A Study of Gröbner Bases and their Application in the Verification of Galois Field Arithmetic Circuits

Hastin Kapoor

A an undergrdaute honors thesis
submitted in partial fulfillment of the
requirements for the degree of

Bachelor of Science *(with Distinction)*

University of Washington

2024

Faculty sponsor: Timothy Duff

Program Authorized to Offer Degree:

BS in Mathematics

University of Washington

Abstract

A Study of Gröbner Bases and their Application in the Verification of Galois Field Arithmetic Circuits

Hastin Kapoor

Supervisor: Dr. Timothy Duff

Arithmetic in finite fields of 2^k elements is extremely important in cryptography, and thus it is just as important to be able to verify that these computations are being performed correctly by a given circuit. This paper aims to introduce the reader to Gröbner bases and their applications in the formal verification of Galois field arithmetic circuits.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge my supervisor, Timothy Duff, who made this all possible. I could not have done this without his help.

I would also like to thank Minseon Shin, who helped me find a topic for my thesis as well as connected me with people in the math department at the start of this project.

Finally, I would like to thank all of my professors over the past four years, as their guidance has led me to where I am today.

DEDICATION

To my dad, who is always full of helpful ideas

To my mom, who made sure I called her at least once a week to let her know how I'm doing

To my friends, who were willing to sit through my ramblings on arbitrary topics

And last but not least, to my cats, who I can't wait to see again

Chapter 1

INTRODUCTION

1.1 Gröbner Bases

In this paper, we use Gröbner bases to solve the problem of ideal membership testing. The problem is defined as such:

Let $k[x_1, \ldots, x_n]$ be a polynomial ring. Given $f \in k[x_1, \ldots, x_n]$ and an ideal $I = \langle f_1, \ldots, f_n \rangle$, determine if $f \in I$ [1].

In other words, given a polynomial in the variables $x_1, \ldots, x_n$, is it possible to find some combination of $f_1, \ldots, f_n$, with coefficients in $k[x_1, \ldots, x_n]$, whose sum is f? A simplified example, using the familiar ring $\mathbb{Z}$ instead of $k[x_1, \ldots, x_n]$, will show the intuition behind the construction and use of Gröbner bases:

Suppose we want to test if 18 is an integer combination of 10 and 6. A greedy algorithm might first divide by 10 and then by 6, seeing that 18 = 10 + 8, so we proceed to divide the remainder of 8 by 6 and obtain 8 = 6 + 2, which cannot be further reduced by this algorithm, thus concluding that 18 is not in the ideal of 10 and 6. Had we divided by 6 first, we would have found 18 = 3 * 6, and concluded that 18 is in the ideal. How we can solve this inconsistency, where the result is dependent on the order of division, is by finding a "better" basis for the ideal generated by 10 and 6. By Bézout's identity, we know that GCD(10, 6) = 2 is some integer combination of 10 and 6, and so we can add 2 to our list of generators. Finding the GCD of each other pair of elements in $\{10, 6, 2\}$ results in no new values, and thus we find our "better" basis to be $\{10, 6, 2\}$. Now when we perform this greedy algorithm, instead of stopping at 8 = 6 + 2, we now continue 2 = 2 + 0, and see that 18 leaves no remainder, i.e. it is an integer combination of the generators of the ideal, and thus the algorithm returns that 18 is in the ideal as we had hoped.

The construction of Gröbner bases follows this same process, comparing pairs of generators of the ideal to find new generators until no new generators are found, and at that point we have a "better" basis that will give consistent results when dividing, independent of the order in which we attempt to divide, thus telling us whether or not the polynomial in question is in the ideal.

1.2 Formal Verification

Formal verification offers a rigorous methodology to ascertain that a system behaves as intended under all possible scenarios. Formally verifying a system exhaustively analyzes all possible states and behaviors, leaving no room for ambiguity or oversight. Attempting to test each possible case of a given system, such as a 128 bit multiplier having 2^{256} possible inputs, is often inefficient and impractical, so instead one can use properties of the system to optimize the verification process. One can expect the input and output of a system to be related in some manner, and these relations can be used in the verification of the system rather than being forced to test each case individually. In Jinpeng Lv, Priyank Kalla, and Florian Enescu's "Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits" [3], the central idea is that Gröbner bases can be used to verify arithmetic circuits over the Galois field $\mathbb{F}_{2^k}$ with 2^k elements, since we can view each gate as a polynomial over its inputs and the specification as a polynomial relating all of the inputs of the entire system to the outputs. The question then becomes whether the specification polynomial is equivalent to the composition of the polynomials corresponding to each gate in the implementation.

