СҮЛЕЙМАН ДЕМИРЕЛЬ УНИВЕРСИТЕТІ

ИНЖЕНЕР ФАКУЛЬТЕТІ

«МАТЕМАТИКА ЖӘНЕ ЖАРАТЫЛЫСТАНУ ҒЫЛЫМДАРЫ»

КАФЕДРАСЫ

НОВИКОВ АЛГЕБРАЛАРЫ ҮШІН ҚАРАСТЫРЫЛҒАН БУХБЕРГЕР АЛГОРИТМІ

КУРСТЫҚ ЖҰМЫС

6М060100 - Математика

Магистрант:_____**Ілияс Д.Т.**

Ғылыми жетекші:_____ **ф.м.ғ.к. Туленбаев К.М.**

АЛМАТЫ – 2013

Аңдатпа

Берілген курстық жұмыс Бухбергер алгоритмі арқылы бикомутативтік алгебралардағы Гребнер базисын табуға арналған.

Аннотация

Данная курсовая работа посвящена описанию реализации алгоритма Бухбергера нахождения базиса Гребнера бикоммутативных алгебр.

Abstract

Given course work describes the Buchberger algorithm to defind Gröbner basis for bicommutative algebra.

Content

Introdaction

Grobner bases allow us to solve algorithmic problems on polynomial ideals. The method of Gröbner bases is implemented for all sufficiently powerful computer algebra systems and is used for the study of polynomial ideals, resulting inapplied problems. One of the important problems of computer algebra is the solution of systems of nonlinear algebraic equations. In practice it is often necessary to solve systems of nonlinear algebraic equations with integer coefficients. One of the methods is to construct a bases Grobner. The theoretical complexity of this algorithm, however, is that one can hardly expect a successful resolution of systems arising in practice. Until recently this was true, and the algorithm could be used mainly for academic purposes. However, in recent years has been significant progress in increasing the performance of the algorithm Buchberger, which allowed us to begin to address and successfully lead to the standard form of previously inconceivable amount. It should be emphasized that progress in this area has been made much more by improving the algorithms, rather than increase the speed of computers. Despite the presence in the theory of systems of equations, which reached the boundary of the worst of Gröbner bases, in practice, for real systems, its performance is significantly higher. In this work we show that what is the reduction and S-polynomial and how to construct a Grobner basis, and also was written Buchberger algorithm.

Chapter 1. Theoretical aspects of Algebras and ideals.

1.1 Basic properties of vector spaces

As we have seen in the introduction, a vector space is a set V with two operations: addition of vectors and scalar multiplication. These operations satisfy certain properties, which we are about to discuss in more detail. The scalars are taken from a field F, where for the remainder of these notes F stands either for the real numbers R or the complex numbers C.

The real and complex numbers are examples of fields. The abstract definition of a field and further examples are studied in algebra courses, such as the MAT 150 series.

Vector addition can be thought of as a map $+ : V \times V \to V$, mapping two vectors $u, v \in V$

to their sum $u+ v \in V$. Scalar multiplication can be described as a map $F \times V \to V$, which assigns to a scalar $a \in F$ and a vector $v \in V$ a new vector av.

Definition 1. A vector space over F is a set V together with the operations of addition $V \times V \to V$ and scalar multiplication $F \times V \to V$ satisfying the following properties:

1. Commutativity: $u + v = v + u$ for all $u, v \in V$;

2. Associativity: $(u + v) + w = u + (v + w)$ and $(ab)v = a(bv)$ for all $u, v, w \in V$ and $a, b \in F$;

3. Additive identity: There exists an element $0 \in V$ such that $0 + v = v$ for all $v \in V$;

4. Additive inverse: For every $v \in V$, there exists an element $w \in V$ such that $v+w = 0$;

5. Multiplicative identity: $1v = v$ for all $v \in V$;

6. Distributivity: $a(u + v) = au + av$ and $(a + b)u = au + bu$ for all $u, v \in V$ and $a, b \in F$.

Usually, a vector space over R is called a real vector space and a vector space over C is called a complex vector space. The elemens $v \in V$ of a vector space are called vectors.

