)$. For a vertex $v \in V$, and two neighbours of it $i, j \in N(v)$ where i and j are not connected, there must exist a unique vertex in $V - N[v]$ connected to i and j, so that the $\mu = 2$ condition between vertices i and j is satisfied. Also, again because of the $\mu = 2$ condition, this vertex cannot be connected to any other vertices in $N(v)$. Label this vertex as $\{i, j\}$. We can now label all vertices in $V - N[v]$ with an unordered pair of non-neighbour vertices in $N(v)$. By labelling these vertices as shown in Figure 4.1, the problem reduces to finding the edges among the $\frac{14}{2} - 7 = 84$ unordered pairs.

4.3 CNF Clauses

In this section, we encode our problem into a CNF instance.

Lemma 4.3.1. *By employing* $\binom{n}{\theta+1}$ *CNF clauses, we can generate a formula that is* `True` *when at most* θ `True` *instances exist among* $x_1, x_2, \ldots, x_n$ *boolean variables.*

Proof. For each subset of size $\theta + 1$ in variables, we add the following clause (without the loss of generality, consider the subset as $\{x_1, \ldots, x_{\theta+1}\}$):

$$(\neg x_1 \vee \neg x_2 \vee \ldots \vee \neg x_{\theta+1}),$$

to the final formula F. The formula validates to `True` if we have θ or less `True` variables. □

Lemma 4.3.2. *By employing* $\binom{n}{\theta-1} + \binom{n}{\theta+1}$ *CNF clauses, we can generate a formula that is* `True` *when exactly* θ `True` *instances exist among* $x_1, x_2, \ldots, x_n$ *boolean variables.*

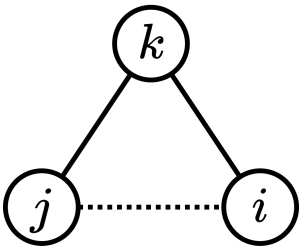

Figure 4.2: A shared neighbour k for vertices $\{i, j\}$. The edge $\{i, j\}$ may or may not be present.

Proof. Similar to the Lemma 4.3.1, we will use $\binom{n}{\theta+1}$ to ensure we have at most θ `True` variables.

To proceed, we utilise $\binom{n}{n-\theta+1}$ more clauses to guarantee that only a maximum of $n - \theta$ variables are marked as `False`. The CNF formula generated employs $\binom{n}{\theta-1} + \binom{n}{\theta+1}$, ensuring that precisely θ variables are identified as `True`. □

We start by encoding the small, simple Paley(9) graph as a starting point for the encoding schemes of strongly regular graphs.

4.3.1 Paley(9)

In order to begin the encoding process, we at least need $\binom{9}{2} = 18$ variables; each variable represents an edge and is denoted by the notation $e_{i,j}$. These boolean variables determine whether an edge exists between vertices i and j (vertices are numbered 1 to n), with a value of `True` indicating its existence.

To ensure the 4-regularity condition holds, for each vertex v, we need to ensure that exactly 4 boolean variables out of the set $\{e_{1,v}, e_{2,v}, \ldots, e_{n,v}\} - \{e_{v,v}\}$ are `True`. Using the approach in Lemma 4.3.2, we can use $9 \cdot (\binom{13}{3} + \binom{13}{5})$ clauses to ensure regularity.

To encode shared neighbours, we use Cherry subgraphs (other names include P_3, claw, and angle). The boolean variables $c_{i,j,k}$, for $i, j, k \in \{1, 2, 3, \ldots, n\}$ and $i < j$, represent the existence of Cherry in Figure 4.2.

To encode Cherries' existence, we need $\binom{n}{2}(n-2)$ number of variables. Each Cherry variable $c_{i,j,k}$ holds, if both edges $e_{i,k}$ and $e_{j,k}$ are `True`, regardless of the variable $e_{i,j}$:

$$(e_{i,k} \wedge e_{j,k}) \longleftrightarrow c_{i,j,k} \equiv (\neg e_{i,k} \vee \neg e_{j,k} \vee c_{i,j,k}) \wedge (\neg c_{i,j,k} \vee e_{j,k}) \wedge (\neg c_{i,j,k} \vee e_{i,k}).$$

If the edge $e_{i,j}$ holds, we need to satisfy the $\lambda = 1$ condition. To do this, we consider $\binom{n-2}{\lambda+1} + \binom{n-2}{\lambda-1}$ number of clauses to make sure exactly $\lambda = 1$ Cherries exist among the vertices i and j. Then, by adding a $\neg e_{i,j}$ literal to these clauses, we make sure this only applies when the edge appears.

Similar to the λ condition, to satisfying the $\mu = 2$ condition involves having $\binom{n-2}{\mu-1} + \binom{n-2}{\mu+1}$ clauses per edge. To ensure these conditions are met when $e_{i,j} =$ `False`, we simply include the literal $e_{i,j}$ in each clause.

Hence, in total for each edge, we have made:

$$\binom{n-2}{\mu-1} + \binom{n-2}{\mu+1} + \binom{n-2}{\lambda+1} + \binom{n-2}{\lambda-1}$$

number of clauses.

4.3.2 Conway-99

Considering this naive approach, the number of clauses needed is huge for this graph and practically cannot be generated. The number of clauses, just to ensure regularity for one vertex, is $\binom{98}{13} + \binom{98}{15}$ which is approximately $\approx 1.8 \times 10^{17}$. However, in Theorem 4.5.3, we prove that k-regularity is an excessive condition; hence, many clauses can be removed.

Theorem 4.3.3. *Let $G = (V, E)$ denote a graph with 99 vertices. If all pairs of neighbouring vertices share $\lambda = 1$ neighbours, and every two non-neighbours have two common neighbours $\mu = 2$, then graph G is 14-regular. Consequently, G is a strongly regular graph.*

Proof. Consider a vertex v with d_v neighbours. Since the induced subgraph of $N(v)$ has to be a complete matching, degree d_v must be even. Also, each pair of non-neighbours in $N(v)$ needs to share a neighbour in $V - N[v]$. These neighbours must be distinct; if not, there would be a vertex $x \in V - N[v]$ that is connected to more than 2 neighbours of $N(v)$. Thus, the following equality must hold:

$$|V - N[v]| = \binom{d_v}{2} - \frac{d_v}{2} \Rightarrow 99 - d_v - 1 = \binom{d_v}{2} - \frac{d_v}{2} \Rightarrow d_v = 14,$$

and all vertices are of degree 14. □

If we limit our consideration to clauses only necessary for the λ and μ parameters, as well as those needed for Cherries, we would still need $\approx 7e8$ number of clauses, which requires 22GB, a huge amount of storage.

