# Counting Zeros over Finite Fields Using Gröbner Bases

Sicun Gao

May, 2009

# Contents


**1 Introduction** **2**

**2 Finite Fields, Nullstellensatz and Gröbner Bases** **6**
- 2.1 Ideals, Varieties and Finite Fields . . . . . 6
- 2.2 Gröbner Bases . . . . . 11
- 2.3 Hilbert's Nullstellensatz . . . . . 21

**3 Counting with Gröbner Bases** **26**
- 3.1 Nullstellensatz in Finite Fields . . . . . 26
- 3.2 $|SM(J + \langle \bar{x}^q - \bar{x} \rangle)| = |V(J)|$ . . . . . 28

**4 Algorithm Analysis** **32**
- 4.1 Analysis of Buchberger's Algorithm . . . . . 32
- 4.2 Counting Standard Monomials . . . . . 35

**5 A Practical #SAT Solver** **37**
- 5.1 DPLL-based Approaches to #SAT . . . . . 37
- 5.2 Gröbner Bases in Boolean Rings . . . . . 39
- 5.3 Experimental Results . . . . . 40

**6 Conclusions and Future Work** **43**

**Bibliography** **45**

# Chapter 1

# Introduction

Given an arbitrary finite field $F_q$ of size $q$ ($q$ a prime power) and a system of multivariate polynomials $f_1, ..., f_m$ in $F_q[x_1, ..., x_n]$, we consider the problem of counting affine zeros of the ideal $\langle f_1, ..., f_m \rangle$ over $F_q$ (in other words, the number of common solutions of $f_1, ..., f_m$ in $F_q^n$). We will call this *the counting problem* throughout.

The counting problem established its importance in number theory since the work of Gauss on the law of reciprocity. The celebrated Riemann Hypothesis for curves over finite fields, formulated by Emil Artin and finally proved by Andre Weil, is concerned with an explicit bound of the number of rational points on curves (i.e., varieties of dimension one) over finite fields [32].

Methods for the counting problem have computational applications both theoretically and practically. In coding theory, for the design of algebraic-geometric codes such as Goppa codes, curves with a large number of rational points are needed, hence methods solving the counting problem are desirable [31]. In several primality testing algorithms, counting the number of points on elliptic curves [16] and hyperelliptic curves [1] over finite fields is a crucial step.

The special case of counting for polynomial systems over $F_2$, which corresponds to the so called *model counting* problem, has even wider applications. Model counting (#SAT) is the problem of computing the number of satisfying assignments to propositional formulas [7]. It is a natural generalization of SAT – many AI and combinatorial problems, lying beyond the capacity of SAT solvers, can be effectively coded as #SAT problems, such as various probabilistic reasoning problems [2, 25, 27]. While solving SAT is only concerned with whether the number of assignments is nonzero, solving the

#SAT problem requires keeping information of all the solutions and is significantly harder. In fact, #SAT is #P-complete, where #P is a complexity class known to contain the Polynomial Hierarchy [30].

The counting problem has attracted considerable algorithmic investigation from two different communities:

On the coding theory and algorithmic algebra side, several exact algorithms for certain special polynomials, as well as theoretical approximation algorithms for more general cases, have been proposed. Schoof [28] gave the first deterministic polynomial-time algorithm for counting rational points on elliptic curves over finite fields. Generalized methods for hyperelliptic curves were devised in [24, 14]. Approximation algorithms have been theoretically successful for single sparse polynomials [17, 21]. However, the general problem for polynomial systems has been less amenable to approximation methods. The best algorithm so far [19, 18], which applies only to prime fields, has time complexity $O(d^{n^{O(n)}}(m \log p)^{O(1)})$ (for prime fields of size $p$, $n$ the number of variables, $d$ the maximum degree of the polynomials and $m$ the number of polynomials).

On the AI and Boolean modeling side, various practical solvers have been developed for #SAT. Currently, approaches for solving #SAT [20, 26, 29] are mostly based on the DPLL algorithm [12] (with the exception of the method of knowledge compilation [11]). The basic idea is to extend the DPLL algorithm with procedures for book-keeping the distribution of solutions in the search space.

There does not exist an algorithm for the counting problem that is both general and practical. The approaches taken by the algorithmic algebra community usually involve sophisticated algebraic-geometric operations and remain only of theoretical interest so far – yet still, the most general theoretical (approximate) algorithm devised for the for polynomial systems [19, 18] is restricted to prime fields, and the worst-case runtime is doubly exponential in the number of variables. On the other hand, although practical solvers have been implemented in the AI community for the #SAT problem, the methods are specifically devised for Boolean variables and do not extend to other fields. Furthermore, the practical solvers are far less successful than DPLL approaches in tackling the SAT problem. In fact, effective heuristics for SAT usually aim at quick zooming-in on a particular satisfying assignment, while for counting the total number of satisfying assignments, the solver needs to take the full solution space into account. For this reason, current DPLL-based #SAT solvers scale orders of magnitude lower than SAT solvers [7].

The main contribution of this thesis is a new method for solving the counting problem which is both general and practical: It can be applied to any polynomial systems over arbitrary finite fields, and has been implemented to solve #SAT, outperforming existing solvers on various benchmarks.

The mathematical correctness of our method relies on a modified Nullstellensatz for finite field. Nullstellensatz is a celebrated theorem in classical algebraic geometry by David Hilbert, initially established for algebraically closed fields (such as the field of complex numbers) [23]. We will show that in the case of finite fields, any ideal $I \subseteq F[x_1, ..., x_n]$ can be easily made into a radical zero-dimensional ideal $I'$ that preserves variety over the finite field $F$. It follows that the Nullstellensatz can be modified to a nice form specifically applicable to finite fields. Then we prove that, the monomials that do not appear in the set of leading monomials of $I'$ (called "standard monomials") generates a vector space over $F$ that is isomorphic to the quotient ring of $F[x_1, ..., x_n]/I'$. As a consequence, the number of zeros of any system of polynomials can be obtained by counting the number of corresponding standard monomials.

On the algorithmic side, our method is based on the powerful method from computational commutative algebra called Gröbner bases [6, 5, 3]. A Gröbner basis $G$ for a system of polynomials $S$ is an equivalent system that possesses useful properties, for example, that a polynomial $f$ is a combination of those in $S$ if and only if the remainder of $f$ with respect to $G$ is 0. (Here, the division algorithm requires an order of a certain type on the monomials.) We will show that Gröbner basis computations provide an algorithmic method for counting the standard monomials, which is hence a way of counting the zeros of the corresponding polynomial systems.

However, a conceivable problem with our method can be the notoriously high computational complexity of Gröbner basis computations. It is known that Gröbner basis computations are at least EXPSAPCE-hard [22], which translates to worst-case running time doubly exponential in the number of variables. But in fact, for the problem that we are solving, a careful analysis of the Buchberger's Algorithm for Gröbner basis construction shows that it stays in single exponential time. Furthermore, we have evaluated the practical Gröbner of our algorithm in tackling practical problems. Experimental results show that our solver outperforms existing search-based solvers significantly on a number of benchmarks, and is competitive in general in terms of the size of problems (number of variables and clauses) that can be handled.

The thesis is organized as follows. In Section 2, we review basic mathematical background, the theory of Gröbner bases and Hilbert's Nullstellen-

satz. The materials are standard and can be found in [3, 9, 23]. In Section 3, we prove the modified Nullstellensatz for finite fields, and how Gröbner bases can be used in the counting problem. In section 4, we analyze the theoretical worst-case complexity of our algorithm. In Section 5, we focus on our practical solver for the #SAT problem and show experimental results compared with existing solvers. Section 6 contains conclusions and discussion of future work.

# Chapter 2

# Finite Fields, Nullstellensatz and Gröbner Bases

## 2.1 Ideals, Varieties and Finite Fields

**Definition 2.1.1.** *A **monoid** is a set $M$ with an associative binary operation "$\cdot$" and an element $e \in G$, such that, for all $a \in G$, $e \cdot a = a$.*

We use $(M, \cdot, e)$ to denote a monoid defined as such, and often use M only when no ambiguity arises. This tuple notation also applies for groups, rings and fields.

**Definition 2.1.2.** *A **group** is a set $G$ with an associative binary operation "$\cdot$", such that*
*(1) $(G, \cdot, e)$ forms a monoid;*
*(2) For all $a \in G$, there exists some $b \in G$ satisfying $b \cdot a = e$.*

**Definition 2.1.3.** *A monoid/group $G$ is called **commutative** if for all $a, b \in G$, $a \cdot b = b \cdot a$.*

**Definition 2.1.4.** *A **commutative ring with unity** is a set $R$ with two binary operations "$+$" and "$\cdot$", as well as two distinct elements $0, 1 \in R$, such that*
*(1) $(R, +, 0)$ forms a commutative group;*
*(2) $(R, \cdot, 1)$ forms a commutative monoid;*
*(3) For all $a, b, c \in R$, $a \cdot (b + c) = a \cdot b + a \cdot c$.*

Since in this paper we do not need to consider any non-commutative rings or rings without unity, we henceforth use "**ring**" to mean commutative ring with unity as defined above.

**Definition 2.1.5.** *An element $a$ in a ring $R$ is called a* ***zero divisor****, if $a \neq 0$ and $ab = 0$ for some $b \neq 0$. An element is called a* ***unit****, or* ***invertible****, if $a \cdot c = 1$ for some $c \in R$.*

**Definition 2.1.6.** *A* ***polynomial ring*** *$R[x_1, ..., x_k]$ over a ring $R$, is the set of polynomials with variables in $x_1, ..., x_k$ and coefficients in $R$, together with ordinary addition and multiplication defined over polynomials. $0, 1 \in R[x_1, ..., x_k]$ are the same as in $R$.*

**Definition 2.1.7.** *A* ***field*** *is a set $F$ with two binary operations $+$ and $\cdot$ and two distinct elements $0, 1 \in F$ such that $(F, +, \cdot, 0, 1)$ forms a ring and every nonzero element $a \in F$ (i.e. $a \neq 0$) is invertible. A field is called a* ***finite field*** *or* ***Galois field*** *when $|F|$ is finite.*

**Definition 2.1.8.** *Let $R$ be a ring. A nonempty set $I \subset R$ is called an* ***ideal*** *of $R$ if:*
*(1) For all $a, b \in I$, $a + b \in I$;*
*(2) For all $a \in I$, for all $r \in R$, $a \cdot r \in I$.*

For any element $r$ in $R$, we define addition and multiplication of $r$ with an ideal $I$ to be

$$r + I = \{r + a : a \in I\},$$

and

$$r \cdot I = \{r \cdot a : a \in I\}.$$

Then we are able to define:

**Definition 2.1.9.** *Let $R$ be a ring and $I$ an ideal in $R$. The* ***quotient ring*** *$R/I$ is defined on the set:*

$$R/J = \{r + J : r \in R\}$$

*together with addition*

$$(r_1 + J) + (r_2 + J) = (r_1 + r_2) + J,$$

*and*

$$(r_1 + J) \cdot (r_2 + J) = (r_1 \cdot r_2) + J.$$

*Zero is naturally defined as $0 + J$ and unity as $1 + J$.*

**Definition 2.1.10.** *An ideal $I \subset R$ is a* ***prime ideal*** *if it is proper and $a \cdot b \in I$ implies $a \in I$ or $b \in I$.*

**Definition 2.1.11.** *An ideal $I \subseteq R$ is a* ***maximal ideal****, if for any other ideal $J \subseteq R$, whenever $I \in J$, either $J = I$ or $J = R$.*

**Theorem 2.1.12.** *Let $I$ be an ideal of $R$. Then the quotient ring $R/M$ is a field if and only if $I$ is a maximal ideal of $R$.*

*Proof.* Left to right:

Suppose that $R/I$ is a field and let $J$ be an ideal of $R$ properly containing $I$. Let $a \in J$ and $a \notin I$. Then $a + I$ is not the zero element of $R/I$, and so there exists $b+I \in R/I$ such that $(a+I)(b+I) = 1+I$. Then $ab-1 \in I \subseteq$. But since $a \in J$, $ab \in J$. Thus $1 \in J$. Hence $I$ is a maximal ideal.

Right to left:

Suppose that $I$ is a maximal ideal of R and let $a+I$ be a non-zero element of $R/I$. We need to show the existence of $b+I \in R/I$ with $(a+I)(b+I) = 1+I$, namely, $ab-1 \in I$.

