Graph Colouring and Sudoku

Applications of Gröbner Bases

Yangda Bei
u7281660

A report presented for
Advanced Studies Course SNCN3101

ANU Mathematical Sciences Institute
The Australian National University
February 21, 2023

Abstract

The Hilbert Basis Theorem states that polynomial rings over a Noetherian ring are also Noetherian, that is, ideals in polynomial rings have a finite generating set. This allows us to use a terminating algorithm to find a unique set of finite elements that generate a polynomial ideal, namely, a reduced Gröbner basis. Using Gröbner bases to describe a system is very powerful and has wide applications in the field of algebraic geometry. We present methods that use Gröbner bases to solve the graph colouring problem as well as Sudoku and its variants.

Contents

1 Introduction

A Gröbner basis is a specific kind of generating set of an ideal in a polynomial ring. It is one of the main practical tools used in computer algebra, computational algebraic geometry, and computational commutative algebra for solving systems of polynomial equations as well as finding the images of algebraic varieties under projections or rotational maps. Introduced in 1965 by Bruno Buchberger in his Ph.D. thesis [3], Buchberger determined an algorithm to find such a basis. In this report, we will discuss the basic properties of Gröbner bases. We also present and implement different methods involving Gröbner bases to problems in graph colouring as well as extending this to solving Sudoku puzzles and its variants. We first give some preliminary definitions and theorems from [5].

Definition 1.1. A subset $I \subseteq k[x_1, \ldots, x_n]$, where k is a field, is an *ideal* if it satisfies:

1. $0 \in I$.
2. If $f, g \in I$, then $f + g \in I$.
3. If $f \in I$ and $h \in k[x_1, \ldots, x_n]$, then $hf \in I$.

Lemma 1.1. *If $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$, then*

$$\langle f_1, \ldots, f_s \rangle = \left\{ \sum_{i=1}^{s} h_i f_i \mid h_1, \ldots, h_s \in k[x_1, \ldots, x_n] \right\}$$

is an ideal of $k[x_1, \ldots, x_n]$.

We will call $\langle f_1, \ldots, f_s \rangle$ the *ideal generated by* $f_1, \ldots, f_s$.

Definition 1.2. If I and J are two ideals in $k[x_1, \ldots, x_n]$, then their *product*, denoted by $I \cdot J$, is defined to be the ideal generated by all polynomials $f \cdot g$ where $f \in I$ and $g \in J$. Thus, the product $I \cdot J$ is the set

$$I \cdot J = \{f_1 g_1 + \cdots + f_r g_r \mid f_1, \ldots, f_r \in I, g_1, \ldots, g_r \in J, r \text{ a positive integer}\}.$$

Definition 1.3. Let $f_1, \ldots, f_s$ be polynomials in $k[x_1, \ldots, x_n]$. Then we set

$$\mathbf{V}(f_1, \ldots, f_s) = \{(a_1, \ldots, a_n) \in k^n \mid f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq s\}.$$

We will call $\mathbf{V}(f_1, \ldots, f_s)$ the *affine variety* defined by $f_1, \ldots, f_s$.

Lemma 1.2. *If $f_1, \ldots, f_s$ and $g_1, \ldots, g_t$ are generators of the same ideal in $k[x_1, \ldots, x_n]$, then we have $\mathbf{V}(f_1, \ldots, f_s) = \mathbf{V}(g_1, \ldots, g_t)$.*

Definition 1.4. An affine variety $\mathbf{V}(I)$ where $I = \langle f_1, \ldots, f_s \rangle$ is defined by

$$\mathbf{V}(I) = \mathbf{V}(f_1, \ldots, f_s) = \{(a_1, \ldots, a_n) \in k^n \mid f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq s\}.$$

Definition 1.5. Let $V \subseteq k^n$ be an affine variety. Then we set

$$\mathbf{I}(V) = \{f \in k[x_1, \ldots, x_n] \mid f(a_1, \ldots, a_n) = 0 \text{ for all } (a_1, \ldots, a_n) \in V\}.$$

Theorem 1.3 (The Weak Nullstellensatz)**.** *Let k be an algebraically closed field and let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal satisifying $\mathbf{V}(I) = \emptyset$. Then $I = k[x_1, \ldots, x_n]$.*

Theorem 1.4 (Hilbert's Nullstellensatz)**.** *Let $I \subseteq \mathbb{C}[k_1, \ldots, x_n]$. Then $f \in \mathbf{I}(\mathbf{V}(I))$ if and only if there exists some integer m such that $f^m \in I$.*

Throughout the report, we give the sample code for each problem. The repository for this project can be found here. The computations will be done in Python using the `sympy` package as it is easier to work with graphs by using the `networkx` package. However, note that using Mathematica is vastly superior in terms of its computational speed for computer algebra.

2 Gröbner Bases

2.1 Monomial Orders and the Division Algorithm

We wish to extend the notion of the division algorithm from $k[x]$ to $k[x_1, \dots, x_n]$. However, it is not so trivial. We will see the complications that arise if we try extending it naïvely in the following section.

Theorem 2.1 (Division Algorithm). *Let k be a field and let g be a non-zero polynomial in $k[x]$. Then ever $f \in k[x]$ can be written as*

$$f = qg + r,$$

where $q, r \in k[x]$, and either $r = 0$ or $\deg(r) < \deg(g)$. Furthermore, q and r are unique, and there is an algorithm for finding q and r.

We also have a notion of a greatest common divisor for two polynomials.

Definition 2.1. A *greatest common divisor* of polynomials $f, g \in k[x]$ is a polynomial h such that:

(i) h divides f and g.

(ii) If p is another polynomial which divides f and g, then p divides h. When h has these properties, we write $h = \gcd(f, g)$.

Using the division algorithm, we have a way of computing the greatest common divisor of two polynomials.

Theorem 2.2 (Euclidean Algorithm). *For $f, g \in k[x]$, $g \neq 0$, we have that $\langle f, g\rangle = \langle \gcd(f, g)\rangle$. We can also explicitly find $\gcd(f, g)$, which is the last non-zero remainder r_n in the sequence of divisions*

$$\begin{aligned} f &= gq_1 + r_1 \\ g &= r_1 q_2 + r_2 \\ r_1 &= r_2 q_3 + r_3 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_n + r_n \\ r_{n-1} &= r_n q_{n+1} + 0. \end{aligned}$$

Using these algorithms, we can analyse the structure of ideals. Questions that arise include the problem of *ideal membership*, i.e., is $f \in \langle f_1, \dots, f_s\rangle$, and *ideal equality*, i.e., does $\langle f_1, \dots, f_s\rangle = \langle g_1, \dots, g_t\rangle$. In the single variable case, we determined the "size" of a polynomial by its degree. This means that the algorithms would eventually terminate as at each stage, the polynomials being produced would have a smaller degree. We now extend this idea of "size" to multivariable polynomials.

