Grobner Bases

Riyaz Ahuja, Dennis Chen, Rohan Jain

Carnegie Mellon University

2025-11-29

Outline

1. Introduction 5
   - 1.1 Setting 6
   - 1.2 Preliminary Definitions . . 7
   - 1.3 Degree Lexicographic Order 8
   - 1.4 Parts of the polynomial . 10
   - 1.5 Gaussian Elimination (v2) 12
   - 1.6 Polynomial Reduction . . 13
   - 1.7 What is the Grobner Basis? 14
   - 1.8 Finding the Grobner Basis 15
   - 1.9 Example of Buchberger's Algorithm . . 17
   - 1.10 Unique Representatives 20
   - 1.11 Computing the representative 21
2. Applications of Grobner Bases . 22
   - 2.1 Ideal Membership Problem 23
   - 2.2 Radicals 24

Outline

2.3 Exercise 4.15 (H2, 3.8) 25
2.4 Elimination: Motivation . 27
2.5 Elimination Theorem . . . 28
2.6 Graph Coloring 29
2.7 Representing Graph Coloring with Ideals . 30
2.8 Coloring the graph 31
2.9 Actually computing a coloring 32
2.10 Chicken McNugget Theorem . . 35
2.11 Representing this as an Ideal 36
2.12 Using the representative 37
2.13 Forcing the representative 38
3. Feasibility 39
3.1 Runtime Analysis 40
3.2 Can we do better? 41
3.3 We can do better 42

Outline

3.4 Avoiding the worst case . . 43
3.5 Where are Grobner bases used? 44
4. Grobner Bases 3: CAS for Theorem Proving 45
4.1 Recap 46
5. Macaulay2 47
5.1 Macaulay2 . . 48
5.2 Macaulay2 . . 49
5.3 Macaulay2 . . 50
5.4 Functionality 51
5.5 Gröbner Base Computation 52
6. Interactive Theorem Proving 53
6.1 Interactive Theorem Proving 54
6.2 Theorem Proving in Lean4 55
6.3 Dependent Type Theory 56
6.4 Computability 57
7. LeanM2 58
7.1 LeanM2 59
7.2 Implementation 60

Outline

7.3 Implementation 61
7.4 Implementation 62
7.5 Implementation 63
7.6 Examples and Demo 64
7.7 Future Work 65

1. Introduction

1.1 Setting

Fix your field k and consider the ring $R = k[x_1, x_2, ..., x_n]$. Remember that by Hilbert Basissatz, any ideal in this ring is finitely generated.

1.1 Setting

Fix your field k and consider the ring $R = k[x_1, x_2, ..., x_n]$. Remember that by Hilbert Basissatz, any ideal in this ring is finitely generated.

Let $f = xy$ and $g = xy - z$ in $k[x, y, z]$ and define $I = \langle f, g \rangle$. Someone may ask whether $z \in I$ or not, and we can respond by saying

$$z = f - g.$$

1.1 Setting

Fix your field k and consider the ring $R = k[x_1, x_2, ..., x_n]$. Remember that by Hilbert Basissatz, any ideal in this ring is finitely generated.

Let $f = xy$ and $g = xy - z$ in $k[x, y, z]$ and define $I = \langle f, g \rangle$. Someone may ask whether $z \in I$ or not, and we can respond by saying

$$z = f - g.$$

But what an expression like z^2? Is that in I as well? This makes us define our problem.

1.1 Setting

Fix your field k and consider the ring $R = k[x_1, x_2, ..., x_n]$. Remember that by Hilbert Basissatz, any ideal in this ring is finitely generated.

Let $f = xy$ and $g = xy - z$ in $k[x, y, z]$ and define $I = \langle f, g \rangle$. Someone may ask whether $z \in I$ or not, and we can respond by saying

$$z = f - g.$$

But what an expression like z^2? Is that in I as well? This makes us define our problem.

Ideal Membership Problem: Given an ideal $I = (f_1, ..., f_n) \subset R$ and a polynomial $f \in R$, is $f \in I$? If it is, what's the linear combination of f_i that is equal to f?

1.2 Preliminary Definitions

Definition (Monomial ordering): Let $\alpha = [\alpha_1, \alpha_2, ..., \alpha_n]$ be a multi-index, meaning

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}.$$

There is a total order $\prec$ on R satisfying:

1) $x^\alpha \prec x^\beta \Longrightarrow x^{\alpha+\gamma} \prec x^{\beta+\gamma}$ for multi-indices α, β, γ.
2) $1 \prec x^\alpha$ for all $\alpha \in \mathbb{N}^n \setminus \{0\}$.

1.3 Degree Lexicographic Order

The previous definition creates a "degree lexicographic order". In simple terms, if we have $x > y > z$ lexicographically, suppose

$$f = x^3 + z^7 + x^2 y + y z^2 + y^2 z + y + x.$$

1.3 Degree Lexicographic Order

The previous definition creates a "degree lexicographic order". In simple terms, if we have $x > y > z$ lexicographically, suppose

$$f = x^3 + z^7 + x^2y + yz^2 + y^2z + y + x.$$

Writing out their multi-indices, we get

$$(3, 0, 0), (0, 0, 7), (2, 1, 0), (0, 1, 2), (0, 2, 1), (0, 1, 0), (1, 0, 0).$$

1.3 Degree Lexicographic Order

The previous definition creates a "degree lexicographic order". In simple terms, if we have $x > y > z$ lexicographically, suppose

$$f = x^3 + z^7 + x^2y + yz^2 + y^2z + y + x.$$

Writing out their multi-indices, we get

$$(3,0,0), (0,0,7), (2,1,0), (0,1,2), (0,2,1), (0,1,0), (1,0,0).$$

We want to sort these by index from left to right, meaning the right order is

$$(3,0,0), (2,1,0), (1,0,0), (0,2,1), (0,1,2), (0,1,0), (0,0,7).$$

1.3 Degree Lexicographic Order

So we get that f should be written as

$$f = x^3 + x^2y + x + y^2z + yz^2 + y + z^7.$$

1.4 Parts of the polynomial

Definition: Fix a monomial order on $k[x_1, ..., x_n]$ and let $f \in k[x_1, ..., x_n]$ written as

$$f = c_1 X^{\alpha_1} + \cdots + c_r X^{\alpha_r}$$

where each α_i is a multiindex such that $X^{\alpha_1} > \cdots > X^{\alpha_r}$ with respect to our monomial ordering. We define:

- $\mathrm{LM}(f) = X^{\alpha_1}$ (the leading monomial)
- $\mathrm{LC}(f) = c_1$ (the leading coefficient)
- $\mathrm{LT}(f) = c_1 X^{\alpha_1} = \mathrm{LC}(f) \cdot \mathrm{LM}(f)$ (the leading term)