Let $I'$ denote the set of elements of R of the form

$$ar + s \text{ for some } r \in R, s \in I$$

. Then $I'$ is an ideal of $R$ and properly contains I since $a \in I'$ but $a \notin I$. Then $I' = R$ because of the maximality of $I$. In particular, then $1 \in M'$ and by definition, $1 = ab + c$ for some $b \in R$ and $c \in I$. Then $ab - 1 \in I$ and $a + I$ has an inverse in $R/I$ as required. □

**Definition 2.1.13** (Field Extension)**.** *Let $k$ be a field. If $k'$ is a subset of $k$ which is closed with respect to the field operations of addition and multiplication in $k$ and the additive and multiplicative inverses of every element in $k'$ are in $k'$, then we say that $k'$ is a subfield of $k$, that $k$ is an* extension field *of $k'$, and that $k/k'$ ("$k$ over $k'$") is a field extension.*

**Definition 2.1.14.** *Let $k$ be is a field extension of $k'$, then an element $a \in k$ is called algebraic over $k'$, if there exists some non-zero polynomial $p(x)$ with coefficients in $k'$ such that $p(a) = 0$. Elements of $k$ which are not algebraic over $k'$ are called transcendental over $k'$.*

**Definition 2.1.15** (Algebraic Extension)**.** *A field extension $k/k'$ is called algebraic if every element of $k$ is algebraic over $k'$.*

**Definition 2.1.16** (Algebraically Closed Field)**.** *A field $k$ is said to be* algebraically closed *if every polynomial in one variable of degree at least 1, with coefficients in $k$, has a root in $k$. An* algebraic closure *of a field $k$ is an algebraic extension of $k$ that is algebraically closed.*

**Theorem 2.1.17.** *Let $k$ be a field. There exists an algebraic closure $k^a$ of $k$.*

The theme of our study will be the relation between ideals in the polynomial ring $k[x_1, ..., x_n]$ over some field $k$ and the points in the $k^n$. The correspondence between the two sets can be captured by two functions: one function takes some set of points in $k^n$ and returns the ideal of polynomials in $k[x_1, ..., x_n]$ that can vanish (i.e., be made zero) at these points; the other takes an arbitrary ideal of polynomials, and returns the points in $k^n$ that can make these polynomials be zero. Formally speaking, (for simplicity we write $R$ for $k[x_1, ..., x_n]$)

**Definition 2.1.18.** *$V : \wp(R) \to k^n$ is defined as*

$$V(J) = \{\bar{a} \in k^n : \forall f(\bar{x}) \in J, f(\bar{a}) = 0\}$$

**Definition 2.1.19.** *$I : \wp(k^n) \to R$, s.t.*

$$I(A) = \{f \in R : \forall \bar{a} \in A, f(\bar{a}) = 0\}$$

**Proposition 2.1.20.** *For any $A \subset k^n$, $I(A)$ forms an ideal in $R$.*

*Proof.* For any A, $0 \in I(A)$, hence it's not empty. For any $f, g \in I(A)$, $f$ and $g$ vanish on $A$, hence $f + g$ also vanishes on $A$. For any $f \in I(A)$ and $h \in R$, $fh$ also vanishes on $A$, hence $fg \in I(A)$.

In all, $I(A)$ forms an ideal in $R$.

□

**Definition 2.1.21.** *For any field $k$ and any ideal $J \subset k[x_1, ..., x_n]$, **the variety of $J$ over the algebraic closure of** $k$, written as $V^a(J)$, is defined as*

$$V^a(J) = \{\bar{a} \in k^a[x_1, ..., x_n] : \forall f \in J, f(\bar{a}) = 0\}$$

**Definition 2.1.22.** *For any field $k$ and a set of points $A \subset (k^a)^n$,*

$$I(A) = \{f \in k[x_1, ..., x_n] : \forall \bar{a} \in A, f(\bar{a}) = 0\}$$

*$I(A)$ is easily verifiable as an ideal in $k[x_1, ..., x_n]$, called **the vanishing ideal of $V$**.*

**Definition 2.1.23.** *The **radical** of an ideal $J$, written as $\sqrt{J}$, is defined as*

$$\sqrt{J} = \{f \in R : \exists r > 0, f^r \in J\}$$

*. $J$ is called a radical ideal if $J = \sqrt{J}$.*

**Proposition 2.1.24.** *The radical $\sqrt{J}$ of an ideal $J$ (in a ring $R$) is an ideal.*

*Proof.* Let $a, b \in \sqrt{J}$ be arbitrary. By definition there exists $n, m \in \mathbb{N}$ such that $a^n, b^m \in J$. Obviously $J \subseteq \sqrt{J}$, hence we only need to prove that $\sqrt{J}$ is closed under addition.

Since $(a+b)^{m+n} = \sum_{i=0}^{n+m} \binom{n+m}{i} a^i b^{m+n-i}$, each term contains either $a^n$ or $b^m$. Thus we have $(a+b)^{m+n} \in J$. By definition, $a + b \in \sqrt{J}$.

□

**Lemma 2.1.25.** *Let $J_i$ be ideals in $k[x_1, ..., x_n]$ and $A_i$ be a subset of $k^n$, then:*

$$(1) V(J_1 + J_2) = V(J_1) \cap V(J_2)$$
$$(2) V(\bigcap_i J_i) = \bigcup_i V(J_i)$$
$$(3) I(\bigcup_i A_i) = \bigcap_i I(A_i)$$

*Proof.* (1) Consider any point $P \in V(J_1 + J_2) \subseteq k^n$, then all $f \in J_1 + J_2$ vanishes on $P$, in particular, $J_1$ and $J_2$ vanish on $P$. Thus $P \in V(J_1) \cap V(J_2)$. Hence

$$V(J_1 + J_2) \subseteq V(J_1) \cap V(J_2)$$

.

On the other hand, consider any point $P \in V(J_1) \cap V(J_2)$. For any $f + g \in J_1 + J_2$, $f \in J_1$ and $g \in J_2$ both vanish on $P$. Hence

$$V(J_1) \cap V(J_2) \subseteq V(J_1 + J_2).$$

(2) Consider any point $P$ in $V(\bigcap_i J_i)$. Suppose $P \notin V(J_i)$ for all $J_i$, then for each $J_i$ there's some $f_i \in J_i$ such that $f_i$ does not vanish on $P$. By definition of ideals, $\prod_i f_i \in \bigcap_i J_i$, which means $P \notin V(\bigcap_i J_i)$, leading to contradiction. Hence

$$V(\bigcap_i J_i) \subseteq \bigcup_i V(J_i).$$

On the other hand, for each $J_i$ we have $V(J_i) \subset V(\bigcap_i J_i)$, for the reason that at any point $P \in V(J_i)$, all $f \in J_i$ must vanish, in particular, all $f \in \bigcap_i J_i$ must vanish on $P$. Hence

$$\bigcup_i V(J_i) \subseteq V(\bigcap_i J_i)$$

(3) For any $f \in I(\bigcup_i A_i)$, $f$ vanishes on all the points in $\bigcup_i A_i$, thus $f \in I(A_i)$ for each $A_i$. Hence

$$I(\bigcup_i A_i) \subset \bigcap_i I(A_i).$$

On the other hand, for any $f \in \bigcap_i I(A_i)$, then f vanishes on all the points in each $A_i$, which means $f_i \in I(\bigcup_i A_i)$.

□

The size of a finite field can only be $p^n$, where $p$ is a prime number and $n$ is a positive integer. When $n = 1$, finite fields of size $p$ (written as $GF(p)$) are isomorphic to natural numbers modulo $p$, i.e., $\{0, 1, ..., (p-1)\}$; when $n > 1$, finite fields of size $p^n$ (written as $GF(p^n)$) are isomorphic to the field of equivalence classes of polynomials whose coefficients belong to $GF(p)$.

**Proposition 2.1.26** (Generalized Fermat's Little Theorem)**.** *For any finite field of size $q$, every element $a \in F$ satisfies:*

$$a^q \equiv a.$$

*Proof.* If $a$ is zero, then $0^q = 0$.

If $a$ is not zero, then $a$ is in the cyclic multiplicative group $(F/\{0\}, \cdot, 1)$[23], by Lagrange's theorem, $a^{q-1} = 1$.

In both cases $a^q = a$. □

## 2.2 Gröbner Bases

**Definition 2.2.1** (Well-ordering)**.** *An order relation $>$ on $\mathbb{N}^n$ is a well-ordering if every strictly decreasing sequence*

$$\alpha(1) > \alpha(2) > \cdots$$

*eventually terminates.*

**Definition 2.2.2** (Monomial Order)**.** *Define the set of* monomials *in $k[x_1, ..., x_n]$ to be: $T = \{x_1^{\alpha_1} \cdots x_n^{\alpha_n} : \alpha_i \in \mathbb{N}\} \subseteq k[x_1, ..., x_n]$. A* monomial order *on $T$ is a total well-ordering on $T$ satisfying*
*(1) For any $t \in T$, $1 \leq t$*
*(2) For all $t_1, t_2, s \in T$, $t_1 \leq t_2$ then $t_1 \cdot s \leq t_2 \cdot s$.*

We give two examples of monomial order:

**Definition 2.2.3** (Lexicographic Order)**.** *Let* $\alpha = (\alpha_1, ..., \alpha_n)$ *and* $\beta = (\beta_1, ..., \beta_n)$*.* $\alpha >_{lex} \beta$ *if in the vector difference* $\alpha - \beta \in \mathbb{Z}^n$*, the leftmost nonzero entry is positive.*

**Definition 2.2.4** (Graded Lexicographic Order)**.** *Let* $\alpha, \beta \in \mathbb{N}^n$*.* $\alpha >_{grlex} \beta$ *if*

$$\sum_{i=1}^{n} \alpha_i > \sum_{i=1}^{n} \beta_i, \text{ or } \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i \text{ and } \alpha >_{lex} \beta.$$

**Definition 2.2.5.** *Let* $f = \sum_\alpha a_\alpha x^\alpha$ *be a nonzero polynomial in* $k[x_1, ..., x_n]$ *and* $>$ *a monomial order. The* multidegree *of* $f$ *is defined as*

$$multideg(f) = \max_{>}(\alpha \in \mathbb{N}^n : a_\alpha \neq 0)$$

*The* leading coefficient *of* $f$ *is* $LC(f) = a_{multideg(f)}$ *and the* leading monomial *is* $LM(f) = x^{multideg(f)}$*. The leading term of* $f$ *is* $LT(f) = LC(f) \cdot LM(f)$*.*

**Proposition 2.2.6.** *Let* $f, g \in k[x_1, ..., x_n]$ *be nonzero polynomials and* $<$ *a monomial order, then:*

*(1)* $multideg(fg) = multideg(f) + multideg(g)$*.*

*(2) If* $f+g \neq 0$ *then* $multideg(f+g) \leq \max_<(multideg(f), multideg(g))$*.*

**Theorem 2.2.7** (Multivariate Polynomial Division)**.** *Fix a monomial order* $>$ *on* $Z^n_{\geq 0}$ *and let* $F = (f_1, ..., f_s)$ *be an ordered s-tuple of polynomials in* $k[x_1, ..., x_n]$ *Then every* $f \in k[x_1, ..., x_n]$ *can be written as*

$$f = a_1 f_1 + ... + a_s f_s + r$$

*where* $a_i, r \in k[x_1, ..., x_n]$ *and either* $r = 0$ *or* $r$ *is a linear combination of monomials not divisible by any of* $LT(f_1), ..., LT(f_s)$ *(with coefficients in* $k$*).* $r$ *is called the remainder of* $f$ *on division by* $F$*.*

*Proof.* The existence of $a_1, ..., a_s$ and $r$ can be shown by an algorithm as follows:

**Algorithm *Multivariate Polynomial Division***
**Input:** $f_1, ..., f_s, f$
**Output:** $a_1, ..., a_s, r$
1. $a_1 \leftarrow 0, ...a_s \leftarrow 0, r \leftarrow 0$
2. $p \leftarrow f$
3. **while** $p \neq 0$
4. **do**

```
5.          i ← 1
6.          division_occured←false
7.          while i ≤ s AND division_occured = false
8.             do
9.                if LT(f_i) divides p
10.                  then
11.                        a_i ← a_i + LT(p)/LT(f_i)
12.                        p ← p − (LT(p)/LT(f_i))f_i
13.                        division_occured ← true
14.                  else
15.                        i ← i + 1 if division_occured=false
16.            then
17.                r ← r + LT(p)
18.                p ← p − LT(p)
```

To prove that the algorithm works, we first show that

$$(\star) \quad f = a_1 f_1 + ... + a_s f_s + p + r$$

holds as every stage. This is clearly true for the initial values.

Now suppose it holds at one step of the algorithm. If next step is a division step, then some $LT(f_i)$ divides $LT(p)$ and the equality

$$a_i f_i + p = (a_i + LT(p)/LT(f_i))f_i + (p - (LT(p)/LT(f_i))f_i)$$

shows that $a_i f_i + p$ is unchanged. Since all other variables are unaffected, $(\star)$ remains true in this case.

On the other hand, if the next step is a remainder step, then $p$ and $r$ will be changed, but the sum $p + r$ is unchanged since

$$p + r = (p - LT(p)) + (r + LT(p)).$$

Hence the equality $(\star)$ is still preserved.

