Анализ и сравнение методов базисов Грёбнера для решения нелинейных полиноминальных систем в конечных полях характеристики 2

автор: Низамов И., перевод: Люлин Д. и Максименко Е.

10 мая 2021 г.

Содержание

Метод базисов Грёбнера — это универсальный инструмент для решения нелинейных полиноминальных уравнений в произвольных алгебраических полях. Этот метод — обобщение метода Гаусса на случай нелинейных уравнений. Эта работа состоит из трёх частей. В первой обсуждается теория базисов Грёбнера и их свойства. Вторая часть описывает алгоритмы построения базисов Грёбнера, их сравнение и анализ. В третьей части обсуждается метод решения систем нелинейных уравнений в конечных алгебраических полях с 2^n элементами, эти поля используются в теории кодирования и криптографии. Также в этой части показаны результаты работы реализованных алгоритмов.

Введение

Преамбула

Сегодня количество задач в точных науках, включающих в себя решение систем алгебраических уравнений, лишь растёт. Например, такие задачи появляются в криптографии и теории кодирования применительно к конечным полям. Очевидно, что решение таких задач в общем случае — нетривиальная задача, хотя для многих систем существует множество различных способов решения.

Если оценивать системы уравнений с точки зрения степеней их уравнений, то для линейных алгебраических уравнений существует множество способов решения (метод Крамера, метод Гаусса, метод прогонки и т. д.). Однако решение систем нелинейных уравнений — намного более сложная задача, требующая больших затрат времени и памяти. Для таких систем также существуют различные методы.

В этой работе описывается метод базисов Грёбнера, появившийся в 1980х годах и привлёкший много внимания благодаря своей универсальности. Этот метод был распространён благодаря развитию ЭВМ, способных производить быстрые вычисления с многочленами. Мы обсудим и сравним разные алгоритмы для построения базисов Грёбнера и их применение для решения систем уравнений. Сравнение будет сделано на базе реализованных нами алгоритмов в конечном поле F_{2^n}. Данный тип алгоритмов не поддерживается каким-либо бесплатным программным обеспечением, использующимся для вычисления базисов Грёбнера. Поэтому в результате нашей работы мы получим инструмент для решения систем уравнений в поле F_{2^n} и построения базисов Грёбнера в них.

Постановка задачи и базовые обозначения.

Целью этой работы является исследование и решение систем нелинейных полиноминальных уравнений в конечном поле F_{2^n}. Задачи этой работы:

1. Дать определение базисам Грёбнера и описать их свойства, которые могут помочь в решении систем уравнений;

2. Найти и описать алгоритм для построения базисов Грёбнера;
3. Найти способы оптимизации данного алгоритма;
4. Применить данный алгоритм к решению систем уравнений;
5. Реализовать все рассматриваемые алгоритмы для разных алгебраических полей;
6. Сравнить реализации в случае поля F_{2^n}.

Для начала, введём базовые определения, используемые в дальнейшем. **Определение.** Группа — это множество G с определённой бинарной операцией $*$, удовлетворяющей следующим аксиомам:

1. Замкнутость операции $*$:
$$a * b \in G \quad \forall a, b \in G;$$
2. Ассоциативность операции $*$:
$$a * (b * c) = (a * b) * c \quad \forall a, b, c \in G;$$
3. Существование нейтрального элемента:
$$a * e = e * a = a \quad \forall a \in G;$$
4. Существование обратного элемента a^{-1} для каждого a:
$$a * a^{-1} = a^{-1} * a = e \quad \forall a \in G.$$

Определение. Группа G называется коммутативной или абелевой, если
$$a * b = b * a \quad \forall a, b \in G.$$

Определение. Группа называется конечной, если она содержит конечное число элементов. Это число называется порядком группы.

Определение. Кольцо - это множество R с двумя определёнными операциями $+$ и $*$, такими что:

1. R является абелевой группой по операции $+$;
2. Операция $*$ ассоциативна;
3. Операция $*$ дистрибутивна по отношению к $+$:
$$(a + b) * c = (a * c) + (b * c) \quad \forall a, b, c \in G;$$
$$a * (b + c) = (a * b) + (a * c) \quad \forall a, b, c \in G.$$

Определение. Поле - это алгебраическая структура F с двумя определёнными операциями $+$ и $*$, такими что:

1. F является абелевой группой по операции $+$;
2. $F\backslash\{0\}$ является абелевой группой по операции $*$;
3. Операции $+$ и $*$ связаны дистрибутивным законом

$$x * (y + z) = (x * y) + (x * z) \quad \forall x, y, z \in F.$$

Далее мы будем пользоваться следующими обозначениями:

1. K - конечное поле;
2. $K[x_1, x_2, ..., x_n]$ - кольцо многочленов n переменных $x_1, x_2, ..., x_n$ над полем K;
3. f, g, h, k, p, q - многочлены из кольца $K[x_1, x_2, ..., x_n]$;
4. I, F, G - конечные подмножества кольца $K[x_1, x_2, ..., x_n]$;
5. s, t, u - одночлены в форме $x_1^{i_1}, ..., x_n^{i_n}$. Предполагается, что $x_i^0 \equiv 1$;
6. a, b, c, d - элементы поля K;
7. i, j, l, m - натуральные числа (или 0).

Определение. Идеал - это подмножество $I \subset K[x_1, x_2, ..., x_n]$, для которого верно:

1. $0 \in I$;
2. $f \in I, g \in I \Rightarrow f + g \in I$;
3. $f \in I, h \in K[x_1, x_2, ..., x_n] \Rightarrow hf \in I$.

Определение. Пусть $f_1, ..., f_s$ - многочлены, $f_i \in K[x_1, x_2, ..., x_n]$. Тогда множество

$$\langle f_1, ..., f_s\rangle = \{\sum_{i=1}^{s} h_i f_i : h_i \in K[x_1, x_2, ..., x_n]\}$$

называется идеалом, порождённым многочленами $f_1, ..., f_s$. Они называются порождающими многочленами или порождающим множеством. Мы будем писать $f \equiv_F g$, если $f \equiv g(mod\langle F\rangle)$, т.е. $f - g \in \langle F\rangle$, где F - подмножество кольца $K[x_1, x_2, ..., x_n]$.

Краткий обзор методов решения систем уравнений

В наши дни в теории кодирования и криптографии используются следующие методы решения систем алгебраических уравнений.

Метод Гаусса. Для решения систем линейных уравнений мы имеем множество различных точных и итеративных методов. Самым распространённым является метод Гаусса. Процесс решения состоит из двух частей. В первой части (прямой ход) мы преобразуем систему, последовательно удаляя переменные из рассмотрения, пока не останется уравнение с одной переменной. Найдя значение данной переменной, мы приступаем ко второй части решения (обратный ход) и находим значения оставшихся переменных последовательно, используя ранее найденные значения.

В случае системы из m уравнений и m переменных, этот метод требует cm^{ω} операций в поле F_q, где c - константа, ω - значение, не превышающее 2,376. Самый быстрый алгоритм, называемый алгоритмом Штрассена, имеет следующие значения: $c = 7$ и $\omega = log_2 7$.

Метод перебора. В этом методе последовательно перебираются все элементы F_q^n, пока не будет найдено подходящее множество. Множество называется подходящим, если при подстановке его элементов в систему не возникает противоречий. В среднем, если $q = 2$, то нужно проверить около $2^{(n-1)}$ множеств.

Метод перебора с префиксами. Этот метод выполняется обходом в глубину соответствующего q-арного префиксного дерева высоты n. Перебираются не все элементы F_q^n, а их префиксы из F_q^k, где k изменяется от 1 до n. Проверка префикса ν происходит следующим способом:

1. Если при подстановке в систему префикса из F_q^k возникает противоречие при каком-либо k, то данная система не подходит. Тогда происходит проверка следующего префикса, который может быть найден по следующему правилу: если последний элемент префикса - это последний элемент поля, то удаляем его из префикса. Иначе заменить его следующим элементом поля.

2. Если система после подстановки префикса не имеет противоречий и линейна, то можно решить её стандартными методами для линейных уравнений, например, методом Гаусса.

3. Если система после подстановки не имеет противоречий и нелинейна, то проверяется префикс $\nu \circ f$ из F_q^{k+1}, где $f \in F_q$ и f - это первый элемент поля.

f - начальный префикс в этом методе.

Метод линеаризации. Пусть $t_1, t_2, ..., t_r$ - все разные одночлены системы степени большей 1. После этого заменим их значениями $y_1, y_2, ..., y_r$ из поля F_q и получим новую линейную систему. Количество переменных в ней меньше или равно $n + r$. Пусть система переопределённая (уравнений больше числа неизвестных). Существует подсистема из линейно независимых уравнений.

Если мы решим её методом Гаусса и подставим полученные значения в исходную систему, то получится система уравнений формы $u_i = a_i, i = 1, ..., s$, где $s \leq r, t_i = y_i, i = 1, ..., r, a_i \in F_q$, и u_i - одночлены. Эта система может быть решена.

В случае поля F_2 решение становится легче. Если $a_i = 1$, то все переменные, содержащиеся в u_i, должны быть равны 1. Иначе хотя бы одна из них должна быть равна 0.

Этот метод не будет работать, если результирующая система станет недоопределённой после подстановки одночленов. В этом случае система будет иметь $q^{n(L)-m(L)}$ решений, где $n(L)$ - количество переменных в новой системе, а $m(L)$ - количество линейно независимых уравнений в ней. Поэтому нужно найти частичное решение и подставить его: $t_i = y_i, i = 1, ..., r$, чтобы сделать систему разрешимой. Это может быть сделано за экспоненциальное время.