To further limit the storage and the number of clauses needed, we decided to stabilise one vertex and its neighbourhood. In figure 4.1, by stabilising one vertex and its neighbours, we end up with exactly 84 vertices, such that each can be represented by an unordered pair $\{i, j\}$, where $i, j \in \{1 \dots 14\}$ and $\{i, j\} \notin N[v]$. We can use this as a starting point to create a viable input for SAT solvers to handle. Unfortunately, the search space is still huge, and CNF SAT solvers cannot entirely explore it.

4.4 Pseudo-Boolean SAT clauses

Unlike the CNF format, the pseudo-boolean (PB) representation is much more efficient in our problem. The pseudo-boolean form is a more generalised form of CNF, which allows for linear equations, equalities or inequalities. In these problems, the goal is to assign boolean values to the variables to satisfy all the equations. The problems in PB format are similar to constraint satisfaction problems (CSP) but with the condition that the domain of their variables is binary bits.

Another usefulness of this format is its optimisation capabilities, where the goal is, along with satisfying the assignments, to minimise or maximise certain linear objective functions.

When dealing with the Conway-99 problem, the regularity condition used to be a stopping point. However, we can simply overcome this problem with linear equations. To better understand this, let us consider the following example:

Example 4.4.1. *In order to express the k-regularity requirement of a vertex v, we can represent it as follows:*

$$+1\ e_{v,1} + 1\ e_{v,2} + 1\ e_{v,3} + \ldots + 1\ e_{v,n} = k.$$

Applying this to all vertices, we can guarantee the k-regularity.

Note that the variable $e_{v,v}$ does not exist due to the absence of self-loops.

In pseudo-boolean encoding, there are two different ways to encode the problem: linear and non-linear. A linear instance is similar to what we previously saw in Example 4.4.1. In non-linear instances, it is possible to have the multiplication of variables in the constraints. In the binary form, this is the same as a conjunction. We now encode our problems in both methods.

Non-linear encoding

The minimum number of variables $\binom{n}{2}$ is enough for non-linear instances.

Lemma 4.4.2. *The equation*

$$-\lambda\ e_{i,j} + 1\ e_{i,j}e_{i,1}e_{1,j} + 1\ e_{i,j}e_{i,2}e_{2,j} + \ldots + 1\ e_{i,j}e_{i,n}e_{n,j} = 0 \tag{4.1}$$

verifies that the λ parameter condition for the pair $\{i, j\}$ holds. Also, the equation

$$\begin{aligned} &+ 1\ e_{i,1}e_{1,j} + 1\ e_{i,2}e_{2,j} + \ldots + 1\ e_{i,n}e_{n,j} \\ &\qquad - 1\ e_{i,j}e_{i,1}e_{1,j} - 1\ e_{i,j}e_{i,2}e_{2,j} - \ldots - 1\ e_{i,j}e_{i,n}e_{n,j} + \mu e_{i,j} = \mu \end{aligned} \tag{4.2}$$

verifies the μ parameter condition.

Proof. The multiplication of $e_{i,k}$ and $e_{k,j}$ represents the existence of Cherry formed by vertex k among vertices $\{i, j\}$.

In case the edge $e_{i,j}$ exists, the equation $\Sigma_k e_{i,k}e_{j,k} = \lambda$ must hold, that is, we need λ Cherries. Otherwise, the equation $\Sigma_k e_{i,k}e_{j,k} = \mu$ needs to hold.

Equation 4.1 validates to 0 when $e_{i,j}$ is `False`. However, when $e_{i,j}$ is `True`, it ensures the $\Sigma_k e_{i,k}e_{j,k} = \lambda$ equality.

Equation 4.2 holds whenever $e_{i,j}$ is `True`. Otherwise, the second part of the equation evaluates to 0, and μ angles must exist between i and j. □

Linear encoding

For a linear pseudo-boolean solver, we define two new sets of variables:

1. Angles $a_{i,j,k}$ evaluates to `True`, when $e_{i,j}$ is `False`, but the Cherry $e_{i,k}e_{k,j}$ exists.
2. Triangles $t_{i,j,k}$, which hold `True` when both the Cherry and the edge exist.

Lemma 4.4.3. *The following inequalities define the angles we require:*

$$\begin{aligned} &- 1\ a_{i,j,k} - 1\ e_{i,j} \geq -1, \\ &- 1\ a_{i,j,k} + 1\ e_{i,k} \geq 0, \\ &- 1\ a_{i,j,k} + 1\ e_{k,j} \geq 0, \\ &+ 1\ a_{i,j,k} + 1\ e_{i,j} - 1\ e_{i,k} - 1\ e_{k,j} \geq -1, \end{aligned}$$

and for triangles:

$$\begin{aligned}
-1\ t_{i,j,k} + 1\ e_{i,j} &\geq 0,\\
-1\ t_{i,j,k} + 1\ e_{i,k} &\geq 0,\\
-1\ t_{i,j,k} + 1\ e_{k,j} &\geq 0,\\
+1\ t_{i,j,k} - 1\ e_{i,j} - 1\ e_{i,k} - 1\ e_{k,j} &\geq -2.
\end{aligned}$$

Proof. The first three angles and triangles equations ensure that $a_{i,j,k}$ and $t_{i,j,k}$ can only be `True` when the corresponding edges satisfy the needs. The fourth line ensures that the angle and the triangle must be `True` when all three edges meet the needs. □

The rest of the encoding is similar to Lemma 4.4.2. For each pair of vertices ij, we need the equation

$$+\mu e_{ij} + 1\ a_{i,j,1} + 1\ a_{i,j,2} \ldots + 1\ a_{i,j,n} = \mu,$$

when the pair is not adjacent, and the equation

$$-\lambda e_{ij} + 1\ t_{i,j,1} + 1\ t_{i,j,2} \ldots + 1\ t_{i,j,n} = 0,$$

when the pair is adjacent.