Next, notice that the algorithm comes to a halt when $p = 0$. In this situation, $(\star)$ becomes

$$f = a_1 f_1 + ... + a_s f_s + r.$$

Since terms are added to $r$ only when they are divisible by none of the $LT(f_i)$, it follows that $a_1, ..., a_s, r$ have the properties when the algorithm terminates.

Now we need to show that the algorithm indeed terminates. Each time we redefine the $p$, either its degree drops with respect to the term ordering

or it becomes 0. To see this, first suppose that during a division step, $p$ is redefined to be

$$p' = p - \frac{LT(p)}{LT(f_i)} f_i.$$

But we know

$$LT(\frac{LT(p)}{LT(f_i)} f_i) = \frac{LT(p)}{LT(f_i)} LT(f_i) = LT(p),$$

so that $p$ and $(LT(p)/LT(f_i))f_i$ have the same leading term. Hence, their different $p'$ must have strictly smaller degree when $p' \neq 0$. Next suppose that during a remainder step, $p$ is redefined to be

$$p' = p - LT(p).$$

Then obviously $\deg(p') < \deg(p)$ when $p' \neq 0$. Thus in either case the degree of $p$ must decrease. But the well-ordering of $>$, the algorithm has to terminate. □

**Definition 2.2.8.** *Let $I$ be an ideal in $F[x_1, ..., x_n]$. Fix any monomial order on $T$.* The ideal of leading monomials *of $I$, $\langle LM(I) \rangle$, is the ideal generated by the leading monomials of all polynomials in $I$.* The ideal of leading terms *of $I$, $\langle LT(I) \rangle$, is the ideal generated by the leading terms of all polynomials in $I$.*

**Lemma 2.2.9** (Dickson's Lemma)**.** *Let $I = \langle x^\alpha : \alpha \in A \rangle \subseteq k[x_1, ..., x_n]$ be a monomial ideal. Then $I$ can be written in the form $I = \langle x^{\alpha(1)}, ..., x^{\alpha(s)}$ where $\alpha(1), ..., \alpha(s) \in A$. That is, $I$ has a finite basis.*

*Proof.* Do induction on the number of variables. If $n = 1$ then $I$ is generated by the univariate monomials, and the smallest $\beta \in A$ satisfies the requirement.

If $n > 1$, assume that the theorem is true for $n - 1$. Suppose $I \subseteq k[x_1, ..., x_n]$ is a monomial ideal. To find generators for $I$, let $J$ be the ideal in $k[x_1, ..., x_{n-1}]$ generated by the monomials $x_1^{a_1} \cdots x_{n-1}^{a_{n-1}}$ for which $x_1^{a_1} \cdots x_{n-1}^{a_{n-1}} x_n^{a_n} \in I$ for some $a_n \in \mathbb{N}$. By inductive hypothesis $J$ has a finite base, i.e., $J = \langle x^{\alpha(1)}, ..., x^{\alpha(s)} \rangle$ where $\alpha(s) \in \mathbb{N}^{n-1}$.

For each $i \in \{1, ..., s\}$, the definition of $J$ says that $x_i^\alpha x_n^{m_i} \in I$ for some $m_i \in \mathbb{N}$. Let $m$ be the largest of $m_i$. Then for each $k \in \{0, ..., m-1\}$, consider the ideal $J_k \subseteq k[x_1, ..., x_n - 1]$ generated by the monomial $x^\beta$ such that $x^\beta x_n^k \in I$. Again by inductive hypothesis we have $J_k = \langle x^{\alpha_k(1)}, ..., x^{\alpha_k(s_k)} \rangle$.

We claim that $I$ is generated by $Jx_n^m, J_0, ..., J_{m-1}x_n^{m-1}$.

Consider any $x^\alpha x_n^p \in I$. If $p \geq m$ then $x^\alpha x_n^p$ is divisible by some $x^{\alpha(i)} x_n^m$ by the construction of $J$. On the other hand, if $p \leq m-1$ then $x^\alpha x_n^p$ is divisible by some $x^{\alpha_p(j)} x_n^p$ by the construction of $J_p$. It follows that the above monomials generate an ideal having the same monomials as $I$. □

**Theorem 2.2.10** (Hilbert's Basis Theorem)**.** *Every ideal $I \subseteq k[x_1, ..., x_n]$ has a finite generating set.*

*Proof.* If $I = \{0\}$ then the generating set is just $\{0\}$. Otherwise, $I$ contains some nonzero polynomial. By Lemma 2.2.9, there are $g_1, ..., g_t \in I$ such that $\langle LT(I)\rangle = \langle LT(g_1), ..., LT(g_t)\rangle$.

We claim that $I = \langle g_1, ..., g_t\rangle$.

Let $f \in I$ be any polynomial. If we apply the multivariate division algorithm to divide $f$ by $\langle g_1, ..., g_t\rangle$, then we get

$$f = a_1 g_1 + ... + a_t g_t + r$$

where $r$ contains no terms divisible by $LT(g_1), ..., LT(g_t)$. Note that

$$r = f - a_1 g_1 - ... - a_t g_t \in I$$

i.e., if $r \neq 0$ then $LT(r) \in \langle LT(I)\rangle = \langle LT(g_1), ..., LT(g_t)\rangle$. Hence $LT(r)$ should be divisible by some $LT(g_i)$ and contradicts the definition of a remainder. Hence $f \in \langle g_1, ..., g_t\rangle$. Since trivially $\langle g_1, ..., g_t\rangle \subseteq I$, we have $I = \langle g_1, ..., g_t\rangle$. □

An immediate application of the Hilbert basis theorem is the Ascending Chain Condition theorem.

**Theorem 2.2.11** (The Ascending Chain Condition)**.** *Let $I_1 \subseteq I_2 \subseteq I_3 \subseteq ...$ be a chain of ideals in $k[x_1, ..., x_n]$. Then there exists some $N$ such that $I_N = I_{N'}$ for any $N' > N$.*

*Proof.* Consider the set $I = \bigcup_{i=1}^{\infty} I_i$. $I$ is an ideal: (1) $0 \in I$ (2) For any $f, g \in I$ we have $f, g \in I_j$ for some $j$ because of the inclusion chain, then $f + g \in I_j$ and $fh \in I_i$ for any $h \in k[x_1, ..., x_n]$.

By Hilbert's basis Theorem, the ideal $I$ has a finite generating set $I = \langle f_1, ..., f_s\rangle$. Let $N$ be the maximum index $j$ such that $f_j \in I_j$, then we have

$$I = \langle f_1, ..., f_s\rangle \subseteq I_N = I_{N+1} = \cdots = I.$$

□

**Definition 2.2.12** (Gröbner bases)**.** *Let $I$ be an ideal in $F[x_1, ..., x_n]$. A* Gröbner basis *for $J$ is defined as a finite set $GB(I) = \{g_1, ..., g_s\}$, $g_i \in I$, such that*

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

**Corollary 2.2.13.** *Every ideal $I \subseteq k[x_1, ..., x_n]$ has a Gröbner basis.*

*Proof.* It follows directly from the proof of Hilbert's basis theorem. □

Now we discuss some properties of Gröbner basis.

**Theorem 2.2.14.** *Let $G = \{g_1, ..., g_t\}$ be a Gröbner basis for an ideal $I \subseteq k[x_1, ..., x_n]$. Then any $f \in k[x_1, ..., x_n]$ has a unique remainder on division by $G$.*

*Proof.* Suppose

$$f = \sum_{i=1}^{t} a_i g_i + r = \sum_{i=1}^{t} a_i' g_i + r'$$

Then $r - r' = \sum_{i=1}^{t}(a_i' - a_i)g_i \in I$. Hence if $r \neq r'$, $LT(r - r') \in \langle LT(g_1), ..., LT(g_t)\rangle$, i.e., $LT(r - r')$ is divisible by some $LT(g_i)$. But this is impossible since no term of $r, r'$ is divisible by any of $LT(g_1), ..., LT(g_t)$. Thus $r - r' = 0$ and uniqueness is proved. □

The remainder $r$ in the previous theorem is also called the *normal form* of $f$. We write this normal form as $\bar{f}^G$.

**Corollary 2.2.15.** *Let $G = \{g_1, ..., g_t\}$ be a Gröbner basis for an ideal $I \subseteq k[x_1, ..., x_n]$ and let $f \in k[x_1, ..., x_n]$. Then $f \in I$ iff the remainder of $f$ on division by $G$ is zero.*

*Proof.* If $f \in I$, then $f = f + 0$ and by uniqueness of the remainder, 0 is the remainder of $f$ on division by $G$. If the remainder is zero, then trivially $f \in I$. □

**Definition 2.2.16.** *Let $f, g \in k[x_1, ..., x_n]$ be nonzero polynomials.*

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

*(2) The* S-polynomial *of $f$ and $g$ is defined as:*

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

*where $\gamma$ is defined as in (1).*

**Lemma 2.2.17.** *Suppose $\sum_{i=1}^{s} c_i f_i$ satisfies $c_i \in k$ and $multideg(f) = \delta \in \mathbb{N}^n$ for all $i$. If $multideg(\sum_{i=1}^{s} c_i f_i) < \delta$ then $\sum_{i=1}^{s} c_i f_i$ is a linear combination of S-polynomials $S(f_j, f_k)$ for $1 \leq j, k \leq s$. Furthermore, each $S(f_j, f_k)$ has multidegree $< \delta$.*

*Proof.* Let $d_i$ be the leading coefficient of $f_i$. Since $multideg(\sum_{i=1}^{s} c_i f_i) < \delta$, we must have $\sum_{i=1}^{s} c_i d_i = 0$.

Define $p_i = f_i/d_i$ and note that $p_i$ has leading coefficient 1. Consider

$$\sum_{i=1}^{s} c_i f_i = \sum_{i=1}^{s} c_i d_i p_i = c_1 d_1 (p_1 - p_2) + (c_1 d_1 + c_2 d_2)(p_2 - p_3) + ... + (c_1 d_1 + ... + c_s d_s) p_s.$$

By assumption, $LT(f_i) = d_i x^\delta$, which implies that the least common multiple of $LT(f_j)$ and $LM(f_k)$ is $x^\delta$. Thus

$$S(f_j, f_k) = \frac{x^\delta}{LT(f_j)} f_j - \frac{x^\delta}{LT(f_k)} f_k = \frac{x^\delta}{d_j x^\delta} f_j - \frac{x^\delta}{d_k x^\delta} f_k = p_j - p_k.$$

It follows that

$$\sum_{i=1}^{s} c_i f_i = c_1 d_1 S(f_1, f_2) + ... + (c_1 d_1 + ... + c_{s-1} d_{s-1}) S(f_{s-1}, f_s),$$

which is a sum of the desired form. Since $p_j$ and $p_k$ have multidegree $\delta$ and leading coefficient 1, the difference $p_j - p_k$ has multidegree less than $\delta$. Thus the same is true for each $S(f_j, f_k)$. □

Now we are ready to understand Buchberger's algorithm.

**Theorem 2.2.18** (Buchberger's Criterion)**.** *Let $I$ be a polynomial ideal. Then a basis $G = g_1, ..., g_t$ is a Gröbner basis for $I$ if and only if for all pairs $i \neq j$, the remainder of $S(g_i, g_j)$ on division by $G$ is zero.*

*Proof.* Left to right is easy, since S-polynomials are in the ideal, and the remainder on division by the Gröbner basis has to be zero.

Right to left:

Let $f \in I$ be a nonzero polynomial. Then since $\{g_1, ..., g_t\}$ is a basis,

$$(\star) \quad f = \sum_{i=1}^{t} h_i g_i.$$

Let $m(i) = multideg(h_i g_i)$, and define $\delta = \max(m(1), ..., m(t))$, then we have $multideg(f) \leq \delta$. Consider all the possible expressions of $f$ in the $(\star)$

form and let fix $\delta$ to be the minimal degree with respect to the monomial order.

Now we show that $multideg(f) = \delta$. Assume $multideg(f) < \delta$. Write $f$ as:

$$\begin{aligned} f &= \sum_{m(i)=\delta} h_i g_i + \sum_{m(i)<\delta} h_i g_i \\ &= \sum_{m(i)=\delta} LT(h_i) g_i + \sum_{m(i)=\delta} (h_i - LT(h_i)) g_i + \sum_{m(i)<\delta} h_i g_j \end{aligned}$$

Thus the assumption implies that $multideg(\sum_{m(i)=\delta} LT(h_i)g_i) < \delta$.