1.4 Parts of the polynomial

Example: Let $f = 42x^3 + 5y^2 + z$. Then,

1.4 Parts of the polynomial

Example: Let $f = 42x^3 + 5y^2 + z$. Then,

$$\mathrm{LM}(f) = x^3,$$

1.4 Parts of the polynomial

Example: Let $f = 42x^3 + 5y^2 + z$. Then,

$$\mathrm{LM}(f) = x^3, \quad \mathrm{LC}(f) = 42,$$

1.4 Parts of the polynomial

Example: Let $f = 42x^3 + 5y^2 + z$. Then,

$$\mathrm{LM}(f) = x^3, \quad \mathrm{LC}(f) = 42, \quad \mathrm{LT}(f) = 42x^3.$$

1.5 Gaussian Elimination (v2)

The motivation for Grobner bases comes from wanting to solve systems of polynomials efficiently. Consider the example below

$$\begin{cases} f := xy^2 + 4 = 0 \\ g := x^2y - 8 = 0 \end{cases}.$$

1.5 Gaussian Elimination (v2)

The motivation for Grobner bases comes from wanting to solve systems of polynomials efficiently. Consider the example below

$$\begin{cases} f := xy^2 + 4 = 0 \\ g := x^2y - 8 = 0 \end{cases}.$$

We want to “eliminate” the first term as we did in classic Gaussian Elimination. This introduces the idea of an S-polynomial, denoted $S(f, g)$. In this case, we get

$$S(f, g) = xf - yg = 4x + 8y.$$

1.5 Gaussian Elimination (v2)

The motivation for Grobner bases comes from wanting to solve systems of polynomials efficiently. Consider the example below

$$\begin{cases} f := xy^2 + 4 = 0 \\ g := x^2 y - 8 = 0 \end{cases}.$$

We want to "eliminate" the first term as we did in classic Gaussian Elimination. This introduces the idea of an S-polynomial, denoted $S(f, g)$. In this case, we get

$$S(f, g) = xf - yg = 4x + 8y.$$

By solving and checking with our equations, we get $(x, y) = (-1, 2)$.

1.6 Polynomial Reduction

Definition: Given $f, g, h \in R$ with $g \neq 0$, we can say f reduces to h modulo g if $\mathrm{LM}(g)$ divides a non-zero term X of f and

$$h = f - \frac{X}{\mathrm{LT}(g)} \cdot g.$$

1.6 Polynomial Reduction

Definition: Given $f, g, h \in R$ with $g \neq 0$, we can say f reduces to h modulo g if $\mathrm{LM}(g)$ divides a non-zero term X of f and

$$h = f - \frac{X}{\mathrm{LT}(g)} \cdot g.$$

- xyz reduces to y^2 modulo $xz - y$ because

$$xyz - y \cdot (xz - y) = y^2.$$

- $x^2 z + 3y^2$ reduces to $-x^3 - 7xy + 3y^2$ modulo $x^2 + xz + 7y$ because

$$x^2 z + 3y^2 - x \cdot \left(x^2 + xz + 7y\right) = -x^3 - 7xy + 3y^2.$$

1.7 What is the Grobner Basis?

Definition: Given $f, h \in R$ and a set $G = \{g_1, ..., g_n\} \subset R$ of nonzero polynomials, we can say f reduces to h modulo G if there exists a sequence of indices $i_1, ..., i_\ell \in \{1, ..., n\}$ and polynomials $h_1, ...h_{\ell-1}$ such that f reduces to h_1 modulo g_{i_1}, h_1 reduces to h_2 modulo g_{i-2}, ..., $h_{\ell-1}$ reduces to h modulo g_{i_ℓ}.

1.7 What is the Grobner Basis?

Definition: Given $f, h \in R$ and a set $G = \{g_1, \ldots, g_n\} \subset R$ of nonzero polynomials, we can say f reduces to h modulo G if there exists a sequence of indices $i_1, \ldots, i_\ell \in \{1, \ldots, n\}$ and polynomials $h_1, \ldots h_{\ell-1}$ such that f reduces to h_1 modulo g_{i_1}, h_1 reduces to h_2 modulo g_{i-2}, …, $h_{\ell-1}$ reduces to h modulo g_{i_ℓ}.

Definition: A polynomial f is called reduced with respect to G if it cannot be reduced modulo G. That is, no term of f is divisible by $\mathrm{LM}(g_i)$ for any i.

1.7 What is the Grobner Basis?

Definition: Given $f, h \in R$ and a set $G = \{g_1, ..., g_n\} \subset R$ of nonzero polynomials, we can say f reduces to h modulo G if there exists a sequence of indices $i_1, ..., i_\ell \in \{1, ..., n\}$ and polynomials $h_1, ...h_{\ell-1}$ such that f reduces to h_1 modulo g_{i_1}, h_1 reduces to h_2 modulo g_{i-2}, ..., $h_{\ell-1}$ reduces to h modulo g_{i_ℓ}.

Definition: A polynomial f is called reduced with respect to G if it cannot be reduced modulo G. That is, no term of f is divisible by $\mathrm{LM}(g_i)$ for any i.

Definition: A set $G = \{g_1, ..., g_n\}$ of non-zero polynomials is a Grobner basis for the ideal $I = (f_1, ..., f_m)$ if for all non-zero $f \in I$, we have that $\mathrm{LM}(g_i) \mid \mathrm{LM}(f)$ for some $g_i \in G$.

1.8 Finding the Grobner Basis

We introduce Buchberger's Algorithm. Let $F = \{f_1, ..., f_m\}$ be a set of polynomials.

1) $G := F$. Construct an initial set of pairs to examine:

$$P := \{(f, g) \mid f, g \in G, f \neq g\}.$$

2) While P is non-empty,
   a) Select an remove a pair $(f, g) \in P$.
   b) Compute $L := \operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g))$.
   c) Compute $S(f, g) = \frac{L}{\operatorname{LT}(f)} f - \frac{L}{\operatorname{LT}(g)} g$.
   d) Reduce $S(f, g)$ with respect to G with the following reduction process:

1.8 Finding the Grobner Basis

- While there is a nonzero term T in $S(f,g)$ for which there exists an $h \in G$ with $\mathrm{LM}(h) \mid T$, write $T = cX$ (with X monomial and c coefficient) and replace

$$S(f,g) := S(f,g) - \frac{c}{\mathrm{LC}(h)} \cdot \frac{X}{\mathrm{LM}(h)} h.$$

Denote the fully reduced polynomial as S'.

e) If S' is nonzero, add it to G. And for every h in G, add the pair (S', h) to P.

3) When no new S-polynomials reduce to a nonzero remainder, (i.e. when P is empty), the current set G is the Grobner basis we are looking for.

1.9 Example of Buchberger's Algorithm

Let $f_1 = x^2 - y$ and $f_2 = xy - 1$. Our goal is to compute the Grobner basis for the ideal $I = \langle f_1, f_2 \rangle$.