Релинеаризация. [1] Этот метод используется, когда в методе линеаризации получается недоопределённая система уравнений. Мы добавляем в систему новые нелинейные уравнения, которые тривиальны для старых переменных, но используем новые переменные. После этого используется метод линеаризации или релинеаризации (повторно).

Расширенная линеаризация. [1] Этот метод использует дополнительный параметр d - натуральное число, $d \geq d_0$. Совершаются два шага:

1. Увеличение: строится система E_d, состоящая из уравнений $f_i(x)t = b_i t$, для $i = 1, ..., m$, где $t = X^k$ и $k = 0, 1, ..., d - d_0$. Степень новой системы не больше d.

2. Линеаризация: полученная система E_d решается методом линеаризации.

Очевидно, что $E_{d_0} = E$, если система уравнений имеет степень d_0. Тогда метод линеаризации и метод расширенной линеаризации при $d = d_0$ совпадают.

Нужно выбрать правильное значение d. Этот метод работает, только когда система E_d переопределена.

Метод линеаризационных множеств. Если мы определим некоторые переменные системы, она станет линейной. Подмножество Z переменных, содержащих эти переменные, называется линеаризационным множеством системы уравнений. Мы можем создать линеаризационное множество и последовательно проверять все множества возможных значений переменных линеаризационного множества.

Каждый раз мы будем получать новую линейную систему. Если она разрешимая, то её можно решить методом Гаусса и найти значения оставшихся переменных.

Сложность этого метода: q^p, где $p = |Z|$. Следовательно, наиболее эффективный алгоритм должен использовать наименьшее линеаризационное множество.

Метод Бухбергера. Базисы Грёбнера применимы к решению задач, относящихся к идеалам колец полиномов. Бруно Бухбергер[5] дал определение этих базисов, а также предложил простой (но не самый эффективный) метод их нахождения. Позже этот метод был усовершенствован многими исследователями, например, Жаном-Шарлем Фожером ([12], [13]), Тиллем Стежерсом ([15]) и Джоном Перри ([2], [11]).

Глава 1. Базисы Грёбнера

Упорядочение мономов в $K[x_1, ..., x_n]$

Пусть K - произвольное поле, а $R = K[x_1, ..., x_n]$ - кольцо полиномов над полем K. Полином - это сумма $c_1M_1 + c_2M_2 + ... + c_mM_m$, где c_i - элементы K, M_i - одночлены. Каждый одночлен (моном) - это произведение $x^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}...x_n^{\alpha_n}$, где α_i - неотрицательные целые числа.

Чтобы определить базис Грёбнера, нам необходимо определить упорядочение мономов $>$, которое удовлетворяет следующим условиям:

$$\forall t \quad t \neq 1 \Rightarrow t > 1$$

$$\forall s, t, u \quad s > t \Rightarrow su > tu$$

Это отношение называется упорядочением мономов.

Как только установлено мономиальное упорядочение, все члены полинома (мономы с ненулевыми коэффициентами) выстраиваются в порядке убывания мономов (в соответствии с их упорядочением). Это позволяет записать полином как упорядоченный список в виде пар *коэффициент-вектор экспонент* - каноническое представление полиномов.

Между мономом $x^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}...x_n^{\alpha_n}$ и его вектором экспонент $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ существует однозначное соответствие. Так, $\alpha > \beta \Rightarrow x^\alpha > x^\beta$.

Примеры возможных мономиальных упорядочений

Лексикографическое упорядочение (lex): пусть есть $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ и $\beta = (\beta_1, \beta_2, ..., \beta_n)$. Говорят, что $\alpha >_{lex} \beta$, если первый слева ненулевой элемент вектора $\alpha - \beta$ больше нуля.

Примеры:

1) $(2, 3, 4) >_{lex} (1, 2, 3)$, потому что $\alpha - \beta = (1, 1, 1)$.

2) $(4, 5, 6) >_{lex} (4, 0, 3)$, потому что $\alpha - \beta = (0, 5, 3)$.

Обратное лексикографическое упорядочение (invlex): пусть есть $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ и $\beta = (\beta_1, \beta_2, ..., \beta_n)$. Говорят, что $\alpha >_{invlex} \beta$, если первый справа ненулевой элемент вектора $\alpha - \beta$ больше нуля.

Примеры:

1) $(2, 3, 4) >_{invlex} (1, 2, 3)$, потому что $\alpha - \beta = (1, 1, 1)$.

2) $(4, 5, 6) >_{invlex} (4, 3, 6)$, потому что $\alpha - \beta = (0, 2, 0)$.

Градуированное лексикографическое упорядочение (grlex): пусть есть $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ и $\beta = (\beta_1, \beta_2, ..., \beta_n)$. Говорят, что $\alpha >_{grlex} \beta$, если

$$|\alpha| = \sum_{i=1}^{n} \alpha_i > |\beta| = \sum_{i=1}^{n} \beta_i \ \vee \ |\alpha| = |\beta| \ \wedge \ \alpha >_{lex} \beta$$

Примеры:

1) $(2, 3, 8) >_{grlex} (5, 2, 3)$, потому что $|(2, 3, 8)| = 13 > |(5, 2, 3)| = 10$.

2) $(4, 6, 5) >_{grlex} (4, 5, 6)$, потому что $|(4, 5, 6)| = |(4, 6, 5)|$, но $\alpha - \beta = (0, 1, -1)$.

Градуированное обратное лексикографическое упорядочение (grevlex): пусть есть $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ и $\beta = (\beta_1, \beta_2, ..., \beta_n)$. Говорят, что $\alpha >_{grevlex} \beta$, если

$$|\alpha| = \sum_{i=1}^{n} \alpha_i > |\beta| = \sum_{i=1}^{n} \beta_i \ \vee \ |\alpha| = |\beta| \ \wedge \ \alpha >_{lex} \beta$$

и первый справа ненулевой элемент вектора $\alpha - \beta$ меньше нуля.

Примеры:

1) $(2, 3, 8) >_{grevlex} (5, 2, 3)$, потому что $|(2, 3, 8)| = 13 > |(5, 2, 3)| = 10$.

2) $(4, 6, 5) >_{grevlex} (4, 5, 6)$, потому что $|(4, 5, 6)| = |(4, 6, 5)|$, но $\alpha - \beta = (0, 1, -1)$.

Заметим, что упорядочения $>_{lex}$, $>_{grlex}$, $>_{grevlex}$ одинаково упорядочивают все единичные векторы переменных $x_1, ..., x_n$:

$$(1, 0, ..., 0) >_{lex} (0, 1, ..., 0) >_{lex} ... >_{lex} (0, 0, ..., 1),$$

$$(1, 0, ..., 0) >_{grlex} (0, 1, ..., 0) >_{grlex} ... >_{grlex} (0, 0, ..., 1),$$

$$(1, 0, ..., 0) >_{grevlex} (0, 1, ..., 0) >_{grevlex} ... >_{grevlex} (0, 0, ..., 1),$$

Пример 1.1 Запишем полином $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2 \in K[x, y, z]$ в каждом из вышеупомянутых упорядочений:

1. Лексикографическое упорядочение (lex):

$$f = -5x^3 + 7x^2z^2 + 4xy^2z + 4z^2$$

2. Обратное лексикографическое упорядочение (invlex):

$$f = 4z^2 + 7x^2z^2 + 4xy^2z - 5x^3$$

3. Градуированное лексикографическое упорядочение (grlex):

$$f = 7x^2z^2 + 4xy^2z - 5x^3 + 4z^2$$

4. Градуированное обратное лексикографическое упорядочение (grevlex):

$$f = 4xy^2z + 7x^2z^2 - 5x^3 + 4z^2$$

Пусть f - полином с установленным упорядочением мономов и

$$f = \sum_{\alpha} \alpha_\alpha x^\alpha$$

Теперь мы можем определить следующие свойства полиномов.

1. Мультистепень полинома

$$multideg(f) = max(\alpha : \alpha_\alpha \neq 0),$$

где максимум находится в соответствии с выбранным упорядочением.

2. Старший коэффициент полинома

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

3. Старший моном полинома

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

4. Старший член полинома

$$(f) = LC(f) * LM(f)$$

Алгоритм деления в $K[x_1, ..., x_n]$

В действительности, алгоритм деления полинома $f \in K[x_{1,...,x_n}]$ на полиномы $f_1, f_2, ..., f_s$ совпадает со случаем одной переменной: нужно удалить старший член полинома f, который определён упорядочением мономов. Это можно сделать следующим способом: умножить один из полиномов f_i на подходящий моном и отнять результат от f.

Теорема 1.1 (Алгоритм деления в $K[x_1, ..., x_n]$). Зафиксируем мономиальное упорядочение > и пусть $F = (f_1, ..., f_s)$ - упорядоченный s-набор полиномов из $K[x_1, ..., x_n]$. Тогда каждый полином $f \in K[x_1, ..., x_n]$ может быть представлен в виде

$$f = \alpha_1 f_1 + ... + \alpha_s f_s + r,$$

где $\alpha_i, r \in K[x_1, ..., x_n]$ и либо $r = 0$, либо r - линейная комбинация мономов (с коэффициентами из K), ни один из которых не делится ни на один из старших членов $(f_1), ..., (f_s)$. Мы называем r остатком от деления полинома f на F. Более того, если $\alpha_i f_i \neq 0$, то

$$multideg(f) \geq multideg(\alpha_i f_i)$$

Псевдокод алгоритма:

Вход: $f_1, ..., f_s$
Выход: $\alpha_1, ..., \alpha_s, r$
$\alpha_1 := 0; ...; \alpha_s := 0; r := 0;$
$p := f;$
WHILE $p \neq 0$ DO
    $i := 1;$
    естьделение := $false$;
    WHILE $i \leq s$ AND естьделение $= false$ DO
        IF $LT(f_i)$ делит (p) THEN
            $\alpha_i := \alpha_i + (\frac{(p)}{(f_i)});$
            $p := p - (\frac{(p)}{(f_i)})f_i;$
            естьделение $= true$;
        ELSE
            $i = i + 1;$
    IF естьделение $= false$ THEN
        $r := r + (p);$
        $p := p - (p);$

Доказательство этой теоремы приведено в [8].