Let $LT(h_i) = c_i x^{\alpha(i)}$. Then the first sum $\sum_{m(i)=\delta} LT(h_i)g_i = \sum_{m(i)=\delta} c_i x^{\alpha(i)}$ has the form described in Lemma 2.2.17. Thus we have (let $t$ be the number of $i$ such that $m(i) = \delta$)

$$\sum_{m(i)=\delta} LT(h_i)g_i = \sum_{1\le j,k\le l} S(x^{\alpha(j)} g_j, x^{\alpha(k)} g_k).$$

Yet we also have

$$S(x^{\alpha(j)} g_j, x^{\alpha(k) g_k}) = \frac{x^\delta}{x^{\alpha(j)} LT(g_j)} x^{\alpha(g)} g_j - \frac{x^\delta}{x^{\alpha(k)} LT(g_k)} x^{\alpha(k)} g_k = x^{\delta - \gamma_{jk}} S(g_j, g_k).$$

where $x^{\gamma_{jk}} = LCM(LM(g_j), LM(g_k))$. Thus, there exist constant $c_{jk} \in k$ satisfying that

$$(1) \qquad \sum_{m(i)=\delta} LT(h_i)g_i = \sum_{1\le j,k\le l} c_{jk} x^{\delta-\gamma_{jk}} S(g_j, g_k).$$

By assumption, the remainder of $S(g_j, g_k)$ on division by $g_1, ..., g_t$ is zero, i.e.,

$$S(g_j, g_k) = \sum_{i=1}^{t} a_{ijk} g_i,$$

where $a_{ijk} \in k[x_1, ..., x_n]$. Furthermore, note that $multideg(a_{ijk} g_i) \le multideg(S(g_j, g_k))$.

Now we multiply the expression for $S(g_j, g_k)$ by $x^{\delta-\gamma_{jk}}$ to obtain

$$(2) \quad x^{\delta-\gamma_{jk}} S(g_j, g_k) = \sum_{i=1}^{t} b_{ijk} g_i,$$

where $b_{ijk} = x^{\delta-\gamma_{ijk}} a_{ijk}$. Applying Lemma 2.2.17 again, we have

$$multideg(b_{ijk} g_i) \le multideg(x^{\delta-\gamma_{jk}} S(g_j, g_k)) < \delta.$$

Now, plugging (2) into (1) we can write

$$\sum_{m(i)=\delta} LT(h_i)g_i = \sum_{i,k} c_{jk}x^{\delta-\gamma_{jk}}S(g_j,g_k) = \sum_{j,k} c_{jk}(\sum_i b_{ijk}g_j)$$

Hence we have $multideg(\sum_{m(i)=\delta} LT(h_i)g_i) < \delta$. But this means $multideg(f) < \delta$, which contradicts our assumption of the minimality of $\delta$. □

Now we are ready to show Buchberger's Algorithm for constructing Gröbner bases.

**Theorem 2.2.19** (Buchberger's Algorithm)**.** *Let $I = \langle f_1, ..., f_s\rangle \neq \{0\}$ be a polynomial ideal. Then a Gröbner basis for $I$ can be constructed in a finite number of steps by the following algorithm:*

***Algorithm***
***Input:*** $f_1, ..., f_s$
***Output:*** *A Gröbner basis* $G = \{g_1, ..., g_t\}$ *for* $I$
1. $G \leftarrow f_1, ..., f_s$
2. ***repeat***
3. $\quad G' \leftarrow G$
4. $\quad$ ***for*** *Each pair* $\{p, q\}$, $p \neq q$ *in* $G'$
5. $\quad\quad$ ***do***
6. $\quad\quad\quad S \leftarrow \overline{S(p,q)}^{G'}$
7. $\quad\quad\quad$ ***if*** $S \neq 0$
8. $\quad\quad\quad\quad$ ***then*** $G \leftarrow G \cup \{S\}$
9. ***until*** $G = G'$

*Proof.* We first show that $G \subseteq I$ holds true at every stage of the algorithm. This is true initially. Whenever we enlarge $G$, it is done by adding remainder $S = \overline{S(p,q)}^{G'}$ for $p, q \in G$. Since by inductive hypothesis $G \subseteq I$, $G \cup S \subseteq I$ holds. We also note that $G$ contains the given polynomials $f_1, ..., f_s$, so $G$ is still a basis for $I$.

The algorithm terminates when $G = G'$, i.e., when $S = \overline{S(p,q)}^{G'} = 0$ for all $p, q \in G$. Hence $G$ is a Gröbner basis following from Theorem 2.2.18.

It remains to prove that the algorithm terminates. In each round of the main loop, we have $\langle LT(G')\rangle \subseteq \langle LT(G)\rangle$ since $G' \subseteq G$. Furthermore, if $G' \neq G$ then $\langle LT(G')\rangle$ is strictly smaller than $\langle LT(G)\rangle$, since any nonzero remainder $r$ of an S-polynomial on division by $G'$ would satisfy $LT(r) \notin LT(G')$.

Hence, the ideals $\langle LT(G')\rangle$ from the successive iterations of the main loop form an ascending chain of ideals in $k[x_1, ..., x_n]$. Thus by the ascending chain

condition in Theorem 2.2.11 implies that after a finite number of iterations the chain has to stabilize. Hence the algorithm terminates. □

**Lemma 2.2.20.** *Let $G$ be a Gröbner basis for the polynomial ideal $I$. Let $p \in G$ be a polynomial such that $LT(p) \in \langle G - \{p\}\rangle$. Then $G - \{p\}$ is still a Gröbner basis for $I$.*

*Proof.* Note that

$$\langle LT(G - \{p\})\rangle = \langle LT(G)\rangle = \langle LT(I)\rangle.$$

Hence $G - \{p\}$ is still a Gröbner basis for $I$. □

**Definition 2.2.21.** *A* minimal Gröbner basis *for an ideal $I$ is a Gröbner basis $G$ for $I$ satisfying that:*
*(1) $LC(p) = 1$ for all $p \in G$.*
*(2) For all $p \in G$, $LT(p) \notin \langle LT(G - \{p\})\rangle$.*

It is easy to obtain a minimal Gröbner basis by deleting all the polynomials whose leading term is subsumed by leading terms of other polynomials. But if we want a canonical representation of the Gröbner bases, we need further require that a basis is "reduced".

**Definition 2.2.22.** *A* reduced Gröbner basis *for a polynomial ideal $I$ is a Gröbner basis $G$ for $I$ such that:*
*(1) $LC(p) = 1$ for all $p \in G$;*
*(2) For all $p \in G$, no monomial of $p$ is contained in $\langle LT(G - \{p\})\rangle$.*

**Theorem 2.2.23.** *Let $I \neq \{0\}$ be a polynomial ideal. Then $I$ has a unique reduced Gröbner basis under a fixed monomial order.*

*Proof.* Let $G$ be a minimal Gröbner basis for $I$. We say that $g \in G$ is reduced for $G$ if no monomials in $g$ is contained in $\langle LT(G - \{g\})\rangle$.

Given $g \in G$, let $g' = \overline{g}^{G-\{g\}}$ and set $G' = (G - \{g\}) \cup \{g'\}$. By minimality of $G$, $LT(g)$ is not divisible by $LT(G - \{g\})$. Hence $LT(g') = LT(g)$ and $G'$ is still a minimal Gröbner basis. Also, $g'$ is reduced for $G'$ by construction. Hence we can just apply the procedure and obtain a reduced Gröbner basis.

To prove uniqueness, suppose that $G$ and $G'$ are reduced Gröbner bases for $I$. Suppose there exists $x^\alpha \in LT(G)/LT(G')$. $x^\alpha$ has to be contained in $\langle LT(G')\rangle$, which means there exists $x^{\alpha'} \in \langle LT(G')\rangle$ that divides $x^\alpha$. Fix $\alpha_0$ to be the minimal such $\alpha'$ then $x^{\alpha_0}$ has to be contained in $LT(G)$, too. Thus $x^\alpha$ is subsumed by $LT(G) - x^\alpha$ which contradicts the minimality of $G'$. Hence

$$LT(G) = LT(G').$$

Thus given $g \in G$, there always exists $g' \in G'$ such that $LT(g) = LT(g')$. We only need to show that $g = g'$.

Consider $g - g'$, which is contained in $I$ and we must have $\overline{g - g'}^G = 0$. Since $LT(g) = LT(g')$, $LT(g - g') \neq LT(g)$. But we know that both $G$ and $G'$ are reduced, hence $LT(g - g')$ can not be reduced by $LT(G)$ or $LT(G')$. This shows that $g - g' = \overline{g - g'}^G = 0$. Hence $G = G'$. □

## 2.3 Hilbert's Nullstellensatz

In this section we state the proof of Hilbert's original Nullstellensatz.

**Definition 2.3.1.** *Let $B$ be a subring of $A$. $A$ is* finitely generated *over $B$, if there are finitely many $a_1, ..., a_n \in A$ such that $A = B[a_1, ..., a_n]$. $A$ is called a* finite B-algebra *if there are $a_1, ..., a_n$ satisfying $A = \sum_i Ba_i$.*

**Lemma 2.3.2.** *Let $C \subseteq B \subseteq A$ be rings, then*

*(1) If $B$ is a finite $C$-algebra and $A$ is a finite $B$-algebra then $A$ is also a finite $C$-algebra.*

*(2) If $A$ is a finite $B$-algebra then $A$ is integral over $B$, i.e., every element $x \in A$ satisfies an equation of the form*

$$x^n + b_{n-1}x^{n-1} + ... + b_1x + b_0.$$

*(3) If $x \in A$ satisfies an equation of the above form then $B[x]$ is a finite $B$-algebra.*

*Proof.* (1) Since $B = \sum_i Cb_i$ and $A = \sum_i Ba_i$, we have $A = \sum_{i,j} Ca_ib_j$.

(2) Suppose $A = \sum_i Ba_i$, then for any $x \in A$ we have $xa_i \in A$ for $i = 1, ..., n$. Thus there are elements $b_{ij} \in B$ such that

$$xa_i = \sum_{j=1}^{n} b_{ij}a_j.$$

To obtain a single polynomial equation for $x$ from these $n$ linear equations, we express this in matrix notation, and take the determinant of the matix. Define a matrix $M$, given by

$$M = (x\delta_{ij} - b_{ij})_{i,j}$$

(where $\delta_{i,j} = 1$ when $i = j$, and 0 otherwise). Then we have $Ma = 0$ where $a^T = (a_1, ..., a_n)$. Let $M^{adj}$ be the adjoint matrix of $M$, we have

$\det Ma = M^{adj}Ma = 0$ and thus $\det Ma_i = 0$ for $i = 1, ..., n$. Since $A$ is generated by $a_i$ from $B$, we have $\det M = 0$. But expanding the determinant we have

$$\det M = x^n + b_{n-1}x^{n-1} + ... + b_0, b_i \in B.$$

Thus $x$ satisfies a polynomial equation with coefficients in $B$.

(3) We have $B[x] = B + Bx + ... + Bx^{n-1}$, hence $B[x]$ is a finite $B$-algebra. □

**Lemma 2.3.3.** *let $A$ be a field and $B \subseteq A$ its subring such that $A$ is a finite $B$-algebra, then $B$ is also a field.*

*Proof.* Let $b \in B$ be a nonzero element. Since A is a field, there exists $b^{-1} \in A$. Lemma 2.3.2(2) shows that

$$b^{-n} + b_{n-1}b^{-(n-1)} + ... + b_1b^{-1} + b_0 = 0$$

where $b_i \in B$. Hence we have

$$b^{-1} = -(b_{n-1} + b_{n-2}b + ... + b_0b^{n-1}) \in B.$$

□

**Lemma 2.3.4.** *Let $f \in k[x_1, ..., x_n]$ be a nonzero element with $\deg f = d$. Then there exists a change of variables $x_i' = x_i - \alpha_i x_n$ for $1 \leq i \leq n-1$ where $\alpha_1, ..., \alpha_{n-1} \in k$ such that the polynomial $f(x_1' + \alpha_1 x_n, ...., x_{n-1}' + \alpha_n x_n, x_n) \in k[x_1', ..., x_{n-1}', x_n]$ has a term $cx_n^d$ for some nonzero $c \in k$.*

*Proof.* Suppose we have chosen some $\alpha_1, ..., \alpha_{n-1}$. We can write $f = F_d + G$ where $F_d$ is a homogeneous polynomial and $\deg G \leq d - 1$. Then

$$f(x_1' + \alpha_1 x_n, ..., x_{n-1}' + \alpha_{n-1}x_n, x_n) = F_d(\alpha_1, ..., \alpha_{n-1}, 1)x_n^d + G'.$$

Since $k$ is infinite, $F_d(\alpha_1, ..., \alpha_{n-1}, 1)$ can not vanish on all the points in $k^{n-1}$. Hence we can always pick $\alpha_1, ..., \alpha_{n-1} \in k$ such that $F_d(\alpha_1, ..., \alpha_{n-1}, 1) \neq 0$. □

**Theorem 2.3.5** (Noether Normalization)**.** *Let $k$ be an infinite field and $A = k[a_1, ..., a_n]$ be a finitely generated $k$-algebra. Then there exists $y_1, ..., y_m \in A$ with $m \leq n$ such that $y_1, ..., y_m$ are algebraically independent over $k$ and $A$ is a finite $k[y_1, ..., y_m]$-algebra.*

*Proof.* We do induction on $n$. Let $k[x_1, ..., x_n]$ be the polynomial ring over $k$ in $n$ variables. Let $\varphi : k[x_1, ..., x_n] \to k[a_1, ..., a_n]$ be the homomorphism induced by $x_i \to a_i$.