Definition 2.2. A *monomial* in $x_1, \dots, x_n$ is a product of the form

$$x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n},$$

where all of the exponents $\alpha_1, \dots, \alpha_n$ are non-zero

Definition 2.3. A *monomial ordering* $\succ$ on $k[x_1, \dots, x_n]$ is a relation on the set of monomials $x^\alpha, \alpha \in \mathbb{Z}^n_{\geq 0}$ (i.e., the monomial exponents), such that:

1. The relation $\succ$ is a total (or linear) ordering.
2. If $\alpha \succ \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \succ \beta + \gamma$.
3. The relation $\succ$ is a well-ordering on $\mathbb{Z}^n_{\geq 0}$ (every non-empty subset has a smallest element).

With respect to a monomial ordering, we can compare the "sizes" of monomials which means we can define a systematic way of division with multivariate polynomials. Here are two basic examples of monomial orderings:

- (lexicographic order) We say $\alpha \succ \beta$ if the left-most non-zero entry of $\alpha - \beta$ is positive.

- (graded lexicographic order) We say $\alpha \succ \beta$ if $|\alpha| \succ |\beta|$, or $|\alpha| = |\beta|$ and $\alpha \succ_{lex} \beta$.

Now for $f \in k[x_1, \ldots, x_n]$, let $LT(f)$ denote the leading term under a monomial ordering $\succ$. Then, we formulate the division algorithm for multivariate polynomials.

Theorem 2.3 (Multivariate Division Algorithm)**.** *Let $\succ$ be a monomial ordering and let $f_1, \ldots, f_s, \in k[x_1, \ldots, x_n]$ be. Then every $f \in k[x_1, \ldots, x_n]$ can be written as*

$$f = q_1 f_1 + \ldots q_s f_s + r,$$

where $q_i, r \in k[x_1, \ldots, x_n]$ where no $LT(f_i)$ is divisible by any term in the remainder r.

2.2 Definition and Properties of Gröbner Bases

The ideal membership problem is difficult to determine if we only consider using the division algorithm. The next example demonstrates that we cannot determine whether f is in $\langle f_1, \ldots, f_s \rangle$ simply by dividing f with the generators.

Example 2.1. *Consider $k[x, y, z]$ with lex order. Let $I = \langle g_1, g_2 \rangle$ where $g_1 = x^2y - z$ and $g_2 = xy - 1$, and let $f = x^3 - x^2y - x^2z + x$. We see that $f \in I$ since*

$$f = x^2 \cdot g_1 + (-x^3 - x) \cdot g_2.$$

However, using polynomial long division to divide f by g_1 gives a remainder of $x^3 - x^2z + x - z$, which is not divisible by g_2. Likewise, dividing by g_2 gives a remainder of $x^3 - x^2z$, which is not divisible by g_1.

This problem can be solved if we find a special generating set that is finite for the ideal I. Such a generating set is known as a *Gröbner basis.* We use the Hilbert Basis Theorem to guarantee that a Gröbner basis always exists for any ideal.

Theorem 2.4 (Hilbert Basis Theorem)**.** *Every polynomial ideal $I \subseteq k[x_1, \ldots, x_n]$ is finitely generated.*

Definition 2.4. Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal. Given a monomial ordering $\succ$, a finite subset $G = \{g_1, \ldots, g_t\} \subseteq I$ different from $\{0\}$ is said to be a *Gröbner basis* if

1. G generates I, and
2. $\langle LT(g_1), \ldots, LT(g_t) \rangle = \langle LT(I) \rangle$, where $LT(I)$ is the set of leading terms of non-zero elements of I.

How does this solve our problem of checking ideal membership? We will use the following proposition.

Proposition 2.1. *Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal and $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis for I. Then given $f \in k[x_1, \ldots, x_n]$, there is a unique $r \in k[x_1, \ldots, x_n]$ such that*

(i) no term of r is divisible by any of $LT(g_1), \ldots, LT(g_t)$, and

(ii) there is a $g \in I$ such that $f = g + r$.

Restated, this proposition says that $f \in I$ if and only if f vanishes when divided by a Gröbner basis for I.

Example 2.2. *From Example 2.1, a Gröbner basis for the ideal $I = \langle g_1, g_2 \rangle$ is given by $G = \{yz - 1, x - z\}$. Note that $yz - 1 = g_1 - x \cdot g_2$ and $x - z = -y \cdot g_1 + (1 + xy) \cdot g_2$, so $I \subseteq \langle yz - 1, x - z \rangle$. Also, $g_1 = x \cdot (yz - 1) + (1 + xy) \cdot (x - z)$ and $g_2 = (yz - 1) + y \cdot (x - z)$, hence, $\langle yz - 1, x - z \rangle = \langle g_1, g_2 \rangle$. Furthermore, it can be checked that $\langle LT(yz - 1, x - z) \rangle = \langle LT(yz - 1), LT(x - z) \rangle$, so it is indeed a Gröbner basis. Since long dividing f by $yz - 1$ and $x - z$, we find that the remainder is zero, so $f \in I$.*

There is also the notion of a minimal Gröbner basis. We call this a *reduced Gröbner basis.*

Definition 2.5. A *reduced Gröbner basis* for a polynomial ideal I is a Gröbner basis G for I such that:

(i) The leading coefficient of p is 1 for all $p \in G$.

(ii) For all $p \in G$, no monomial of p lies in $\langle LT(G \setminus \{p\}) \rangle$.

There are many algorithms to determine a Gröbner basis for an ideal such as Buchberger's algorithm (refer to Chapter 2 §7 of [5]), however they are quite technical and outside the scope of this report.

3 Graph Colouring

A large portion of graph theory is concerned with the colouring of graphs such that no two adjacent vertices are coloured with the same colour. The Four Colour theorem [9] is one of the most celebrated theorem in the field of graph colouring, which provided the motivation for developing algebraic graph theory. It is also said to be the first theorem proved using significant help from computers. Graph colouring has many applications in computer science, information theory, and complexity theory, as well as many everyday problems, such as minimising conflicts in scheduling.

The accompanying notebook for this section can be found at here. Before we formulate a graph colouring problem to an ideal membership problem, we first give some key definitions.

Definition 3.1. A *graph* is an ordered pair $G = (V, E)$, which consists of a nonempty set V of *vertices* and a set E of paired vertices whose elements are called *edges*.