1.9 Example of Buchberger's Algorithm

Let $f_1 = x^2 - y$ and $f_2 = xy - 1$. Our goal is to compute the Grobner basis for the ideal $I = \langle f_1, f_2 \rangle$.

We start by computing the S polynomial

$$S(f_1, f_2) = y f_1 - x f_2 = x - y^2.$$

1.9 Example of Buchberger's Algorithm

Let $f_1 = x^2 - y$ and $f_2 = xy - 1$. Our goal is to compute the Grobner basis for the ideal $I = \langle f_1, f_2 \rangle$.

We start by computing the S polynomial

$$S(f_1, f_2) = yf_1 - xf_2 = x - y^2.$$

Since $x - y^2$ cannot be reduced by f_1 or f_2, we add it to our basis:

$$f_3 := x - y^2.$$

1.9 Example of Buchberger's Algorithm

So now we have $G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}$. Now we want to calculate $S(f_1, f_3)$ and $S(f_2, f_3)$.

1.9 Example of Buchberger's Algorithm

So now we have $G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}$. Now we want to calculate $S(f_1, f_3)$ and $S(f_2, f_3)$. Firstly,

$$S(f_1, f_3) = f_1 - xf_3 = xy^2 - y.$$

1.9 Example of Buchberger's Algorithm

So now we have $G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}$. Now we want to calculate $S(f_1, f_3)$ and $S(f_2, f_3)$. Firstly,

$$S(f_1, f_3) = f_1 - x f_3 = xy^2 - y.$$

However, we can see that $S(f_1, f_3) = y f_2$, so we don't add it.

1.9 Example of Buchberger's Algorithm

So now we have $G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}$. Now we want to calculate $S(f_1, f_3)$ and $S(f_2, f_3)$. Firstly,

$$S(f_1, f_3) = f_1 - xf_3 = xy^2 - y.$$

However, we can see that $S(f_1, f_3) = yf_2$, so we don't add it.

$$S(f_2, f_3) = f_2 - yf_3 = y^3 - 1.$$

1.9 Example of Buchberger's Algorithm

So now we have $G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}$. Now we want to calculate $S(f_1, f_3)$ and $S(f_2, f_3)$. Firstly,

$$S(f_1, f_3) = f_1 - xf_3 = xy^2 - y.$$

However, we can see that $S(f_1, f_3) = yf_2$, so we don't add it.

$$S(f_2, f_3) = f_2 - yf_3 = y^3 - 1.$$

If we take f_3 and replace $x = y^2$ into this polynomial, we get that

$$y^3 - 1 = xy - 1 = f_2.$$

As such, we don't want to add this polynomial to our basis either.

1.9 Example of Buchberger's Algorithm

As we have considered every S polynomial of every pair of polynomials in our basis, we are done and we have that our Grobner basis is

$$G = \{f_1 = x^2 - y, f_2 = xy - 1, f_3 = x - y^2\}.$$

1.10 Unique Representatives

A basis $\{g_1, ..., g_n\}$ of I is a Grobner basis iff every element of $A(X) = k[\boldsymbol{x}]/I$ has exactly one representative with none of its terms divisible by any $\mathrm{LM}(g_i)$.

1.10 Unique Representatives

A basis $\{g_1, ..., g_n\}$ of I is a Grobner basis iff every element of $A(X) = k[\boldsymbol{x}]/I$ has exactly one representative with none of its terms divisible by any $\mathrm{LM}(g_i)$.

Proof: Follows from the definition of a Grobner Basis.

1.10 Unique Representatives

A basis $\{g_1, ..., g_n\}$ of I is a Grobner basis iff every element of $A(X) = k[\boldsymbol{x}]/I$ has exactly one representative with none of its terms divisible by any $\mathrm{LM}(g_i)$.

Proof: Follows from the definition of a Grobner Basis.

- Grobner Basis $\Longrightarrow$ Unique Representative: For the sake of contradiction suppose some polynomial has two representatives r_1 and r_2. But then $r_1 - r_2 \in I$, and the leading term from $r_1 + (r_2 - r_1)$ comes from I.

1.10 Unique Representatives

A basis $\{g_1, ..., g_n\}$ of I is a Grobner basis iff every element of $A(X) = k[\boldsymbol{x}]/I$ has exactly one representative with none of its terms divisible by any $\mathrm{LM}(g_i)$.

Proof: Follows from the definition of a Grobner Basis.

- Grobner Basis $\Longrightarrow$ Unique Representative: For the sake of contradiction suppose some polynomial has two representatives r_1 and r_2. But then $r_1 - r_2 \in I$, and the leading term from $r_1 + (r_2 - r_1)$ comes from I.
- Unique Representative $\Longrightarrow$ Grobner Basis: The unique representative of 0 is 0.

1.11 Computing the representative

Just use the division algorithm the exact same way as computing the Grobner basis.

2. Applications of Grobner Bases

2.1 Ideal Membership Problem

The biggest use of Grobner Bases for mathematicians is the *ideal membership problem.*

2.1 Ideal Membership Problem

The biggest use of Grobner Bases for mathematicians is the *ideal membership problem.*

Solution: Compute a Grobner Basis for I and the representative of f. If it is 0, then $f \in I$; otherwise, we know for sure that $f \notin I$.

2.2 Radicals

Suppose $I = \langle f_1, ..., f_n \rangle$ and our Grobner Basis is G.

We overload notation a little and define $\mathrm{LM}(I)$ to be the ideal of I generated by the leading monomials $\mathrm{LM}(f)$ for all $f \in I$.

2.2 Radicals

Suppose $I = \langle f_1, \ldots, f_n \rangle$ and our Grobner Basis is G.

We overload notation a little and define $\mathrm{LM}(I)$ to be the ideal of I generated by the leading monomials $\mathrm{LM}(f)$ for all $f \in I$.

1) Radical Membership Problem: Recall that $f \in \sqrt{I} \Longleftrightarrow 1 \in \langle f_1, \ldots, f_n, 1 - yf \rangle$.
2) Is I radical: It is a fact that $\mathrm{LM}(G)$ generates $\mathrm{LM}(I)$ and G being square free implies I is radical (since G generates I).

2.3 Exercise 4.15 (H2, 3.8)

Consider the projection of the twisted cubic (i.e. the Veronese embedding $\mathbb{P}^1 \ni [x : y] \mapsto [x^3 : x^2y : xy^2 : y^3] \in \mathbb{P}^3$) from (i) the point $[1 : 0 : 0 : 1]$ and from (ii) the point $[0 : 1 : 0 : 0]$. In each case, show the image is an irreducible curve in $\mathbb{P}^2$, and find the defining equation.