Пример 1.2 Разделим полином $f = x^2y + xy^2 + y^2$ на $f_1 = xy - 1$ и $f_2 = y^2 - 1$, используя лексикографическое упорядочение, в котором $x > y$. Если будет возможен выбор между f_1 и $f2_2$, будем делить на f_1. В начале $p = f$.

Шаг 1. $(p) = x^2y$ делится только на $(f_1) = xy$. Результат деления x прибавим к α_1. После вычитания $xf_1 = x^2y - x$ из p мы получим $p = xy^2 + x + y^2$.

Шаг 2. $(p) = xy^2$ делится и на $(f_1) = xy$, и на $(f_2) = y^2$. Выберем первый полином. Результат деления y прибавим к α_1. Теперь $\alpha_1 = x + y$. После вычитания $yf_1 = xy^2 - y$ из p получим $p = x + y^2 + y$.

Шаг 3. $(p) = x$, и этот член не делится ни на один из главных членов. Но p - не остаток, потому что y^2 делится на (f_2). Поэтому прибавляем x к α_1 и после вычитания получаем $p = y^2 + y$.

Шаг 4. $(p) = y^2$ делится только на $(f_2) = y^2$. Результат деления равен 1 прибавим его к α_2. После вычитания $f_2 = y^2 - 1$ из p получим $p = y + 1$.

Шаг 5. Ни один из членов p не делится на главные члены, поэтому прибавим их к остатку r.

Алгоритм успешно завершился. f можно записать в виде:

$$f = x^2y + xy^2 + y^2 = (x + y)(xy - 1) + 1(y^2 - 1) + x + y + 1$$

Видно, что ни один моном остатка не делится на старшие члени делителей. Также заметим, что значения $\alpha_1, ..., \alpha_s, r$ зависят от порядка делителей в F. Если изменим порядок, получим

$$f = x^2y + xy^2 + y^2 = (x + 1)(y^2 - 1) + x(xy - 1) + 2x - 1$$

Дальше мы увидим, что есть такие наборы полиномов, для которых остаток не зависит от порядка делителей.

Мономиальные идеалы.

Дадим определение мономиального идеала.

Определение 1.1. Идеал $I \subset K[x_1, ..., x_n]$ называется мономиальным, если I состоит из всех конечных сумм формы $\sum_{\alpha \in A} h_\alpha x^\alpha$, где A - подмножество $\mathbb{Z}^n_{\geq 0}$, и каждый полином $h_\alpha \in K[x_1, ..., x_n]$. Такой идеал будет обозначаться через $\langle x^\alpha : \alpha \in A \rangle$.

Пример такого идеала: $\langle x^4y^2, x^3y^4, x^2y^5 \rangle \in K[x_1, ..., x_n]$.

Можно охарактеризовать все мономы, принадлежащие данному мономиальному идеалу, с помощью двух лемм.

Лемма 1.1. Пусть $I = \langle x^\alpha : \alpha \in A \rangle$ - мономиальный идеал. Моном x^β принадлежит I тогда и только тогда, когда какой-либо моном $x^\alpha, \alpha \in A$ делит x^β.

Доказательство.

Если моном x^β делится на $x^\alpha, \alpha \in A$, то x^β принадлежит I по определению. Докажем обратное. Пусть x^β принадлежит I. Тогда $x^\beta = \sum_{i=1}^s h_i x^{\alpha(i)}$, где $h_i \in K[x_1, ..., x_n], \alpha(i) \in A$. Каждый множитель h_i - линейная комбинация мономов, так что каждый член правой части равенства делится на некоторый моном $x^{\alpha(i)}$. Следовательно, x^β тоже обладает этим свойством и может быть членом в каком-либо $h_i x^{\alpha(i)}$. ■

Лемма 1.2. Пусть I - мономиальный идеал, и $f \in K[x_1, ..., x_n]$. Тогда следующие три утверждения эквивалентны:

1. f принадлежит I;
2. Каждый член f принадлежит I;
3. f может быть выражен как линейная комбинация мономов из I.

Доказательство.

Последовательность доказательств $(3) \Rightarrow (2) \Rightarrow (1)$ очевидна. Доказательство $(1) \Rightarrow (3)$ выглядит так же, как и доказательство второй части леммы 1.1. ■

Таким образом, мономиальный идеал однозначно определяется своими мономами.

Следствие 1.1. Два мономиальных идеала равны тогда и только тогда, когда равны множества мономов, принадлежащие соответствующим идеалам.

Из леммы 1.2 и следствия 1.1 получается важное заключение.

Теорема 1.2 (Лемма Диксона). Любой мономиальный идеал $I = \langle x^\alpha : \alpha \in A\rangle \subset K[x_1, ..., x_n]$ может быть представлен в виде $I = \langle x^{\alpha(1)}, ..., x^{\alpha(s)}\rangle$, где $\alpha(1), ..., \alpha(s) \in A$. В частности, I имеет конечный базис.

Доказательство леммы Диксона приведено в [8].

Следовательно, каждый идеал в $K[x_1, ..., x_n]$ является конечно порождаемым.

Определение 1.2. Пусть $I \subset K[x_1, ..., x_n]$ - ненулевой идеал.

1. Обозначим множество старших членов из I как (I)

$$(I) = \{cx^\alpha : \exists f \in I((f) = cx^\alpha)\}$$

2. Обозначим идеал, порождаемый элементами (I) как $\langle(I)\rangle$

Замечание 1.1. Равенство $\langle(f_1), ..., (f_s)\rangle = \langle(I)\rangle$ не всегда верно, несмотря на то что I был порождён $f_1, ..., f_s$. Однако включение всегда имеет место: $\langle(f_1), ..., (f_s)\rangle \subset \langle(I)\rangle$

Пример 1.3. Пусть $I = \langle f_1, f_2\rangle$, где $f_1 = x^3 - 2xy$, $f_2 = x^2y - 2y^2 + x$, и на мономах из $K[x, y]$ задано grlex-упорядочение. Тогда

$$x^2 = x(x^2y - 2y^2 + x) - y(x^3 - 2xy)$$

Таким образом, x^2 принадлежит идеалу I, а $x^2 = (x^2)$ принадлежит идеалу $\langle(I)\rangle$. Но ни $(f_1) = x^3$, ни $(f_2) = x^2y$ не делятся на x^2. Отсюда x^2 не принадлежит идеалу $\langle(f_1), (f_2)\rangle$ по лемме 1.

Базисы Грёбнера и их свойства.

Чтобы дать определение базисам Грёбнера, докажем утверждение.

Предложение 1.1. Пусть $I \subset K[x_1, ..., x_n]$ - идеал. Тогда:

1. $\langle(I)\rangle$ - мономиальный идеал;
2. $\exists g_1, ..., g_s \in I : \langle(g_1), ..., (g_s)\rangle = \langle(I)\rangle$.

Доказательство.

Пусть каждый полином $g \in I\backslash\{0\}$ имеет старший моном $LM(g)$. Эти старшие мономы порождают мономиальный идеал $\langle LM(g) : g \in I\backslash\{0\}\rangle$. $LM(g)$ отличается от (g) только ненулевым коэффициентом. Тогда новый мономиальный идеал равен $\langle(I)\rangle$.

$\langle(I)\rangle$ порождён мономами $LM(g), g \in I\backslash\{0\}$, поэтому существует конечная последовательность полиномов $g_1, ..., g_s$, для которых $\langle LM(g_1), ..., LM(g_2)\rangle =$

$\langle(I)\rangle$. Опять, $LM(g)$ отличается от (g) только ненулевым коэффициентом. Отсюда $\langle(I)\rangle = \langle(g_1), ..., (g_s)\rangle$. ■

Используя эти утверждения и приведённый ранее алгоритм деления, докажем второе важное утверждение.

Теорема 1.3 (Теорема Гильберта о Базисе). Каждый идеал $I \subset K[x_1, ..., x_n]$ является конечно порождённым, то есть $\exists g_1, ..., g_s \in I : I = \langle(g_1), ..., (g_s)\rangle$.

Доказательство этой теоремы приведено в [8]. Подмножество элементов $g_1, ..., g_s$ из предложения 1.1 в точности формируют базис Грёбнера идеала I, центральный элемент этой работы.

Определение 1.3. Пусть определено некоторое мономиальное упорядочение. Базис Грёбнера идеала I - это конечное подмножество $G = g_1, ..., g_s$ элементов из I, такое что

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

Следующее следствие вытекает из доказательства теоремы Гильберта о базисе.

Следствие 1.2. Пусть определено некоторое мономиальное упорядочение. Тогда каждый идеал $I \subset K[x_1, ..., x_n]$, не содержащий нулей, имеет базис Грёбнера G, и G - базис I.