For this report, we only consider *simple graphs*, that is, graphs that do not have more than one edge between two vertices and no edges that start and end at the same vertex.

Definition 3.2. Let $G = (V, E)$ be an n-vertex graph. For a vertex $v \in V$, the *neighbourhood* of v is given by $N(v) = \{u \in V \mid \{u, v\} \in E\}$.

Definition 3.3. Let $G = (V, E)$ be a graph containing an edge $e = \{u, v\}$ with $u \neq v$. Let f be a function that is the identity on $V \setminus \{u, v\}$ and sends u and v to w. The *edge contraction* of e, denoted by G/e, results in a new graph $G' = (V', E')$ where $V' = V \setminus \{u, v\} \cup \{w\}$ and $E' = E \setminus \{e\}$. The *deletion* of e, denoted by $G - e$, is the graph $G' = (V, E \setminus \{e\})$.

Put simply, the edge contraction of $e = \{u, v\}$ removes the e and merges the two vertices u and v, and edge deletion removes an edge between two vertices.

Definition 3.4. For a graph $G = (V, E)$, a *colouring* is a function $\mathcal{C} : V \to \{1, 2, \dots\}$ such that for all $u \in N(v)$, $\mathcal{C}(u) \neq \mathcal{C}(v)$. G is k-colourable if G can be coloured with k distinct colours. A subset of vertices with the same colour forms a *colour class*.

We now ask the question, given a graph, is it k-colourable?

3.1 Determining k-colourability

3.1.1 Quotient Method

We introduce a tool that helps us transform a graph colouring problem into a set of polynomial equations that we can solve. The theorem after gives the relation between the decidability of a k-colouring of a graph and ideal membership.

Definition 3.5. The *graph polynomial* f_G associated to the graph $G = (V, E)$ is an element of the ring $\mathbb{C}[x_1, \dots, x_n]$, given by:

$$f_G := \prod_{\{u,v\} \in E} (u - v)$$

Theorem 3.1. *Fix k a positive integer. Let I be the ideal generated by the polynomials $x^k - 1$ for $x \in V$. The graph G is k-colourable if and only if $f_G \notin I$.*

Proof. Let G be k-colourable. Then there is an assignment of colours to the vertices such that no two adjacent vertices have the same colour. This corresponds to a point $a \in \mathbf{V}(I)$ such that $f_G(a) \neq 0$. Hence, $f_G \notin I$.

Conversely, if G is not k-colourable, then there is at elast one pair of adjacent vertices that share a colour. This means that f_G vanishes for any assignment of colours, that is, f_G vanishes on $\mathbf{V}(I)$. By Hilbert's Nullstellensatz, there is some m such that $f_G^m \in I$. This implies that $f_G^m(a) = 0$ for $a \in \mathbf{V}(I)$. Because $\mathbb{C}$ is a field, $f_G^m(a) = 0$ implies that $f_G(a) = 0$. Hence, $f_G \in I$. □

One can view the above theorem in this way: for each $v \in V$, the polynomials that generate I represent the k-th roots of unity. We assign a colour to each vertex, where each colour corresponds to a k-th root of unity. Then, if two adjacent vertices are assigned the same colour, f_G will vanish. This criterion allows us to perform an algorithm to determine if a graph is k-colourable: compute a Gröbner basis for I, and then divide f_G by this basis.

3.1.2 Roots of Unity Method

We can also generate the ideal in a way such that we can check whether there is a solution in the variety corresponding to the ideal. Let $x = e^{\frac{2\pi i}{k}} \in \mathbb{C}$ be the k-th root of unity. Represent the k colours by the k distinct roots of unity, so each vertex is assigned $1, x, x^2, \ldots, x^{k-1}$. For $1 \leq i \leq n$, we can model this as

$$x_i^k - 1 = 0. \tag{1}$$

Now, if x_i and x_j are connected by an edge, they need to be a different colour. Since $x_i^k = x_j^k$, we have that $(x_i - x_j)(x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1})$. We require x_i and x_j to be different k-th roots of unity so

$$x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1} = 0. \tag{2}$$

Let the ideal I be generated by the polynomials in Equation 1 as well as by the polynomials in Equation 2 for each pair of adjacent vertices x_i, x_j . We call this system of polynomial equations the *roots of unity* system.

Definition 3.6. The k-colouring of an n-vertex graph G given by the *roots of unity system* is the ideal $I \subseteq \mathbb{C}[x_1, \ldots, x_n]$ generated by

$$\begin{aligned} \text{for all } i \in V(G): \quad & x_i^k - 1 \\ \text{for all } \{i,j\} \in E(G): \quad & x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1} \end{aligned}$$

Since $\mathbf{V}(I) = \emptyset$ implies that $I = \mathbb{C}[x_1, \ldots, x_n]$ by the Weak Nullstellensatz, we can check whether $1 \in I$. If this is true, then there are no solutions to the graph colouring problem.

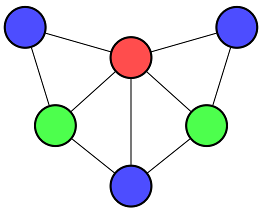


Figure 1: *A graph colouring for a graph with six vertices. It can be coloured with three colours so it is 3-colourable. However, it cannot be coloured with two colours, so it has chromatic number $\chi(G) = 3$.*

3.2 The Chromatic Number

Another interesting question that arises from the colouring problem is what the smallest number of colours required to colour a graph. The accompanying notebook to this subsection can be found here. We first give the following definition.

Definition 3.7. The *chromatic number* $\chi = \chi(G)$ is the smallest number such that the graph can be coloured with χ colours. A k-colourable graph is k-chromatic if its chromatic number is k.

Figure 1 gives the chromatic number for an example graph. We will now discuss the covering ideal of a graph and its connection to its chromatic number.

Definition 3.8. For a graph $G = (V, E)$, an *independent set* is a set of vertices $U \subseteq V$ such that there are no edges between any two vertices in U. The independence number is the size of the largest independent set. A subset $W \subseteq V$ is a *vertex cover* such that for all $\{u, v\} \in E$, $u \in W$ or $v \in W$.

It follows that a vertex cover of a graph is the complement of an independent set. Hence, a maximal independent set corresponds to a minimal vertex cover. Furthermore, each vertex in an independent set can be coloured the same colour. Therefore, from definitions, we get the following lemma.