Solution: For the sake of time we only do (i)

- Take the projection $[a : b : c : d] \mapsto [b : c : a - d]$. The image has parametrization $[x, y] \mapsto [x^2y : xy^2 : x^3 - y^3]$.
- A point $[a : b : c]$ is in the image if some $[x : y : a : b : c]$ is in the ideal

$$I := \langle a - x^2y, b - xy^2, c - (x^3 - y^3) \rangle.$$

2.3 Exercise 4.15 (H2, 3.8)

- Eliminate x and y from the ideal to get a single equation in terms of a, b, and c. (**How?** We will cover this right after!)
- If we really wanted to we could use M2 to check irreducibility, but that's kind of silly in this case: the image of a (non-constant) dominant rational map is irreducible.

What is the point? We no longer have to make ad-hoc arguments that the image is the vanishing ideal of some polynomial; we (or M2) can mindlessly perform some calculations.

2.4 Elimination: Motivation

We would like to write I in the form

$$\langle f, g_1, g_2 \rangle$$

where f depends entirely on a, b, and c, and g_1 and g_2 yield solutions x and y after we plug in a, b, and c which satisfy f.

2.5 Elimination Theorem

For $I \subseteq k[\boldsymbol{x}]$ with Grobner basis G (with respect to lexicographic ordering $x_n \prec \ldots \prec x_1$),

$$G_\ell := G \cap k[x_{\ell+1}, \ldots, x_n]$$

is a Grobner basis of

$$I_\ell := I \cap k[x_{\ell+1}, \ldots, x_n].$$

2.5 Elimination Theorem

For $I \subseteq k[\boldsymbol{x}]$ with Grobner basis G (with respect to lexicographic ordering $x_n \prec ... \prec x_1$),

$$G_\ell := G \cap k[x_{\ell+1}, ..., x_n]$$

is a Grobner basis of

$$I_\ell := I \cap k[x_{\ell+1}, ..., x_n].$$

Idea: Just show that $\mathrm{LM}(I_\ell) = \mathrm{LM}(G_\ell)$.

2.5 Elimination Theorem

For $I \subseteq k[\boldsymbol{x}]$ with Grobner basis G (with respect to lexicographic ordering $x_n \prec \ldots \prec x_1$),

$$G_\ell := G \cap k[x_{\ell+1}, \ldots, x_n]$$

is a Grobner basis of

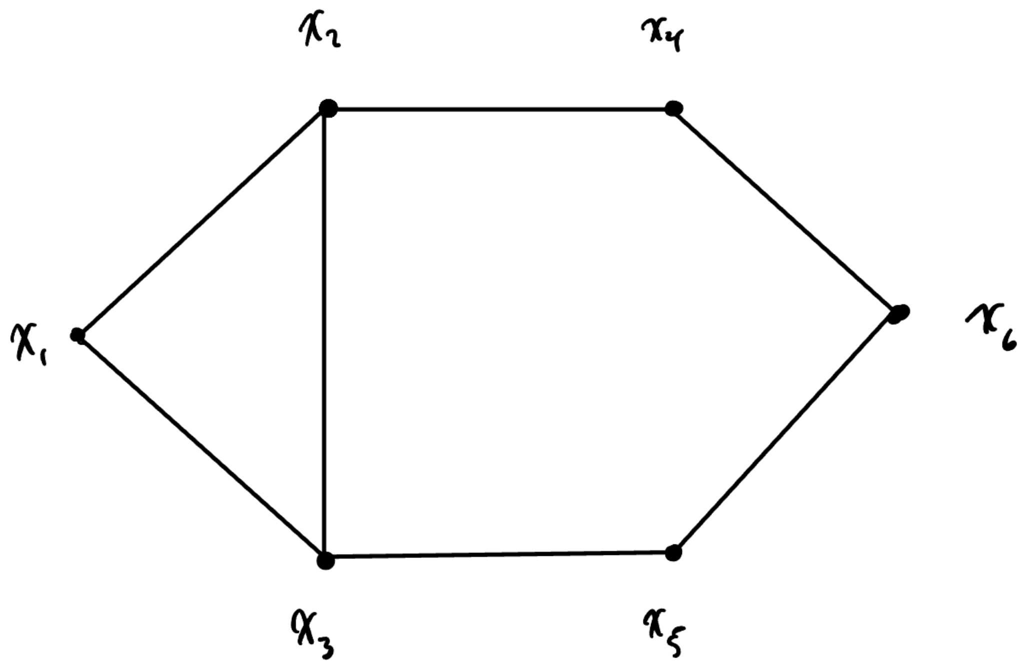
$$I_\ell := I \cap k[x_{\ell+1}, \ldots, x_n].$$

Idea: Just show that $\mathrm{LM}(I_\ell) = \mathrm{LM}(G_\ell)$.

See also: Extension Theorem. (This is how we recover a full solution from a partial solution.)

2.6 Graph Coloring

Let's analyze the graph below. We are wondering whether this graph is three-colorable.



2.7 Representing Graph Coloring with Ideals

Work in $\mathbb{F}_3$ (integers mod 3) and let $\{-1, 0, 1\}$ be the colors.

2.7 Representing Graph Coloring with Ideals

Work in $\mathbb{F}_3$ (integers mod 3) and let $\{-1, 0, 1\}$ be the colors. We'll assign one variable to each vertex (6 vertices means $x_1, x_2, ..., x_6$).

2.7 Representing Graph Coloring with Ideals

Work in $\mathbb{F}_3$ (integers mod 3) and let $\{-1, 0, 1\}$ be the colors. We'll assign one variable to each vertex (6 vertices means $x_1, x_2, ..., x_6$).

We are subject to the constraint that $x_i^3 - x_i = 0$ for all i, which is saying that each vertex gets assigned exactly one color.

2.7 Representing Graph Coloring with Ideals

Work in $\mathbb{F}_3$ (integers mod 3) and let $\{-1, 0, 1\}$ be the colors. We'll assign one variable to each vertex (6 vertices means $x_1, x_2, ..., x_6$).

We are subject to the constraint that $x_i^3 - x_i = 0$ for all i, which is saying that each vertex gets assigned exactly one color. Additionally, for each edge (i, j), $x_i \neq x_j$.

2.7 Representing Graph Coloring with Ideals

Work in $\mathbb{F}_3$ (integers mod 3) and let $\{-1, 0, 1\}$ be the colors. We'll assign one variable to each vertex (6 vertices means $x_1, x_2, ..., x_6$).

We are subject to the constraint that $x_i^3 - x_i = 0$ for all i, which is saying that each vertex gets assigned exactly one color. Additionally, for each edge (i, j), $x_i \neq x_j$. Consider the adjacency polynomial

$$f(x_i, x_j) = x_i^2 + x_i x_j + x_j^2 - 1.$$

This is zero if and only if they are different colors.

2.8 Coloring the graph

Now we claim that solutions to

$$V\left(\left\{x_i^3 - x_i \mid \forall i = 1, ..., n\right\}, \left\{f\left(x_i, x_j\right) \mid (i, j) \in E_\Gamma\right\}\right)$$

will correspond to valid colorings of the graph.