4.4.1 Triangular view

In Section 3.4.3, we showed that the existence of a $(99, 14, 1, 2)$ strongly regular graph depends on a graph with parameters $(231, 18, 5, \mu \leq 3)$. To find this graph, we can encode it using our previous approaches; however, for the $\mu \leq 3$ condition, Equation 4.2 must turn into inequality:

$$\begin{aligned}
+1\ e_{i,1}e_{1,j} + 1\ e_{i,2}e_{2,j} + \ldots + 1\ e_{i,n}e_{n,j}&\\
- 1\ e_{i,j}e_{i,1}e_{1,j} - 1\ e_{i,j}e_{i,2}e_{2,j} - \ldots - 1\ e_{i,j}e_{i,n}e_{n,j} + \mu e_{i,j} &\leq \mu. \quad (4.3)
\end{aligned}$$

4.4.2 Paley(9) subgraph

In Theorem 3.4.3, we showed there cannot be 11 separate Paley(9) subgraphs. To encode this, we simply divided vertices into 11 sets of 9 vertices and set the edge boolean variables to `True` for edges and `False` for non-edges. However, the SAT solver was unable to infer the unsatisfiability (after a 12-hour running time). This gives us a good example of SAT solvers' weaknesses when it comes down to unsatisfiable formulas, as they will be doing an exhaustive search with minor optimisations. We will go through our experiments more deeply in Section 5.

Corollary 4.4.4. *Since we know the Paley(9) pattern cannot hold, we can adjust our initial configuration and assign edge values to ensure that there is at least one vertex that does not follow the Paley(9) pattern.*

4.5 Parameter redundancy

In Theorem 4.3.3, we saw that the 14-regular condition is redundant; hence, the Conway-99 problem based only on λ and μ parameters is equivalent to the main problem. A question that arises is that can we generalise this?

The Petersen graph and its parameters $(10, 3, 0, 1)$ serve us as a counterexample.

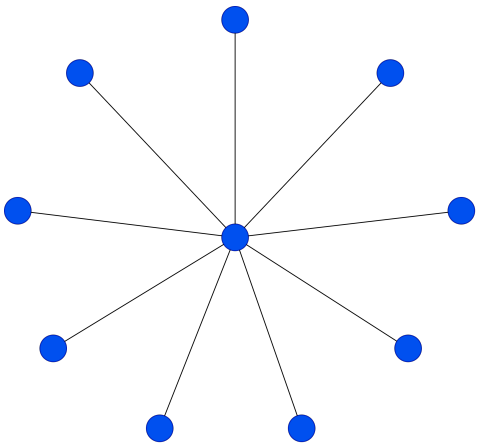

Figure 4.3: Star graph $K_{1,9}$, a graph with the same λ and μ parameters as the Petersen graph but not regular.

Example 4.5.1. *The graph depicted in 4.3 satisfies both $\lambda = 0$ and $\mu = 1$ conditions. However, it is not a regular graph.*

Although this is a counterexample, its nature is similar to trivial strongly regular graphs.

Lemma 4.5.2. *Let $G = (V, E)$ be a graph with 10 vertices with parameters $\lambda = 1$ and $\mu = 2$, and a vertex v with degree $deg(v) = 3$. Then G must be 3-regular.*

Proof. Every two non-adjacent vertices x and y share one neighbour. Since each vertex $z \in N(x) - N(y)$ must share exactly $\mu = 1$ neighbour with y, there must be a matching among their neighbours.

There must be 6 edges among the two sets $V - N[v]$ and $N(v)$. For every pair of v's neighbours, as they are non-adjacent, there must be a matching between their neighbours. Ergo, each of them is connected to two vertices in $V - N[v]$, and all of them have a degree of 3. By continuously applying this, we can infer the graph is 3-regular. □

Although the Petersen graph served as a counterexample, the statement holds for $\mu > 1$:

Theorem 4.5.3. *Let $G = (V, E)$ denote a graph with n vertices. If the graph has λ and μ parameters, and $\mu > 1$, then the graph is regular and, consequently, strongly regular.*

Proof. Let u and v be two adjacent vertices. They share λ neighbours. Let e_v be the number of edges between $N(v) \cup N(u)$ and $N(v) - N(u)$, and let e_u be the number of edges between $N(v) \cup N(u)$ and $N(u) - N(v)$. Also, e_λ is the number of edges within the set $N(v) \cup N(u)$ and e_μ is the number of edges between $N(u) - N(v)$ and $N(v) - N(u)$.

Due to the λ parameter, the induced subgraphs $N(u)$ and $N(v)$ form a λ-regular graph, and thus,

$$e_v + 2e_\lambda = e_u + 2e_\lambda = \lambda(\lambda - 1).$$

We can infer $e_u = e_v$. Also, because of the μ condition

$$(\mu - 1)(deg(v) - \lambda - 1) = e_v + e_\mu, \text{ and } (\mu - 1)(deg(u) - \lambda - 1) = e_u + e_\mu,$$

must hold. So,

$$(\mu - 1)(deg(u) - deg(v)) = 0;$$

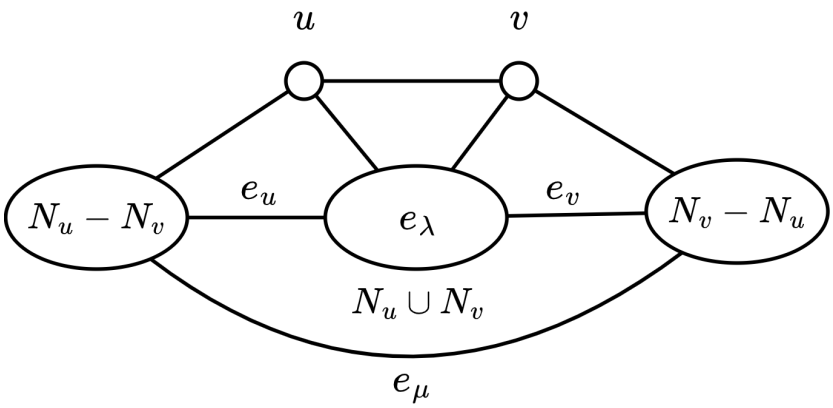

Figure 4.4: An adjacent pair of vertices $\{u, v\}$ in a graph with λ and μ parameters.

hence $\mu > 1$, the degrees of adjacent vertices u and v are the same. Since the graph is connected, by continuing this, we infer all vertices must have the same degree, and this means the graph is regular. □

One question that arises is whether other parameters, λ and μ, are also redundant. That is, if we suppose graph G is regular and every two neighbours share λ neighbours, then all non-neighbours share the same number of neighbours μ, and vice-versa.