Lemma 3.2. *Let $G = (V, E)$ be a graph and $U \subseteq V$ be a subset of vertices. Then U is an independent set if and only if $V \setminus U$ is a vertex cover.*

Definition 3.9. For a graph G, the *cover ideal* $I_{cover}(G)$ is the monomial ideal

$$I_{cover}(G) := \bigcap_{\{u,v\}\in E} \langle u, v\rangle$$

We will see that the minimal generators for $I_{cover}(G)$ will correspond to minimal vertex covers for G. Since the minimal vertex cover is the complement of a maximal independent set, then $I_{cover}(G)$ will be related to the independence number of G.

Proposition 3.1. *Let G be a graph and $W \subseteq V$ a minimal vertex cover. Let J be the cover ideal of G. Then J is generated by monomials of the form $\prod_{w\in W} w$.*

Proof. Let x_W denote the product of variables for any $W \subseteq V$, and let I be the ideal generated by $\{x_{W_{min}}\}$, where W_{min} is a minimal vetex cover. Suppose that W is a minimal vetex cover. Then for any edge $\{u, v\} \in E$, either $u \in W$ or $v \in W$. This means that either u divides x_W or v divides x_W, and so, $x_W \in \langle u, v\rangle$. Hence, $x_W \in J$, so $I \subseteq J$. Conversely, let $f \in J$ be any monomial generator and let W be the set of variables which can divide f. Since $f \in J$, then $f \in \langle u, v\rangle$ for each edge $\{u, v\} \in E$. This means that either u divides f or v divides f, so either $u \in W$ or $v \in W$. Hence, W is a vertex cover. Let $W' \subseteq W$ be a minimal vertex cover. Since $x_{W'}$ divides f, $f \in I$. Hence, $J \subseteq I$. □

We have already established that minimal vertex covers are the compliments of maximal independent sets, so the generators of $I_{cover}(G)$ also correspond to the maximal independent sets of G. Using this fact, we prove the following theorem from [8].

Theorem 3.3. *Let G be a graph a vertex set $V = \{x_1, \ldots, x_n\}$ with cover ideal J. Then $x_V^{d-1} \in J^d$ if and only if $\chi(G) \leq d$, where $x_V = x_1 \ldots x_n$. In particular,*

$$\chi(G) = \min\{d \mid x_V^{d-1} \in J^d\}.$$

Proof. First, let $U_1 \cup \cdots \cup C_{\chi(G)}$ be a $\chi(G)$-colouring of G, where each U_i contains vertices in the same colour class. It is also an independent set. By Lemma 3.2, we have that for each $i = 1, \ldots, \chi(G)$,

$$W_i = U_1 \cup \cdots \cup U_{i-1} \cup U_{i+1} \cup \cdots \cup U_{\chi(G)}$$

is a vertex cover of G. Hence, $x_{W_i} \in J$ for $i = 1, \ldots, \chi(G)$. It follows that

$$\prod_{i=1}^{\chi(G)} x_{W_i} = \left(\prod_{i=1}^{\chi(G)} x_{U_i}\right)^{\chi(G)-1} = (x_1 \ldots x_n)^{\chi(G)-1} = x_V^{\chi(G)-1} \in J^{\chi(G)}.$$

The first equality comes from the fact that the colour classes U_i are disjoint so if a vertex $x_i \in U_j$, then $x_i \in W_k$ for all $k \neq j$. Hence each vertex appears $\chi(G) - 1$ times in $x_{W_1} \ldots x_{W_{\chi(G)}}$. Hence, $x_V^{d-1} = x_V^{\chi(G)-1} x_V^{d-\chi(G)} \in J^{\chi(G)} \cdot J^{d-\chi(G)} = J^d$ since $x_V \in J$.

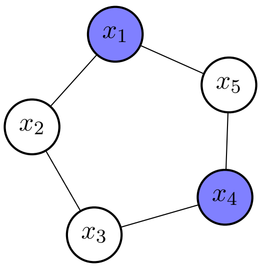
Conversely, suppose that $x_V^{d-1} \in J^d$. Then there exists d minimal vertex covers $W_1, \ldots, W_d$, which are not necessarily disinct, such that $x_{W_1} \ldots x_{W_d}$ can divide x_V^{d-1}. Using a similar argument from above, this is because for each $x_i \in \{x_1, \ldots, x_n\}$, there exists some W_j such that $x_i \notin W_j$, since if $x_i \in W_j$ for all $1 \leq j \leq d$, the power of x_i is d in $x_{W_1} \ldots x_{W_d}$. This gives a contradiction since it cannot divide x_V^{d-1}. Now, we want to form complements of W_i in a way that gives a d-colouring of G. Form the following d sets as follows:

$$\begin{aligned}
U_1 &= V \setminus W_1 \\
U_2 &= (V \setminus W_2) \setminus U_1 \\
U_3 &= (V \setminus W_3) \setminus (U_1 \cup U_2) \\
&\vdots \\
U_d &= (V \setminus W_d) \setminus (U_1 \cup \cdots \cup U_{d-1}).
\end{aligned}$$

It suffices to show that $U_1, \ldots, U_d$ form a d-colouring of G. First, note that each U_i's are disjoint by construction. Also, because each $U_i \subseteq V \setminus W_i$, each U_i is an independent set.

Then it remains to show that $V = U_1 \cup \dots \cup U_d$. If we have $x \in V$, then there exists some W_j such that $x \notin W_j$, so $x \in V \setminus W_j$. Hence, either $x \in U_j$ or $x \in (U_1 \cup \dots \cup U_{j-1})$ for $j \leq d$. □

Example 3.1. *Consider the graph $G = C_5$, which is the cyclic graph with five nodes. The vertices coloured in blue form an independent set since they share no edges. The uncoloured vertices form a vertex cover since any edge has an endpoint among the uncoloured vertices.*



Let J be the cover ideal for this graph. Apply Proposition 3.1 and we have that $J = \langle x_1, x_2 \rangle \cap \langle x_2, x_3 \rangle \cap \langle x_3, x_4 \rangle \cap \langle x_4, x_5 \rangle \cap \langle x_5, x_1 \rangle$. Using a computer algebra simplifier, we have that

$$J = \langle x_2x_4x_5, x_2x_3x_5, x_1x_3x_5, x_1x_3x_4, x_1x_2x_4 \rangle,$$

which corresponds to the minimal vertex covers of G.

By inspection, we see that $\chi(G) = 3$. However, we wish to confirm Theorem 3.3. What is the minimal d such that $(x_1x_2x_3x_4x_5)^{d-1} \in J^d$? For $d = 1$, we have that $1 \notin J$, so G is not 1-colourable. For $d = 2$, J^2 is generated by pairwise products of the generators of J, so J^2 is generated by monomials of degree 6. Hence, $x_1x_2x_3x_4x_5 \notin J^2$. Finally, for $d = 3$, J^3 is generated by products of any three of the monomial generators of J. We find that $x_1x_2^2x_3^2x_4^2x_5^2 = (x_2x_4x_5) \cdot (x_2x_3x_5) \cdot (x_1x_3x_4)$. Hence, $x_1 \cdot x_1x_2^2x_3^2x_4^2x_5^2 = x_1^2x_2^2x_3^2x_4^2x_5^2 \in J^3$ so the $\chi(G) = 3$.