Vector spaces are very fundamental objects in mathematics. Definition 1 is an abstract definition, but there are many examples of vector spaces. You will see many examples of vector spaces throughout your mathematical life. Here are just a few:

Example 1. Consider the set Fn of all n-tuples with elements in F. This is a vector space. Addition and scalar multiplication are defined componentwise. That is, for $u = (u1, u2, \ldots, un)$, $v = (v1, v2, \ldots, vn) \in Fn$ and $a \in F$, we define

$$u + v = (u1 + v1, u2 + v2, \ldots, un + vn),$$
$$au = (au1, au2, \ldots, aun).$$

It is easy to check that all properties of Definition 1 are satisfied. In particular the additive identity $0 = (0, 0, \ldots, 0)$ and the additive inverse of u is

$$-u = (-u1, -u2, \ldots, -un).$$

Special cases of Example 1 are Rn, in particular R2 and R3. We have already seen in the introduction that there isgtb a geometric interpretation for elements in R2 and R3 as points in
the plane and 3-space, respectively.

Example 2. Let $F\infty$ be the set

$$F\infty = \{(u1, u2, \ldots) \mid uj \in F \text{ for } j = 1, 2, \ldots\}.$$

Addition and scalar multiplication are defined as expected

$$(u1, u2, \ldots) + (v1, v2, \ldots) = (u1 + v1, u2 + v2, \ldots),$$
$$a(u1, u2, \ldots) = (au1, au2, \ldots).$$

You should verify that with these operations F∞ becomes a vector space.

Example 3. Verify that V = {0} is a vector space!

Example 4. Let P(F) be the set of all polynomials p : F → F with coefficients in F. More precisely, p(z) is a polynomial if there exist a0, a1, . . . , an ∈ F such that

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0. \quad (1)$$

Addition and scalar multiplication are defined as

$$(p + q)(z) = p(z) + q(z),$$

$$(ap)(z) = ap(z),$$

where p, q ∈ P(F) and a ∈ F. For example, if p(z) = 5z + 1 and q(z) = 2z2 + z + 1, then (p + q)(z) = 2z2 + 6z + 2 and (2p)(z) = 10z + 2. Again, it can be easily verified that P(F) forms a vector space over F. The additive identity in this case is the zero polynomial, for which all coefficients are equal to zero. The additive inverse of p(z) in (1) is $-p(z) = -a_n z^n - a_{n-1} z^{n-1} - \cdots - a_1 z - a_0$.

2 Elementary properties of vector spaces

We are going to prove several important, yet simple properties of vector spaces. From now on V will denote a vector space over F.

Proposition 1. Every vector space has a unique additive identity.

Proof. Suppose there are two additive identities 0 and 0′. Then

$$0' = 0 + 0' = 0,$$

where the first equality holds since 0 is an identity and the second equality holds since 0′ is an identity. Hence 0 = 0′ proving that the additive identity is unique.

Proposition 2. Every v ∈ V has a unique additive inverse.

Proof. Suppose w and w′ are additive inverses of v, so that v +w = 0 and v +w′ = 0. Then

$$w = w + 0 = w + (v + w') = (w + v) + w' = 0 + w' = w'.$$

Hence $w = w'$ as desired.

Since the additive inverse of v is unique as just shown, it will from now on be denoted tby $-v$. We define $w - v$ to mean $w + (-v)$. We will in fact show in Proposition 5 that $-v = -1v$.

Proposition 3. $0v = 0$ for all $v \in V$.

Note that the 0 on the left hand side in Proposition 3 is a scalar, whereas the 0 on the right hand side is a vector.

Proof. For $v \in V$ we have

$$0v = (0 + 0)v = 0v + 0v,$$

using distributivity. Adding the additive inverse of $0v$ to both sides we obtain

$$0 = 0v - 0v = (0v + 0v) - 0v = 0v.$$

Proposition 4. $a0 = 0$ for every $a \in F$.

Proof. Similarly to the proof of Proposition 3, we have for $a \in F$

$$a0 = a(0 + 0) = a0 + a0.$$

Adding the additive inverse of $a0$ to both sides we obtain $0 = a0$ as desired.

Proposition 5. $(-1)v = -v$ for every $v \in V$.

Proof. For $v \in V$ we have

$$v + (-1)v = 1v + (-1)v = (1 + (-1))v = 0v = 0,$$

which shows that $(-1)v$ is the additive inverse $-v$ of v.

3 Subspaces

Definition 2. A subset $U \subset V$ of a vector space V over F is a subspace of V if U itself is

a vector space over F.

To check that a subset $U \subset V$ is a subspace, it suffices to check only a couple of the conditions of a vector space.

Lemma 6. Let $U \subset V$ be a subset of a vector space V over F. Then U is a subspace of V if and only if

1. **additive identity:** $0 \in U$;
2. **closure under addition:** $u, v \in U$ implies $u + v \in U$;
3. **closure under scalar multiplication:** $a \in F$, $u \in U$ implies that $au \in U$.

Proof. 1 implies that the additive identity exists. 2 implies that vector addition is welldefined and 3 ensures that scalar multiplication is well-defined. All other conditions for a vector space are inherited from V since addition and scalar multiplication for elements in U are the same viewed as elements in U or V .