Chapter 2

BACKGROUND KNOWLEDGE

Before we discuss the specific details on how Gröbner bases are defined and the motivation behind doing so, we first start with a few definitions and properties of polynomials.

Definition 2.1. *Let k be a field, and let $f_1, \ldots, f_s$ be polynomials in $k[x_1, \ldots, x_n]$. Then we set*

$$V(f_1, \ldots, f_s) = \{(a_1, \ldots, a_n) \in k^n : f_i(a_1, \ldots, a_n) = 0 \ \forall \ 1 \leq i \leq s\}.$$

*We call $V(f_1, \ldots, f_s)$ the **affine variety** defined by $f_1, \ldots, f_s$. [1]*

This definition states that the affine variety of a set of polynomials is the set of all solutions to the system of equations that sets each polynomial to zero. For example, consider $\mathbb{R}^2$ and the affine variety $V(x-y, x^2+y^2-1)$. Setting each polynomial to zero, we see $x^2+y^2-1=0$ defines the unit circle in $\mathbb{R}^2$ and $x-y=0$ defines a line through the origin. The intersection of these sets is the set of points that are solutions to each equation, and thus the affine variety $V(x-y, x^2+y^2-1)$ is the two points $\{(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}), (-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})\}$. See Figure 2.1.

Next, we define how one can generate an ideal from an affine variety. Recall that an ideal is defined as follows.

Definition 2.2. *A subset $I \subset k[x_1, \ldots, x_n]$ 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 $hf \in I$. (See [1, p. 30].)*

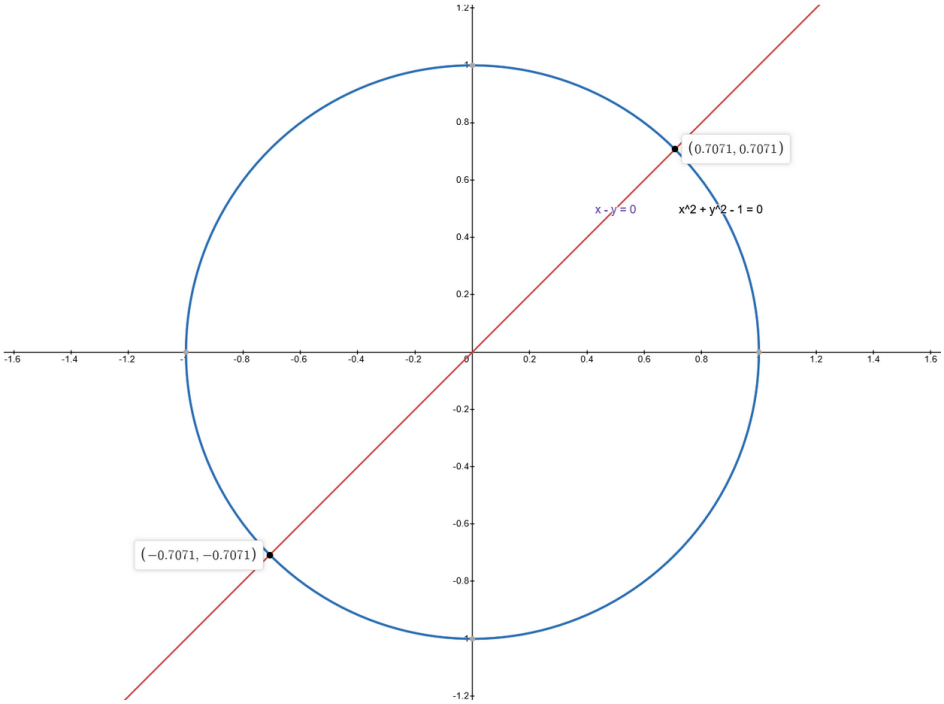

Figure 2.1: The affine variety $V(x-y, x^2+y^2-1)$ consists of the two labeled points.

A standard construction in algebra is the ideal generated by a finite set of polynomials $\{f_1, \ldots, f_s\} \subset k[x_1, \ldots, x_n]$, which we denote by

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

The proof that this is an ideal is straightforward.

Note that for an ideal $\langle f_1, \ldots, f_s \rangle$, since every polynomial in the ideal is a polynomial combination of basis elements, if all basis polynomials are zero then so is every element of the ideal, thus $V(f_1, \ldots, f_s) = V(\langle f_1, \ldots, f_s \rangle)$.

Given an affine variety, the way one constructs an ideal is slightly different.