Докажем одно из самых важных свойств базиса Грёбнера.

Предложение 1.2. Пусть $G = \{g_1, ..., g_s\}$ - базис Грёбнера идеала $I \subset K[x_1, ..., x_n]$, и пусть $f \in K[x_1, ..., x_n]$. Тогда существует единственный полином $r \in K[x_1, ..., x_n]$, который обладает следующими свойствами:

1. Никакой из главных членов $(g_1), ..., (g_s)$ не делит никакой из членов полинома r.
2. $\exists g \in I : f = g + r$.

Таким образом, r - это остаток от деления полинома f на G. Более того, r не зависит от порядка делителей в G. В этом случае r называется нормальной формой f.

Доказательство.

Можно записать f в следующей форме, используя алгоритм деления:

$$f = \alpha_1 g_1 + ... + \alpha_s g_s + r,$$

где r удовлетворяет первому условию. Пусть $g = \alpha_1 g_1 + ... + \alpha_s g_s$. Полином g принадлежит I, так что второе условие тоже выполено.

Пусть существуют другие полиномы r' и g', для которых выполняется условие $f = g' + r'$. В этом случае $r - r' = g - g' \in I$. Поэтому если $r \neq r'$, то

$$(r - r') \in \langle(g_1), ..., (g_s)\rangle = \langle(I)\rangle.$$

Из леммы 1.1 мы получим, что $(r - r')$ делит некоторый моном (g_i), что противоречит первому условию. Тогда $r = r'$. ■

Следствие 1.3. Пусть $G = \{g_1, ..., g_s\}$ - базис Грёбнера идеала $I \subset K[x_1, ..., x_n]$, и пусть $f \in K[x_1, ..., x_n]$. f принадлежит I тогда и только тогда, когда остаток от деления полинома f на G равен 0.

Доказательство.

Пусть остаток равен 0. В этом случае f принадлежит I. Докажем обратное утверждение. Пусть f принадлежит I. Можно записать f в форме $f = f + 0$, что удовлетворяет обоим условием из предложения 1.2. Также нулевой полином - единственный возможный остаток от деления полинома f на G. ■

Таким образом, мы можем установить, принадлежит ли полином идеалу. Нужно просто найти остаток от деления полинома на базис Грёбнера. Теперь нужно найти способ построения этих базисов.

Определение 1.4. Пусть $\overline{f}^F$ - остаток от деления полинома f на упорядоченный набор полиномов F. Этот набор можно считать неупорядоченным, если F - базис Грёбнера.

Пример 1.4:

Пусть $F = (x^2y - y^2, x^4y^2 - y^2)$. Мы используем лексикографическое упорядочение. В этом случае $\overline{x^5y}^F = xy^3$.

Определение 1.5. Пусть $f, g \in K[x_1, ..., x_n]$ - ненулевые элементы, где $multideg(f) = \alpha, multideg(g) = \beta$. Пусть $\gamma = (\gamma_1, ..., \gamma_n)$, где $\gamma_i = max(\alpha_i, \beta_i)$ дл всех i. Тогда x^γ - наименьшее общее кратное $LM(f)$ и $LM(g)$. Обозначим его как $x^\gamma = LCM(LM(f), LM(g))$.

S-полином полиномов f и g - следующее выражение:

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

Легко заметить, что это выражение удаляет главные члены полиномов.

Лемма 1.3. Рассмотрим сумму $\sum_{i=1}^{n} c_i f_i$, где мультистепень f_i равна δ, и c_i принадлежит K для всех i. Если мультистепень этой суммы меньше δ, то эта сумма - линейная комбинация S-полиномов $S(f_i, f_l), 1 \leq j, l \leq s$ с коэффициентами из K. Более того, мультистепень каждого S-полинома $S(f_i, f_l)$ меньше γ.

Доказательство леммы 1.3 приведено в [8]. Мы доказали, что если комбинация полиномов с равной мультистепенью удаляет свои старшие члены, это отражается и на их удалении в S-полиномах. Чтобы вычислить базисы Грёбнера, Бухбергер ввёл следующий критерий, основанный на последней лемме в [6].

Теорема 1.4 (Критерий Бухбергера). Пусть I - полиномиальный идеал. Базис $G = \{g_1, ..., g_s\}$ идеала I является базисом Грёбнера этого идеала тогда и только тогда, когда для каждой из пар $i \neq j$ остаток от деления S-полинома $S(g_i, g_j)$ на G равен 0.

Этот критерий называется критерием S-пар. Он помогает составить алгоритм для нахождения базисов Грёбнера.

Алгоритм Бухбергера.

Алгоритм Бухбергера основан на критерии S-пар. Мы можем получить базис Грёбнера из набора полиномов, последовательно прибавляя $\overline{S(f_i, f_j)}^F$ к F. **Теорема 1.5 (Алгоритм Бухбергера [6]).** Пусть $I = \langle f_1, ..., f_s \rangle$, и I не содержит нулей. Тогда базис Грёбнера идеала I строится по следующему алгоритму с конечным числом шагов:

Вход: $F = (f_1, ..., f_s)$
Выход: базис Грёбнера $G = \{g_1, ..., g_t\}$ идеала I, где $F \subset G$
$G := F$;
REPEAT
    $G' := G$;
    FOR каждая пара $\{p, q\}, p \neq q$ в G' DO
        $S := \overline{S(p, q)}^{G'}$;
        IF $S \neq 0$ THEN $G = G \cup \{S\}$;
UNTIL $G = G'$;

Пример 1.5. Пусть $I = \langle f_1, f_2 \rangle = \langle x^3 - 2xy, x^2y - 2y^2 + x \rangle$. Используем grlex-упорядочение с $x > y$. Ранее мы показали, что порождающие элементы этого базиса не образуют базис Грёбнера. Расширим набор. В начале $G = \{f_1, f_2\}$.

Шаг 1. $S(f_1, f_2) = yf_1 - xf_2 = -x^2, \overline{S(f_1, f_2)}^G = -x^2 \neq 0$. Поэтому $G = \{f_1, f_2, f_3\}$, где $f_3 = -x^2$.

Шаг 2. $\overline{S(f_1, f_2)}^G = 0, S(f_1, f_3) = f_1 + xf_3 = -2xy$, но $\overline{S(f_1, f_3)}^G = -2xy \neq 0$. Поэтому $G = \{f_1, f_2, f_3, f_4\}$, где $f_4 = -2xy$.

Шаг 3. $\overline{S(f_1, f_3)}^G = 0, S(f_1, f_4) = yf_1 + \frac{1}{2}x^2 f_4 = -2xy^2 = yf_4$. Отсюда $\overline{S(f_1, f_4)}^G = 0$.

Шаг 4. $S(f_2, f_3) = f_2 + yf_3 = -2y^2 + x$, но $\overline{S(f_2, f_3)}^G = -2y^2 + x \neq 0$. Поэтому $G = \{f_1, f_2, f_3, f_4, f_5\}$, где $f_5 = -2y^2 + x$.

Шаг 5. Легко проверить, что $\overline{S(f_i, f_j)}^G = 0$ для всех $1 \leq i < j \leq 5$.

В итоге, используя теорему 1.4, получим результирующий базис Грёбнера для grlex-упорядочения:

$$G = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}$$

Это самая простая версия алгоритма Бухбергера. На практике она мало полезна. Но мы можем улучшить данный алгоритм. Заметим, что если остаток $\overline{S(p, q)}^G = 0$, то он не изменится после расширения G'.

Однако, этот алгоритм строит избыточные базисы, потому что у них есть избыточные элементы. Можно от них избавиться, используя следующую лемму.

Лемма 1.4. Пусть G - базис Грёбнера идеала I, и есть полином $p \in I$, главный член которого $(p) \in \langle (G \backslash \{p\}) \rangle$. Тогда $G \backslash \{p\}$ - тоже базис Грёбнера.

Доказательство.

Известно, что $\langle(G)\rangle = \langle(I)\rangle$. Тот факт, что $(p) \in \langle(G\backslash\{p\})\rangle$, значит, что $\langle(G\backslash\{p\})\rangle = \langle(S)\rangle$. Отсюда, $G\backslash\{p\}$ - базис Грёбнера по определению. ■

Таким образом, удалив такие полиномы из G и нормализовав оставшиеся полиномы, мы получим минимальный базис Грёбнера. Но эти операции можно проводить, только если G - базис Грёбнера.

Определение 1.6. Минимальным базисом Грёбнера идеала I называется базис Грёбнера, удовлетворяющий следующим условиям:

1. Все главные коэффициенты элементов базиса равны 1;
2. Для любых $p \in G(p) \notin \langle(G\backslash\{p\})\rangle$.

В нашем примере минимальный базис Грёбнера

$$G = \{x^2, xy, y^2 - \frac{1}{2}x\}$$

Каждый идеал может иметь несколько разных минимальных базисов Грёбнера. В нашем примере минимальным базисом Грёбнера является также

$$G = \{x^2 - \alpha xy, xy, y^2 - \frac{1}{2}x\}$$

для любых $\alpha \in K$. Нужно выбрать наилучший. Такой базис называется редуцированным

Определение 1.7. Редуцированным называется базис Грёбнера, удовлетворяющий следующим условиям:

1. Все главные коэффициенты элементов базиса равны 1;
2. Для любых $p \in G$ никакой из мономов не принадлежит $\langle(G\backslash\{p\})\rangle$.

В нашем примере первый из минимальных базисов также является редуцированным базисом Грёбнера.