Example 5. In every vector space V , the subsets $\{0\}$ and V are trivial subspaces.

Example 6. $\{(x1, 0) \mid x1 \in R\}$ is a subspace of R2.

Example 7. $U = \{(x1, x2, x3) \in F3 \mid x1 + 2x2 = 0\}$ is a subspace of F3. To see this we need to check the three conditions of Lemma 6. The zero vector $(0, 0, 0) \in F3$ is in U since it satisfies the condition $x1 + 2x2 = 0$. To show that U is closed under addition, take two vectors $v = (v1, v2, v3)$ and $u = (u1, u2, u3)$. Then by the definition of U we have $v1+2v2 = 0$

and $u1 + 2u2 = 0$. Adding these two equations it is not hard to see that then the vector $v + u = (v1 + u1, v2 + u2, v3 + u3)$ satisfies $(v1 + u1) + 2(v2 + u2) = 0$.

Hence $v + u \in U$.

Similarly, to show closure under scalar multiplication, take $u = (u1, u2, u3) \in U$ and $a \in F$. Then $au = (au1, au2, au3)$ satisfies the equation

$au1 + 2au2 = a(u1 + 2u2) = 0$, so that $au \in U$.

Example 8. $\{p \in P(F) \mid p(3) = 0\}$ is a subspace of P(F).

Example 9. The subspaces of R2 are $\{0\}$, all lines through the origin, and R2. The subspaces of R3 are $\{0\}$, all lines through the origin, all planes through the origin, and R3. In fact, these exhaust all subspaces of R2 and R3, respectively. To prove this we will need further tools such as the notion of bases and dimensions to be

discussed soon. In particular this shows that lines or planes that do not pass through the origin are not subspaces (this is not so hard to show!).

For all examples above, check that the conditions of Lemma 6 are satisfied.

Note that if U and U′ are subspaces of V , then their intersection $U \cap U'$ is also a subspace(see Homework 2 and Figure 2). However, the union of two subspaces is not necessarily a
subspace. Think for example of the union of two lines in R2.

4 Sums and direct sums

Throughout this section V is a vector space over F and U1, $U2 \subset V$ denote subspaces.

Definition 3. Let U1, $U2 \subset V$ be subspaces of V . Define the sum of U1 and U2 as

$$U1 + U2 = \{u1 + u2 \mid u1 \in U1, u2 \in U2\}.$$

Check as an exercise that U1 + U2 is a subspace of V . In fact, U1 + U2 is the smallest subspace of V that contains both U1 and U2.

Example 10. Let

$$U1 = \{(x, 0, 0) \in F3 \mid x \in F\},$$
$$U2 = \{(0, y, 0) \in F3 \mid y \in F\}.$$

Then

$$U1 + U2 = \{(x, y, 0) \in F3 \mid x, y \in F\}. \quad (2)$$

If alternatively $U2 = \{(y, y, 0) \in F3 \mid y \in F\}$ then (2) still holds.

If U = U1+U2, then for any $u \in U$ there exist $u1 \in U1$ and $u2 \in U2$ such that u = u1+u2. If it so happens that u can be uniquely written as u1 + u2, then U is the direct sum of U1 and U2.

Definition 4. Suppose every $u \in U$ can be uniquely written as u = u1 +u2 for $u1 \in$ U1 and $u2 \in U2$. Then

$U = U1 \oplus U2$

is the direct sum of U1 and U2.

Example 11. Let

$$U1 = \{(x, y, 0) \in R3 \mid x, y \in R\},$$
$$U2 = \{(0, 0, z) \in R3 \mid z \in R\}.$$

Then $R3 = U1 \oplus U2$. However, if instead

$U2 = \{(0,w, z) \mid w, z \in R\}$,

then $R3 = U1 + U2$, but it is not the direct sum of U1 and U2.

Example 12. Let

$$U1 = \{p \in P(F) \mid p(z) = a0 + a2z2 + \ldots\ldots + a2mz2m\},$$
$$U2 = \{p \in P(F) \mid p(z) = a1 + a3z3 + \ldots\ldots + a2m+1z2m+1\}.$$

Then $P(F) = U1 \oplus U2$.

Proposition 7. Let $U1, U2 \subset V$ be subspaces. Then $V = U1 \subset U2$ if and only if

1. $V = U1 + U2$;

2. If $0 = u1 + u2$ with $u1 \in U1$ and $u2 \in U2$, then $u1 = u2 = 0$.

Proof.