If $ker(\varphi) = \{0\}$ then we take $m = n$ and $y_1 = a_1, ..., y_n = a_n$ and the conclusion is immediately true.

Otherwise, there exists some nonzero $f \in I$. When $n = 1$, we have $f(a_1) = 0$ and following Lemma 2.3.2(3), $m = 0$ and the conclusion follows.

When $n > 1$, assume that the result is true for $n-1$. From Lemma 2.3.4 we know that there exist $\alpha_1, ..., \alpha_{n-1} \in k$ such that setting $\alpha_i' = a_i - \alpha_i a_n$ and $A' = k[a_1', ..., a_{n-1}'] \subseteq A$, we have that for some nonzero constant $c \in k$ the polynomial

$$F(x_n) = \frac{1}{c} f(a_1' + \alpha_1 x_n, ..., a_{n-1}' + \alpha_{n-1} x_n, x_n)$$

is a monic polynomial in $A'[x_n]$ and $F(a_n) = 0$. By Lemma 2.3.2(3) this implies that $a_n$ is integral over $A'$. By inductive hypothesis, there exist $y_1, ..., y_m \in A'$ such that $y_1, ..., y_m$ are algebraically independent over $k$ and $A'$ is a finite $k[y_1, ..., y_m]-$algebra. Now by Lemma 2.3.2(3) we have that $A = A'[a_n]$ is a finite $A'$-algebra and by 2.3.2(1) we have that A is also a finite $k[y_1, ..., y_m]$-algebra.

□

**Theorem 2.3.6.** *Let $k$ be an infinite field and $A = k[a_1, ..., a_n]$ a finitely generated $k$-algebra. If $A$ is a field then $A$ is algebraic over $k$.*

*Proof.* By Noether Normalization, we have $y_1, ..., y_m \in A$ for some $m \leq n$ such that $B = k[y_1, ..., y_m] \subseteq A$ and $A$ is a finite $B$-algebra. By Lemma 2.3.3, we know $B$ is a field. But this can only be true when $m = 0$. Hence $A$ is a finite extension of $k$, which means it is algebraic over $k$. □

**Theorem 2.3.7.** *Let $k$ be an algebraically closed field and $A = k[x_1, ..., x_n]$. Every maximal ideal $m \subseteq A$ is of the form $m = \langle x_1 - a_1, ..., x_n - a_n \rangle$ where $(a_1, ..., a_n) \in A_k^n$.*

*Proof.* Any ideal of the form $\langle x_1 - a_1, ..., x_n - a_n \rangle$ is maximal: Consider the evaluation homomorphism $\varphi : k[x_1, ..., x_n] \to k, \varphi(f) = f(a_1, ..., a_n)$. By linear transformation we may assume $a_i$ are all zero. Then $\varphi$ just maps each polynomial to its constant term, and the kernel is the set of polynomials with zero constant, which are precisely contained in the ideal $m$. Hence $k[x_1, ..., x_n]/m \cong k$.

Conversely, let $m \subseteq k[x_1, ..., x_n]$ be any maximal ideal. Then $k[x_1, ..., x_n]/m$ is a field as well as a finitely generated k-algebra. Following Theorem

2.3.6, $k[x_1, ..., x_n]/m$ is algebraic over $k$. But since $k$ is algebraically closed, $k[x_1, ..., x_n]/m$ is isomorphic to $k$. let $\varphi$ be the isomorphism between them. Let $b_i = x_i \mod m \in k[x_1, ..., x_n]/m$ and $a_i = \varphi^{-1}$, then we have

$$x_i - a_i \in ker\varphi = m,$$

hence

$$\langle x_1 - a_1, ..., x_n - a_n \rangle \subseteq m.$$

But we already know that $\langle x_1 - a_1, ..., x_n - a_n \rangle$ is maximal. Thus $\langle x_1 - a_1, ..., x_n - a_n \rangle = m$. □

**Theorem 2.3.8** (Weak Nullstellensatz)**.** *Let $k$ be any algebraically closed field. A system of polynomials $f_1, ..., f_m \in k[x_1, ..., x_n]$ have no common zero over $k$ if and only if $1 \in J$.*

*Proof.* Left to right:

Suppose $1 \notin J$, which means $J$ is a proper ideal of $k[x_1, ..., x_n]$ and is contained in some maximal ideal $J'$. By Theorem 2.3.7, $J'$ must be of the form $(x_1 - a_1, ..., x_n - a_n)$ for some $\bar{a} \in k^n$. Thus $V(J) \subseteq V(J') = \{\bar{a}\}$. That is, $f_1, ..., f_m$ has a common zero $\bar{a}$. Contradiction.

Right to left:

$1 \in J$ then there exists $g_1, ..., g_m \in k[x_1, ..., x_n]$ such that

$$1 = f_1 g_1 + ... + f_m g_m.$$

If $f_1, ..., f_m$ have a common zero then evaluating the equation on $\bar{a}$ we have the contradiction $1 = 0$.

□

We now prove the generalized version of the weak Nullstellensatz, with a view from Gröbner bases.

**Theorem 2.3.9** (Generalized Weak Nullstellensatz)**.** *Let $k$ be any field. A system of polynomials $f_1, ..., f_m \in k[x_1, ..., x_n]$ has no common zero over the algebraic closure of $k$, if and only if, $1 \in \langle f_1, ..., f_m \rangle$.*

*Proof.* Right to left:

Easy. If $1 \in \langle f_1, ..., f_m \rangle$ then $1 = \sum f_i g_i$ which implies $f_i$ can't have common zeros to make $1 = 0$.

Left to right:

Suppose $f_1, ..., f_m$ have no common zero over $k^a$. Consider $f_1, ..., f_m$ as polynomials in $k^a[x_1, ..., x_n]$, and directly apply Hilbert's weak Nullstellensatz. We have $1 \in \langle f_1, ..., f_m \rangle$.

Now consider the ideal $\langle f_1, ..., f_m \rangle$ as a subset of $k^a[x_1, ..., x_n]$, then it has a finite Gröbner basis $G$ satisfying that $1 \in G$. Consider the algorithm of Gröbner basis. Since $f_1, ..., f_m$ contain only coefficients in $k$, all the arithmetic operations involved in the Gröbner basis computation can only generate coefficients from $k$. In other words, the Gröbner basis of $\langle f_1, ..., f_m \rangle$ remains the same over any field extension of $k$. Thus $1 \in \langle f_1, ..., f_m \rangle$, which is now considered as a subset of $k[x_1, ..., x_n]$. □

Now it's not hard to prove the generalized strong Nullstellensatz using the Robinowitsch trick.

**Theorem 2.3.10** (Generalized Strong Nullstellensatz)**.** *Let $J$ be any ideal in $k[x_1, ..., x_n]$. Let $f$ be a polynomial in $k[x_1, ..., x_n]$ such that $f(\bar{a}) = 0$ for every zero $\bar{a}$ of $J$ over $k^a$. Then there exists an integer $m > 0$ such that $f^m \in J$.*

*Proof.* We introduce a new variable $y$ and consider the ideal $J' = J + \langle fy - 1 \rangle \subset k[x_1, ..., x_n, y]$.

Notice that since $f$ vanishes on all the zeros of $J$, $yf - 1$ does not vanish any of these zeros. Hence the new ideal $J'$ can not vanish on any points. Now we call the generalized weak Nullstellensatz which we just proved, then:

$$1 = g_0(fy - 1) + \sum_i g_i h_i \quad (\star)$$

where $h_i \in J$ and $g_i \in k[x_1, ..., x_n, y]$. Since the new indeterminant only appears in $g_i$, we multiply a sufficient power of $f$, say $f^m$, to both sides of the equation, such that every $y$ appearing in $g_i$ is paired with an $f$, and get:

$$f^m = g_0(1 - fy) + \sum_i (g_i f^m) \cdot h_i$$

Now consider the equation modulo the principle ideal $\langle fy - 1 \rangle$, we have

$$f^m = \sum_i g_i' \cdot h_i \ mod \ \langle fy - 1 \rangle$$

where each appearance of $yf$ in $g_i$ is substituted by 1; hence on both sides of the equation, only polynomials in $k[x_1, ..., x_n]$ can appear. But that means both sides can't be reduced by $fy - 1$ (since they are "coefficients" for $y$), and we get

$$f^m = \sum_i g_i' \cdot h_i.$$

Hence $f \in \sqrt{J}$. □

# Chapter 3

# Counting with Gröbner Bases

In this section, we will prove the following theorems to show how Gröbner basis computation can be used to solve #SAT. We write $F_q$ for a finite field of size $q$, and $F_q[x_1, ..., x_n]$ for the $n$-variate polynomial ring over $F_q$. First, we show that any ideal $J \subseteq F_q[x_1, ..., x_n]$ can be easily turned into a radical ideal $J'$ by adding polynomials of the form $x_i^q - x_i$, without changing the zero set $V(J)$ over $F_q$ (Lemma 4.1). This implies the strong Nullstellensatz over finite fields (Theorem 4.1), which generalizes a result in [15]. Next, we show that the strong Nullstellensatz guarantees that computing the Gröbner basis of $J'$ suffices for determining the size of $V(J)$. In fact, the number of monomials that are not reducible by the leading monomials in the Gröbner basis of $J'$ (Definition 4.1) is exactly the number of zeros of $J$ (Lemma 4.2, Lemma 4.3 and Theorem 4.3).

## 3.1 Nullstellensatz in Finite Fields

**Lemma 3.1.1.** *For any ideal $J \subseteq F_q[x_1, ..., x_n]$, $J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$ is radical.*

*Proof.* We need to show

$$\sqrt{J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle} = J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle.$$

Since by definition, any ideal is contained in its radical, we only need to prove

$$\sqrt{J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle} \subseteq J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle.$$

For brevity we write $R = F_q[x_1, ..., x_n]$ and $S = \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$.

Consider an arbitrary polynomial $f \in \sqrt{J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle}$. By definition, for some integer $s$, $f^s \in J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$. Let $[f]$ and $[J]$ be the images of, respectively, $f$ and $J$, in $R/\langle x_1^q - x_1, ..., x_n^q - x_n \rangle$ under the canonical homomorphism from $R$ to $R/\langle x_1^q - x_1, ..., x_n^q - x_n \rangle$. Now we have $[f]^s \in [J]$, and we only need to have $[f] \in [J]$. We prove, by induction on the structure of polynomials, that for any $[g] \in R/S$, $[g]^q = [g]$.

If $[g] = cx_1^{a_1} \cdots x_n^{a_n} + S$ $(c \in F_q, a_i \in N)$, then

$$[g]^q = (cx_1^{a_1} \cdots x_n^{a_n} + S)^q = (cx_1^{a_1} \cdots x_n^{a_n})^q + S = cx_1^{a_1} \cdots x_n^{a_n} + S = [g].$$

If $[g] = [h_1] + [h_2]$, by inductive hypothesis, $[h_1]^q = [h_1], [h_2]^q = [h_2]$, and, since any element divisible by $p$ is zero in $F_q$ $(q = p^r)$, then

$$[g]^q = ([h_1] + [h_2])^q = \sum_{i=0}^{q} \binom{q}{i} [h_1]^i [h_2]^{q-i} = [h_1]^q + [h_2]^q = [h_1] + [h_2] = [g]$$

Hence $[g]^q = [g]$ for any $[g] \in R/S$, without loss of generality we can assume $s < q$ in $[f]^s$. Then, since $[f]^s \in [J]$,

$$[f]^s \cdot [f]^{q-s} = [f]^q = [f] \in [J].$$

□

A nice form of the strong Nullstellensatz for finite fields, which generalizes the result in [15] from prime fields to arbitrary finite fields, follows from this observation.