The triangular view of the Berlekamp–Van Lint–Seidel graph is a counter-example to this. This graph has parameters $(891, 30, 9, \leq 3)$ and is not strongly regular. In general, for a 30-regular 891 vertex graph with $\lambda = 9$, we know no such strongly regular graph exists as no integer μ can satisfy the equation in Theorem 2.1.1.

Chapter 5

Studying the experiments

In this chapter, we will review the experiments we had, based on how we encoded the problems in Chapter 4. The full set of codes and experiments can be found on GitHub. The main SAT solver used was a pseudo-boolean solver named `clasp` [19].

Multiple 12-hour tests were run on parameter sets of strongly regular graphs with less than 65 vertices. Meanwhile, multiple dedicated SAT solver processes also searched for a $(99, 14, 1, 2)$ strongly regular graph. Considering the granted resources, a longer run was not possible.

To make the final comparison, we used pseudo-boolean encodings instead of CNF encodings, as the latter are too large and impractical. Also, non-linear instances were chosen over linear instances. SAT solvers linearise their non-linear inputs, and since the SAT solvers linearise inputs based on their needs, they will give us better and more efficient results. For example, the strongly regular graph with parameters $(26, 10, 3, 4)$ was found ≈ 2.5 times faster with a non-linear method.

During the execution of these tasks, in some instances, the SAT solver generated results for larger graphs but led to no results for smaller ones. We theoretically analyse these results.

5.1 Automorphism

The order of automorphism group $|Aut(G)|$ serves us as a measurement of a graph's symmetry. As we studied spectrum and automorphisms in Chapter 2, all strongly regular graphs with the same set of parameters are cospectral. However, non-isomorphic instances of these graphs exist, which implies not all cospectral graphs are isomorphic. These non-isomorphic instances lead to multiple automorphism groups for strongly regular graphs with the same parameter set.

Although in the realm of strongly regular graphs, spectrums do not provide us with useful information about isomorphisms and automorphism groups, they are a useful tool to determine non-isomorphic graphs in general.

We now study a few parameter sets for strongly regular graphs that were successfully found using SAT solvers. Our initial analysis focuses on smaller graphs 5.1.1, followed by an examination of bigger graphs 5.1.2. At last, we study those that were not attainable using our methods 5.1.3.

5.1.1 Small graphs

Schläfli graph

The Schläfli graph, named after Ludwig Schläfli, is a strongly regular graph with parameters $(27, 16, 10, 8)$. The building method of this graph is geometrical, but we won't delve into it here. Based on its construction method, it can be guessed that this graph has a considerable automorphism group order, specifically 51840. This is notably large when compared to other strongly regular graphs with a close number of vertices. For example, Paley(25) has an automorphism group of order 600.

Paulus graphs

There are two sets of non-isomorphic strongly regular graphs known as Paulus graphs. One set has parameters of $(25, 12, 5, 6)$ and consists of 15 graphs, one of which is isomorphic to Paley(25), while the other set has parameters of $(26, 10, 3, 4)$ and contains 10 graphs.

For parameters $(26, 10, 3, 4)$, the largest automorphism group is of order 120.

Parameters (28,9,0,4)

Krein bounds, named after Mark Krein, consist of two inequalities that strongly regular graphs must meet in relation to their eigenvalues. They stem from Krein parameters.

Theorem 5.1.1 (Krein bounds)**.** *A strongly regular graph with eigenvalues r, s, and k, where k is its regularity, must satisfy the following bounds:*

$$(r+1)(k+r+2rs) \leq (k+r)(s+1)^2,$$
$$(s+1)(k+s+2rs) \leq (k+s)(r+1)^2.$$

Parameters $(28, 9, 0, 4)$ cannot form a strongly regular graph. Applying the same approach as we used in 3.1.2 gives us a spectrum of $\{9^1, 1^{21}, (-5)^6\}$. However, using the Krein bounds, it can be deduced that these values are not possible for a strongly regular graph.

Chang graphs

In [13], Chang showed four non-isomorphic strongly regular graphs with the parameter set $(28, 12, 6, 4)$ exist. One of these graphs, named the Triangular graph $T(8)$, has an automorphism group of order 40320. We study Triangular graphs further in 5.1.2.

In the comparison chart shown in Table 5.1, we have analysed the previous graphs and the time taken by the SAT solver to arrive at a solution. The parameter set $(28, 9, 0, 4)$ failed to yield any result even after approximately 12 hours. This example highlights a scenario where we are aware that the parameters are not feasible, using a close Krein bound, and the SAT solver cannot reach a final state. We can also see that the Schläfli graph parameter set received an answer in less than two seconds while finding a Paulus graph with fewer vertices takes longer. The large automorphism group of the $T(8)$ graph is another example where our SAT solver performed fast.