Предложение 1.3. Пусть I - полиномиальный идеал, не содержащий нулей, и пусть определено некоторое мономиальное упорядочение. В таком случае, существует единственный редуцированный базис Грёбнера.

Доказательство.

Пусть полином $p \in G$ редуцирован по отношению к G, если ни один из его мономов не принадлежит $\langle(G\backslash\{g\})\rangle$. Заметим, что если g редуцирован по отношению к G, то он также редуцирован по отношению к любому другому минимальному базису Грёбнера G', который содержит g и оперирует тем же множеством главных членов, что и G.

Пусть $G' = (G - \{g\}) \cup \{\overline{g}^{G\backslash\{p\}}\}$ для какого-то полинома $g \in G$. Это множество также является минимальным базисом Грёбнера идеала I, потому что $(\overline{g}^{G\backslash\{p\}}) = (g)$, и $\langle(G')\rangle = \langle(G)\rangle$. Полином $g' = \overline{g}^{G\backslash\{p\}}$ также редуцирован по отношению к G'.

Если мы проведём эту процедуру для каждого элемента базиса G, то полученный базис Грёбнера будет редуцированным, потому что все его элементы редуцированы по отношению к нему.

Теперь нужно доказать единственность редуцированного базиса. Пусть у нас есть два редуцированных базиса G и G'. У них общее множество членов ([8]), поэтому для каждого полинома $g \in G$ есть элемент $g' \in G'$, такой что $(g) = (g')$.

Рассмотрим полином $g - g'$ для каждой пары g, g'. Он принадлежит I, поэтому $\overline{g - g'}^G = 0$. Ни один из мономов $g - g'$ не принадлежит (G) или (G'), потому что G и G' редуцированны, и главные члены g и g' уничтожились при вычитании. Поэтому $\overline{g - g'}^G = g - g' = 0$. Это показывает, что $G = G'$. Таким образом, редуцированный базис единственый. ■

Несмотря на то что алгоритм Бухбергера не применяется на практике, есть множество эффективных алгоритмов, основанных на нём. В действительности, наиболее ресурсозатратной операцией является редуцирование (деление) S-полинома на набор полиномов. Поэтому новые алгоритмы используют специальный критерий, чтобы уменьшить количество рассматриваемых S-полиномов и, следовательно, количество редуцирований.

Чтобы обозначить такой критерий, необходимо новое определение.

Определение 1.8. Пусть $G = \{g_1, ..., g_s\} \subset K[x_1, ..., x_n]$ - множество полиномов, для которого определено некоторое мономиальное упорядочение. Тогда функция f, принадлежащая $K[x_1, ..., x_n]$, называется редуцируемой к нулю по модулю $G (f \to_G 0)$, если она может быть выражена как

$$f = \alpha_1 g_1 + ... + \alpha_s g_s$$

где мультистепень функции больше или равна мультестепеней ненулевых компонентов.

Верна следующая теорема

Теорема 1.6. Базис $G = \{g_1, ..., g_s\}$ идеала I является базисом Грёбнера тогда и только тогда, когда все S-полиномы, созданные из его элементов, редуцируемы к нулю по модулю G.

Доказательство этой теоремы было получено во время доказательства теоремы 1.4. Этот факт может быть использован для первого критерия.

Предложение 1.4. Пусть есть множество $G \subset K[x_1, ..., x_n]$. Пусть f, g - элементы G, и их главные члены взаимнопросты. Тогда S-полином $S(f, g)$ редуцируем к 0 по модулю G.

Введём ещё одно определение, чтобы обозначить второй критерий.

Определение 1.9. Пусть $F = (f_1, ..., f_s)$ - набор полиномов. *Сизигией* главных членов набора F - набор многочленов $(h_1, ..., h_s) \subset (K[x_1, ..., x_n])^s$, такой что

$$\sum_{i=1}^{s} h_i * (f_i) = 0.$$

Обозначим подмножство $(K[x_1, ..., x_n])^s$, содержащее все сизигии главных членов набора F как $S(F)$. У этого подмножества есть конечный базис.

Обозначим сизигию, порождённую S-полиномом $S(f_i, f_j)$, следующим способом:

$$S_{ij} = \frac{x^\gamma}{(f_i)} e_i - \frac{x^\gamma}{(f_j)} e_j,$$

где e_i - i-й единичный вектор из $(K[x_1, ..., x_n])^s$. Можно показать, что сизигии также являются базисом $S(F)$, поэтому каждая сизигия S из этого множества может быть представлена как

$$S = \sum_{i<j} u_{ij} S_{ij},$$

где $u_{ij} \in K[x_1, ..., x_n]$.

Используя это определение, введём второй критерий для конструкции базисов Грёбнера.

Теорема 1.7. Базис $G = \{g_1, ..., g_s\}$ идеала I является базисом Грёбнера тогда и только тогда, когда выполняется следующее условие для всех элементов базиса $S = (h_1, ..., h_s)$ пространства сизигий $S(G)$:

$$S * G = \sum_{i=1}^{s} h_i g_i \to_G 0.$$

Чтобы использовать этот критерий, необходимо обозначить один факт.

Предложение 1.5. Пусть подмножество $S \subset \{S_{ij} : 1 \leq i < j \leq s\}$ - базис $S(G)$, где $G = \{g_1, ..., g_s\}$. Также предположим, что есть три полинома g_i, g_j, g_l, принадлежащие G, и такие что наименьшее общее кратное главных членов g_i и g_j делится на старший член g_l. Если S_{il} и S_{jl} принадлежат S, то $S \backslash \{S_{ij}\}$ является базисом $S(G)$.

Теперь мы можем составить улучшенную версию алгоритма Бухбергера, используя новые критерии. Доказательства всех вышеупомянутых критериев и следующего алгоритма приведены в [8].

Теорема 1.8. Базис Грёбнера полиномиального идеала $I = \langle f_1, ..., f_s \rangle$ можно построить с помощью следующего алгоритма за конечное количество шагов.

Вход: $F = (f_1, ..., f_s)$

Выход: базис Грёбнера $G = \{g_1, ..., g_t\}$ идеала I.

$B := \{(i, j) | 1 \leq i < j \leq s\}$;

$G := F$;

$t := s$;

WHILE $B \neq 0$ DO

Выбрать $(i, j) \in B$;

IF $LCM((f_i), (f_j)) \neq (f_i) * (f_j)$ AND Критерий (f_i, f_j, B) не выполнен THEN

$S := \overline{S(f_i, f_j)}^G$;

IF $S \neq 0$ THEN
$t := t + 1$;
$f_t := S$;
$G := G \cup f_t$;
$B := B \cup \{(i, j) | 1 \leq i < j \leq t - 1\}$
$B := B \backslash \{(i, j)\}$;

Выполнение критерия (f_i, f_j, B) означает, что есть $l \notin \{i, j\}$, для которого (i, l) и (j, l) не принадлежат B, и главный член f_l делит наименьшее общее кратное главных членов f_i и f_j.

Этот алгоритм намного оптимальнее, но всё ещё оставляет возможность для улучшения. Например, если мы расставим делители в порядке возрастания главных членов, то сможем сэкономить вычислительные ресурсы, потому что будет сокращено число сравнений в операции редуцирования. В 1985 году Бухбергер предложил выбирать пары (i, j) полиномов, таких что наименьшее общее кратное их старших членов минимально ([6]). Эта стратегия называется стратегией нормального выбора. Позже она была улучшена в так называемой сахарной стратегии. Все эти стратегии начали детально изучаться с начала 1980х. Более того, в 1985 году Бухбергер показал, как редуцировать G во время его расширения ([6]), чтобы сразу получить редуцированны базис.

Самые эффективные алгоритмы для нахождения базисов Грёбнера - это F4 и F5 Фожера, изобретённые Жаном-Шарлем Фожером, соответственно, в 1999 ([12]) и 2005 ([13]). Они значительно сокращают количество рассматриваемых S-полиномов и операций редукции во время вычисления базиса Грёбнера. В следующей главе мы рассмотрим алгоритм F5 и его модификации F5R и F5C.

Верхняя оценка времени исполнения алгоритма Бухбергера и используемой им памяти в обычном случае была найдена Томасом Дюбе в 1990 [10]. Исследование было опубликовано под названием "Структура полиномиальных идеалов и базисы Грёбнера". Дюбе доказал, что степени элементов базиса Грёбнера всегда ограничены сверху значением

$$2(\frac{d^2}{2} + d)^{2^{n-1}},$$

где n - количество переменных, d - максимальная полная степень входных полиномиальных уравнений. Это показывает, что алгоритм Бухбергера имеет двойную экспоненциальную сложность

$$d^{2^{n+o(1)}}.$$

С другой стороны, есть примеры, когда базис содержит степень

$$d^{2^{\Omega(n)}}.$$

Тем не менее, такие примеры очень редки. Поэтому это выражение можно считать нижней оценкой, и она не может быть уменьшена. Иными словами,

даже самые быстрые алгоритмы могут потребовать колоссальные объёмы памяти и времени для вычислений. Но эти случаи крайне редки. В дополнение стоит отметить, что в случае порождающих полиномов малой степени были достигнуты хорошие оценки для степеней элементов базиса Грёбнера.

Глава 2. Алгоритмы для вычисления базисов Грёбнера, основанные на сигнатурах.

Как говорилось ранее, самой затратной по времени операцией в алгоритме Бухбергера является редуцирование S-полинома. Более того, мы заинтересованы только в тех парах полиномов, чей S-полином не равен нулю после редуцирования. Такие пары называются критическими. Так как необходимое условие для базисов Грёбнера - это сведение к нулю всех возможных S-полиномов, видно, что большинство вычислений бесполезны и избыточны. Поэтому нужно понять, какие S-полиномы редуцируются в ноль, перед вычислением. Один из таких критериев был описан в улучшенном алгоритме Бухбергера.