3.3 Chromatic Polynomials

As an aside, we explore the k-colourability of a graph by introducing the chromatic polynomial which is a key object in the study of algebraic graph theory. This section describes the problem in a more graph theoretic manner.

Theorem 3.4. *The chromatic function of a graph $G = (V, E)$ is a polynomial.*

Proof. Before our discussion on chromatic polynomials, we want to show that they are indeed polynomials. Construct colourings of G by partitioning V into independent sets and assigning a unique to colour to each independent set. Given k colours, there are k ways to choose a colour for the first set, $(k-1)$ ways for the second, and so on. Hence, there are $k(k-1)(k-2)\dots$ possible ways to colour a graph G such that it results in a proper colouring. This is a polynomial in k. □

Definition 3.10. The *chromatic polynomial* $P(G, x)$ for a graph G is a unique polynomial, which evaluated at any integer $k \geq 0$ coincides with $P(G, k)$, the number of G's proper k-colourings.

Theorem 3.5. *Let G be an n-vertex graph. Let C be a partial proper colouring of c vertices of G using d_0 colours. Let $P(G_C, k)$ be the number of ways of completing this colouring using k colours to obtain a proper colouring of G. Then $P(G_C, k)$ is a monic polynomial in k with integer coefficients of degree $n - c$ for $k \geq d_0$.*

Proof. We prove the above theorem using induction on the number of edges of the graph G with a partial colouring C. Consider the three cases:

Case 1. Suppose that $\{u, v\}$ is an edge connecting vertices u and v of G where at most one of which is contained in C. Now, each proper colouring of G is also a proper colouring of $G - e$ since the deletion of the edge has no effect on an already coloured graph. Each proper colouring of $G - e$ is also a proper colouring of G if and only if the endpoints of e are distinct vertices u and v. Hence, the number of proper colourings $P(G_C, k)$

can be obtained from $P(G_C - e, k)$ by subtracting the colourings that assigns the same colour to u and v which is given by $P(G_C/e, k)$. We get that

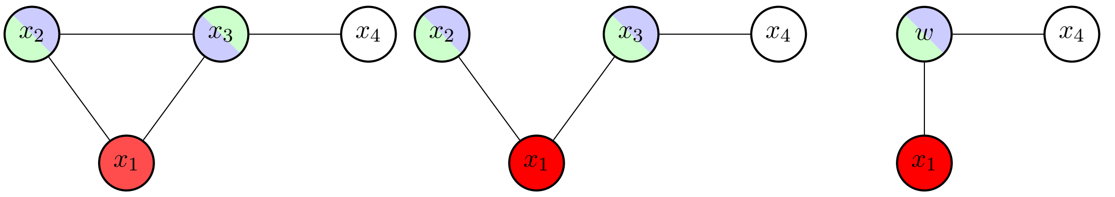
$$P(G_C, k) = P(G_C - e, k) - P(G_C/e, k).$$

(This can also be shown from the deletion-contraction recurrence formula or **Fundamental Reduction Theorem** [6]). Since $G - e$ and G/e have fewer edges than G, we can apply induction to complete the proof in this case.

Case 2. Suppose that G has one vertex v_0 not contained in C. If this vertex is not adjacent to any vertex of G, then $G = C \cup \{v_0\}$, which is a disjoint union. Hence, we can colour v_0 any of the k colours. If this vertex is adjacent to d vertices of C which use d_0 colours, then $P(G_C, k) = \max\{k - d_0, 0\}$.

Case 3. Suppose that ever vertex of G is contained in C. Then, we already have a colouring of G and $P(G_C, k) = 1$.

Hence, by induction the number of edges, the theorem is proved. □



(a) *Partial colouring of G. If x_2 is green, x_3 must be blue and x_4 can be either red or green, so we get two 3-colourings. We get two more if we set x_2 to be blue, give a total of four 3-colourings*

(b) *$G - \{x_2, x_3\}$. Now, x_2 and x_3 can be the same colour. There are two options for x_2, x_3, and x_4, giving eight 3-colourings.*

(c) *$G/\{x_2, x_3\}$. The vertices x_2 and x_3 has been replaced by w. There are two ways of colouring w and two ways of colouring x_4, giving four 3-colourings.*

Figure 2: *3-colourings of a graph G given a partial colouring with deletion and contraction.*

Case 1 in the above proof uses the deletion-contraction formula. We see an example in Figure 2 where $P(G_C, 3) = P(G_C - \{x_2, x_3\}, 3) - P(G_C/\{x_2, x_3\}, 3) = 8 - 4 = 4$. Next, it would be very interesting to understand the conditions under which a partial colouring can be extended to a unique colouring. This motivates the following result.

Theorem 3.6. *Let G be a graph with chromatic number $\chi(G)$ and C be a partial colouring of G using $\chi(G) - 2$ colours. Then there are at least two ways of extending the colouring if the partial colouring can be completed to a total proper colouring of G.*

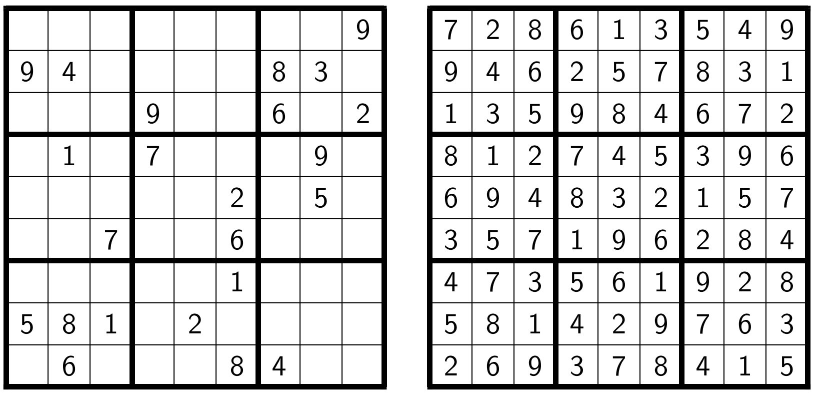
Proof. If the partial colouring only uses $\chi(G) - 2$ colours, then the two colours left can be used interchangeably to complete the partial solution to a total proper colouring. Hence, there are at least two ways of extending to a full solution by swapping the two colours used. □

Theorem 3.6 implies that there exists a unique solution for a partial colouring C if C uses at least $\chi(G) - 1$ colours. We introduce Sudoku puzzles in the next section, but it is insightful to know that for any 9 × 9 Sudoku puzzle, at least eight numbers must be used in the clues [11]. In general, for an $n^2 \times n^2$ Sudoku puzzle, at least $n^2 - 1$ numbers must be used in a given partial solution for it to have a unique solution.