Definition 2.3. *Let $V \subset k^n$ be an affine variety. Then we set (cf. [1, p. 32])*

$$I(V) = \{ f \in k[x_1, \ldots, x_n] : f(a_1, \ldots, a_n) = 0 \; \forall \; (a_1, \ldots, a_n) \in V \}.$$

Following [1, p.32–33], we show that the set $I(V)$ is an ideal of $k[x_1, \ldots, x_n]$:

1. $0 \in I(V)$, since 0 vanishes on the entirety of k^n and thus on every point in V.

2. Let $f, g \in I(V)$ and $(a_1, \ldots, a_n)$ an arbitrary point in V. $f(a_1, \ldots, a_n) + g(a_1, \ldots, a_n) = 0 + 0 = 0$, thus $f + g \in I(V)$.

3. Let $f \in I(V)$ and $h \in k[x_1, \ldots, x_n]$. Then

$$h(a_1, \ldots, a_n) f(a_1, \ldots, a_n) = h(a_1, \ldots, a_n) \cdot 0 = 0,$$

   thus $hf \in I(V)$.

Thus $I(V)$ is an ideal. Continuing with the previous example, we have the ideal generated by the variety $V(x - y, x^2 + y^2 - 1)$ is the set of all polynomials which have zeros at $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. From this alone it is not clear what the generators of such an ideal might be, but we will present a useful relationship between the initial ideal and the ideal of the variety of the initial ideal in the case where k a finite field.

It is clear that $\langle f_1, \ldots, f_s \rangle \subset I(V(f_1, \ldots, f_s))$ since by definition the polynomials that define the variety vanish on the variety, but the reverse is not always true. In the field $\mathbb{F}_q$, polynomials of the form $x^q - x$ are zero at all points in the field, so the ideal of any nonempty variety will contain $x_i^q - x_i$ for all $1 \leq i \leq n$ since these polynomials vanish everywhere, and thus vanish everywhere within the variety as well [3]. These vanishing polynomials allow us to relate the two ideals $\langle f_1, \ldots, f_s \rangle$ and $\subset I(V(f_1, \ldots, f_s))$, and are used in the strong Nullstellensatz theorem as defined in [3], which relates a variety and its ideal. Note that this is not the same as the strong Nullstellensatz over the complex numbers defined in [1].

Theorem 2.4. ***(Strong Nullstellensatz over $\mathbb{F}_q$, see eg. [3].)*** *For any Galois field $\mathbb{F}_q$, let $J \subset \mathbb{F}_q[x_1, \ldots, x_d]$ be an ideal, and let $J_0 = \langle x_1^q - x_1, \ldots, x_d^q - x_d \rangle$ be the ideal of all vanishing polynomials. Let $V_{\mathbb{F}_q}(J)$ denote the variety of J over $\mathbb{F}_q$. Then,*

$$I(V_{\mathbb{F}_q}(J)) = J + J_0 = J + \langle x_1^q - x_1, \ldots, x_d^q - x_d \rangle$$

This simplifies the computation of the ideal of the variety, since this result tells us exactly what the generators of the ideal of the variety should be: take the generators of the initial ideal and append the vanishing polynomials for each variable.

The next important definition is monomial ordering. In [1], the authors define monomial ordering as such:

Definition 2.5. *A **monomial ordering** $>$ on $k[x_1, \ldots, x_n]$ is any relation $>$ on $\mathbb{Z}^n_{\geq 0}$, or equivalently, any relation on the set of monomials $x^\alpha, \alpha \in \mathbb{Z}^n_{\geq 0}$, satisfying:*

1. *$>$ is a total (or linear) ordering on $\mathbb{Z}^n_{\geq 0}$*
2. *If $\alpha > \beta$ and $\gamma \in \mathbb{Z}\geq 0^n$, then $\alpha + \gamma > \beta + \gamma$*
3. *$>$ is a well-ordering on $\mathbb{Z}^n_{\geq 0}$. This means that every nonempty subset of $\mathbb{Z}^n_{\geq 0}$ has a smallest element under $>$. (See [1, p. 55].)*

In other words, a monomial ordering gives us a way to compare any two monomials. This is used in defining the leading term of a polynomial, for example we could choose a monomial order where $x^2 > xy$ so that $x^2 + xy$ is expressed as such, or we could choose one that has $xy > x^2$ so that it is expressed as $xy + x^2$. The two poynomials are equivalent, but their leading terms differ: $x^2 + xy$ has a leading term of $LT(x^2 + xy) = x^2$ whereas $xy + x^2$ has the leading term $LT(xy + x^2) = xy$. Defining an appropriate monomial order will be important for Gröbner bases, as they are reliant on the leading terms of polynomials. An example of a monomial ordering is lexicographic order.