В этой главе мы рассмотрим алгоритм F5, разработанный в 2002 году Жаном-Шарлем Фожером ([13]). Критерий этого алгоритма считается одним из самых эффективных. Алгоритм F5 использует так называемые полиномиальные сигнатуры.

Сигнатуры и помеченные полиномы.

Пусть $F = (f_1, ..., f_m)$ - упорядоченный набор полиномов из $K[x_1, ..., x_n]$ с определённым мономиальным упорядочением. Тогда $F \in K[x_1, ..., x_n]^m$. Пусть T обозначает множество всех возможных мономов.

Обозначим i-й единичный вектор в $K[x_1, ..., x_n]^m$ за F_i. Вектор $g = (g_1, ..., g_m)$ называется сизигией, если

$$\sum_{i=1}^{m} g_i f_i = 0.$$

С другой стороны,

$$g = \sum_{i=1}^{m} g_i F_i.$$

Введём новое упорядочение в $K[x_1, ..., x_n]^m$, определённое следующим способом:

$$\sum_{k=i}^{m} g_k F_k < \sum_{k=j}^{m} g_k F_k \Leftrightarrow \begin{cases} i > j \wedge h_j \neq 0 \\ i = j \wedge LM(g_i) < LM(h_i) \end{cases}$$

Заметим, что $F_i < F_j$ при $i > j$.

Для всех $g \in K[x_1, ..., x_n]^m$ есть индекс i, такой что $g = \sum_{k=i}^{m} g_k F_k$ и $g_i \neq 0$. Определим новую функцию $index(g)$, чьё значение равно этому индексу.

Далее, по новому упорядочению, $LM(g) = (g_i)F_i$. Пусть T_i - множество $\{tF_i | t \in T\}$. Объединение всех T_i - это множество индексов полиномов из идеала $\langle F \rangle$.

Пусть t - моном. Обозначим за $W(t)$ следующее множество:

$$\left\{ g \in K[x_1, ..., x_n]^m \quad | \quad LM\left(\sum_{i=1}^{m} g_i f_i\right) = t \right\}$$

Верны следующие утверждения:

Утверждение 2.1. Пусть $w(t)$ - минимальный элемент $W(t)$. Если мономы t_1 и t_2 различаются, то $LM(w(t_1)) \neq LM(w(t_2))$.

Утверждение 2.2. Определим новую функцию $\nu(p) = LM(w(LM(p)))$, где p - полином из идеала $\langle F \rangle$. Если полиномы p_1 и p_2 идеала $\langle F \rangle$ имеют разные главные мономы, то $\nu(p_1) \neq \nu(p_2)$.

Определение 2.1. Функция $\nu(p) = LM(w(LM(p)))$ исползуется в семействе алгоритмов F5 и называется полиномиальной сигнатурой.

Определение 2.2. Помеченный полином - это пара $r = (tF_i, f)$, где второй элемент - оригинальный полином, а первый - сигнатура этого полинома.

Пример 2.1. Предположим, что есть идеал $I = \langle g_1, g_2 \rangle \subset Q[x, y, z]$, где $g_1 = xy - z^4, g_2 = y^2 - z$. Используем лексикографическое мономиальное упорядочение. В этом случае, сигнатуры g_1 и g_2 равны соответственно $(1, 1)$ и $(1, 2)$.

Определим следующие операции над сигнатурами и помеченными полиномами, используемые в дальнейшем:

1. Тело помеченного полинома: $poly(r) = f$;
2. Сигнатура помеченного полинома: $S(r) = tF_i$;
3. Индекс помеченного полинома: $index(r) = i$;
4. Главный моном помеченного полинома: $LM(r) = LM(f)$;
5. Главный коэффициент помеченного полинома: $LC(r) = LC(f)$;
6. Редукция помеченного полинома по множеству полиномов G: $\overline{r}^G = \left(S(r), \overline{f}^G\right)$;
7. Произведение помеченного полинома на ненулевой элемент поля $\lambda \in K$: $\lambda r = (tF_i, \lambda f)$;
8. Произведение помеченного полинома на моном u: $ur = (utF_i, uf)$;
9. Произведение сигнатуры $w = tF_i$ на моном ν: $\nu w = (\nu t)F_i$;

Помеченный полином называется допустимым, если существует $g = (g_1, ..., g_m)$, такой что $LM(g) = S(r)$.

Обозначим множество всех помеченных полиномов как R.

Алгоритм F5.

Алгоритм F5 был изобретён Жаном-Шарлем Фожером в 2002 году. Этот алгоритм является поэтапным, он последовательно находит базис Грёбнера для наборов $(f_m), (f_{m-1}, f_m), (f_{m-2}, f_{m-1}, f_m), ..., (f_1, ..., f_m)$. Алгоритм не использует критерий Бухбергера, чтобы находить критические пары полиномов. Вместо этого он использует свой критерий.

Введём следующие обозначения, которые важны для алгоритма.

Определение 2.3. Пусть P - конечное подмножество R, и s, t - помеченные полиномы, для которых $poly(s) = f, poly(t) = g$, где $f, g \neq 0$. Если f можно выразить в форме

$$f = \sum_{p \in P} \mu_p poly(p),$$

где для каждого $p \in P$ с ненулевым телом выполняется следующее условие:

$$LM(\mu_p)LM(p) \leq LM(t); \quad LM(\mu_p)S(p) \leq S(s),$$

то это представление называется t-представлением помеченного полинома s по отношению к P. Это свойство обозначается как $f = O_p(t)$. Если есть другой помеченный полином t', такой что $LM(t') \leq LM(t), S(t') \leq S(t)$ и $f = O_p(t')$, то это свойство обозначается как $f = o_p(t)$.

Определение 2.4a. Помеченный полином r с сигнатурой tF_k нормализован по отношению к набору полиномов $F = (f_1, ..., f_m)$, если $t \notin LM\langle F\rangle$.

Определение 2.4b. Пара из помеченного полинома и монома нормализована по отношению к набору полиномов, если их произведение нормализовано.

Определение 2.4c. Пара помеченных многочленов (r_i, r_j) нормализована по отношению к набору полиномов, если выполняется следующее:

1. $S(r_j) < S(r_i)$;
2. Пары (u_i, r_i) и (u_j, r_j) нормализованы, где $u_i = \frac{w_{i,j}}{LM(r_i)}, u_j = \frac{w_{i,j}}{LM(r_i)}, w_{i,j} = LCM(LM(r_i), LM(r_j))$.

S-полином нормализованной пары помеченных полиномов (r_i, r_j) - это следующий помеченный полином:

Пример 2.2. Для примера выше

$$g_3 = S(g_2, g_1) = xg_2 - yg_1 = x(y^2 - z) - y(xy - z^4) = -xz + yz^4$$

$$g_4 = S(g_3, g_1) = yg_3 + zg_1 = y(-xy + yz^4) + z(xy - z^4) = y^2z^4 - z^5$$

$$S(g_3) = xS(g_2) = x(1, 2) = (x, 2)$$

$$S(g_4) = yS(g_3) = y(x, 2) = (xy, 2)$$

Алгоритм F5 рассматривает только нормализованные пары помеченных полиномов (r_i, r_j), для которых $S(u_j, r_j) < S(u_i, r_i)$, так что он использует определённое упорядочение. Также мы увидим, что алгоритм использует только сизигии, порождённые S-полиномами.

Используя введённые термины, сформулируем новый критерий. Алгоритм F5 будет его использовать, чтобы выбросить необязательные помеченные полиномы.

Теорема 2.1. Пусть $F = (f_1, ..., f_m)$ - упорядоченный набор полиномов, и $R = (r_1, ..., r_s)$ - набор допустимых помеченных полиномов, таких что $F \subset poly(R)$. Предположим, что для каждой нормализованной пары $(r_i, r_j) \in R^2$

$$S\left(poly(r_i), poly(r_j)\right) = o_{poly(R)}(u_i r_i) \quad (= 0),$$

где $u_i = \frac{LCM(LM(poly(r_i)), LM(poly(r_j)))}{LM(poly(r_i))}$. В этом случае множество $G = poly(R)$ является базисом Грёбнера идеала, порождённого F.

Этот критерий является основным, но не единственным в F5. В дополнение, алгоритм создаёт правила в процессе выполнения, в зависимости от того, какой новый помеченный полином может быть выражен через произведение старого и монома. Если результирующий S-полином может быть выражен через какой-либо из его двух аргументов, то пара отбрасывается.

Иначе, простыми словами, алгоритм F5 использует следующие критерии:

1. Критерий F5: S-полином $S(p, q) = LC(q)u_p p - LC(p)u_q q$ может быть пропущен при вычислении базиса Грёбнера во время итерации l, если:
   (a) При выражении $S(p) = tF_l$ существует $g \in G_{l-1}$, такое что $LM(g)$ делит $u_p t$;
   (b) При выражении $S(q) = tF_l$ существует $g \in G_{l-1}$, такое что $LM(g)$ делит $u_q t$.
2. Критерий перезаписывания: S-полином $S(p, q) = LC(q)u_p p - LC(p)u_q q$ может быть пропущен при вычислении базиса Грёбнера во время итерации l, если:
   (a) При выражении $S(p) = tF_l$ существует $g \in G_{l-1}$, вычисленное после p и такое, что $S(g) = \nu F_l$, и ν делит $u_p t$;
   (b) При выражении $S(q) = tF_l$ существует $g \in G_{l-1}$, вычисленное после q и такое, что $S(g) = \nu F_l$, и ν делит $u_q t$;

Пример 2.3. В примере выше $S(g_4) = S(S(g_3, g_1)) = (xy, 2)$ и $LM(g_1) = xy$. Моном xy делит xy, поэтому не нужно вычислять g_4, так как он будет отредуцирован к нулю.