**Theorem 3.1.2** (Strong Nullstellensatz in Finite Fields)**.** *For an arbitrary finite field $F_q$, let $J \subseteq F_q[x_1, ..., x_n]$ be an ideal, then*

$$I(V(J)) = J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle.$$

*Proof.* For an arbitrary ideal $J \subseteq F_q[x_1, ..., x_n]$, applying Hilbert's Nullstellensatz to $J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$ and using the previous lemma, we have:

$$I(V^a(J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle)) = J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$$

But since $V^a(\langle x_1^q - x_1, ..., x_n^q - x_n \rangle) = F_q^n$,

$$V^a(J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle) = V^a(J) \cap F_q^n = V(J)$$

and we reach the clean form,

$$I(V(J)) = J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle.$$

□

As an easy corollary we have:

**Theorem 3.1.3** (Weak Nullstellensatz in Finite Fields)**.** *For an arbitrary finite field $F_q$, given $m$ polynomials $f_1, ..., f_m \in F_q[x_1, ..., x_n]$, $f_1, ..., f_m$ have no common zero in $F_q^n$ if and only if $1 \in \langle f_1, ..., f_m, x_1^q - x_1, ..., x_n^q - x_n \rangle \subseteq F_q[x_1, ..., x_n]$.*

*Proof.* Right to left is easy. Left to right: If $f_1, ..., f_m$ have no common zero then

$$V(\langle f_1, ..., f_m \rangle) = \emptyset.$$

Hence

$$\langle f_1, ..., f_m, x_1^q - x_1, ..., x_n^q - x_n \rangle = I(\emptyset) = \langle 1 \rangle.$$

□

## 3.2 $|SM(J + \langle \bar{x}^q - \bar{x} \rangle)| = |V(J)|$

Next we prove that the size of the variety over $F_q$ of any ideal can be obtained from its Gröbner basis. Write $Span(\{f_1, ..., f_m\})$ to denote the vector space formed by linear combinations of $f_1, ..., f_m$ with scalars from the base field $F_q$. We will keep using the notation $R = F_q[x_1, ..., x_n]$. The method is standard for algebraically closed fields [**?**, 9], we will make sure that it works for finite fields.

**Definition 3.2.1** (Standard Monomials)**.** *The set of* standard monomials *of any ideal $J$ is defined as:*

$$SM(J) = \{x^\alpha : x^\alpha \notin \langle LM(J) \rangle\}.$$

When an ideal $J$ has a Gröbner basis $G$, we also write the standard monomial set of $J$ as $SM(G)$, and call it the standard monomial set of $G$.

**Lemma 3.2.2.** *Let $J$ be any ideal in $R$, we have $Span(SM(J)) \cong R/J$.*

*Proof.* Let $[f]$ denote the image of $f$ under the canonical homomorphism $R \to R/J$. Since $Span(SM(J)) \subseteq R$, we restrict the canonical homomorphism to $Span(SM(J))$ and get $\phi : Span(SM(J)) \to R/J$, such that $\phi(f) = [f]$.

Since $\phi$ is already a homomorphism, we only need to show that it is bijective.

1. $\phi$ is injective: Take $f, g \in Span(SM(J))$, $f \neq g$. Then $f - g \in Span(SM(J))$. Since, by the definition of standard monomials, none of the monomials appearing in $f - g$ can be divided by any monomial in $LM(J)$, $f - g$ cannot be reduced by $J$. Thus, $f - g = 0$ if and only if $[f - g] = [0]$. Hence $\phi(f) - \phi(g) \neq [0]$.

2. $\phi$ is surjective: For any $[f] \in R/J$, suppose $[f] = f' + J$ where $f'$ cannot be reduced by $J$, then $f' \in Span(SM(J))$ and $\phi(f') = [f]$.

In all, $Span(SM(J)) \cong R/J$. □ □

**Lemma 3.2.3.** *Let $J = I(V) \subseteq R$ for some $V = \{P_1, ..., P_m\} \subseteq F_q^n$. Consider $R/J$ as a finite dimensional vector space over $F_q$, then we have* $\dim R/J = m$.

*Proof.* For any point $P \in F_q^n$, we write its $i$-th coordinate as $(P)_i$.

1. When $m = 1$: It is easy to see that $J$ vanishes on a single point $P$ if and only if it is of the form $\langle x_1 - (P)_1, ..., x_n - (P)_n \rangle$:

   Any polynomial $f \in \langle x_1 - (P)_1, ..., x_n - (P)_n \rangle$ naturally vanishes on P. On the other hand, for any polynomial $f \in J$, divided by $x_i - (P)_i$ for all $i \in \{1, ..., n\}$, we have

   $$f = \sum_{i=1}^{n} h_i \cdot (x_i - (P)_i) + r,$$

   where $h_i \in R$ and $r \in F_q$. Since $f(P) = 0$, $r = 0$. That is,

   $$f \in \langle x_1 - (P)_1, ..., x_n - (P)_n \rangle.$$

   Hence $\dim R/J = 1$.

2. When $m > 1$: Consider $P_1 \in V$. Suppose, for any $P_j, j \in 2, ..., m$, $P_1$ and $P_j$ differ at the $i$-th coordinate. We construct

   $$g_j(\vec{x}) = ((P_1)_i - (P_j)_i)^{-1}(x_i - (P_j)_i)$$

   and let

   $$f_1(\vec{x}) = \prod_{j=2}^{n} g_j(\vec{x}).$$

   Then we have

   $$f_1(P_1) = 1, f_1(P_2) = ... = f_1(P_m) = 0.$$

Let $\{f_1, ..., f_m\}$ be the set of such functions defined for each point in $V$.

We show that $\{[f_1], ..., [f_m]\}$ is a basis of $R/J$:

(a) Independence:
Suppose for some $a_i \in F_q, i \in \{1, ..., m\}$,

$$a_1[f_1] + ... + a_m[f_m] = [0].$$

It follows that

$$a_1 f_1 + ... + a_m f_m \in J.$$

That is, for $j \in \{1, ..., m\}$,

$$a_1 f_1(P_j) + ... + a_m f_m(P_j) = 0.$$

Since $f_j(P_j) = 1$, and $f_j(P_i) = 0$ when $i \neq j$, we have

$$a_1 f_1(P_j) + ... + a_m f_m(P_j) = a_j f_j(P_j) = a_j.$$

Hence $a_j = 0$ for all $j \in \{1, ..., m\}$.

(b) Spanning:
Consider any $[h] \in R/J$. Let $q_i = h(P_i) \in F_q$, then we have

$$h - (q_1 f_1 + ... + q_m f_m) \in J.$$

Hence

$$[h] - (q_1[f_1] + ... + q_m[f_m]) = [0].$$

That is,

$$R/J = Span(f_1, ..., f_m).$$

In all, $\dim R/J = m$. □

**Theorem 3.2.4** (Counting with Gröbner bases)**.** *Let $J \subseteq R$ be any ideal and $G = \{g_1, ..., g_s\}$ be the Gröbner basis of $J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$. Suppose $|SM(G)| = m$, then $J$ vanishes on exactly $m$ distinct points in $F_q^n$.*

*Proof.* By the strong Nullstellensatz in Theorem 4.1, we have

$$I(V(J)) = \langle g_1, ..., g_s \rangle.$$

Then, using Lemma 4.2,

$$|SM(G)| = \dim(R/\langle g_1, ..., g_s \rangle).$$

Finally, Lemma 4.3 ensures that

$$\dim(R/I(V(J))) = \dim(R/\langle g_1, ..., g_s\rangle) = |V(J)|.$$

Hence

$$|V(J)| = |SM(G)| = m.$$

□

# Chapter 4

# Algorithm Analysis

## 4.1 Analysis of Buchberger's Algorithm

Gröbner bases are well-known to be very expensive to compute. It was shown to be at least EXPSPACE-hard in general [22], requiring time doubly exponential in the number of variables. Nevertheless, given the special structure of our problem, we show that Buchberger's Algorithm halts in single exponential time, and is hence reasonable to be used for solving #SAT.

**Algorithm** ***Counting Zeros with Buchberger's Algorithm***
**Input:** $f_1, ..., f_m, x_1^q - x_1, ..., x_n^q - x_n \in F_q[x_1, ..., x_n]$ and a monomial ordering $<_M$
**Output:** $N$ - the number of satisfying assignments of $\varphi$

1. $G \leftarrow \{f_1 - 1, ..., f_m - 1, x_1^2 - x_1, ..., x_n^2 - x_n\}$
2. Preprocess G so that $f_1, ..., f_m$ are reduced with respect to $x_i^q - x_i$ and have no duplicated leading monomials.
3. **repeat**
4. $\quad S \leftarrow \emptyset$
5. $\quad$ Order the polynomials in G as $(g_1, ..., g_t)$ in lexicographic order
6. $\quad$ **for** $1 \leq i < j \leq t$
7. $\quad\quad h \leftarrow \frac{LCM(LT(g_i), LT(g_j))}{LT(g_j)} g_i - \frac{LCM(LT(g_i), LT(g_j))}{LT(g_i)} g_j$
8. $\quad\quad r \leftarrow$the remainder after reducing $h$ with respect to $g_1, ..., g_t$
9. $\quad\quad$ **if** $r \neq 0$
10. $\quad\quad\quad$ **then** $S \leftarrow S \cup \{r\}$
11. $\quad\quad\quad\quad G \leftarrow G \cup S$
12. **until** $S = \emptyset$
13. $M_1, ..., M_t \leftarrow$leading monomials of the polynomials in $G$
14. $N \leftarrow$the number of monomials that are not divisible by $M_1, ..., M_t$

15. **return** N

**Proposition 4.1.1.** *For ideals of the form* $\langle f_1, ..., f_m, x_1^q - x_1, ..., x_n^q - x_n \rangle \subseteq F_q[x_1, ..., x_n]$*, where* $f_1, ..., f_m$ *are given in sparse form and have longest length of* $l$*, the basic Buchberger Algorithm halts in time* $q^{O(n)} + O(m^2 l)$*.*

*Proof.* At the beginning, we need to calculate the steps needed for preprocessing (Line 2). Preprocessing is not always necessary, and the reason for doing this is to avoid having some $f_i$ with degree higher than $q$ on some variable, as well as duplicated leading monomials in $LM(f_1), ..., LM(f_m)$. First, we divide each $f_i$ by the field polynomials, which can be easily done by replacing each occurence of $x_i^q$ by $x_i$, if there are any. This requires at most $O(ml)$ steps. Then we need to make sure all the duplicated leading monomials in $f_1, ..., f_m$ are deleted, as follows. After ordering the polynomials with respect to their leading monomials and $<_M$, we search for duplicated leading monomials. Suppose $f_i$ and $f_j$ have the same leading monomial, then we let $f_j'$ to be the result of subtracting $(LT(f_j)/LT(f_i))f_i$ from $f_j$, so that the leading monomial of $f_j$ is canceled. Now $f_j'$ is a new polynomial whose leading monomial is strictly smaller than that of $f_j$. We delete $f_j$ and put $f_j'$ back into the polynomial list. If $f_j'$ still has duplicated leading monomial with other polynomials down the list, we repeat the subtraction and inserting operations. Completing this process requires $O(ml)$ at most for each polynomial, and at most $O(m^2 l)$ for all the polynomials. In all, completing preprocessing requires $T_{preprocessing} = O(m^2 l)$.

Now we analyze the main loop (Line 3 to 18). Consider round $k+1$ of the main loop. Let $T_{k+1}$ be the time taken by this round and $L_{k+1}$ the size of $G$ after this round. We immediately have the following relation between $T_{k+1}$ and $L_k$:

$$T_{k+1} \leq \binom{L_k}{2} \cdot T_{Spoly} \cdot T_{reduce},$$

where $T_{Spoly}$ is the time taken by computing the S-polynomial from a pair of polynomials in $G$ (Line 7) and $T_{reduce}$ is the maximum time taken by reducing an S-polynomial with respect to polynomials in $G$ (Line 8).

The standard reduction process uses multivariate polynomial division [9, 13]. Since during the division process, after each division round, the leading monomial of the dividend polynomial strictly decreases with respect to the monomial ordering, the number of division rounds is bounded by the longest downward monomial chain leading from $LM(h)$ ($h$ as defined on Line 7) that can appear in the division. Now, the existence of the field polynomials $x_i^q - x_i$ in the divisor pool ensures two properties of the division process (Line 8):

First, after division, any variable appearing in the remainder must have degrees lower than $q$, otherwise the division should not halt. Hence the longest possible remainder is shorter than $q^n$, which is the maximum number of different monomials with degree on each variable lower than $q$. Notice that we have preprocessed the input polynomials to have no duplicated leading monomials, and the multivariate division procedure ensures that the remainder can only have a strictly smaller leading monomial compared to the dividend, which is different from all the leading monomials of the divisors. Hence, after each round the length of the basis (the number of polynomials in $G$) is always smaller than $q^n + n$, since we can at most have $q^n$ different monomials with degree on each variable smaller than $q$, plus $n$ extra field polynomials with degree $q$.

Second, when an S-polynomial or intermediate polynomial appearing during the division as dividend has a degree higher than $q$ on variable $x_i$, the degree is reduced to lower than $q$, through division by $x_i^q - x_i$. Since the length of any downward chain from the leading monomial of the dividend is at most $q^n$, and at each division round the reduction by $x_i^q - x_i$ may take up another step, we have the maximum number of division rounds bounded by $2q^n$.

Further, notice that when operating over a finite field, the positive characteristic ensures that the coefficients never grow exponentially as in the case for rationals or complex numbers. We can assume arithmetic among coefficients takes unit time. Each division round, say dividing $f$ by $g$, consists of multiplying $LT(f)/LT(g)$ to $g$, and subtracting $(LT(f)/LT(g)) \cdot g$ from $f$. Thus each round during the division process takes time at most $2q^n$, since $q^n$ is the longest possible length of $f$ and $g$.