5.1.2 Large graphs

To further investigate SAT solver's capabilities, we consider its performance on larger graphs. Table 5.2 contains larger graphs where our SAT solver was able to find a solution.

	n	k	λ	μ	time(s)	result
1	27	16	10	8	1.982	$\exists$
2	26	10	3	4	10.584	$\exists$
3	28	9	0	4	-	No result
4	28	12	6	4	0.48	$\exists$

Table 5.1: Parameter sets with around 27 vertices. Based on the Krein bounds, the set $(28, 9, 0, 4)$ is unsatisfiable, where SAT solvers cannot yield any results.

	n	k	λ	μ	time(s)	comment
1	36	10	4	2	10.271	Rook
2	36	14	7	4	577.235	Triangular
3	45	16	8	4	1749.487	Triangular
4	49	12	5	2	2.994	Rook
5	50	7	0	1	2.773	Hoffman-Singleton
6	56	10	0	2	13473.1	Sims-Gewirtz
7	64	14	6	2	19.04	Rook

Table 5.2: Larger strongly regular graphs found using SAT solver

Rook's graph

Rook graphs are the Cartesian product of two complete graphs $K_m \square K_n$. In these graphs, edges represent the legal moves that can be made by a rook chess piece, that is, moving vertically or horizontally.

When $m = n$ and the order of both complete graphs is the same, the graph becomes strongly regular, with a parameter set of $(n^2, 2n-2, n-2, 2)$. Each vertex in this graph is connected to $n-1$ vertices on two complete subgraphs on its coordinate, which means it is $2n-2$-regular.

We name the vertices based on their coordinates, i.e. (i, j) is where the i-th horizontal and j-th vertical graphs meet. If two vertices share a coordinate, they are adjacent and share $n-2$ neighbours. For two non-adjacent vertices (i_1, j_1) and (i_2, j_2), they share two neighbours, which are (i_1, j_2) and (i_2, j_1). This implies the graph's strong regularity with parameters $(n^2, 2n-2, n-2, 2)$.

As we can see, these graphs are highly symmetrical. The order of $Aut(G)$ for a Rook graph G is $2(n!)^2$. Entries 1,4,and 7 of Table 5.2 are Rook graphs. We can see that even for a very large order of 64, the SAT solver was able to obtain a result. Again, this is due to the SAT solver's symmetry-breaking techniques.

Triangular graphs

These highly symmetrical graphs can be built in many different ways. One possible approach is using *Line graphs*. Line graphs are similar to the triangular approach we employed in Subsection 3.4.3. A line graph of a graph G, denoted with $L(G)$, represents the adjacency of *edges*. Two edges are adjacent if they share a vertex.

Triangular graphs can be defined as the Line graph of a complete graph, i.e. $L(K_n)$. Let us now focus on how this graph is strongly regular. This graph consists of $\binom{n}{2}$ vertices, which represent the edges of K_n. In K_n, each edge shares a vertex with $2n-2$ edges, and

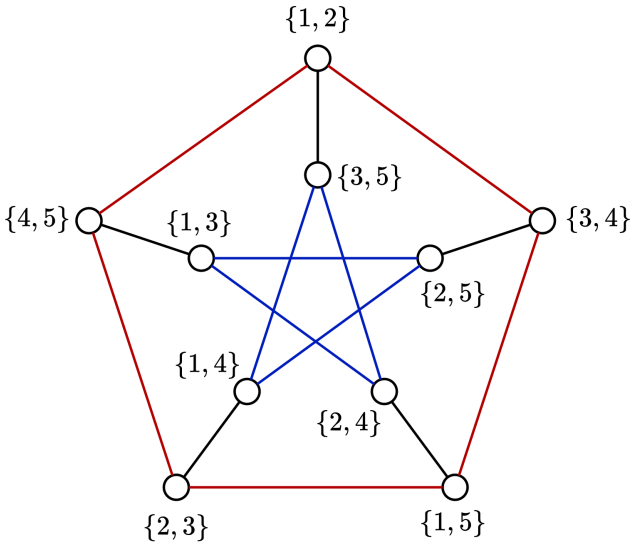

Figure 5.1: The Petersen or $K(5, 2)$ graph. Two vertices are adjacent if their respective sets share no element.

every two edges that share a vertex are adjacent to $n - 2$ other edges. If two edges do not share a vertex, name them uv and xy, which means they are non-adjacent, they share four neighbours. These neighbours are edges xu, xv, yu, and yv. Therefore, the graph obtained is a strongly regular graph with parameters $(\binom{n}{2}, 2n - 2, n - 2, 4)$. These graphs are also sometimes referred to as *perfect line graphs*.

The way we have built this graph already shows the high order of its automorphism group. The $Aut(L(K_n))$ contains $n!$ permutations, which are based on the initial K_n graph.

There are more ways to build these graphs as well. For example, the complement of $T(n) = L(K_n)$ can be represented by **Kneser graphs**. A Kneser graph $K(n, k)$ is a graph whose vertices are k-element subsets of a set with n elements. Two vertices are connected if their respective subsets are disjoint. The Petersen graph is also the Kneser graph $K(5, 2)$, as shown in Figure 5.1.

The adjacency matrices of triangular graphs $T(n)$ are uniquely determined, up to isomorphism, by their spectrum and eigenvalues when n is not 8. The other three Chang graphs have the same parameters as $T(8)$.

Hoffman-Singleton

Only one strongly regular graph with parameters $(50, 7, 0, 1)$ exists. This graph is known as the Hoffman–Singleton graph. Its automorphism group is of order 252000.

Sims-Gewirtz graph

This is a unique, strongly regular graph with parameters $(56, 10, 0, 2)$. Its automorphism group is of order 80640.

The results show that all the graphs that were found have large automorphism groups, which makes them highly symmetrical. SAT solvers, by applying their symmetry-breaking techniques, are able to reduce the search space.

5.1.3 Graphs that were not found

We previously saw that for the set of parameters $(28, 9, 0, 4)$, the SAT solver failed to yield any results while it was already unsatisfiable. We now study satisfiable parameters where the SAT solver still fails. Do they all have small automorphism groups?

	n	k	λ	μ	comment
1	29	14	6	7	Paley(29)
2	35	16	6	8	-
3	36	14	4	6	-
4	36	16	6	6	-
5	36	25	16	20	$\overline{K_6 \square K_6}$
5	37	18	8	9	Paley(37)
6	41	20	9	10	Paley(41)

Table 5.3: Parameter sets where SAT solver was unable to find a solution.