Definition 2.6. ***(Lexicographic Order)*** *Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$. We say $\alpha >_{lex} \beta$ if, in the vector difference $\alpha - \beta \in \mathbb{Z}^n$, the leftmost nonzero entry is positive. We write $x^\alpha >_{lex} x^\beta$ if $\alpha >_{lex} \beta$.*

Intuitively, lexicographic order is "dictionary order". For $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots$ and $x^\beta = x_1^{\beta_1} x_2^{\beta_2} \ldots$ two monomials, and order $x_1 > \cdots > x_n$, one compares the largest x_i for which

the power differs between x^α and x^β. If, for example, x^α has the greater power of x_i, then we say $x^\alpha > x^\beta$. This is similar to how one determines which word comes first between "variety" and "variable": the first letter that differs between the two is the "e" in "variety" versus the "a" in "variable". An intuition as to why monomial orders matter in the context of [3] can be seen in an example of polynomial long division. When doing polynomial long division, one compares the leading terms of the divisor and dividend to determine a part of the quotient, which is why it is important to define what the leading monomials are by setting a monomial order satisfying Definition 2.5. Should we choose a "bad" monomial order, our long division might require more steps than if we had chosen a "good" monomial order. In fact, if we don't have a Gröbner basis, the monomial order can change the resulting remainder when we try to divide by polynomials of more than one variable.

The last definitions and related theorems necessary before defining Gröbner bases are about monomial ideals. We will use the following terminology.

Definition 2.7. *Let $f = \sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $k[x_1, \ldots, x_n]$ and let $>$ be a monomial order. As in [1], we define:*

1. *The **multidegree** of f is*

$$multideg(f) = \max(\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0)$$

(the maximum is taken with respect to $>$.)

2. *The **leading coefficient** of f is*

$$LC(f) = a_{multideg(f)} \in k.$$

3. *The **leading monomial** of f is*

$$LM(f) = x^{multideg(f)}$$

(with coefficient 1.)

4. The ***leading term*** *of f is*

$$LT(f) = LC(f) \cdot LM(f).$$

For example, consider the polynomial $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2$. Using the lexicographical order with $x > y > z$, we have:

$$multideg(f) = (3, 0, 0)$$

$$LC(f) = -5$$

$$LM(f) = x^3$$

$$LT(f) = -5x^3$$

Now we define monomial ideals.

Definition 2.8. *([1, p. 70]) An ideal $I \subset k[x_1, \ldots, x_n]$ is a* ***monomial ideal*** *if there is a subset $A \subset \mathbb{Z}^n_{\geq 0}$ (possibly infinite) such that I consists of all polynomials which are finite sums of the form $\sum_{\alpha \in A} h_\alpha x^\alpha$, where $h_\alpha \in k[x_1, \ldots, x_n]$. In this case, we write $I = \langle x^\alpha : \alpha \in A \rangle$.*

In other words a monomial ideal is an ideal which can be generated by a basis of monomials, but this basis is not necessarily finite. However, Dickson's lemma shows that a finite basis exists for any monomial ideal.

Theorem 2.9. ***(Dickson's Lemma [1, p. 71])*** *Let $I = \langle x^\alpha : \alpha \in A \rangle \subset k[x_1, \ldots, x_n]$ be a monomial ideal. Then I can be written in the form $I = \langle x^{\alpha(1)}, \ldots, x^{\alpha(s)} \rangle$, where $\alpha(1), \ldots, \alpha(s) \in A$. In particular, I has a finite basis.*

Dickson's lemma can be used to prove the Hilbert basis theorem, which states that not only monomial ideals but all ideals in $k[x_1, \ldots, x_n]$ have a finite generating set, however these proofs are omitted for brevity's sake. These proofs, along with proofs of all of the following theorems, can be found in the second chapter of [1].

Here is the formal statement of the Hilbert basis theorem:

Theorem 2.10. ***(Hilbert Basis Theorem)*** *Every ideal $I \subset k[x_1, \ldots, x_n]$ has a finite generating set. That is, $I = \langle g_1, \ldots, g_t \rangle$ for some $g_1, \ldots, g_t \in I$.*

This theorem is useful in some of the proofs regarding Gröbner bases, since it guarantees that a finite basis exists, from which we can build a Gröbner basis.

Now, that we've defined monomial ideals, we can define Gröbner bases:

Definition 2.11. *Fix a monomial order. A finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal I is said to be a* ***Gröbner basis*** *if*

$$\langle LT(g_1), \ldots LT(g_t) \rangle = \langle LT(I) \rangle$$

From this definition alone it is not clear how Gröbner bases are useful. However, the next proposition tells us an important property of Gröbner bases.