Finally, S-polynomials are defined as (cf. [9]):

$$S(g_i, g_j) = \frac{LCM(LT(g_i), LT(g_j))}{LT(g_i)} \cdot g_i - \frac{LCM(LT(g_i), LT(g_j))}{LT(g_j)} \cdot g_j$$

Hence the number of arithmetic steps taken by computing S-polynomial, $T_{S-polynomial}$, is less than the sum of lengths of the two polynomials, which is also bounded by $2q^n$.

In all, we have $T_{reduce} \leq 2q^n \cdot 2q^n$, $L_k \leq q^n + n$ and $T_{Spoly} \leq 2q^n$. Hence

$$T_{k+1} \leq \binom{L_k}{2} \cdot T_{Spoly} \cdot T_{reduce} \leq (q^n + n)^2 \cdot 2q^n \cdot 2q^n \cdot 2q^n$$

Now observe that the final length of the Gröbner basis is bounded by $q^n + n$ (again, because we can have at most $q^n$ different leading monomials

appearing in $G$, and $n$ extra field polynomials). At each round at least one polynomial with a new leading monomial needs to be generated, otherwise the process halts. Hence the maximum number of rounds is bounded by $(q^n+n)-(m+n)$. (We preprocessed the input polynomials $f_1,...,f_m$ to be mutually irreducible and already reduced by $x_i^q - x_i$, hence $m < q^n$.)

In all, the overall worst-case complexity of the Buchberger Algorithm is

$$\begin{aligned} T &\leq (q^n+n-(m+n)) \cdot T_k + T_{preprocessing} \\ &\leq q^n \cdot (q^n+n)^2 \cdot 2q^n \cdot 2q^n \cdot 2q^n + O(m^2 l) = q^{O(n)} + O(m^2 l). \end{aligned}$$

At last, since there are only $q^n$ possible different standard monomials, the worst-case complexity of the counting step is $O(q^n)$, subsumed by the complexity of Buchberger's Algorithm. We'll discuss practical heuristics for the counting step in the next section. Here we settle for the overall worst-case complexity, which is $q^{O(n)} + O(m^2 l)$. □

## 4.2 Counting Standard Monomials

**Proposition 4.2.1.** *For each $x_i$ there exists $d_i \in \{0,...,q\}$ such that $x_i^{d_i} \in LM(G)$. The set of standard monomials can be listed after enumerating at most $\prod_{i=1}^n d_i$ monomials.*

*Proof.* Write $J + \langle x_1^q - x_1, ..., x_n^q - x_n \rangle$ as $J'$. For each $1 \leq i \leq n$, consider the ideal $J_i = J' \cap F_q[x_i]$. Since $V^a(J') \subseteq V^a(\langle x_1^q - x_1, ..., x_n^q - x_n^q \rangle) = F_q^n$, $V^a(J')$ is finite. Hence $J_i$ is a nonempty ideal consisting of univariate polynomials, which must be a principal ideal and has a unique generator $g_i(x_i)$.

Since $g_i \in J_i \subseteq J'$, $LM(g_i) \in \langle LM(J') \rangle = \langle LM(G) \rangle$. By the definition of Gröbner basis, $LM(g_i)$ is reducible to 0 with respect $LM(G)$. Thus, there exists $x_i^{d_i} \in LM(G)$, $0 \leq d_i \leq q$, and $LM(g_i)$ is divisible by $x_i^{d_i}$.

Any monomial $x_1^{a_1} \cdots x_n^{a_n}$, with $a_i \geq d_i$ for some $i$, is divisible by ${x_i}^{d_i}$, and cannot appear in the standard monomial set. Thus the search for standard monomials halts after enumerating at most $\prod_{i=1}^n d_i$ monomials, which is bounded by $q^n$ and does not increase the overall worst-case complexity of our algorithm. □

In practice, $d_i$ is much smaller than $q$ for polynomial systems with sparse solutions, since $d_i$ must be smaller than the total number of standard monomials, which is the number of solutions as we just proved.

Here we mention the special case of bivariate polynomials. The number of standard monomials of any ideal can be explicitly formulated from the shape of its reduced Gröbner basis:

**Proposition 4.2.2.** *Let $J_{(x,y)} \subseteq F_q[x,y]$ be an ideal containing the field polynomials $x^q - x$ and $y^q - y$. Suppose the set of leading monomials of the reduced Gröbner basis of $J_{(x,y)}$, when ordered with respect to the lexicographical ordering $x > y$, has the form:*

$$LM(G) = \{x^{a_0}, x^{a_1}y^{b_1}, ..., x^{a_k}y^{b_k}, y^{b_{k+1}}\}.$$

*Then the set of standard monomials of $G$ has size (set $a_{k+1} = 0$)*

$$|SM(G)| = \sum_{i=0}^{k} (a_i - a_{i+1}) \cdot b_{i+1}.$$

*Proof.* Since $G$ is a reduced Gröbner basis, for any $i, j$, $x^{a_j}y^{b_j}$ should not divide $x^{a_i}y^{b_i}$ (if we started from an arbitrary Gröbner basis, we can first delete any monomial reducible by other monomials in the set of leading monomials). Thus, for any $i, j$, $0 \leq i < j \leq k+1$, we must have $a_i > a_j$ (because of the lexicographical ordering) and $b_i < b_j$.

Consider any monomial $x^s y^t$, with $a_{i+1} \leq s < a_i$ for some $0 \leq i \leq k$.

First, $x^s y^t$ cannot be divisible by any monomial $x^{a_l}y^{b_l}$ with $0 \leq l \leq i$, since $a_l \geq a_i > s$.

If $t < b_{i+1}$, then $x^s y^t$ is not divisible by monomials $x^{a_l}y^{b_l}$ with $l > i$ either, since $b_l \geq b_{i+1}$; thus $x^s y^t$ is not divisible by any monomial in $LM(G)$ and is a standard monomial.

On the contrary, if $t \geq b_{i+1}$ then it is divisible by $x^{a_{i+1}}y^{b_{i+1}}$.

Hence the standard monomial set can be explicitly written as (set $a_{k+1} = 0$):

$$SM(G) = \bigcup_{i=0}^{k} \{x^s y^t : a_{i+1} \leq s < a_i, 0 \leq t < b_{i+1}\}.$$

And $|SM(G)| = \sum_{i=0}^{k} (a_i - a_{i+1}) \cdot b_{i+1}$. □

# Chapter 5

# A Practical #SAT Solver

## 5.1 DPLL-based Approaches to #SAT

The first, as well as the most widely studied and used, algorithm for SAT is the DPLL algorithm [12]. The algorithm is a branch and search algorithm, whose search space can be easily represented as a binary tree, where each node represents the Boolean formula under a certain variable assignment. A formula is satisfiable if and only if there exists a leaf node evaluated to true. The DPLL algorithm can be presented as a recursive procedure:

**Algorithm *DPLL(formula, assignment)***

```
1.  necessary = deduction(formula, assignment)
2.  new_assignment = union(necessary, assignment)
3.  if is_satisfied(formula, new_assignment)
4.     then SATISFIABLE
5.     else  if is_conflicting(formula, new_assignment)
6.              then CONFLICT
7.                   branching_var = choose_free_variable(formula, new_assignment)
8.  assign1 = union(new_assignment, assign(branching_var, 1))
9.  result1 = DPLL(formula, assign1)
10. if result1 == SATISFIABLE
11.    then SATISFIABLE
12.    else  assign2 = union(new_assignment, assign(branching_var,0))
13.          DPLL(formula, assign2)
```

The top-level function DPLL is called with a CNF formula and a set of assignments. The function deduce() returns with necessary assignment forced by the formula (so that the formula is not directly falsified). If the

formula evaluates to 1 or 0, the recursion terminates with an answer SATISFIABLE or CONFLICT. Otherwise the a new variable will be branched upon, and the procedure is recursively called with the new assigments.

The basic idea of extending DPLL for model counting is quite straightforward. The solver can explore the complete search tree for an n-variable formula as in usual DPLL, pruning unsatisfiable branches based on falsified clauses and declaring a branch to be satisfied when all clauses have at least one true literal. When a branch is declared satisfied and the partial variable assignment at that point has t fixed variables (fixed either through the branching heuristic or by unit propagation), we associate $2^{n-t}$ solutions with this branch corresponding to the partial assignment being extended by all possible settings of the $n - t$ yet unassigned variables, backtrack to the last decision variable that can be flipped, flip that variable, and continue exploring the search space. The model count for the formula is finally computed as the sum of such $2^{n-t}$ counts obtained over all satisfied branches.

One heuristic used for avoiding exhaustive enumeration of all the paths is called component analysis [20, 26]. Suppose the constraint graph G of a CNF formula F can be partitioned into disjoint components where there is no edge connecting a vertex in one component to a vertex in another component, i.e., the variables of F corresponding to vertices in two different components do not appear together in any clause. Since the components are disjoint, it follows that every clause of F appears in a unique component, the sub-problems captured by the components are independent. Thus, #F can be evaluated by identifying disjoint components of F, computing the model count of each component, and multiplying the results together. Practically, components are identified dynamically as the underlying DPLL procedure attempts to extend a partial assignment. With each new extension, several clauses may be satisfied so that the constraint graph simplifies dynamically depending on the actual value assignments to variables.

Another existing approach for #SAT is called knowledge compilation, which is implemented as part of the c2d compiler [10, 11]. The idea is that Boolean formulas can be represented in a way that the number of solution is easily obtainable. In the c2d compiler, Boolean formulas are compiled into a representation form called d-DNNF (deterministic, decomposable negation normal form). The decomposability and determinism of d-DNNF allow the number of solutions of a formula, as well as many other queries, to be read-off easily once it is compiled. The c2d compiler was shown to outperform DPLL-based solvers in a number of benchmark [11].

As is explained in previous sections, approximate counting solvers use SAT solvers as oracles [**?**], and usually scale better than exact solvers, push-

ing the limit to instances with several thousand variables. This is reasonable, since in [**?**] it is proved that approximately solving #SAT can be done in $\mathsf{BPP}^{\mathsf{NP}}$. That is, with an $\mathsf{NP}$ oracle (such as a SAT solver), there exists a probabilistic polynomial-time algorithm that can approximate the number of solutions with an arbitrarily small error-probability. However, for the same reason the capability of approximate #SAT solvers is much lower than exact #SAT solvers. For example, by Toda's Theorem [30], any bounded Quantified Boolean Formula ($PH$) can be reduced to a #SAT instance in polynomial time, while approximate counting ($BPP^{NP}$) is much lower on the complexity hierarchy.

## 5.2 Gröbner Bases in Boolean Rings

As an alternative to DPLL, the method of Gröbner basis computation [5] has been investigated previously for solving SAT [8]. The basic idea is to apply the weak Nullstellensatz (Theorem 3.1.3). The Gröbner basis proof system [8] for SAT has been proved to be theoretically as efficient as systems based on resolution methods (which includes DPLL). However, because of the tremendous success of DPLL-based SAT solvers and the relatively early stage of optimization on Gröbner basis computation, in practice no SAT solver based on Gröbner bases has been used.

A propositional formula $\varphi(p_1, ..., p_n)$ with $n$ propositional variables can always be translated to a multivariate polynomial $f(x_1, ..., x_n)$ over the finite field $F_2 = \{0, 1\}$, in the sense that $\varphi$ and $f$ define the same function from $F_2^n$ to $F_2$. Note that in $F_2$ we have $1 + 1 = 0$, hence addition behaves like "xor", and multiplication behaves like "and". Concretely, the translation function $\tau$ between the set of formulas to the polynomial ring $F_2[x_1, ..., x_n]$ can be defined as:

$$\begin{aligned}
\tau(p_i) &= x_i; \\
\tau(\neg\varphi) &= \tau(\varphi) + 1; \\
\tau(\varphi_1 \wedge \varphi_2) &= \tau(\varphi_1) \cdot \tau(\varphi_2); \\
\tau(\varphi_1 \vee \varphi_2) &= \tau(\neg(\neg\varphi_1 \wedge \neg\varphi_2)) = 1 + (1 + \tau(\varphi_1)) \cdot (1 + \tau(\varphi_2)) \\
&= \tau(\varphi_1) \cdot \tau(\varphi_2) + \tau(\varphi_1) + \tau(\varphi_2).
\end{aligned}$$

A conceivable problem with the translation is that, when the polynomials are expanded, an exponential number of monomials (products of variables) may occur. But if the Boolean formulas are in CNF form, say $\varphi = \bigwedge_{i=1}^{m} C_i$, we can translate each clause $C_i$ into a single polynomial $f_i$, and obtain a

*polynomial system* $\{f_1, ..., f_m\}$. In this way, for a $k$-CNF formula where each clause contains at most $k$ literals, the size of each polynomial is bounded by a constant, and hence the overall length of the polynomial system is linear in the size of the original formula. Following the translation rules, it is easy to see that $\varphi$ is satisfiable if and only if there exists $\bar{a} \in F_2^n$ such that $\bigwedge_{i=1}^m f_i(\bar{a}) = 1$. We call this $\bar{a}$ a *zero* of the polynomial system $\{f_1 - 1, ..., f_m - 1\}$.