Suppose $V = U1 \subset U2$. Then 1 holds by definition. Certainly $0 = 0 + 0$ and since by uniqueness this is the only way to write $0 \in V$ we have $u1 = u2 = 0$.

Suppose 1 and 2 hold. By 1 we have that for all $v \in V$ there exist $u1 \in U1$ and $u2 \in U2$ such that $v = u1 + u2$. Suppose $v = w1 + w2$ with $w1 \in U1$ and $w2 \in U2$. Subtract the two equations to obtain

$$0 = (u1 - w1) + (u2 - w2),$$

where $u1 - w1 \in U1$ and $u2 - w2 \in U2$. By 2 this implies $u1 - w1 = 0$ and $u2 - w2 = 0$, or

equivalently $u1 = w1$ and $u2 = w2$ as desired.

Proposition 8. Let $U1, U2 \subset V$ be subspaces. Then $V = U1 \subset U2$ if and only if

1. $V = U1 + U2$;
2. $U1 \cap U2 = \{0\}$.

Proof.

Suppose $V = U1 \oplus U2$. Then 1 holds by definition. If $u \in U1 \cap U2$,

then $0 = u + (-u)$ with $u \in U1$ and $-u \in U2$ (why?).

By Proposition 7 we have $u = 0$ and $-u = 0$, so that $U1 \cap U2 = \{0\}$.

Suppose 1 and 2 hold. To prove that $V = U1 \oplus U2$ holds, suppose that

$$0 = u1 + u2 \quad \text{where } u1 \in U1 \text{ and } u2 \in U2. \ (3)$$

By Proposition 7 it suffices to show that $u1 = u2 = 0$. Equation (3) implies that $u1 = -u2 \in U2$. Hence $u1 \in U1 \cap U2$ which in turn implies that $u1 = 0$. Hence also $u2 = 0$ as desired. Everything in this section can be generalized to m subspaces U1, U2, . . .Um, except Proposition 8. To see this consider the following example:

Example 13. Let

$$U1 = \{(x, y, 0) \in F3 \mid x, y \in F\},$$
$$U2 = \{(0, 0, z) \in F3 \mid z \in F\},$$
$$U3 = \{(0, y, y) \in F3 \mid y \in F\}.$$

Then certainly $F3 = U1 + U2 + U3$, but $F3 \neq U1 \oplus U2 \oplus U3$ since for example $(0, 0, 0) = (0, 1, 0)+(0, 0, 1)+(0,-1,-1)$. But $U1 \cap U2 = U1 \cap U3 = U2 \cap U3 = \{0\}$, so that the analogonof Proposition 8 does not hold.

1.2 Definition and properties of algebras.

Algebra (from Arabic al-jebr meaning "reunion of broken parts") is the branch of mathematics concerning the study of the rules of operations and relations, and the constructions and concepts arising from them, including terms, polynomials, equations and algebraic structures. Together with geometry, analysis, topology, combinatorics, and number theory, algebra is one of the main branches of pure mathematics.

Elementary algebra, often part of the curriculum in secondary education, introduces the concept of variables representing numbers. Statements based on these variables are manipulated using the rules of operations that apply to numbers, such as addition. This can be done for a variety of reasons, including equation solving. Algebra is much broader than elementary algebra and studies what happens when different rules of operations are used and when operations are devised for things other than numbers. Addition and multiplication can be generalized and their precise definitions lead to structures such as groups, rings and fields, studied in the area of mathematics called abstract algebra.

Classification

Algebra may be divided roughly into the following categories:

Elementary algebra, in which the properties of operations on the real number system are recorded using symbols as "place holders" to denote constants and variables, and the rules governing mathematical expressions and equations involving these symbols are studied. This is usually taught at school under the title algebra (or intermediate algebra and college algebra in subsequent years). University-level courses in group theory may also be called elementary algebra.

Abstract algebra, sometimes also called modern algebra, in which algebraic structures such as groups, rings and fields are axiomatically defined and investigated.

Linear algebra, in which the specific properties of vector spaces are studied (including matrices);

Universal algebra, in which properties common to all algebraic structures are studied.

Algebraic number theory, in which the properties of numbers are studied through algebraic systems. Number theory inspired much of the original abstraction in algebra.

Algebraic geometry applies abstract algebra to the problems of geometry.

Algebraic combinatorics, in which abstract algebraic methods are used to study combinatorial questions.

In some directions of advanced study, axiomatic algebraic systems such as groups, rings, fields, and algebras over a field are investigated in the presence of a geometric structure (a metric or a topology) which is compatible with the algebraic structure. The list includes a number of areas of functional analysis:

Normed linear spaces, Banach spaces,Hilbert spaces,Banach algebras,Normed algebras,Topological algebras,Topological groups.