Proposition 2.12. *Let $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis for an ideal $I \subset k[x_1, \ldots, x_n]$ and let $f \in k[x_1, \ldots, x_n]$. Then there is a unique $r \in k[x_1, \ldots, x_n]$ with the following two properties:*

1. *No term of r is divisible by any of $LT(g_1), \ldots, LT(g_t)$.*

2. *There is a $g \in I$ such that $f = g + r$.*

In particular, r is the remainder on division of f by G no matter how the elements of G are listed when using the division algorithm [1].

This proposition removes the ambiguity caused when dividing using different listed orders of the basis of an ideal.

In order to prove this proposition, we require one additional lemma about monomial ideals:

Lemma 2.13. *Let $I = \langle x^\alpha : \alpha \in A \rangle$ be a monomial ideal. Then a monomial x^β lies in I if and only if x^β is divisible by x^α for some $\alpha \in A$ [1].*

With this lemma, we go on to prove Proposition 2.12:

Proof of Proposition 2.12. The division algorithm gives $f = a_1g_1 + \cdots + a_tg_t + r$, where r satisfies 1. We can also satisfy 2 by setting $g = a_1g_1 + \cdots + a_tg_t \in I$. This proves the existence of r.

To prove uniqueness, suppose that $f = g + r = g' + r'$ satisfy 1 and 2. Then $r - r' = g' - g \in I$, so that if $r \neq r'$, then $LT(r - r') \in \langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_t)\rangle$. By the previous lemma, it follows that $LT(r - r')$ is divisible by some $LT(g_i)$. This is impossible since no term of r, r' is divisible by one of $LT(g_1), \ldots, LT(g_t)$. Thus $r - r'$ must be zero, and uniqueness is proved.

The final part of the proposition follows from the uniqueness of r [1]. □

Proposition 2.12 also leads directly into the following corollary, which states that the division by a Gröbner basis results in a remainder of zero if and only if the polynomial is in the ideal, which is a property we had hoped for in a basis.

Corollary 2.14. *Let $G = g_1, \ldots, g_t$ be a Gröbner basis for an ideal $I \subset k[x_1, \ldots, x_n]$ and let $f \in k[x_1, \ldots, x_n]$. Then $f \in I$ if and only if the remainder on division of f by G is zero [1].*

This corollary follows directly from Proposition 2.12, since if the remainder is zero then $f \in I$ and if $f \in I$ then $f = f + 0$ satisfies the conditions of the proposition and thus zero is the remainder of f on division by G. Therefore we can use Gröbner bases to test if f is in an ideal, without ambiguity since listed order does not matter for a Gröbner basis.

The next corollary tells us which ideals we can guarantee the existence of a Gröbner basis for:

Corollary 2.15. *Fix a monomial order. Then every ideal $I \subset k[x_1, \ldots, x_n]$ other than $\{0\}$ has a Gröbner basis. Furthermore, any Gröbner basis for an ideal I is a basis of I [1].*

Note that the leading term of 0 is undefined, which is why we assume $\{0\} \subsetneq I$.

Conveniently then, for all ideals other than $\{0\}$, a Gröbner basis exists.

In order to produce a Gröbner basis for a given ideal, one can use Buchberger's algorithm. Connecting back to the first example in the introduction, Buchberger's algorithm compares generators of the ideal, adding more generators until the conditions for the set of generators being a Gröbner basis are met. Before we can define the algorithm, we must introduce a few definitions and lemmas.

Definition 2.16. *Let $f, g \in k[x_1, \ldots, x_n]$ be nonzero polynomials.*

1. *If multideg(f) $= \alpha$ and multideg(g) $= \beta$, then let $\gamma = (\gamma_1, \ldots, \gamma_n)$, where $\gamma_i = max(\alpha_i, \beta_i)$ for each i. We call x^γ the **least common multiple** of LM(f) and LM(g), written $x^\gamma = LCM(LM(f), LM(g))$.*

2. *The **S-polynomial** of f and g is the combination*

$$S(f, g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g.$$

These S-Polynomials will be used in defining our equivalent of the GCD as it is used in our first example, as we compare two generators by using their S-polynomials.

The next theorem, known as Buchberger's criterion, proves our stopping condition for Buchberger's algorithm. In our first example, we stopped when no pair of generators gave us a new GCD. Here, Buchberger's criterion tells us that when no new generators are produced, we have a Gröbner basis, and also that when we have a Gröbner basis, we cannot use this algorithm to produce any more generators, although this second direction is of less importance to us.