Нужно отметить, что конечность алгоритма F5 была доказана только для регулярных последовательностей полиномов.

Определение 2.5. Последовательность полиномов $F = (f_1, ..., f_m) \in K[x_1, ..., x_n]^m$ называется регулярной, если для каждого i, такого что $1 \leq i \leq m$,

$$\forall g \in K[x_1, ..., x_n] \quad gf_i \in \langle f_{i+1}, ..., f_m \rangle \Rightarrow g \in \langle f_{i+1}, ..., f_m \rangle$$

Определить, является ли последовательность полиномов регулярной, - сложная задача. Например, известно, что любая переопределённая система уравнений не регулярна. Это вызывает определённые сложности при работе с системами уравнений в конечных полях, потому что там используются дополнительные уравнения в форме $x^q + x = 0$. В алгоритме F5 есть специальный способ, позволяющий определить, регулярна ли последовательность полиномов. Если последовательность не реглярна, то операция редуцирования результирует с нулём. Фожер доказал, что в случае регулярной последовательности все операции редуцирования во время нахождения базиса Грёбнера дадут ненулевой результат, который потом используется для получения новых элементов базиса. Также Фожер отметил, что алгоритм можно модифицировать так, чтобы он останавливался в случае нерегулярной последовательности.

Здесь не приведён псевдокод алгоритма из-за его большого объёма. Он полностью доступен в работах Фожера и Стёжерса. Вместо этого дадим краткий обзор его главных процедур и функций.

1. IncrementalF5 - главное тело алгоритма. На вход поступает последовательность полиномов $F = (f_1, ..., f_m) \in K[x_1, ..., x_n]^m$. На выходе принимается базис Грёбнера $G = \langle F \rangle$. В начале процедура создаёт список для правил, по которым с помощью старых полиномов будут выражаться новые. После этого совершается m итераций, на каждой из которых происходит вычисление промежуточнго базиса Грёбнера $G_i = \langle f_{m+1-i}, ..., f_m \rangle$.

2. AlgorithmF5 - вычисляет промежуточный базис Грёбнера, используя следующие действия:

   - Создание списка критических пар помеченных полиномов.
   - Создание списка S-полиномов, используя критические пары с наименьшей степенью.
   - Редуцирование S-полиномов, используя ранее полученный базис Грёбнера.
   - Создание нового списка критических пар, используя результаты редуцирования S-полинома.

   Процедура повторяет последние три шага, пока не обработает все возможные критические пары помеченных полиномов.

3. CritPair - создаёт критические пары полиномов, удаляя те, которые не принимаются алгоритмом F5.

4. Spol - создаёт S-полиномы, используя критические пары и удаляя те, которые не подходят по критерию перезаписывания. Также создаёт правила для новых полиномов.

5. Reduction - совершает редуцирование S-полиномов, используя ранее найденный базис Грёбнера, основанный на критериях F5 и перезаписывания.

6. AddRule - вызывается каждый раз, когда создаётся новый S-полином, и добавляет новое правило в список.

7. Rewritten? - проверяет, проходит ли критерий перезаписывания.

Алгоритм F5R.

Несмотря на то что алгоритм F5 довольно эффективен, у него есть существенный недостаток. Он не редуцирует промежуточные базисы Грёбнера. В результате, промежуточные базисы содержат слишком много избыточных элементов к концу алгоритма. Из этого следует, что

1. Алгоритм тратит больше времени на операции редуцирования, потому что ему нужно проверить больше делителей.

2. Создаётся большое количество S-полиномов.

Получается, что надо использовать редуцированные промежуточные базисы Грёбнера в алгоритме F5. Основная сложность в том, что для этого нужно отслеживать сигнатуры полиномов в промежуточных базисах. Одна из версий улучшения алгоритма была предложена Тиллем Стёжерсом в 2005 году ([15]) и была названа F5R. Отличие от F5 в том, что после каждой итерации промежуточный базис G_i редуцируется к базису B_i. Новый базис не меняет сигнатуры полиномов и используется для дальнейшего редуцирования, пока с помощью G_i создаются новые S-полиномы. Операции редуцирования используются в двух подалгоритмах F5: CritPair и Reduction. Это значительно сокращает количество операций редуцирования, но количество рассматриваемых S-полиномов остаётся прежним.

Алгоритм F5C.

Алгоритм F5C был представлен Джоном Перри и Кристианом Эдером в 2009 году ([11]). Как и F5R, он использует редуцированные версии промежуточных базисов Грёбнера для расчётов. Но редуцированные базисы используются не только для редуцирования S-полиномов, но и для создания новых критических пар помеченных полиномов. F5C решил проблему отслеживания

сигнатур элементов редуцированного базиса. В алгоритме F5C правила для помеченных полиномов стираются после получения каждого промежуточного базиса Грёбнера, а затем составляются заново для элементов редуцированного базиса в процедуре SetupReducedBasis:

1. Пусть G_i - промежуточный базис Грёбнера и результат i-й итерации алгоритма. Саначала вычисляется редуцированный базис B_i.

2. Для каждого элемента b_j редуцированного базиса B_i определяется сигнатура F_{i+j-1}.

3. Все правила стираются.

4. Добавляются новые правила для каждого возможного S-полинома в базисе B_i. Пусть (b_j, b_k) - пара полиномов. Тогда правило будет иметь вид $(uF_{i+j-1}, 0)$, где $u = \frac{LCM(LM(b_j),LM(b_k))}{LM(b_j)}$, и 0 - сигнатура нулевого полинома. Те правила, которые используют сигнатуру нулевого полинома, более предпочтительны, поэтому они будут проверены раньше других во время проверки критерия перезаписывания.

В наши дни алгоритм F5C является одним из наиболее эффективных для получения базисов Грёбнера. В отличие от алгоритма F5, который может находить базисы для систем типов Cyclic-8 и Cyclic-9, алгоритм F5C может получать базисы для систем типа Cyclic-10

$$C_{10}(x) = \left\{ \sum_{j=0}^{9} \prod_{k=j}^{j+i-1} x_{kmod10} = 0, i = \overline{1,10} \right\}$$

достаточно быстро ([11]).

Глава 3. Решение систем уравнений в F_q.

Метод Бухбергера.

Предположим, у нас есть система уравнений

$$\begin{cases} f_1 = 0 \\ f_2 = 0 \\ ... \\ f_s = 0 \end{cases},$$

где $f_1, ..., f_s \in F_q[x_1, ..., x_n]$. Пусть $F = (f_1, ..., f_s)$ - упорядоченный набор полиномов из $F_q[x_1, ..., x_n]$ с лексикографическим упорядочением. Например, в случае 2-х переменных это будет выглядеть так:

$$1 < x_2 < x_2^2 < x_1 < x_1x_2 < x_1x_2^2 < x_1^2 < x_1^2x_2 < x_1^2x_2^2 < ...$$

Определим также упорядочение для главных членов полиномов и их произведений с элементами F_q. Допустим, что каждый из полиномов в системе имеет следующую форму:

$$f = c_{\alpha_1} x^{\alpha_1} + ... + c_{\alpha_m} x^{\alpha_m}, \quad c_{\alpha_i} \in F_q,$$

где все мультистепени расставлены в порядке убывания $\alpha_1 > \alpha_2 > ... > \alpha_m$. Теперь можно определить все главные коэффициенты, главные мономы и главные члены.

$$LC(f) = c_{\alpha_1},$$
$$LM(f) = x^{\alpha_1},$$
$$(f) = c_{\alpha_1} x_{\alpha_1}$$

Пусть $I = \langle f_1, ..., f_n \rangle$ - идеал системы. Используем алгоритм для нахождения базиса Грёбнера этой системы. В результате получим новую систему G

$$\begin{cases} g_1 = 0 \\ g_2 = 0 \\ ... \\ g_s = 0 \end{cases},$$

которая является треугольной и возможно содержит уравнение для одной переменной $x_i \in X$. Если мы решим это уравнение, например, методом Берлекампа [4], то сможем подставить полученные значения x_i в G и повторить процесс, пока система не решится.

Возникает вопрос, почему базис Грёбнера всегда получается в треугольной форме. Это следствие из теоремы об исключении. Введём следующее обозначение.

Определение 3.1. Пусть $I = \langle f_1, ..., f_s \rangle \in K[x_1, ..., x_n]$. В этом случае идеал $I_j = I \cap K[x_{i+1}, ..., x_n] \subset K[x_{i+1}, ..., x_n]$ называется l-м идеалом исключения.

В действительности, идеал состоит из всех следствий решения уравнения $f_1 = ... = f_s = 0$, которые зависят только от переменных $x_{l+1}, ..., x_b$. Идеал I равен I_0. Если поменять порядок переменных, получим другие l-е идеалы исключения. Тогда чтобы избавиться от переменных $x_1, ..., x_l$, нужно найти ненулевые полиномы в l-м идеале исключения I_l. Для этого как-то нужно получить элементы I_l. Это можно сделать, используя следующую теорему.