4 Sudoku

In this section, we will explain how the solutions of a Sudoku puzzle can be represented as the points in the vanishing locus of a polynomial in 81 variables. The unique solution of a well-posed Sudoku puzzle can be read off from the reduced Gröbner basis of that ideal. We can formulate the algebraic approach to solving a Sudoku puzzle as a graph colouring problem where the aim is to construct a 9-colouring of a particular graph given a partial solution. It is worth noting that since we can regard solving a Sudoku as a graph problem, graph theoretical methods are much more efficient at solving a Sudoku puzzle than using Gröbner bases. The general problem of solving a Sudoku puzzle on an $n^2 \times n^2$ grid with $n \times n$ blocks is NP-complete and has been subsequently converted to various other NP-complete problems such as constraint satisfaction [15], integer programming [1], boolean satifiability [12], and the Hamiltonian cycle problem [10].

A *Sudoku board* is a particular example of a Latin square. A *Latin square* of order n is an $n \times n$ square grid filled with n distinct symbols such that no symbol appears more than once in each row and column. Typically, Sudoku boards are 9×9 Latin squares filled with the integers 1 to 9 with the additional constraint that the numbers appears only once in each of the nine distinguished 3×3 blocks. We say that a *Sudoku puzzle* is a partial solution to a Sudoku board. A *well-posed* Sudoku puzzle is one that uniquely determines the rest of the board. We take Sudoku puzzle to mean well-posed Sudoku puzzle in the remainder of the report. Figure 3 is an example of a Sudoku puzzle and its corresponding Sudoku board. It has been conjectured that the minimum number of clues given in a Sudoku puzzle for it to be solved uniquely is 17 but has not been proven.



								9
9	4					8	3	
			9			6		2
	1		7				9	
					2		5	
		7			6			
					1			
5	8	1		2				
	6				8	4		

7	2	8	6	1	3	5	4	9
9	4	6	2	5	7	8	3	1
1	3	5	9	8	4	6	7	2
8	1	2	7	4	5	3	9	6
6	9	4	8	3	2	1	5	7
3	5	7	1	9	6	2	8	4
4	7	3	5	6	1	9	2	8
5	8	1	4	2	9	7	6	3
2	6	9	3	7	8	4	1	5

Figure 3: *Well-posed Sudoku puzzle and board*

First, we give an outline of what the technique to solve a Sudoku puzzle using Gröbner basis involves. Then, we provide a mathematical model relating the solution to the vanishing locus of a set of polynomials generated by the constraints of the puzzle. We apply it to Shidoku, a 4×4 variant of Sudoku to make computations easier. The Python notebook containing the code for this section can be found here. Note that the variables in the code are 0-indexed.

4.1 Formulating Sudoku as a Graph Colouring Problem

We can formulate a Sudoku puzzle as a graph where the vertices are coloured with the numbers in the 81 squares. This gives us a graph with 81 nodes. From here on, *nodes*, *variables*, and *cells* will be used interchangeably as they are equivalent unless otherwise stated. There is an edge between two vertices if the corresponding squares are in the same region, that is, if they are

- in the same row,
- in the same column, and
- in the same 3×3 block.

Hence, the degree of each node is $6 + 6 + 8 = 20$. We use a well known formula to calculate the total number of edges for this graph.

Theorem 4.1. *The degree sum formula states that for a given graph $G = (V, E)$,*

$$\sum_{v \in V} \deg(v) = 2|E|.$$

By Theorem 4.1, a typical Sudoku graph has $(81 \times 20)/2 = 810$ edges. Because of the high number of variables that naturally arise from such a large graph, we investigate and apply techniques to solve a smaller variant of Sudoku later.

To model a Sudoku board using polynomial equations, we take inspiration from the graph polynomial defined in Section 3.5. We can represent each of the cells with a different variable, say $x_1, \ldots, x_{81}$, on a graph G_S with 81 nodes. Every entry a_i in the ith cell of a Sudoku board satisfies $a \in \{1, \ldots, 9\}$ if and only if a_i is a root of the univariate polynomial $f_i(x_i)$

defined as

$$f_i(x_i) := \prod_{k=1}^{9}(x_i - k). \tag{3}$$

Next, for $i \neq j$, the polynomial $f_i(x_i) - f_j(x_j)$ vanishes on $\mathbf{V}(x_i - x_j)$, so $x_i - x_j$ is a factor of $f_i(x_i) - f_j(x_j)$. Then, for $i \neq j$, we define the quotient of $f_i(x_i) - f_j(x_j)$ when divided by $x_i - x_j$ to be

$$g_{ij}(x_i, x_j) = \frac{f_i - f_j}{x_i - x_j}, \tag{4}$$

where $g_{ij} \in \mathbb{Q}[x_i, x_j]$. Next, we join two vertices with an edge as described from earlier, and so the set E is defined to be

$$E = \{(i,j) \mid 1 \leq i < j \leq 81, i\text{th and } j\text{th cell are in the same region}\}.$$

Let $I \subseteq \mathbb{Q}[x_1, \ldots, x_{81}]$ be the ideal generated by the 891 polynomials f_i, $i = 1, \ldots, 81$ and g_{ij}, $(i,j) \in E$. Then, solving a Sudoku puzzle is equivalent to finding a 9-colouring of a graph given a partial colouring, which are the clues to the Sudoku. We will call this representation of Shidoku by the polynomial Equations 3 and 4 the *quotient Shidoku system.* Section 4.2.2 provides a similar method for finding a solution to a puzzle.