Theorem 2.17. ***(Buchberger's Criterion)*** *Let I be a polynomial ideal. Then a basis $G = \{g_1, \ldots, g_t\}$ for I is a Gröbner basis for I if and only if for all pairs $i \neq j$, the remainder on division of $S(g_i, g_j)$ by G (listed in some order) is zero.*

This then leads us into Buchberger's algorithm, which, given any basis of an ideal I, will produce a Gröbner basis for I.

Theorem 2.18. ***(Buchberger's Algorithm)*** *Let $I = \langle f_1, \ldots, f_s \rangle \neq 0$ be a polynomial ideal. Then a Gröbner basis for I can be constructed in a finite number of steps by Algorithm 1.*

Data: $\{f_1, \ldots, f_s\} \subset k[x_1, \ldots, x_n]$
Result: G, a Gröbner basis for $\langle f_1, \ldots, f_s \rangle$
$G \leftarrow \{f_1, \ldots, f_s\}$;
$S \leftarrow \{(f_i, f_j) \mid 1 \leq i < j \leq n\}$; `/* Initial set of S-pairs */`
while $S \neq \emptyset$ **do**
  pick $(f, g) \in S$;
  $r \leftarrow \text{remainder}(S(f, g), G)$;
  if $r \neq 0$ **then**
    $G \leftarrow G \cup \{r\}$;
    $S \leftarrow (S - (f, g)) \cup \{S_{f,r} \mid \forall f \in G\}$
  end
end

Algorithm 1: Buchberger's Algorithm [1, p. 90].

In our first example, we found the GCD of two generators, and if it wasn't in our basis we added it to the basis. In this case, we compare two generators by looking at their S-polynomial. If, when dividing by our current basis of the ideal, the remainder of the S-polynomial is nonzero, we add this remainder to our basis. Once we have compared each pair, and no more new generators are found, the resulting basis is a Gröbner basis.

This algorithm is guaranteed to terminate by applying the ascending chain condition on the leading term ideal. Further details of this proof can be found in [1].

Another concern we have yet to address is why we can add the remainders to the set of generators. The S-polynomial $S(f, g)$ is some combination of its two polynomials f and g, both of which are in the ideal since they are generators of the ideal. Then we know

$S(f,g) \in I$, and by performing division by our basis to obtain a remainder, we are subtracting multiples of elements of the basis and thus remain within the ideal by definition, thus the remainder is in the ideal and can be added to the set of generators without changing the ideal generated by it.

This algorithm constructs a Gröbner basis for an ideal I, but there exists more than one such generating set that fits the definition for a Gröbner basis. The next few lemmas define what a better choice for a Gröbner basis might be, considering that in order to test ideal membership we need to divide by the basis so having fewer polynomials means less computation is required in general.

The following lemma helps us reduce the number of polynomials in a Gröbner basis by showing that if a particular property holds for an element in the basis, it can be removed and we would still have a Gröbner basis for the same ideal.

Lemma 2.19. *Let G be a Gröbner basis for the polynomial ideal I. Let $p \in G$ be a polynomial such that $LT(p) \in \langle LT(G - \{p\})\rangle$. Then $G - \{p\}$ is also a Gröbner basis for I.*

Using the above lemma, we see that it can be possible to reduce the number of elements in a Gröbner basis. We define a minimal Gröbner basis as follows.

Definition 2.20. *A **minimal Gröbner basis** for a polynomial ideal I is a Gröbner basis G for I such that:*

1. *$LC(p) = 1$ for all $p \in G$, and*

2. *For all $p \in G$, $LT(p) \notin \langle LT(G - \{p\})\rangle$.*

However, it is still the case that multiple minimal Gröbner bases can exist. The next definition is even more specific, and defines what is called a reduced Gröbner basis.

Definition 2.21. ***A reduced Gröbner basis** for a polynomial ideal I is a Gröbner bsis G for I such that:*

1. $LC(p) = 1$ *for all* $p \in G$.

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

Every ideal has a unique reduced Gröbner basis. This can be used to prove the equality of two ideals given a generating set for each, since if their reduced Gröbner bases are identical, then the ideals they define must be the same.

Chapter 3

APPLICATION IN VERIFICATION

In order to explain how Gröbner bases are used in the verification of arithmetic circuits over Galois fields, we will apply the verification process to the following implementation of multiplication over the Galois field $\mathbb{F}_{2^2}$, defined by the irreducible polynomial $P(x) = x^2 + x + 1 \in \mathbb{F}_2[x]$, with root $P(\alpha) = 0$, as described in Jinpeng Lv, Priyank Kalla, and Florian Enescu's "Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits" [3]. Figure 3.1 illustrates this multiplier.