2.9 Actually computing a coloring

Consider all the relevant polynomials:

$$x_1^3 - x_1,\ x_2^3 - x_2,\ x_3^3 - x_3,\\ x_4^3 - x_4,\ x_5^3 - x_5,\ x_6^3 - x_6$$

Now for adjacency:

$$x_1^2 + x_1x_2 + x_2^2 - 1,\ x_1^2 + x_1x_3 + x_3^2 - 1,\\ x_2^2 + x_2x_3 + x_3^2 - 1,\ x_2^2 + x_2x_4 + x_4^2 - 1,\\ x_3^2 + x_3x_5 + x_5^2 - 1,\ x_4^2 + x_4x_6 + x_6^2 - 1,\\ x_5^2 + x_5x_6 + x_6^2 - 1$$

2.9 Actually computing a coloring

Now if we include $x_1 + 1$ and $x_2 - 1$, we have the polynomials to our coloring ideal for this Γ. If we use Macaulay2 to compute the Grobner basis

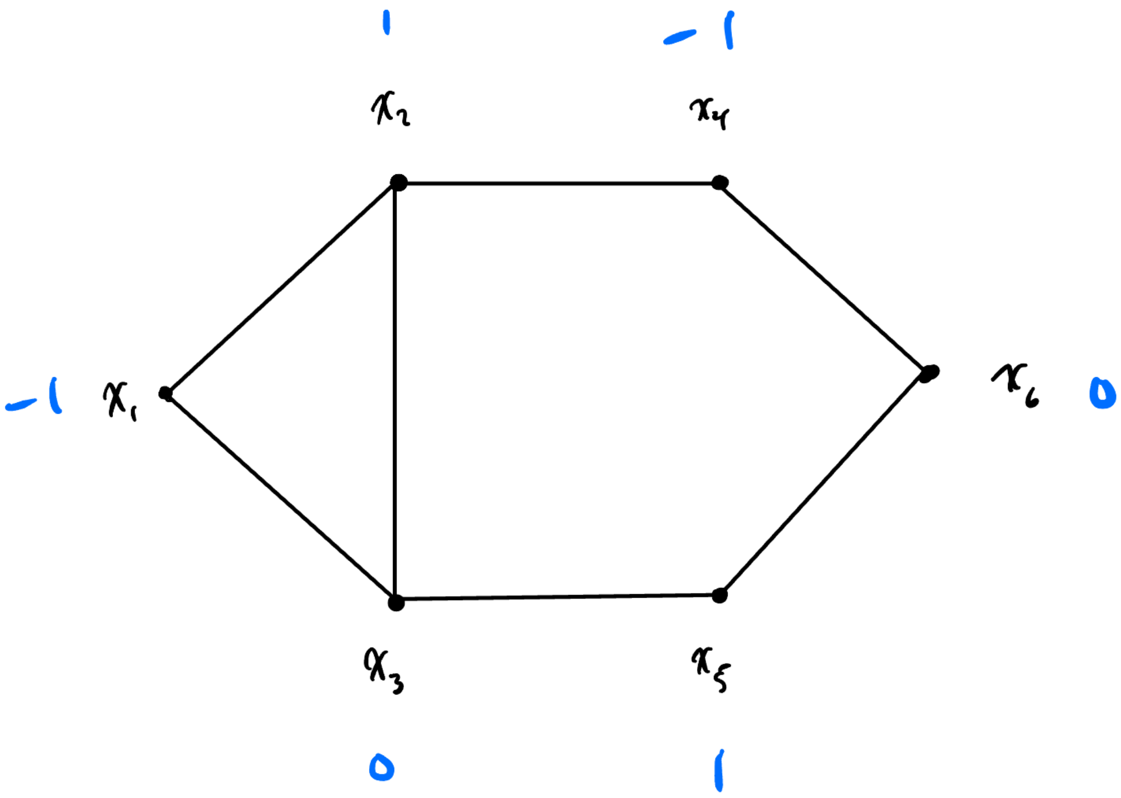
$$G(I_\Gamma) = \{x_1 + 1, x_2 - 1, x_3, x_5x_6 + x_6^2, x_4x_6 + x_6^2 - x_4 - 1, x_5^2 - 1,$$
$$x - 4x_5 - x_6^4 + x_4 + x_5 + x_6 + 1, x_4^2 + x_4, x_6^3 - x_6\}.$$

This gives us a multitude of possible assignments, one of which is

$$x_1 = -1, x_2 = 1, x_3 = 0, x_4 = -1, x_5 = 1, x_6 = 0.$$

Next slide shows that this is a valid coloring.

2.9 Actually computing a coloring



2.10 Chicken McNugget Theorem

The Oakland McDonald's sells Chicken McNuggets in sizes of 4, 6, 10, and 20. However, suppose when someone buys a 20-piece, they get lazy and only put 19. I'm wondering if I can buy 849 pieces because I love the class 21-849 so much. If I can do this, I also want to know how I can do it in the least number of boxes.

2.11 Representing this as an Ideal

Idea: Consider the ideal

$$I = \langle x_4 - z^4, x_6 - z^6, x_{10} - z^{10}, x_{19} - z^{19} \rangle \subseteq \mathbb{Q}[z, x_4, x_6, x_{10}, x_{19}].$$

2.11 Representing this as an Ideal

Idea: Consider the ideal

$$I = \langle x_4 - z^4, x_6 - z^6, x_{10} - z^{10}, x_{19} - z^{19} \rangle \subseteq \mathbb{Q}[z, x_4, x_6, x_{10}, x_{19}].$$

Note that

$$x_4^a x_6^b x_{10}^c x_{19}^d = z^{849}$$

as an element of $A(X)$ precisely when $4a + 6b + 10c + 19d = 849$.

2.12 Using the representative

We want our representative of z^{849} to satisfy two properties:

2.12 Using the representative

We want our representative of z^{849} to satisfy two properties:

1) If possible (i.e. if there is a member of the equivalence class satisfying this property) we do not want any term of our representative to be divisible by z.

2.12 Using the representative

We want our representative of z^{849} to satisfy two properties:

1) If possible (i.e. if there is a member of the equivalence class satisfying this property) we do not want any term of our representative to be divisible by z.
2) The degree of the representative is minimal.

2.12 Using the representative

We want our representative of z^{849} to satisfy two properties:

1) If possible (i.e. if there is a member of the equivalence class satisfying this property) we do not want any term of our representative to be divisible by z.
2) The degree of the representative is minimal.

It can be seen that any representative of z^{849} satisfying these conditions must be a monomial.

2.13 Forcing the representative

Computing the Grobner Basis does not require us to use the lexiographic ordering on monomials!

2.13 Forcing the representative

Computing the Grobner Basis does not require us to use the lexiographic ordering on monomials!

We only need the ordering to respect divisibility.

2.13 Forcing the representative

Computing the Grobner Basis does not require us to use the lexiographic ordering on monomials!