Paley graphs

We studied the construction method of Paley graphs in 3.2. Consider a Galois field $GF(q)$, where $q = p$, a prime number, and the set of its quadratic residues $QR(q)$. Define a function $f_{a,b} : GF(q) \to GF(q)$:

$$f_{a,b}(x) = ax + b,$$

where $a \in QR(q)$ and $b \in GF(q)$. There are $\frac{1}{2}(q-1)$ quadratic residues; hence, coefficients a and b have $\frac{1}{2}(q-1)q$ possible assignments to form linear functions. In a Paley graph $G = (GF(q), E)$, two vertices are connected if their difference is a quadratic residue. Since the product of two quadratic residues is itself a quadratic residue,

$$(x - y) \in QR(q) \Leftrightarrow (ax + b - ay - b) \in QR(q)$$

holds, as $a, (x - y) \in QR(q)$. This gives us $\frac{1}{2}(q-1)q$ automorphisms. According to Carlitz's theorem [12], the Galois Field $GF(q)$ of a prime power $q = p^d$ possesses d automorphisms. Thus, by combining these automorphisms, an automorphism group of order $\frac{d}{2}(q-1)q$ is found for Paley(q). It can also be proved no more automorphisms exist.

In Table 5.3, we can see three sets of parameters that Paley graphs can satisfy. For parameter set $(29, 14, 6, 7)$, a total of 41 non-isomorphic graphs have been found by computer search. A list of these graphs is compiled here by Ted Spence. Therefore, the Paley(41) graph was not the only answer to this SAT problem, and even more answers could have been found.

Based on these observations, we can see even for highly structured graphs like the Paley set, SAT solver fails to solve the problem. The two sets of graphs where SAT solver did not fail both have exponential automorphism orders, while the Paley graphs' automorphism set is polynomial.

Rook graph's complement

Rook graphs possessed a large automorphism group. However, SAT solver was unable to solve this instance $\overline{K_6 \square K_6}$, which is the Rook graph's complement, with high exponential symmetry.

Graphs with 36 vertices

McKay and Spence studied two parameter sets of strongly regular graphs in [27], which are $(36, 15, 6, 6)$ and $(36, 14, 4, 6)$.

For $(36, 14, 4, 6)$, a total of 180 non-isomorphic instances were found. The parameter set $(36, 15, 6, 6)$ has 32548 possible graphs. There are several possible solution instances available, however, they lack high symmetry, which poses a challenge for the SAT solver. The solver is unable to overcome this, and its performance is similar to an exhaustive search.

The Conway-99 problem

Based on our observation, we can infer that if $(99, 14, 1, 2)$ strongly regular graphs exist, then each of these graphs will likely have a very small automorphism group.

5.2 Alternative and additional methods

5.2.1 Additional constraints

It is beneficial to have additional constraints to guide the SAT solver towards the correct solution. During our experiments to study parameter redundancy 4.5, we ran multiple tests to see whether the SAT solver gives us a strongly regular graph based on only two parameters. For parameters $(28, 12, 6, 4)$, based on $\lambda = 6, \mu = 4$ and $k = 12, \mu = 4$, and $k = 12, \lambda = 6$ the time it took was approximately ≈ 2.5 minutes, ≈ 2.5 hours, ≈ 9 hours respectively.

After analysing this, it may be advantageous to include more restrictions in our SAT formula. A potential approach is to introduce constraints related to vertex colouring. It is common knowledge that all graphs can be coloured using $\Delta(G) + 1$ colours. Nevertheless, it may not be ideal to impose a strict limitation as the SAT solver will need to tackle both the chromatic number $\chi(G)$ problem and find the strongly regular graphs. In some instances, leaving the SAT solver to colour the graph with $2\Delta(G)$ colours sped up the process.

Also, more feasibility conditions exist for the class of strongly regular graphs as we studied them. Studying these conditions in more detail and encoding them into the SAT formula might make SAT solvers yield results faster.

5.2.2 MAX-SAT and almost strongly regular graphs

SAT solvers used in our studies are strict on satisfying all clauses. In order to obtain an almost strongly regular graph, we can remove some of the neighbouring or regularity constraints. We observed that even for small unsatisfiable constraints, they did not reach any terminal state. This could be the result for the Conway-99 parameters as well.

The maximum boolean satisfiability problem, **MAX-SAT**, asks for the maximum number of clauses that can be satisfied in the SAT formula. This problem is categorised as an NP-Hard problem. Although exact MAX-SAT solvers exist, approximate MAX-SAT solvers are of our interest here. While the solver is searching for the solution, we want the solver to provide us with the best solution it has found so far.

The combination of SAT and MAX-SAT problems gives us the **partial MAX-SAT problem**. In these formulas, some clauses are categorised as *hard* clauses and others are categorised as *relaxable* or *soft* ones. The solver is forced to satisfy all hard clauses, while it looks for the maximum number of soft clauses it can satisfy.

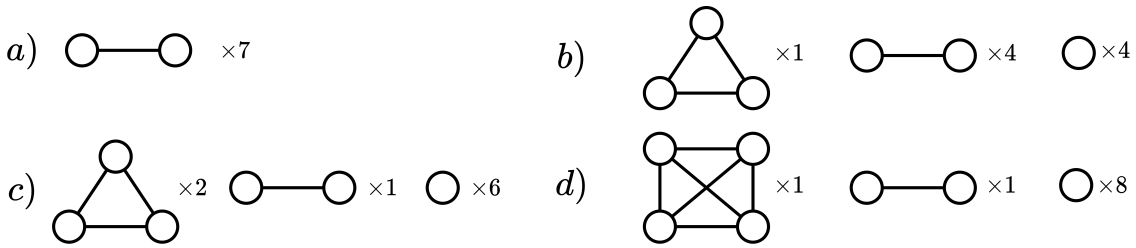

Figure 5.2: Possible $N(v)$ induced subgraphs

By employing the MAX-SAT method, we can find solutions that are almost strongly regular. For example, we force the SAT solver to return a k-regular graph while it tries to satisfy the maximum number of neighbouring conditions. SAT solvers that can provide us with the best solution they have found so far can be great to use. Examining where they hit a dead-end and need to backtrack might give us new insights into this problem and let us come up with other general feasibility conditions.

The following theorem shows us another approach by removing the λ conditions.