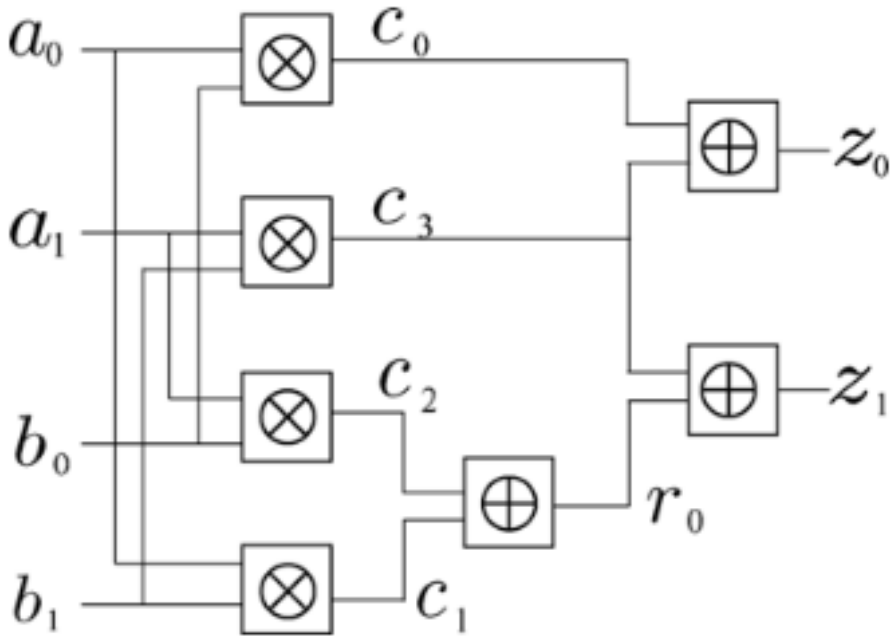

Figure 3.1: 2 bit multiplier example from [3].

The specification for this multiplier is $A \cdot B = Z$ or equivalently $A \cdot B - Z = 0$, where A, B are two bit inputs, and Z is the resulting value when multiplying A and B in the Galois field $\mathbb{F}_{2^2}$. The task of verification is to verify whether or not the above circuit will output the intended result for any valid input. An overview of the process is that one considers each gate as a equation on its inputs and outputs, moving all terms to one side to have the other side set to zero to obtain a polynomial which is zero when the relation between input and output is satisfied. Then we consider all values of the variables for which all polynomials are satisfied, and see if the specification lies on this variety. If that is the case, then for any input and output satisfying the specification, we know that the gates are satisfied as well, and thus the input to the circuit will produce the intended output, as determined by the specification polynomial.

In our example, each of $a_0, a_1, b_0, b_1, c_0, c_1, c_2, c_3, r_0, z_0$, and z_1 are elements of $\mathbb{F}_2 = \{0, 1\}$, and are referred to as bit-level variables. In addition, we have three word-level variables, A, B, and Z, corresponding to what (a_0, a_1), (b_0, b_1), and (z_0, z_1) represent in $\mathbb{F}_{2^2}$. Specifically, the polynomials relating the word-level variables and bit-level polynomials are

$A = a_0 + a_1 \cdot \alpha$

$B = b_0 + b_1 \cdot \alpha$

$Z = z_0 + z_1 \cdot \alpha$

We then write out the equations corresponding to each non-input gate:

$c_0 = a_0 \cdot b_0$

$c_1 = a_0 \cdot b_1$

$c_2 = a_1 \cdot b_0$

$c_3 = a_1 \cdot b_1$

$r_0 = c_1 + c_2$

$z_0 = c_0 + c_3$

We then rewrite the equations to have one side as 0 by subtracting the right from the left, noting that addition and subtraction are equivalent in $\mathbb{F}_{2^k}$:

$0 = c_0 + a_0 \cdot b_0$

$0 = c_1 + a_0 \cdot b_1$

$\vdots$

Finally, we consider the ideal of the variety generated by the above polynomials. By the strong Nullstellensatz (Theorem 2.4), this ideal is equal to the ideal generated by the polynomials along with the vanishing polynomials for every gate:

$A^4 + A$

$B^4 + B$

$Z^4 + Z$

$a_0^4 + a_0$

$\vdots$

However, a result of Lv, Kalla, and Enescu's work is that one can further reduce the generators to a minimal Gröbner basis by replacing the vanishing polynomials with the following, using only the bit-level input variables:

$a_0^2 + a_0$

$a_1^2 + a_1$

$b_0^2 + b_0$

$b_1^2 + b_1$

Intuitively, one can view this as constraining the values of the inputs to $\mathbb{F}_2$, or in terms of the circuit confining their values to being individual bits of value 0 or 1, and since each other variable is dependent on these values, they too are constrained to their intended possible values.