We only need the ordering to respect divisibility.

There exists an ordering such that

- if $\alpha_z < \beta_z$ then $\alpha \prec \beta$,
- and if $\alpha_z = \beta_z$ and $\deg \alpha < \deg \beta$, then $\alpha < \beta$.

Such an ordering suffices.

3. Feasibility

3.1 Runtime Analysis

How fast is Buchberger's algorithm?

3.1 Runtime Analysis

How fast is Buchberger's algorithm? In the worst case, doubly exponential (i.e. $O\left(d^{2^{\Omega(n)}}\right)$).

3.1 Runtime Analysis

How fast is Buchberger's algorithm? In the worst case, doubly exponential (i.e. $O\left(d^{2^{\Omega(n)}}\right)$).

Thus Grobner Bases are not particularly useful for producing theoretically fast algorithms (i.e. fast under worst-case analysis) to solve computational questions.

3.2 Can we do better?

A paper by Mayr and Meyer from 1982 shows the answer is (from an asymptotic perspective!) **NO**.

3.2 Can we do better?

A paper by Mayr and Meyer from 1982 shows the answer is (from an asymptotic perspective!) **NO**.

Reason: there exist Grobner Bases with polynomials of degree $d^{2^{\Omega(n)}}$. Just returning your result takes that long.

3.3 We can do better

But in the real world, *constant factors matter*.

3.3 We can do better

But in the real world, *constant factors matter*.

State of the art: Faugere F4/F5 algorithms

3.3 We can do better

But in the real world, *constant factors matter*.

State of the art: Faugere F4/F5 algorithms

How do they do better?

3.3 We can do better

But in the real world, *constant factors matter.*

State of the art: Faugere F4/F5 algorithms

How do they do better?

- F4 uses matrix multiplication to parallelize the computation of remainders

3.3 We can do better

But in the real world, *constant factors matter*.

State of the art: Faugere F4/F5 algorithms

How do they do better?

- F4 uses matrix multiplication to parallelize the computation of remainders
- F5 computes Grobner Bases incrementally

3.4 Avoiding the worst case

Fortunately, this worst case behavior does not happen often.

- With a small number of generators, computing the Grobner basis is typically not too slow.
- Certain constraints can also be coded into the ideal to reduce the number of eliminations to 0

3.5 Where are Grobner bases used?

Robotics. (Inverse kinematics, i.e. “how much force does the robot need to apply to end up in a certain position?”)

4. Grobner Bases 3: CAS for Theorem Proving

4.1 Recap

- Grobner Bases are a particular kind of generating set of an ideal in a polynomial ring with "nice" properties.
  - Can be seen as a generalization of Euclid's gcd algorithm and Gaussian elimination.
- Grobner Bases allow for the explicit computation of:
  - Ideal membership
  - Elimination Theory
  - Graph colorings (3-coloring, sudoku)
  - Robotics (reverse kinematics)
  - and many other applications
- Poor theoretical worst-case complexity, but in practice, highly optimized algorithms exist.

5. Macaulay2

5.1 Macaulay2

Macaulay2 is:

- a free CAS for commutative algebra and algebraic geometry
- Designed to provide algebraic algorithms with fast and efficient implementations
- designed to be useful for mathematicians, with core functionality including:
  - arithmetic on rings, modules, and matrices
  - algorithms for Grobner bases, Hilbert series, determinants, etc.

1. A gem of the modular universe, by Bruce Hunt; arXiv:alg-geom/9503018v1.
2. Local cohomology of bivariate splines, by Hal Schenck and Mike Stillman, J. Pure Appl. Algebra 117/118 (1997) 535-548.
3. A spectral sequence for splines, by Hal Schenck, Adv. in Appl. Math. 19 (1997) 183-199.
4. A family of ideals of minimal regularity and the Hilbert series of $C^r(\Delta$ hat), by Hal Schenck and Mike Stillman, Adv. in Appl. Math. 19 (1997) 169-182.
5. Homotopy types of complements of 2-arrangements in R^4, by Daniel Matei and Alexander I. Suciu; arXiv:math/9712251v2; appeared in: Topology 39 (2000), no. 1, 61-88.
6. How many squares in an infinite chess board can a knight reach in d moves?, by Mordechai Katzman, preprint.
7. Four Counterexamples in Combinatorial Algebraic Geometry, by Bernd Sturmfels, preprint, July 8, 1998.
8. Fat points, inverse systems, and piecewise polynomial functions, Anthony V. Geramita and Henry K. Schenck, J. Algebra 204 (1998) 116-128.
9. Examples of non-trivial roots of unity at ideal points of hyperbolic 3-manifolds, by Nathan M. Dunfield; arXiv:math/9801064v2; appeared in: Topology, Vol 38, No. 2, pp 457-465, 1999.
10. The Pfaffian Calabi-Yau, its Mirror, and their link to the Grassmannian G(2,7), by Einar Andreas Rodland; arXiv:math/9801092v1.