Elementary algebra is the most basic form of algebra. It is taught to students who are presumed to have no knowledge of mathematics beyond the basic principles of arithmetic. In arithmetic, only numbers and their arithmetical operations (such as +, −, ×, ÷) occur. In algebra, numbers are often denoted by symbols (such as a, x, or y). This is useful because:

It allows the general formulation of arithmetical laws (such as $a + b = b + a$ for all a and b), and thus is the first step to a systematic exploration of the properties of the real number system.

It allows the reference to "unknown" numbers, the formulation of equations and the study of how to solve these. (For instance, "Find a number x such that $3x + 1 =$

10" or going a bit further "Find a number x such that ax + b = c". This step leads to the conclusion that it is not the nature of the specific numbers that allows us to solve it, but that of the operations involved.)

It allows the formulation of functional relationships. (For instance, "If you sell x tickets, then your profit will be 3x − 10 dollars, or f(x) = 3x − 10, where f is the function, and x is the number to which the function is applied.")

Abstract algebra extends the familiar concepts found in elementary algebra and arithmetic of numbers to more general concepts.

Sets: Rather than just considering the different types of numbers, abstract algebra deals with the more general concept of sets: a collection of all objects (called elements) selected by property, specific for the set. All collections of the familiar types of numbers are sets. Other examples of sets include the set of all two-by-two matrices, the set of all second-degree polynomials (ax2 + bx + c), the set of all two dimensional vectors in the plane, and the various finite groups such as the cyclic groups which are the group of integers modulo n. Set theory is a branch of logic and not technically a branch of algebra.

Binary operations: The notion of addition (+) is abstracted to give a binary operation. The notion of binary operation is meaningless without the set on which the operation is defined. For two elements a and b in a set S, a * b is another element in the set; this condition is called closure. Addition (+), subtraction (-), multiplication (×), and division (÷) can be binary operations when defined on different sets, as is addition and multiplication of matrices, vectors, and polynomials.

Identity elements: The numbers zero and one are abstracted to give the notion of an identity element for an operation. Zero is the identity element for addition and one is the identity element for multiplication. For a general binary operator * the identity element e must satisfy a * e = a and e * a = a. This holds for addition as

$a + 0 = a$ and $0 + a = a$ and multiplication $a \times 1 = a$ and $1 \times a = a$. Not all set and operator combinations have an identity element; for example, the positive natural numbers (1, 2, 3, ...) have no identity element for addition.

Inverse elements: The negative numbers give rise to the concept of inverse elements. For addition, the inverse of a is written $-a$, and for multiplication the inverse is written a^{-1}. A general two-sided inverse element a^{-1} satisfies the property that $a * a^{-1} = 1$ and $a^{-1} * a = 1$.

Associativity: Addition of integers has a property called associativity. That is, the grouping of the numbers to be added does not affect the sum. For example: $(2 + 3) + 4 = 2 + (3 + 4)$. In general, this becomes $(a * b) * c = a * (b * c)$. This property is shared by most binary operations, but not subtraction or division or octonion multiplication.

Commutativity: Addition and multiplication of real numbers are both commutative. That is, the order of the numbers does not affect the result. For example: $2 + 3 = 3 + 2$. In general, this becomes $a * b = b * a$. This property does not hold for all binary operations. For example, matrix multiplication and quaternion multiplication are both non-commutative.

Chapter 2. Buchberger algorithm for Polynomial ring.

2.1 Associative algebras and ideals.

Associative algebra

In mathematics, an associative algebra A is an associative ring that has a compatible structure of a vector space over a certain field K or, more generally, of a module over a commutative ring R. Thus A is endowed with binary operations of addition and multiplication satisfying a number of axioms, including associativity

of multiplication and distributivity, as well as compatible multiplication by the elements of the field K or the ring R.

In some areas of mathematics, associative algebras are typically assumed to have a multiplicative unit, denoted 1. To make this extra assumption clear, these associative algebras are called unital algebras.

Formal definition

Let R be a fixed commutative ring. An associative R-algebra is an additive abelian group A which has the structure of both a ring and an R-module in such a way that ring multiplication is R-bilinear:

$r*(xy)=(r*x)y=x(r*y)$

for all $r \in R$ and $x, y \in A$. We say A is unital if it contains an element 1 such that

$1x=x=x1$

for all $x \in A$.

If A itself is commutative (as a ring) then it is called a commutative R-algebra.