Our implementation of the counting algorithm is based on PolyBori [4], which is an efficient Gröbner basis computation package specifically designed for Boolean rings. A distinguished feature of PolyBori is its use of BDD as the data structure of storing the polynomials, which significantly reduces the space cost [4].

## 5.3 Experimental Results

We implemented our algorithm as a #SAT solver in the computer algebra environment SAGE based on PolyBori [4], a package for efficient Gröbner basis computation in Boolean rings.[1]

We compared our solver with the other exact #SAT solvers on benchmarks of several different types, as listed in Table 1. All tests were performed on a laptop machine with 1.46GHz Intel Celeron CPU and 2GB memory running 32-bit Linux (c2d was ran in windows on the same machine). The time-out limit is set as 15 minutes. We write "T" when a solver fails in the time limit, and "NA" when no answer is returned (in relsat, there is a limit on the number of solutions).

The first two sets of benchmarks (urquhart, xor) are from "small but difficult" benchmarks collected in [**?**]. The instances suffixed with "-sat" are the ones with the last constraint deleted from the original instance. We use them to test whether GBsolver can efficiently count a large number of solutions. It can be seen that while DPLL-based solvers scale exponentially on these benchmarks, both Gröbner basis solver (GBsolver) and c2d performed well.

The next two sets (rand3bip-sat, pmg) are from hard benchmarks used in 2007 SAT competitions [**?**, **?**]. In the "rand3bip-sat" set, GBsolver instantly solves all formulas while all the other solvers fail to terminate. In the "pmg" set, GBsolver easily handles the unsatisfiable ones, but runs into trouble when the instances become satisfiable.

[1]The code and benchmark can be found at http://www.cs.cmu.edu/ sicung/sharp-sat.html

Table 5.1: Performance comparison with Relsat, Cachet, sharpSAT, and c2d on selected benchmarks

| Benchmark | #Vars | #Cls | #Sols | **GBsolver** | Relsat | Cachet | sharpSAT | c2d |
|---|---|---|---|---|---|---|---|---|
| urquhart | | | | | | | | |
| 2-25bis | 36 | 96 | 0 | **0.07** | < 1 | 0.08 | 0.26 | 0.27 |
| 2-25bis-sat | 36 | 95 | 2048 | **0.16** | NA | 0.28 | 0.35 | 0.25 |
| 2-25 | 60 | 160 | 0 | **0.12** | 7 | 1.37 | 86.72 | 0.47 |
| 2-25-sat | 60 | 159 | 524288 | **0.34** | NA | 1.06 | 86.97 | 0.44 |
| 3-25bis | 99 | 264 | 0 | **0.38** | T | T | T | 5.87 |
| 3-25bis-sat | 99 | 263 | 4.29e+09 | **2.81** | T | T | T | 2.91 |
| xor | | | | | | | | |
| X1.1-16 | 46 | 122 | 0 | **0.04** | < 1 | 0.30 | 0.17 | 0.36 |
| X1.1-16-sat | 46 | 121 | 16384 | **0.18** | NA | 0.22 | 0.26 | 0.41 |
| X1.1-24 | 70 | 186 | 0 | **0.06** | 63 | 4.12 | 42.51 | 0.56 |
| X1.1-24-sat | 70 | 185 | 4.19e+06 | **0.53** | NA | 9.58 | 37.49 | 0.78 |
| X1.1-32 | 94 | 250 | 0 | **0.08** | T | 42.46 | T | 1.22 |
| X1.1-32-sat | 94 | 249 | 1.07e+09 | 1.93 | NA | 177.23 | T | **1.81** |
| X1.1-36 | 106 | 282 | 0 | **0.1** | T | 249.92 | T | 5.78 |
| X1.1-36-sat | 106 | 281 | 1.72e+10 | **2.42** | T | 179.18 | T | 2.69 |
| rand3bip | | | | | | | | |
| 200-3 | 200 | 800 | 1 | **0.88** | T | T | T | T |
| 200-3-sat | 200 | 879 | 2 | **0.90** | T | T | T | T |
| 220-4 | 220 | 880 | 1 | **0.94** | T | T | T | T |
| 220-4-sat | 220 | 879 | 2 | **1.03** | T | T | T | T |
| 250-1 | 250 | 1000 | 1 | **1.10** | T | T | T | T |
| 250-1-sat | 250 | 999 | 1 | **1.17** | T | T | T | T |
| 300-15 | 300 | 1200 | 1 | **7.39** | T | T | T | T |
| 300-15-sat | 300 | 1199 | 1 | **6.82** | T | T | T | T |
| pmg | | | | | | | | |
| Pmg-12 | 190 | 632 | 0 | **0.22** | T | T | T | T |
| Pmg-12-sat | 190 | 631 | NA | T | T | T | T | T |
| Pmg-13 | 409 | 1362 | 0 | **6.21** | T | T | T | T |
| Pmg-13-sat | 409 | 1361 | NA | T | T | T | T | T |
| Pmg-14 | 577 | 19227 | 0 | **49.41** | T | T | T | T |
| Pmg-14-sat | 577 | 1921 | NA | T | T | T | T | T |
| misc | | | | | | | | |
| grid-0010 | 110 | 191 | 5.94e+23 | 90.12 | NA | 0.50 | **0.24** | 2.19 |
| c499.isc | 243 | 714 | 2.20e+12 | T | T | T | T | **13.92** |
| tire-1 | 352 | 1038 | 7.26e+8 | T | NA | 0.11 | **0.06** | 4.59 |
| log-1 | 939 | 3785 | 5.64e+20 | T | NA | **0.09** | **0.09** | 18.65 |

The last group of benchmarks shows that GBsolver is complementary to but not a replacement for DPLL-based and d-DNNF solvers, since GBsolver fails on these benchmarks while the other solvers have no difficulties in handling them. Whether this is because of the inherent limit of Gröbner basis computation, or can be remedied with more optimization work, remains an open question.

It is worth noting that the Gröbner basis method should have an edge when the number of solutions of a formula is small, which is often the hard case for search-based approaches. In fact, a small number of solutions would imply a small set of standard monomials (Theorem 4.2), which corresponds to a simple structure of the Gröbner basis. For example, in the extreme case, when a formula is unsatisfiable, the standard monomial set is empty. Then its corresponding Gröbner basis must be of the simplest form $\{1\}$, which can often be detected early in the computation process (this may explain the data on "pmg" benchmarks). In contrast, DPLL-based methods may need to make a lot of futile search when the solution is very sparsely distributed. Thus, it is likely that the Gröbner basis approach will turn out to be a useful complement for DPLL-based methods.

# Chapter 6

# Conclusions and Future Work

The main contribution of this thesis is a new method for solving the counting problem which is both general and practical: It can be applied to any polynomial systems over arbitrary finite fields, and has been implemented to solve #SAT. The mathematical correctness of our method relies on a Nullstellensatz modified for finite fields and properties of Gröbner Bases. Our algorithm is proved to halt in single exponential time in the number of variables. We have implemented a practical #SAT solver based on our algorithm. Experimental results show that our solver outperforms existing search-based solvers significantly on a number of benchmarks, and is competitive in general in terms of the size of problems (number of variables and clauses) that can be handled.

The complexity of Gröbner Basis computation for ideals with field polynomials should remain a topic of further research. It is conceivable that the Gröbner Basis computation may only require polynomial space, since, after all, $P^{\#P} \subseteq PSPACE$. If it turns out that Gröbner Bases for ideals with the field polynomials can indeed be computed using polynomial space, then a very interesting question may follow, that is: Would knowing the number of zeros help computing the Grobner Basis of an ideal? The question arises because given the theorems we proved, counting the number of solutions (a $\#P$ problem) reveals information about the shape of the Gröbner Basis (for example, trivially, when the number of solutions is 0, the Gröbner Basis is fixed to be 1). If the answer is positive, then it would confirm hardness of the $P^{\#P}$ class. If the answer is negative, then it may, on the contrary, give more evidence about separating $P^{\#P}$ and $PSPACE$. Evidently these are

distant goals, but to our best knowledge the connections have not been made before.

Major improvement on the performance of the #SAT solver can be gained through optimizing Gröbner basis computation in Boolean rings. In particular, while the order of variables has a big influence on the computation, few heuristics for choosing the right variable order for Gröbner basis computation have been investigated. It should be mentioned that while the DPLL algorithm has gone through more than twenty years of improvement, work on Gröbner basis in Boolean rings has just started recently. Our result demonstrates the potential of using algebraic methods in Boolean problems that are harder than SAT. We are convinced that more optimizations for Gröbner basis computation will lead to progress on these problems, where DPLL-based approaches have not been working very well. Also, it is likely that the Gröbner basis and DPLL approaches are complementary in general, and a careful combination of the different approaches may lead to powerful solvers that are able to handle a wide variety of benchmarks.

# Bibliography


[1] L. M. Adleman and M.-D. A. Huang. *Primality testing and Abelian varieties over finite fields*, volume 1512 of *Lecture notes in mathematics*. Springer, 1992.

[2] F. Bacchus, S. Dalmao, and T. Pitassi. Algorithms and complexity results for #sat and bayesian inference. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS 2003)*, pages 340–351, 2003.

[3] T. Becker and V. Weispfenning. *Gröbner Bases*. Springer, 1998.

[4] M. Brickenstein and A. Dreyer. Polybori: A framework for gröbner-basis computations with boolean polynomials. *J. Symb. Comput.*, 44(9):1326–1345, 2009.

[5] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bulletin*, 10(3):19–29, 1976.

[6] B. Buchberger. A note on the complexity of constructing Gröbner-bases. In J. A. van Hulzen, editor, *EUROCAL*, volume 162 of *Lecture Notes in Computer Science*, pages 137–145. Springer, 1983.

[7] B. S. Carla P. Gomes, Ashish Sabharwal. Model counting. *Handbook of Satisfiability*, 2008.

[8] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner Basis algorithm to find proofs of unsatisfiability. In *Proc. 28th Annual ACM Symposium on Theory of Computing*, pages 174–183, 1996.

[9] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 1997.

[10] A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[11] A. Darwiche. New advances in compiling CNF into decomposable negation normal form. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 328–332, 2004.

[12] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[13] J. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[14] P. Gaudry and R. Harley. Counting points on hyperelliptic curves over finite fields. In W. Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 313–332. Springer, 2000.

[15] R. Germundsson. Basic results on ideals and varieties in finite fields. *Technical Report LiTH-ISY-I-1259, Linkoping University, S-581 83*, 1991.

[16] S. Goldwasser and J. Kilian. Almost all primes can be quickly certified. In *Proc. 18th Annual ACM Symposium on Theory of Computing*, pages 316–329. ACM, 1986.

[17] D. Grigoriev and M. Karpinski. An approximation algorithm for the number of zeros of arbitrary polynomials over GF[q]. In *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 662–669. IEEE, 1991.

[18] M.-D. A. Huang and Y.-C. Wong. Solving systems of polynomial congruences modulo a large prime (extended abstract). In *Proc. 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 115–124, 1996.

[19] M.-D. A. Huang and Y.-C. Wong. An algorithm for approximate counting of points on algebraic sets over finite fields. In J. Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 514–527. Springer, 1998.

[20] R. J. B. Jr. and J. D. Pehoushek. Counting models using connected components. In *AAAI/IAAI*, pages 157–162, 2000.

[21] M. Karpinski and M. Luby. Approximating the number of zeroes of a GF[2] polynomial. *J. Algorithms*, 14(2):280–287, 1993.

[22] K. Kühnle and E. W. Mayr. Exponential space computation of Gröbner Bases. In *Proc. ISSAC*, pages 63–71, 1996.

[23] S. Lang. *Algebra*. Springer, 3rd edition, 2005.

[24] J. Pila. Frobenius maps of Abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192):745–763, 1990.

[25] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

[26] T. Sang, F. Bacchus, P. Beame, H. A. Kautz, and T. Pitassi. Combining component caching and clause learning for effective model counting. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT 2004), LNCS 3542*, 2004.

[27] T. Sang, P. Beame, and H. Kautz. Solving bayesian networks by weighted model counting. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 475–482, 2005.

[28] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483–494, 1985.

[29] M. Thurley. sharpSAT - counting models with advanced component caching and implicit BCP. pages 424–429, 2006.

[30] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comp.*, 20(5):865 – 877, 1991.

[31] J. van Lint et al. *Introduction to Coding Theory and Algebraic Geometry*. Springer, 1988.

[32] A. Weil. *Sur les courbes algébriques et les variétés qui s'en déduisent*. Hermann, 1948.