Proposition 4.1. *Let $\mathbf{V}(I)$ be the vanishing locus of I in $\mathbb{Q}^{81}$, and let $a = (a_1, \ldots, a_{81})$ be a point. Then $a \in \mathbf{V}(I)$ if and only if $a_i \in \{1, \ldots, 9\}$, for $i = 1, \ldots, 81$, and $a_i \neq a_j$, for $(i,j) \in E$.*

Proof. If $a_i \in \{1, \ldots, 9\}$ for $i = 1, \ldots, 81$ and and $a_i \neq a_j$, for $(i,j) \in E$, then we have already seen that f_i and g_{ij} vanish at a, so $a \in \mathbf{V}(I)$. Conversely, let $a \in \mathbf{V}(I)$. Then $f_i(a) = 0$ so $a_i \in \{1, \ldots, 9\}$ for all i. Now, suppose for contradiction that $a_i = a_j =: b$ for some $(i,j) \in E$. We have that

$$g_{ij}(x_i, x_j) = \frac{f_i(x_i) - f_j(x_j)}{x_i - x_j} \implies f_i(x_i) = (x_i - x_j)g_{ij}(x_i, x_j) + f_j(x_j).$$

Substituting in b for x_j gives $f_i(x_i) = (x_i - b)g_{ij}(x_i, b)$, and since $g_{ij}(b,b) = 0$ by assumption, this implies that b is a zero of f_i of order at least two, which is impossible. □

Next, we show that for a well-posed puzzle, the unique solution corresponds to a single point.

Proposition 4.2. *Let S be an explicitly given, well-posed Sudoku puzzle with preassigned numbers (clues) $\{a_i\}_{i \in L}$ for some subset $L \subset \{1, \ldots, 81\}$. We associate to S the ideal $I_S = I + \langle \{x_i - a_i\}_{i \in L} \rangle$ (the clues of the puzzle). Then the reduced Gröbner basis of I_S is given by $G_{red} = \langle x_1 - a_1, \ldots, x_{81} - a_{81} \rangle$, and $(a_1, \ldots, a_{81})$ is the solution of the Sudoku.*

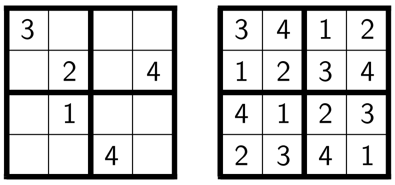
Proof. Given that S is well-posed, assume that S has a unique solution $(a_1, \ldots, a_{81})$. By Proposition 4.1 and the Nullstellensatz, the radical ideal $\sqrt{I_S}$ is the maximal ideal $\langle x_1 - a_1, \ldots, x_{81} - a_{81} \rangle$. This means that I_S contains a suitable power of $x_i - a_i$ for each i. Since I_S contains the square-free polynomials $f_i(x_i) = \prod_{k=1}^{9}(x_i - k)$, we get that the elimination ideals $I_S \cap K[x_i]$ are generated by the $x_i - a_i$. This, $I_S = \langle x_1 - a_1, \ldots, x_{81} - a_{81} \rangle$, which is the form of the reduced Gröbner basis. □

We now apply this property to finding the reduced Gröbner basis for an example Shidoku puzzle using different methods.

4.2 Shidoku

A *Shidoku board* is a 4×4 Latin square that whose regions, and similarly, a *well-posed Shidoku puzzle* is a partial solution to a Shidoku board that uniquely determines the board.

Each of the methods presented for solving the puzzle uses values from different number systems to represent the n variables given by the n cells: the roots of unity method solves for solutions in $\mathbb{C}^n$, sum-product method in $\mathbb{Z}^n$, and boolean method in $\mathbb{Z}_2^n$.



3			
	2		4
	1		
		4	

3	4	1	2
1	2	3	4
4	1	2	3
2	3	4	1

Figure 4: *Well-posed Shidoku puzzle and board*

4.2.1 Roots of Unity Method

We can reformulate the techniques used to solve graph colouring in the context of Sudoku. Similar to the graph colouring solution in Section 3.1.2, we can represent pairs of cells that share a region rather than a whole region in itself.

To start, we replace the numbers $1, 2, 3$ and 4 with the fourth roots of unity, ± 1 and $\pm i$. Note that we can make any arbitrary choice for the root of unity associated with a number. Then, we can encode this information in each cell x_i for $1 \leq i \leq 16$ which takes on values from the fourth roots of unity in 16 polynomial equations of the form

$$x_i^4 - 1 = 0. \tag{5}$$

Next, if we fix x_i, for x_j in the same region as x_i, we have another set of polynomial equations that encodes the puzzle. Since $x_i^4 - x_j^4 = 0$, factoring gives $(x_i - x_j)(x_i + x_j)(x_i^2 + x_j^2) = 0$. Now, x_j cannot be the same number as x_i because they must be distinct roots of unity so $x_i - x_j \neq 0$. Hence, we also have polynomials of the form

$$(x_i + x_j)(x_i^2 + x_j^2) = 0. \tag{6}$$

The 56 polynomials of this form, along with the 16 from Equation 5, gives 72 polynomials total that we can generate our ideal I_{RoI}. Lastly, we add in the constraints given by the clues. If we set the symbols $\{1, 2, 3, 4\}$ to the complex fourth roots of unity $\{i, -1, -i, 1\}$, we also add the equations $x_1 + i = 0, x_6 + 1 = 0, x_8 - 1 = 0, x_{10} - i = 0, x_{15} - 1 = 0$ to the ideal. We can then check for the solution of the corresponding variety or read off the reduced Gröbner basis.

4.2.2 Sum-Product Method

We can also have a representation of the board based on its regions. Every cell can take on a number from $\{1, 2, 3, 4\}$. Fix i for $1 \leq i \leq 16$. For each x_i, we can encode this in the polynomial equation

$$(x_i - 1)(x_i - 2)(x_i - 3)(x_i - 4) = 0 \tag{7}$$

since the x_i must be one of $\{1, 2, 3, 4\}$. Now suppose we have four cells w, x, y, z in the same region. It turns out that the only way to choose four numbers that sum to 10 and multiply to 24 from the set $\{1, 2, 3, 4\}$ is to use each number once. This means that for w, x, y, z in each region, we have polynomial equations of the form

$$w + x + y + z - 10 = 0, \text{ and} \tag{8}$$

$$wxyz - 24 = 0. \tag{9}$$

We get 16 equations from Equation 7 and 24 equations from Equation 8 and Equation 9 since there are 12 regions, giving a total of 40 initial polynomials that we can use to generate the ideal I_{SP}. We also add constraints based on the cells that have been filled on any given Shidoku puzzle. For example, if we consider the Shidoku puzzle in Figure 4 (left), we would also add $x_1 - 3 = 0$, $x_6 - 2 = 0$, $x_8 - 4 = 0$, $x_{10} - 1 = 0$, and $x_{15} - 4 = 0$. Similarly, we find the point in the corresponding variety if it has a unique solution or read off the reduced Gröbner basis.