From R-modules

Starting with an R-module A, we get an associative R-algebra by equipping A with an R-bilinear mapping $A \times A \rightarrow A$ such that

$x(yz)=(xy)z$

for all x, y, and z in A. This R-bilinear mapping then gives A the structure of a ring and an associative R-algebra. Every associative R-algebra arises this way.

Moreover, the algebra A built this way will be unital if and only if

$\exists 1 \in A,, 1x=x1=x$

This definition is equivalent to the statement that a unital associative R-algebra is a monoid in R-Mod (the monoidal category of R-modules).

From rings

Starting with a ring A, we get a unital associative R-algebra by providing a ring homomorphism $\eta:R\to A$ whose image lies in the center of A. The algebra A can then be thought of as an R-module by defining

$r*x=\eta(r)x$

for all $r \in R$ and $x \in A$.

If A is commutative then the center of A is equal to A, so that a commutative R-algebra can be defined simply as a homomorphism $\eta:R\to A$ of commutative rings.

Algebra homomorphisms

A homomorphism between two associative R-algebras is an R-linear ring homomorphism. Explicitly, $\varphi:A_1\to A_2$ is an associative algebra homomorphism if

$\varphi(r*x)=r*\varphi(x)$

$\varphi(x+y)=\varphi(x)+\varphi(y)$

$\varphi(xy)=\varphi(x)\varphi(y)$

For a homomorphism of unital associative R-algebras, we also demand that

$\varphi(1)=1$

The class of all unital associative R-algebras together with algebra homomorphisms between them form a category, sometimes denoted R-Alg.

The subcategory of commutative R-algebras can be characterized as the coslice category R/CRing where CRing is the category of commutative rings.

2.2 Definition of Lie algebra.

Definition 1.1.1. By an (nonassociative) algebra over a field F we mean a vector space A together with an F-bilinear operation A×A → A which is usually written (x, y) #→ xy.

The adjective "nonassociative" means "not necessarily associative". An associative algebra is an algebra A whose multiplication rule is associative: x(yz) = (xy)z for all x, y, z ∈ A. The existence of a unit 1 is not assumed.

Definition 1.1.2. Let L be a vector space over a field F. Then a bilinear operation $[\cdot\cdot] : L \times L \to L$ sending (x, y) to [xy] is called a bracket if it satisfies the following two conditions. [L1] [xx] = 0 for all x ∈ L.

[L2] (Jacobi identity) [x[yz]] + [y[zx]] + [z[xy]] = 0 for all x, y, z ∈ L.

A vector space L with a bracket [··] is called a Lie algebra. This is an example of a nonassociative algebra.

Let us analyze the two conditions. Condition [L1] implies:

[L1'] [xy] = −[yx] for all x, y ∈ L. (Bracket is skew commutative.)

Proof: [(x + y)(x + y)] = 0 = [xx] + [xy] + [yx] + [yy] = [xy] + [yx].

Conversely, if the characteristic of the field F is not equal to 2 then [L1'] implies that 2[xx] = 0 implies [L1]. So, [L1] and [L1'] are equivalent when char F = 2. &

The second condition [L2] can be rewritten as follows:[x[yz]] = [[xy]z] + [[zx]y].

The term [[zx]y] prevents L from being associative. Since z , x, y are arbitrary we obtain:

Proposition 1.1.3. A Lie algebra is associative if and only if [[LL]L] = 0.

The notation [[LL]L] indicates the vector subspace of L generated by all expressions [[xy]z].

Definition 1.1.4. A (Lie) subalgebra of a Lie algebra L is defined to be a vector subspace K so that [K K] $\subseteq$ K.

For example, [LL] is always a Lie subalgebra of L.

Definition 1.1.5. A homomorphism of Lie algebras is a linear map $\varphi : L \to L'$ so that $\varphi([xy]) = [\varphi(x)\varphi(y)]$ for all $x, y \in L$.

1.2. Examples.

Example 1.2.1. The simplest example of a Lie algebra is given by letting [xy] = 0 for all $x, y \in L$ where L is any vector space over F. All conditions are clearly satisfied. A Lie algebra satisfying this condition (usually written as [LL] = 0) is called abelian. The word "abelian" comes from one standard interpretation of the bracket. Suppose that A is an associative algebra over F. Then the commutator [xy] is defined by [xy] = xy − yx. This is easily seen to be a bracket and is also called the Lie bracket of the associative algebra.

Example 1.2.2. Suppose that V is any vector space over F. We define gl(V) to be the Lie algebra of all F-linear endomorphisms of V under the Lie bracket operation. A Lie subalgebra of gl(V) is called a linear Lie algebra.