5.2 Macaulay2

1. A gem of the modular universe, by Bruce Hunt; arXiv:alg-geom/9503018v1.
2. Local cohomology of bivariate splines, by Hal Schenck and Mike Stillman, J. Pure Appl. Algebra 117/118 (1997) 535-548.
3. A spectral sequence for splines, by Hal Schenck, Adv. in Appl. Math. 19 (1997) 183-199.
4. A family of ideals of minimal regularity and the Hilbert series of C^r(Δ hat), by Hal Schenck and Mike Stillman, Adv. in Appl. Math. 19 (1997) 169-182.
5. Homotopy types of complements of 2-arrangements in R^4, by Daniel Matei and Alexander I. Suciu; arXiv:math/9712251v2; appeared in: Topology 39 (2000), no. 1, 61-88.
6. How many squares in an infinite chess board can a knight reach in d moves?, by Mordechai Katzman, preprint.
7. Four Counterexamples in Combinatorial Algebraic Geometry, by Bernd Sturmfels, preprint, July 8, 1998.
8. Fat points, inverse systems, and piecewise polynomial functions, Anthony V. Geramita and Henry K. Schenck, J. Algebra 204 (1998) 116-128.
9. Examples of non-trivial roots of unity at ideal points of hyperbolic 3-manifolds, by Nathan M. Dunfield; arXiv:math/9801064v2; appeared in: Topology, Vol 38, No. 2, pp 457-465, 1999.
10. The Pfaffian Calabi-Yau, its Mirror, and their link to the Grassmannian G(2,7), by Einar Andreas Rodland; arXiv:math/9801092v1.
11. Varieties of sums of powers, by Kristian Ranestad and Frank-Olaf Schreyer, J. Reine Angew. Math. 525, 147-181 (2000), arXiv:math/9801110.
12. The Projective Geometry of the Gale Transform, by David Eisenbud and Sorin Popescu, July 23, 1998, arXiv:math/9807127.
13. Computing Global Extension Modules for Coherent Sheaves on a Projective Scheme, by Gregory G. Smith; arXiv:math/9807170v1; appeared in: Journal of Symbolic Computation 29 (2000) 729-746.
14. On the divisor class group of double solids, by Stephan Endrass; arXiv:math/9809158v2.
15. Cohomology rings and nilpotent quotients of real and complex arrangements, by Daniel Matei and Alexander I. Suciu; arXiv:math/9812087v3; appeared in: Arrangements-Tokyo 1998, 185-215, Advanced Studies in Pure Mathematics, vol. 27, Kinokuniya, Tokyo, 2000.
16. Moduli of (1,7)-polarized abelian surfaces via syzygies, by Nicolae Manolache and Frank-Olaf Schreyer, Math. Nachr. 226 (2001) 177-203, arXiv:math/9812121.
17. Inbreeding depression in a zygotic algebra, by J. A. Vargas and R. F. del Castillo, Communications in Algebra, 27 (1999) 4425-4432.
18. Ideals with a given Hilbert function, by Graham Evans, preprint, 1999.
19. The moduli space of (1,11)-polarized abelian surfaces is unirational, by Mark Gross and Sorin Popescu; arXiv:math/9902017v1.
20. Standard bases with respect to the Newton filtration, by Stephan Endrass; arXiv:math/9904071v1.
21. Weighted Tango Bundles And Their Moduli Spaces, by Paolo Cascini; arXiv:math/9904091v4; appeared in: Forum Math. 13, No.2, 251-260 (2001).
22. Real Schubert Calculus: Polynomial systems and a conjecture of Shapiro and Shapiro, by Frank Sottile; arXiv:math/9904138v1; appeared in: Experimental Mathematics, 9, Number 2, (2000), pp. 161-182.
23. Enriques Surfaces and other Non-Pfaffian Subcanonical Subschemes of Codimension 3, by David Eisenbud, Sorin Popescu, and Charles Walter; arXiv:math/9906171v1.
24. Bezout's theorem and Cohen-Macaulay modules, by J. Migliore, U. Nagel, and C. Peterson; arXiv:math/9907074v1.
25. Determinantal hypersurfaces, by A. Beauville; arXiv:math/9910030v2.
26. A smooth space of tetrahedra, by Eric Babson, Paul E. Gunnells, and Richard Scott; arXiv:math/9910049v2.
27. Constructing irreducible representations of finitely presented algebras, by Edward S. Letzter; arXiv:math/9910132v4.
28. Quartic 3-fold: Pfaffians, instantons and half-canonical curves, by A. Iliev and D. Markushevich; arXiv:math/9910133v2.
29. Geometry and algebra of prime Fano 3-folds of genus 12, by Frank-Olaf Schreyer, Compos. Math. 127 (2001) 297-319, arXiv:math/9911044.
30. Computation of maximal reachability submodules, by Wiland Schmale; arXiv:math/9911211v1.
31. Hyperplane Arrangement Cohomology and Monomials in the Exterior Algebra, by David Eisenbud, Sorin Popescu, and Sergey Yuzvinsky; arXiv:math/9912212v3.
32. Translated tori in the characteristic varieties of complex hyperplane arrangements, by Alexander I. Suciu; arXiv:math/9912227v3; appeared in: Topology and Appl. 118 (2002), 209-223.
33. Hardy-Weinberg theory for tetraploidy with mixed mating, by J. A. Vargas, Advances in Applied Mathematics 24 (2000) 369-383.
34. Non-general type surfaces in $\mathbf{P}^4$: Some remarks on bounds and constructions, by Wolfram Decker and Frank-Olaf Schreyer. J. Symb. Comput. 29 (2000) 545-582.
35. A rank two vector bundle associated to a three arrangement, and its Chern polynomial, Henry K. Schenck, Adv. Math. 149 (2000) 214-229.
36. Cellular binomial ideals. Primary decomposition of binomial ideals, by Ignacio Ojeda Martínez de Castilla; Ramón Piedra-Sánchez, Journal of Symbolic Computation, 30 (2000) 383-400, MR 2001g:13058.
37. Jacobian ideals of trilinear forms: an application of 1-genericity, by A. Guerrieri and I. Swanson, Journal of Algebra, **226** (2000) 410-435.
38. Possible resolutions for a given Hilbert function, by E. Graham Evans, Jr., and Benjamin P. Richert, preprint, 2000.
39. Polynomial and rational solutions of holonomic systems, by T. Oaku, N. Takayama, and H. Tsai; arXiv:math/0001064v1.
40. Calabi-Yau Threefolds and Moduli of Abelian Surfaces I, by Mark Gross and Sorin Popescu; arXiv:math/0001089v1.
41. Cohomology on Toric Varieties and Local Cohomology with Monomial Supports, by David Eisenbud, Mircea Mustata, and Mike Stillman; arXiv:math/0001159v1; appeared in: J. Symbolic Comput. 29 (2000), no. 4-5, 583-600.
42. Deformation of singular lagrangian subvarieties, by Duco van Straten and Christian Sevenheck, arXiv:math/0002083.
43. Computer Algebra and Algebraic Geometry - Achievements and Perspectives, by Gert-Martin Greuel; arXiv:math/0002247v1; appeared in: Centre for Computer Algebra, University of Kaiserslautern, Reports On Computer Algebra, No. 29 (2000).
44. *Macaulay2* and the geometry of schemes, by Gregory G. Smith and Bernd Sturmfels; arXiv:math/0003033v1; appeared in: In Computations in algebraic geometry with Macaulay 2, pp 55-70, Springer-Verlag, 2001.
45. Relations in the cohomology ring of the moduli space of rank 2 Higgs bundles, by Tamas Hausel and Michael Thaddeus; arXiv:math/0003094v2.
46. Sheaf Cohomolog and Free Resolutions over Exterior Algebras, by David Eisenbud and Frank-Olaf Schreyer; arXiv:math/0005055v1.
47. Some conjectures about the Hilbert series of generic ideals in the exterior algebra, by Jan Snellman and Guillermo Moreno-Socias; arXiv:math/0007089v3; appeared in: Homology, homotopy and applications, volume 4, no 2, 2002, pages 409-426.
48. Computing homomorphisms between holonomic D-modules, by Harrison Tsai and Uli Walther; arXiv:math/0007139v1.
49. An excursion from enumerative geometry to solving systems of polynomial equations with *Macaulay2*, by Frank Sottile; arXiv:math/0007142v2; appeared in: Computations in Algebraic Geometry with Macaulay 2, edited by D. Eisenbud, D. Grayson, M. Stillman, and B. Sturmfels. ACM 8, Springer-Verlag, 2001. pp. 101-129.
50. The Center Variety of Polynomial Differential Systems, by Abdul Salam Jarrah, Reinhard Laubenbacher, and Valery Romanovski; arXiv:math/0009061v2.