From these polynomials, we can use Buchberger's algorithm to compute a Gröbner basis for the circuit polynomials, and then divide the specification polynomial by the Gröbner basis to determine if it is in the ideal. However, the second optimization of Jinpeng Lv, Priyank Kalla, and Florian Enescu's work is obviating the need for Buchberger's algorithm. By using a particular monomial order, one can set the leading terms in such a way that one can guarantee that the S-polynomials will have remainder zero over division by the basis, in

other words the set of circuit polynomials and vanishing polynomials already form a Gröbner basis without any computation required.

How this is done is for a gate c with $a + b = c$, we rewrite the polynomial $a + b - c$ to have c as the leading monomial. Since no other gate will have c in its leading monomial, the S-polynomial of c with any other gate will divide to zero. The only pairs that remain are the input gates a_i and their vanishing polynomials $a_i^2 - a_i$, however their S-polynomials divide to zero as well. Thus by Buchberger's criterion 2.17, we have a Gröbner basis. The monomial order required for this simplification to work is a reverse topological order: for each gate $a + b = c$, we want $c > a$ and $c > b$, so a reverse topological sort of the gates gives us exactly this. Not only this, but give the chronological order of the creation of the gates, where each gate must have its inputs defined before itself, we already have a topological sort of the gates and thus we can take this order and reverse it to obtain the proper monomial order.

Here is the complete code for this circuit, using the computer algebra system Macaulay2 [2]:

```
-*
Lv et al Example 5
*-
restart
F = GF(4, Variable => a)

R = F[Z,A,B,z0,z1,r0,c0,c3,c1,c2,a0,a1,b0,b1,MonomialOrder=>Lex]

-- Specification polynomial
f = Z+A*B

-- Gate polynomials
f1 = c0+a0*b0
```

```
f2 = c1+a0*b1

f3 = c2+a1*b0

f4 = c3+a1*b1

f5 = r0+c1+c2

f6 = z0+c0+c3

f7 = z1+r0+c3

f8 = A+a0+a1*a

f9 = B+b0+b1*a

f10 = Z+z0+z1*a


-- Vanishing polynomials

f11 = a0^2+a0

f12 = a1^2+a1

f13 = b0^2+b0

f14 = b1^2+b1


-- Ideal of polynomials + vanishing polynomials

I = ideal(f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14)


-- Forcing Macaulay2 to divide using the given generators, which we
   claim form a Gr\"obner basis

forceGB(gens I);

f % I
```

Running this code in Macaulay2 returns the result that f divided by the listed generators for the ideal I is 0, and thus f is in the ideal I and the multiplier is verified.

Chapter 4

RESULTS

We have applied the algorithm and its optimizations to a small example, as outlined in Chapter 3, by using Macaulay2 and were able to verify the circuit. We attempted to verify a larger circuit, specifically a 128-bit multiplier, from David McGrew and John Viega's "The Galois/Counter Mode of Operation (GCM)" [4], but we were unable to successfully verify their algorithm for multiplication. However, we made some observations regarding the process. Firstly, despite our attempt at verification being insufficiently optimized, the number of polynomials constructed in the process of verification required significant memory to store, exceeding the 32 gigabytes available before terminating, and conjecture that even with further optimization this algorithm would still require significant amounts of memory to store all of the polynomials. We also observed that, having written the code that defines each gate in the circuit by requiring that its inputs are already defined as the outputs of previous gates, the chronological order of the construction of the gates provided a topological sort of the gates, which was exactly the reverse of the requirements for the monomial order. By creating the gates, polynomials, and their corresponding variables in parallel, then reversing the order of the list of variables, we have guaranteed that the monomial order defined by this list (using lexicographical order) meets the requirements for eliminating the need for Buchberger's algorithm.

BIBLIOGRAPHY

[1] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, third edition, 1997.

[2] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at `http://www2.macaulay2.com`.

[3] Jinpeng Lv, Priyank Kalla, and Florian Enescu. Efficient Gröbner Basis Reductions for Formal Verification of Galois Field Arithmetic Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(9):1409–1420, 2013.

[4] David A. McGrew and John Viega. The Galois/Counter Mode of Operation (GCM). 2005.