Definition 1.2.3. A representation of the Lie algebra L is defined to be a Lie algebra homomorphism $L \to gl(V)$ for some vector space V . The representation is called faithful if this homomorphism is injective: $L \hookrightarrow gl(V)$.

1.2.1. linear Lie algebras. There is a well-known theorem (due to Ado in characteristic 0 and Iwasawa in characteristic p) what every finite dimensional Lie algebra has a faithful finite dimensional representation. I.e., it is isomorphic to a

linear Lie algebra. So, our finite dimensional examples are all linear. What are the finite dimensional linear Lie algebras?

If $V = F^n$ then $gl(V)$ is denoted $gl(n, F)$. This is the vector space of all $n \times n$ matrices with coefficients in F with Lie bracket given by commutator: $[xy] = xy - yx$. A subalgebra is given by a subset of $gl(n, F)$ which is closed under this bracket and under addition and scalar multiplication.

Example 1.2.4. Let $sl(n, F) \subseteq gl(n, F)$ denote the set of all $n \times n$ matrices with trace equal to zero.

(1) $Tr([xy]) = Tr(xy) - Tr(yx) = 0$. So, $sl(n, F)$ is closed under $[\ ]$.

(2) $Tr(x + y) = Tr(x) + Tr(y) = 0$.

(3) $Tr(ax) = a\,Tr(x) = 0$

Therefore, $sl(n, F)$ is a linear Lie algebra.

Proposition 1.2.5. Suppose that $f : V \times V \to F$ is a bilinear form. Then the set of all $x \in gl(V)$ so that

$f(x(v), w) + f(v, x(w)) = 0$ or all $v, w \in V$ is a Lie subalgebra of $gl(V)$ which we denote $o(V, f)$

Proof. It is clear that $o(V, f)$ is a vector subspace since the defining equation is linear in x. The following calculation shows that it is closed under Lie bracket.

$f(xy(v), w) + f(y(v), x(w)) = 0$

$f(yx(v), w) + f(x(v), y(w)) = 0$

$f(v, xy(w)) + f(x(v), y(w)) = 0$

$f(v, yx(w)) + f(y(v), x(w)) = 0$

If we take the alternating sum (+ - + -) of these equations we see that

$f([xy](v), w) + f(v, [xy](w)) = 0$

Example 1.2.6. Particular examples of the above definition are as follows.

(1) Suppose that f is a nondegenerate symmetric bilinear form on V . Then o(V , f) is called the orthogonal Lie algebra relative to f.

(2) Suppose that f is a nondegenerate skew symmetric form on V : $f(v,v) = 0$ for all $v \in V$. (If char F = 2 this is equivalent to the condition that & $f(v,w) = -f(w, v)$ for all v,w.) In this case dim V = 2n (even) and o(V , f) is called the symplectic Lie algebra relative to f. We will look at these examples in more detail later.

Example 1.2.7. Other easy examples of linear Lie algebras are:

(1) $t(n, F) \subseteq gl(n, F)$, the set of upper triangular $n \times n$ matrices over F

(2) $n(n, F) \subseteq t(n, F)$, the set of strictly upper triangular matrices (with 0 on the diagonal).

(3) $d(n, F) \subseteq t(n, F)$, the set of diagonal $n \times n$ matrices with coefficients in F.

1.3. Derivations.

Definition 1.3.1. Suppose that A is a nonassociative algebra over F. Then a derivation on A is a linear function $\delta : A \to A$ so that

$\delta(xy) = \delta(x)y + x\delta(y)$

for all $x, y \in A$. The set of all derivations on A is denoted Der(A).

Proposition 1.3.2. Der(A) is a Lie subalgebra of gl(A).

Proof.

Go back to the definition of a Lie algebra. Using the skew symmetry condition [L1'], Condition [L2] can be rephrased as:

$[z[xy]] = [[zx]y] + [x[z\ y]$

In other words, the bracket by z operation $adz(\cdot) = [z(\cdot)]$ satisfies:

$adz[xy] = [adz(x)y] + [xadz(y)]$

So any Lie algebra acts on itself by derivations. This gives a homomorphism:

$ad : L \rightarrow Der(A)$ called the adjoint representation.

2.3 Polynomial ring and ideals.

Polynomial ring

In mathematics, especially in the field of abstract algebra, a polynomial ring is a ring formed from the set of polynomials in one or more variables with coefficients in another ring. Polynomial rings have influenced much of mathematics, from the Hilbert basis theorem, to the construction of splitting fields, and to the understanding of a linear operator. Many important conjectures involving polynomial rings, such as Serre's problem, have influenced the study of other rings, and have influenced even the definition of other rings, such as group rings and rings of formal power series.