5.3 Macaulay2

There's a few (3000+) papers that use M2.

5.4 Functionality

```
i1 : R = QQ[a..d]

o1 = R

o1 : PolynomialRing
```

```
i2 : I = ideal(a^3-b^2*c, b*c^2-c*d^2, c^3)

                3    2      2      2   3
o2 = ideal (a  - b c, b*c  - c*d , c )

o2 : Ideal of R
```

```
i3 : G = gens gb I

o3 = | c3 bc2-cd2 a3-b2c c2d2 cd4 |

             1      5
o3 : Matrix R  <-- R
```

Docs | Grobner Example

5.5 Gröbner Base Computation

Buchberger

1) For current basis elements f, g, compute the S-polynomial $S(f, g)$
2) Reduce $S(f, g)$ using polynomial division w.r.t current basis
3) If nonzero remainder, add it to the basis and update the choice of f, g
4) repeat until all S-polys reduce to 0.

F4 (M2′s algorithm)

1) Select and compute a set of basis pairs $\{(f_i, g_i)\}_I$ whose S-polynomials share a common (minimal) degree.
2) For each (f_i, g_i), compute $t_f \cdot \mathrm{LM}(f_i) = t_g \cdot \mathrm{LM}(g_i) = \mathrm{lcm}(\mathrm{LM}(f_i), \mathrm{LM}(g_i))$ and store $t_f \cdot f$ and $t_g \cdot g$ (for quickly computing S-polynomials)
3) Arrange these S-polys as the rows of a matrix (columns corresponding to ordered monomials) and perform row reduction to get row-echelon form
4) Nonzero rows correspond to new basis elements. Add to the grobner basis.
5) Repeat until no new elements are formed.

6. Interactive Theorem Proving

6.1 Interactive Theorem Proving

Interactive theorem provers are software that allow users to interactively work with the computer to construct *formal* mathematical proofs.

- Includes Lean, Rocq, Isabelle/HOL, etc.
- Guaranteed correctness provided by the language kernel
  - (Strong reward signal for AI applications: AlphaProof, STP, etc.)
- Recent successes in collaborative formalization:
  - Characterization of Equational Magmas [Tao] (*Heavily* relied on Grobner Bases!)
  - Polynomial Friedman-Ruzsa Conjecture
  - Kepler Conjecture [Hales]
  - Perfectoid Spaces [Buzzard]
  - […]

6.2 Theorem Proving in Lean4

Quick Demo!

6.3 Dependent Type Theory

- By Curry-Howard, proofs of mathematical propositions are isomorphic to types (at a high level…)
- Lean is founded on dependent type theory, where propositions are encoded as types, and a proof of a proposition is simply an inhabitant of the corresponding type.
- Dependent types allow the encoding of complex mathematical statements with embedded invariants.
  - Lean's type theory includes a countable hierarchy of universes, avoiding paradoxes and inductive types, quotient types, etc.

6.4 Computability

- Lean is generally constructive, but not inherently so.
  - Classical logic allows for greater expressiveness at the cost of noncomputability at the hands of AoC, etc.

For example:

```
structure Real where ofCauchy ::
  /-- The underlying Cauchy completion -/
  cauchy : CauSeq.Completion.Cauchy (abs : ℚ → ℚ)
```

```
/-- A finite field with `p ^ n` elements.
Every field with the same cardinality is (non-canonically)
isomorphic to this field. -/
def GaloisField := SplittingField (X ^ p ^ n - X : (ZMod p)[X])
```

7. LeanM2

7.1 LeanM2

- Lean has type-theoretically perfect verification of proofs, but small support for computation tactics.
- Macaulay2 provides extremely efficient, locally hosted, and extensible computation tools

Therefore, we propose *LeanM2*, which aims to upgrade Macaulay2 to form formal proofs for use in Lean.

7.2 Implementation

1) Convert current Lean hypotheses + goals into M2 command
2) Receive M2 response
3) Convert response into proof certificate and Lean syntax

7.3 Implementation

1) Convert current Lean hypotheses + goals into M2 command
   a) Parse proof state in `TacticM` monad and extract relevant structures
   b) Synthesize metavariables and types into computable structures
   c) Combine new structures into M2 command
2) Receive M2 response
   a) parse messy M2 response into proof witness
3) Convert response into proof certificate and Lean syntax
   a) Build Mathlib API for GB proof certification
   b) Create parser for M2 outputs into syntactic, computable structures
   c) Reinstance structures as `Lean.Expr` and parse into valid proof
   d) Create and apply tactics to automatically use certificates to close goals.

7.4 Implementation

- `M2Type` instances the semantic (often noncomputable) meaning of Lean code to the corresponding syntactic (computable) type.
  - Explicitly constructs partial isomorphisms between the types, with formal proofs of invertibility
  - Encapsulates `Repr`, `UnRepr` for easy conversion to/from M2.
- Syntactic representations include:
  - $\mathbb{R}$: Cauchy Completion $\mapsto$ rationals + trancendental fns
  - $\mathbb{C}$: See above.
  - $GF(p^n)$: Splitting field (AoC) $\mapsto$ Conway w/ algebraic equivalence pf.
  - […]

Polynomial rings (syntactically: `_root_.Expr`) are represented abstractly w/ base ring, atoms, and lifting fn to output ring.

7.5 Implementation

Command:

- ```
  R=QQ[x0, x1]
  f=((x0)^2 + (x1)^2)
  I=ideal(x0,x1)
  G=gb(I,ChangeMatrix=>true)
  f % G
  (getChangeMatrix G)*(f// groebnerBasis I)
  ```

Response:

- ```
  i1 : R=QQ[x0, x1]
  o1 = R
  o1 : PolynomialRing

  i2 : f=((x0)^2 + (x1)^2)
         2     2
  o2 = x0  + x1
  o2 : R

  i3 : I=ideal(x0,x1)
  o3 = ideal (x0, x1)
  o3 : Ideal of R

  i4 : G=gb(I,ChangeMatrix=>true)
  o4 = GroebnerBasis[status: done; S-pairs encountered up to
  degree 0]
  o4 : GroebnerBasis

  i5 : f % G
  o5 = 0
  o5 : R

  i6 : (getChangeMatrix G)*(f// groebnerBasis I)
  o6 = {1} | x0 |
       {1} | x1 |
               2       1
  o6 : Matrix R  <--- R
  ```

7.6 Examples and Demo

Demo time!

See: https://github.com/riyazahuja/lean-m2

7.7 Future Work

- June 2025
  - Implement Grobner Basis API into Mathlib
  - add support for Exterior algebras, Weyl algebras, and other noncomputables
  - Stabilize UI and extend beyond ideal membership (elimination theory, etc.)
- Aug 2025
  - Generalize proof certification to standard M2 library (once type synthesis is done and API is built, the rest is easy!)
- ???
  - ITP