Theorem 5.2.1. *In a partial MAX-SAT formula, for a graph G containing 99 vertices, if the 14-regularity and the $\mu = 2$ conditions are defined as hard clauses, and a $(99, 14, 1, 2)$ strongly regular graph exists, then minimising the number of Claw subgraphs (as soft clauses) must satisfy the $\lambda = 1$ condition. If it does not, no such strongly regular graph exists.*

Proof. In the graph G obtained as a solution, consider an arbitrary vertex v. In our first step, we demonstrate $N(v)$ has a total of 7 edges. There are 84 vertices in $V - N[v]$, each connecting to exactly two vertices in $N(v)$ ($k = 14$ and $\mu = 2$ hard clauses). This implies 84×2 outgoing edges from $N[v]$ exist. Also, there are 14 edges among v and $N(v)$. We can infer that

$$\frac{14 \times 14 - 84 \times 2 - 14}{2} = 7$$

edges must be in $N(v)$.

Each non-neighbour pair $\{u, w\} \subset N(v)$ must be connected to exactly one vertex in $V - N[v]$, outside v's neighbourhood. Otherwise, there must exist a non-neighbouring pair of vertices sharing three neighbours, which breaks the $\mu = 2$ condition. Based on this, we can infer no Cherry induced subgraphs ($K_{1,2}$) can exist within $N(v)$. Therefore, in subgraph $N(v)$, all components must be complete (K_i for $i \in \{1, 2, 3, 4\}$). Figure 5.2 contains all possible $N(v)$ induced subgraphs.

We now count the number of Claw subgraphs in possible subgraphs. Let c_i be the number of K_is in $N(v)$. The number of Claws is obtained by

$$\binom{14}{3} - 12 \times c_2 - 11 \times 3c_3 - 10 \times 6c_4 - c_3 - 4c_4,$$

which would be minimised in case a) of Figure 5.2. In this case, graph G has the $\lambda = 1$ parameter and is strongly regular.

□

Chapter 6

Conclusions

In “Approaching the Conway-99 problem using SAT solvers", we studied strongly regular graphs both combinatorically and algebraically. We studied structures similar to the graph that the Conway-99 problem asks for and proved these patterns cannot hold in any solution for this problem.

To search for this problem, we decided to use SAT solvers to see whether they could find any solution to this problem. Therefore, we encoded this problem and transformed it into a SAT formula. As the SAT solver failed, we studied the reason behind this failure theoretically.

Countless ideas came to our minds on this beautiful research path. Unfortunately, we hit the wall of a deadline; otherwise, there is much to do research on. Some of these ideas will be listed here. The topics and ideas we gather from these sources will be valuable for our future research endeavours.

6.1 Future work

- As we studied, the structure of strongly regular graphs could be applied to many areas. Can we use this structure to develop better SAT solvers? Are these structures useful for detecting symmetries faster? Are their irregular structures useful for solving asymmetries faster?
- While approaching this, because of the short research period, we considered SAT solvers as black boxes. In the future, by studying the methods used in SAT solvers and combining them with our problem-specific methods, we can generate a better search tool for this problem.
- In 3.4.4, we conjectured that a Paley(9) subgraph cannot exist in a possible $(99, 14, 1, 2)$ strongly regular graph. If a Paley(9) subgraph appears, it would be a dominating set of the whole graph. To approach this problem, we can consider dominating sets and see whether we can have a 9-vertex Paley(9) dominating set.
- In Table 3.1, we listed the possible instances of strongly regular graphs with properties $\lambda = 1$ and $\mu = 2$. The Berlekamp–Van Lint–Seidel graph was built based on $GF(3)$ and Paley(9). Another possible graph to be discovered is a $(6273, 112, 1, 2)$ strongly regular graph.

  Considering the prime factorisation of 6273

$$6273 = 3^2 \times 17 \times 41,$$

we can see all three factors are congruent to 1 modulo 4, which means three Paley(9), Paley(17), and Paley(41) graphs exist. Can we use these three graphs to build a $(6273, 112, 1, 2)$ strongly regular graph?

- In 3.4.2 we proved a $(99, 14, 1, 2)$ strongly regular graph cannot follow the Paley(9) pattern. We used this in our SAT formula and preset the value of multiple edge boolean variables. It is highly likely we can prove a bound on the number of vertices following this pattern. Using this, we can reduce our search space by presetting more edge boolean variables.
- In 1990, Paul Seymour conjectured that in every directed graph, there is a vertex whose second neighbourhood, that is, the set of vertices of minimum distance 2, is as large as its first neighbourhood. Fisher, in [18], partially proved this by proving it for only tournaments, a simple, directed, complete graph. The graph K_n is a highly structured graph and is a trivial strongly regular graph.

  A question that arises is whether we can generalise this for all strongly regular graphs. This seems likely as they are highly structured. We proved this theorem for strongly regular graphs with $\lambda = 1$.

6.2 Acknowledgements

I am grateful for the guidance and support provided by Prof. Anuj Dawar throughout this project. I would like to express my heartfelt appreciation to my family, friends, and the esteemed members of Churchill College who motivated and assisted me on this journey.

Bibliography

[1] Majid Behbahani and Clement Lam. Strongly regular graphs with non-trivial automorphisms. *Discrete Mathematics*, 311(2):132–144, 2011.

[2] ER Berlekamp, JH Van Lint, and JJ Seidel. A strongly regular graph derived from the perfect ternary golay code. In *A survey of combinatorial theory*, pages 25–30. Elsevier, 1973.

[3] Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In Tomas Balyo, Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, editors, *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, volume B-2020-1 of *Department of Computer Science Report Series B*, pages 51–53. University of Helsinki, 2020.

[4] Armin Biere, Matti Järvisalo, and Benjamin Kiesl. Preprocessing in sat solving. *Handbook of Satisfiability*, 336:391–435, 2021.

[5] Norman Biggs. *Finite groups of automorphisms: course given at the University of Southampton, October-December 1969*, volume 6. CUP Archive, 1971.

[6] Raj Chandra Bose. Strongly regular graphs, partial geometries and partially balanced designs. 1963.

[7] Andries E. Brouwer. Parameters of Strongly Regular Graphs. Brouwer's website.