Теорема 3.1 (об исключении, [8]). Пусть $I \in K[x_1, ..., x_n]$ - идеал, а G - его базис Грёбнера относительно лексикографического мономиального упорядочения $x_1 > x_2 > ... > x_n$. Тогда для каждого l от 0 до n множество $G_l = g \cap K[x_{l+1}, ..., x_n]$ является базисом Грёбнера l-го исключающего идеала I_l.

Поэтому построение базисов Грёбнера в случае lex-упорядочения последовательно исключает все переменные, и результат имеет треугольную форму.

Чтобы получить такой результат, необязательно использоать lex-упорядочение. Можно использовать какое-нибудь так называемое l-е исключающее упорядочение, такое что каждый моном, содержащий какую-либо из переменных $x_1, ..., x_l$, больше любого монома из $K[x_{l+1}, ..., x_n]$. Существует более общая версия теоремы 3.1, основанная на каком-либо l-ь исключающем упорядочении, включая и lex-упорядочение.

Описанный метод решения нелинейных систем уравнений завершается либо если найдёт значения всех переменных, либо если придёт к треугольной форме, не содержащей выражения с одной переменной. Верно следующее утверждение: система не имеет решений тогда и только тогда, когда соответствующий базис Грёбнера содержит ненулевую константу.

Поэтому можно считать, что этот метод является нелинейным обобщением метода Гаусса, который используется для нахождения решения систем линейных уравнений. Стоит отметить, что использование лексикографического мономиального упорядочения приводит к образованию базиса Грёбнера самой высокой степени и требует самых высоких затрат времени и памяти, по сравнению с другими упорядочениями.

Пример решения системы уравнений в F_2.

В теории кодирования и криптографии часто нужно иметь дело с уравнениями с коэффициентами в конечных полях. Предположим, что нужно решить следующее уравнение для взлома криптосистемы:

$$x^9 + \alpha x + \alpha^{13} = 0.$$

Корни этого уравнения находятся в F_{16}. Поле F_{16} является кольцом вычетов $F_2[x]$ с I, где I - идеал, порождённый нередуцируемым полиномом $f = x^4 + x + 1 \in F_2[x]$. Элементы поля могут быть представлены либо в степенном базисе $A = (1, \alpha, \alpha^2, \alpha^3)$, либо в нормальном базисе $B = (\beta, \beta^2, \beta^4, \beta^8)$, где $\beta = \alpha^3, \alpha$ - примитивный элемент F_{16}. Тогда в нормальном базисе

$$\forall x \in F_{16} \quad x = x_0\beta + x_1\beta^2 + x_2\beta^4 + x_3\beta^8,$$

$$\alpha = \beta + \beta^8,$$

$$\alpha^{13} = \beta + \beta^4 + \beta^8$$

Подставляя эти значения в исходное уравнение, получим следующую систему нелинейных квадратичных уравнений с переменными $x_0, x_1, x_2, x_3 \in F_2$:

$$x_0x_1 + x_1x_2 + x_1x_3 + x_1 + x_2x_3 + x_3 = 1$$

$$x_0x_2 + x_0x_3 + x_0 + x_1x_2 + x_1 + x_2x_3 + x_3 = 0$$

$$x_0x_1 + x_0x_3 + x_1x_3 + x_1 + x_2x_3 + x_2 + x_3 = 1$$

$$x_0x_1 + x_0x_2 + x_0x_3 + x_1x_2 + x_1 + x_2 = 1$$

Также добавим к этим уравнениям другие, которые переопределяют систему свойствами элементов F_2.

$$x_i^2 + x_i = 0, \quad i = \overline{0,3}$$

Используя лексикографическое мономиальное упорядочение $x_0 > x_1 > x_2 > x_3$, получим базис Грёбнера системы:

$$x_2^2 + x_2 = 0$$

$$x_3^2 + x_3 = 0$$

$$x_0 + x_3 = 1$$

$$x_1 = 1$$

$$x_2x_3 + x_2 + x_3 = 1$$

Теперь можно легко получить три решения оригинального уравнения:

- $x_0 = 0, x_1 = 1, x_2 = 1, x_3 = 0 \Rightarrow x = \alpha^3 + \alpha^6 + \alpha^{12} = \alpha^7$
- $x_0 = 0, x_1 = 1, x_2 = 0, x_3 = 1 \Rightarrow x = \alpha^6 + \alpha^9 = \alpha^5$
- $x_0 = 0, x_1 = 1, x_2 = 1, x_3 = 1 \Rightarrow x = \alpha^9 + \alpha^6 + \alpha^{12} = \alpha^{14}$

Вывод.

В этой работе была изучена задача нахождения решения системы нелинейных полиномиальных уравнений. Во-первых, были рассмотрены известные методы решения таких систем. Были отмечены их главные свойства и области применения. В дополнение, изучена сложность этих методов.

Во-вторых, мы дали определение базисам Грёбнера полиномиального идеала. Чтобы это сделать, необходимо было ввести определение мономиального упорядочения и показать, что все полиномиальные идеалы конечно порождаемы с помощью леммы Диксона и теоремы Гильберта о базисе. Также для построения этих базисов был введён алгоритм деления полиномов многих переменных. Дав определение S-полинома, мы смогли дать определение базисам, которым посвящена данная работа. Также были описаны их главные свойства.

В-третьих, мы описали алгоритм для построения требуемого базиса Грёбнера, введённый Бухбергером. Этот алгоритм позволил получаать базисы Грёбнера, в идеале, порождённом любым множеством полиномов, за конечное время. Этот алгоритм позволил преобразовать системы линейных уравнений в форму, в которой есть уравнения с одной переменной, но, возможно, в больших степенях. Как было описано, сложность таких алгоритмов зависит от количества вычислений остатков от деления на набор полиномов. Поэтому рассматривался вопрос о сокращении количества S-полиномов.

В-четвёртых, мы описали и сравнили между собой семейство алгоритмов F5 для построения базисов Грёбнера. Мы дали описание критериям для исключения критических пар, основанным на сигнатурах полиномов в оригинальном алгоритме F5, введённом Жаном-Шарлем Фожером, и показали, как можно улучшить данный алгоритм, описав его модификации F5R и F5C, предложенные Тиллем Стёжерсом и Джоном Перри.

В-пятых, мы описали метод решения систем уравнений в конечных полях с помощью создания базиса Грёбнера. Этот метод использует лексикографическое мономиальное упорядочение, которое позволяет построить базис в треугольной форме уравнений, последовательно исключая переменные. Единственное неудобство заключается в том, что это упорядочение создаёт базис Грёбнера с большими степенями.

Мы узнали, что время и сложность алгоритма построения базиса Грёбнера очень зависит от количества и степени переменных в системе, а также от мономиального упорядочения. Известно, что эти методы имеют очень большую сложность в худшем случае. Однако, также известно, что вероятность такого случая крайне мала, и большинсво алгоритмов для построения базисов Грёбнера намного быстрее. С появлением таких алгоритмов, как F5, F5C и улучшенный алгоритм Бухбергера появляется надежда, что распространённое мнение о сложности и неудобства работы с базисами Грёбнера, изменится, и этот пессимизм пройдёт.

Источники

[1] Агибалов Г.П. Методы решения систем полиномиальных уравнений над конечным полем // Вестник ТГУ. Приложение. 2006. No 17. С. 4 - 9

[2] Арри, А., Перри Дж. The F5 Criterion revised, 2011. Journal of Symbolic Computation, 46(2):1017–1029, June 2011. Доступно онлайн на arxiv.org/abs/1012.3664.

[3] Аржанцев И.В., 2003. Базисы Грёбнера и системы алгебраических уравнений. ISBN 5-94057-095-X

[4] Берлекэмп Э., 1968. Алгебраическая теория кодирования. McGraw Hill. ISBN 0-89412-063-8.

[5] Бухбергер Б., 1965. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph.D. Thesis, University of Innsbruck. (на немецком) Перевод на английский в работе Бухбергера (2006).

[6] Бухбергер Б., 1985. Gröbner bases: An algorithmic method in polynomial ideal theory. In: (Bose, N. K., ed.) Recent Trends in Multidimensional Systems Theft T, D. Reidel.

[7] Бухбергер Б., 2006. Bruno Buchberger's PhD thesis 1965: an algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. J. Symbolic Comput. 41 (3–4), 475–511. Перевод работы 1965 года с немецкого на английский - Абрамсон М. П.

[8] Кокс Д., Литтл Дж., О'Ши Д., 1997. Идеалы, многообразия и алгоритмы: Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. Springer. ISBN 0-387-94680-2.

[9] Дэвенпорт Дж., Сирэ И., Турнье Э., 1991. Компьютерная алгебра. ISBN 978-0-12-204230-0

[10] Дюбе Т., 1990. The Structure of Polynomial Ideals and Gröbner Bases. SIAM J. Comput. 19(4): 750-773

[11] Эдер, Ч., Перри, Дж., 2010. F5C: A Variant of Faugere's F5 Algorithm With Reduced Gröbner Bases. Journal of Symbolic Computation, 45(12), 1442-1458.

[12] Фожер Ж.-Ш., 1999. A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139 (1-3), 61-88

[13] Фожер Ж.-Ш., 2002. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: ISSAC '02: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation. ACM Press, New York, NY, USA, pp. 75–83.

[14] Лидль Р.; Нидеррейтер Г., 1997. Finite Fields (2е изд.), Cambridge University Press, ISBN 0-521-39231-4

[15] Стежерс Т., 2006. Faugere's F5 algorithm revisited. Cryptology ePrint Archive, Report 2006/404.