Remark 4.1. *For a normal 9×9 Sudoku, there is more than one choice of selecting numbers from $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ that sum to 45 and multiply to $9! = 362880$, namely $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $\{1, 2, 4, 4, 4, 5, 7, 9, 9\}$. We can instead assign each cell a number from $\{-2, -1, 1, 2, 3, 4, 5, 6, 7\}$ since it is the smallest set in magnitude for which each of the nine elements are picked exactly once to make the sum and product.*

4.2.3 Boolean Variable Method

Lastly, we explain the Boolean method. For each cell, we introduce the four variables for each cell, $x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}$ for $1 \leq i \leq 16$, where we set $x_{i,k} = 1$ when cell x_i takes the value k, and $x_{i,k} = 0$ otherwise. Encoding the individual cells for the puzzle now takes 64 variables instead of 16 (it is suggested by Bernasconi et al. [2] and Sato et al. [14] that the cost of finding a Gröbner basis is greatly reduced using Boolean variables). For each i, we then get polynomials of the form

$$x_{i,k}(x_{i,k} - 1) = 0 \tag{10}$$

Because each $x_{i,j}$ can only take on values 0 and 1, we also get 16 polynomials of the form

$$x_{i,1} + x_{i,2} + x_{i,3} + x_{i,4} - 1 = 0 \tag{11}$$

Finally, we require any two cells x_i and x_j in the same region to have different values. Therefore, for each possible k, at least one of $x_{i,k}$ and $x_{j,k}$ must be 0. We have 56 polynomial equations of the form

$$x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + x_{i,3}x_{j,3} + x_{i,4}x_{j,4} = 0. \tag{12}$$

We get a total of 136 initial polynomials that we can use to generate our ideal I_{BV}. Lastly, we need to add the equations $x_{1,3} - 1, x_{6,2} - 1, x_{8,4} - 1, x_{10,1}01$, and $x_{15,4} - 1$ to the ideal which corresponds to the values in the cell from the clues given. We then find the reduced Gröbner basis to find the solution.

4.3 Kropki Sudoku

Whilst the Gröbner basis method might be overshadowed by classical Sudoku solving heuristics, it can provide a solution to variants of Sudoku. We look at a Sudoku with Kropki dots. The normal rules of Sudoku apply with two added constraints: if the absolute difference between any two digits in neighbouring cells is 1, the cells are separated by a white dot, and if the digit is half of the digit in the neighbouring cell, they are separated by a black dot. The dots between 1 and 2 can be either white or black. We can model the Kropki dots using polynomial equations. If cells x_i and x_j are separated by a white dot, then we get polynomials of the form

$$(x_i - x_j)(x_j - x_i) + 1 = 0 \tag{13}$$

since either $x_i - x_j = -1$ or $x_j - x_i = 1$ and the other giving 1. Now, if x_i and x_j are separated by a black dot, we can model this with polynomials of the form

$$(x_i - 2x_j)(2x_i - x_j) = 0. \tag{14}$$

This is because we are not sure which one of x_i or x_j is double the other. We get that either $x_i - 2x_j = 0$ or $2x_i - x_j$. Along with the polynomials generated from the usual constraints of Sudoku, we add Equations 13 and 14 to our ideal and compute a Gröbner basis.

However, using the `sympy` package in Python is very inefficient and is not feasible to find a Gröbner basis and solve for over 900 polynomials. We provide the notebook file here.

5 Future Directions

There are many future directions for work with Gröbner bases not limited to computational algebra. Fields such as robotics [13], error-correcting codes [4], and cryptography [7] are key areas of Gröbner basis research. Perhaps progress can also be made into solving the open problem of determining the minimum number of clues for a Sudoku puzzle to have a unique solution.

Acknowledgements

I would like to thank my supervisor Angus McAndrew for taking time out of his summer break to supervise a reading course in algebraic geometry. I would also like to thank Lekh Bhatia, Kieran Lelis, and Tushar Muralidharan for their support as well as being nerdy enough to complete a summer course in mathematics wih me.

References

[1] Andrew Bartlett, Timothy P Chartier, Amy N Langville, and Timothy D Rankin. An integer programming model for the sudoku problem. *Journal of Online Mathematics and its Applications*, 8(1), 2008.

[2] Anna Bernasconi, Bruno Codenotti, Valentino Crespi, and Giovanni Resta. Computing groebner bases in the boolean setting with applications to counting. In *WAE*, pages 209–218, 1997.

[3] Bruno Buchberger. Bruno buchberger's phd thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of symbolic computation*, 41(3-4):475–511, 2006.

[4] Xuemin Chen, I.S. Reed, T. Helleseth, and T.K. Truong. Use of grobner bases to decode binary cyclic codes up to the true minimum distance. *IEEE Transactions on Information Theory*, 40(5):1654–1661, 1994. doi: 10.1109/18.333885.

[5] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[6] Fengming Dong, Khee-Meng Koh, and Kee L Teo. *Chromatic polynomials and chromaticity of graphs*. World Scientific, 2005.

[7] Jean-Charles Faugere and Antoine Joux. Algebraic cryptanalysis of hidden field equation (hfe) cryptosystems using gröbner bases. In *Advances in Cryptology-CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings 23*, pages 44–60. Springer, 2003.

[8] Christopher A Francisco, Huy Tài Hà, and Adam Van Tuyl. Colorings of hypergraphs, perfect graphs, and associated primes of powers of monomial ideals. *Journal of Algebra*, 331(1):224–242, 2011.

[9] Georges Gonthier et al. Formal proof–the four-color theorem. *Notices of the AMS*, 55 (11):1382–1393, 2008.

[10] Michael Haythorpe. Reducing the generalised sudoku problem to the hamiltonian cycle problem. *AKCE International Journal of Graphs and Combinatorics*, 13(3):272–282, 2016.

[11] Agnes M Herzberg and M Ram Murty. Sudoku squares and chromatic polynomials. *Notices of the AMS*, 54(6):708–717, 2007.

[12] Ines Lynce Ist, Inês Lynce, and Joël Ouaknine. Sudoku as a sat problem. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (AIMATH)*, pages 1–9, 2006.

[13] Antonio Montes. A new algorithm for discussing gröbner bases with parameters. *Journal of Symbolic Computation*, 33(2):183–208, 2002.

[14] Yosuke Sato, Akira Nagai, and Shutaro Inoue. On the computation of elimination ideals of boolean polynomial rings. In *Computer Mathematics: 8th Asian Symposium, ASCM 2007, Singapore, December 15-17, 2007. Revised and Invited Papers*, pages 334–348. Springer, 2008.

[15] Helmut Simonis. Sudoku as a constraint problem. In *CP Workshop on modeling and reformulating Constraint Satisfaction Problems*, volume 12, pages 13–27. Citeseer, 2005.