[8] Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[9] Andries E. Brouwer and H. Van Maldeghem. *Strongly Regular Graphs*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2022.

[10] Frans C Bussemaker, Willem H Haemers, R Matron, and Henny A Wilbrink. A (49, 16, 3, 6) strongly regular graph does not exist. *European Journal of Combinatorics*, 10(5):413–418, 1989.

[11] Peter J Cameron. Strongly regular graphs. *Topics in Algebraic Graph Theory*, 102:203–221, 2004.

[12] L Carlitz. A theorem on permutations in a finite field. *Proceedings of the American Mathematical Society*, 11(3):456–459, 1960.

[13] Li-Chien Chang. Association schemes of partially balanced block designs with parameters v= 28, n1= 12, n2= 15 and p2 11= 4, sci. *Record*, 4:12–18, 1960.

[14] John Conway. Five $1,000 problems (update 2017). https://oeis.org/A248380/a248380.pdf.

[15] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[16] Kris Coolsaet, Jan Degraer, and Edward Spence. The strongly regular $(45, 12, 3, 3)$ graphs. *the electronic journal of combinatorics*, pages R32–R32, 2006.

[17] Dean Crnković, Bernardo Gabriel Rodrigues, Sanja Rukavina, and Loredana Simčić. Ternary codes from the strongly regular (45, 12, 3, 3) graphs and orbit matrices of 2-(45, 12, 3) designs. *Discrete Mathematics*, 312(20):3000–3010, 2012.

[18] David C Fisher. Squaring a tournament: a proof of dean's conjecture. *Journal of Graph Theory*, 23(1):43–48, 1996.

[19] Martin Gebser, Benjamin Kaufmann, and Torsten Schaub. Conflict-driven answer set solving: From theory to practice. *Artif. Intell.*, 187:52–89, 2012.

[20] Marcel JE Golay. Notes on digital coding. *Proc. IEEE*, 37:657, 1949.

[21] Oleg Gritsenko. On strongly regular graph with parameters (65; 32; 15; 16). *arXiv preprint arXiv:2102.05432*, 2021.

[22] Derek Allan Holton and John Sheehan. *The Petersen graph*, volume 7. Cambridge University Press, 1993.

[23] Jonatan Janmark, David A Meyer, and Thomas G Wong. Global symmetry is unnecessary for fast quantum search. *Physical Review Letters*, 112(21):210502, 2014.

[24] Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.

[25] Leonid A. Levin. Universal'nye pereborchiki i otsenki slozhnosti nekotorykh algoritmov. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[26] Eugene Luks. Symmetry-breaking predicates for search problems. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR'96)*, volume 5, page 148. Morgan Kaufmann Pub, 1996.

[27] Brendan D McKay and Edward Spence. Classification of regular two-graphs on 36 and 38 vertices. *Australasian Journal of Combinatorics*, 24:293–300, 2001.

[28] Edward Spence. The strongly regular $(40, 12, 2, 4)$ graphs. *the electronic journal of combinatorics*, 7:R22–R22, 2000.

[29] H.A. Wilbrink. *On the (99,14,1,2) strongly regular graphs*, pages 342–355. EUT-Report. Technische Hogeschool Eindhoven, 1984.

Appendix A

Runtime test experiments

n	k	λ	μ	time(s)	Comment
5	2	0	1	0.0	
9	4	1	2	0.001	
10	3	0	1	0.001	
10	6	3	4	0.002	
13	6	2	3	0.004	
15	6	1	3	0.007	
15	8	4	4	0.007	
16	5	0	2	0.006	
16	6	2	2	0.012	
16	9	4	6	0.012	
16	10	6	6	0.049	
17	8	3	4	0.407	
21	10	3	6	0.068	
21	10	4	5	-	$\emptyset$
21	10	5	4	0.32	
25	8	3	2	1.717	
25	12	5	6	67.719	
25	16	9	12	18.991	
26	10	3	4	10.584	
26	15	8	9	22.224	
27	10	1	5	0.587	
27	16	10	8	1.982	
28	9	0	4	-	$\emptyset$
28	12	6	4	0.48	
28	15	6	10	0.233	
28	18	12	10	-	$\emptyset$
29	14	6	7	-	
33	16	7	8	-	$\emptyset$
35	16	6	8	-	
35	18	9	9	-	
36	10	4	2	10.271	
36	14	4	6	-	
36	14	7	4	577.235	
36	15	6	6	-	
36	20	10	12	-	
36	21	10	15	-	
36	21	12	12	-	

36	25	16	20	-	
37	18	8	9	-	
40	12	2	4	-	
40	27	18	18	-	
41	20	9	10	-	
45	12	3	3	-	
45	16	8	4	1749.487	
45	22	10	11	-	
45	28	15	21	-	
45	32	22	24	-	
49	12	5	2	2.994	
49	16	3	6	-	∅
49	18	7	6	-	
49	24	11	12	-	
49	30	17	20	-	
49	32	21	20	-	∅
49	36	25	30	-	
50	7	0	1	2.773	
50	21	4	12	-	∅
50	21	8	9	-	
50	28	15	16	-	
50	28	18	12	-	∅
50	42	35	36	-	
53	26	12	13	-	
55	18	9	4	-	
55	36	21	28	-	
56	10	0	2	13473.1	
56	22	3	12	-	∅
56	33	22	15	-	∅
56	45	36	36	-	
57	14	1	4	-	∅
57	24	11	9	-	
57	28	13	14	-	∅
57	32	16	20	-	
57	42	31	30	-	
61	30	14	15	-	
63	22	1	11	-	∅
63	30	13	15	-	
63	32	16	16	-	
63	40	28	20	-	∅
64	14	6	2	19.04	
64	18	2	6	-	
64	21	0	10	-	∅
64	21	8	6	-	
64	27	10	12	-	
64	42	26	30	-	
64	42	30	22	-	∅
64	45	32	30	-	
64	49	36	42	-	

Table A.1: All SAT tasks. SAT solver clasp was used. The character '-' means after ≈ 12 hours, no result was obtained. Character ∅ denotes that no graphs with the respective parameters exist.