Polynomials

A polynomial in X with coefficients in a field K is an expression of the form

$$p=p_mX^m+p_{m-1}X^{m-1}+....+p_1X+p_0,$$

where $p_0, ..., p_m$ are elements of K, the coefficients of p, and X, X^2, ... are formal symbols ("the powers of X"). Such expressions can be added and multiplied, and then brought into the same form using the ordinary rules for manipulating

algebraic expressions, such as associativity, commutativity, distributivity, and collecting the similar terms. Any term $p_k X^k$ with zero coefficient, $p_k = 0$, may be omitted. The product of the powers of X is defined by the familiar formula

$X^k X^l = X^{k+l}$ where k and l are any natural numbers. Two polynomials are considered to be equal if and only if the corresponding coefficients for each power of X are equal. By convention, $X_1 = X$, $X_0 = 1$, and the sum defining the polynomial p may be viewed as the linear combination of the symbols $X_m, \ldots, X_1, X_0$ with coefficients $p_m, \ldots, p_1, p_0$. Using the summation symbol, the same polynomial is expressed more compactly as follows:

$p=pmXm+pm-1Xm-1+....+p1X+p0=\sum_{k=0}^{m} p_k X^k$.

The summation limits are frequently omitted, so that the same polynomial is written as

$p=\sum_k p_k X^k$,

and it is understood that only finitely many terms are present, i.e. p_k is zero for all large enough values of k, in our case, for $k > m$. The degree of a polynomial is the largest k such that the coefficient of X^k is not zero. In the special case of zero polynomial, all of whose coefficients are zero, the degree is undefined, or sometimes defined to be the symbol $-\infty$.

2.4. The polynomial ring K[X]

The set of all polynomials with coefficients in the field K forms a commutative ring denoted K[X] and is called the ring of polynomials over K. The symbol X is commonly called the "variable", and this ring is also called the ring of polynomials in one variable over K, to distinguish it from more general rings of polynomials in several variables. This terminology is suggested by the important cases of polynomials with real or complex coefficients, which may be alternatively viewed

as real or complex polynomial functions. However, in general, X and its powers, X k, are treated as formal symbols, not as elements of the field K. One can think of the ring K[X] as arising from K by adding one new element X that is external to K and requiring that X commute with all elements of K. In order for K[X] to form a ring, all powers of X have to be included as well, and this leads to the definition of polynomials as linear combinations of the powers of X with coefficients in K.

A ring has two binary operations, addition and multiplication. In the case of the polynomial ring K[X], these operations are explicitly given by the following formulas:

$$(\sum^{n}_{i=0} a_i X^i) + (\sum^{n}_{i=0} b_i X^i) = \sum^{n}_{i=0} (a_i + b_i) X^i$$

And

$$(\sum^{n}_{i=0} a_i X^i) * (\sum^{m}_{j=0} b_j X^j) = \sum^{m+n}_{k=0} (\sum_{i+j=k} a_i b_j) X^k$$

In the first formula, one of the polynomials may be extended by adding "dummy terms" with zero coefficients, so that the same set of powers formally appears in both summands. In the second formula, the inner summation in the right hand side is only extended over indices within bounds, $0 \le i \le m$ and $0 \le j \le n$. Alternative expressions of addition and multiplication, without using explicit bounds in the sums, are as follows:

$$(\sum_i a_i X^i) + (\sum_i b_i X^i) = \sum_i (a_i + b_i) X^i$$

And

$$(\sum_i a_i X^i) * (\sum_j b_j X^j) = \sum_k (\sum_{l,j:i+j=k} a_i b_j) X^k$$

Since only finitely many coefficients a_i and b_j are non-zero, all sums in effect have only finitely many terms, and hence represent polynomials from K[X].

Since a polynomial from K[X] can be multiplied by a "scalar" k from K to yield a new polynomial, K[X] actually constitute an associative algebra over K. Viewed as a vector space, K[X] has a basis consisting of the countably infinite set

{1, X, X 2, X 3, ...}.

More generally, the field K can be replaced by any commutative ring R, giving rise to the polynomial ring over R , which is denoted R[X].

Properties of K[X]

The polynomial ring K[X] is remarkably similar to the ring Z of integers in many respects. This analogy and the arithmetic of the ring of polynomials were thoroughly investigated by Gauss and his theory served as a model for development of abstract algebra in the second half of the nineteenth century in the works of Kummer, Kronecker, and Dedekind.

Hilbert's basis theorem

In mathematics, specifically commutative algebra, Hilbert's basis theorem states that every ideal in the ring of multivariate polynomials over a Noetherian ring is finitely generated. This can be translated into algebraic geometry as follows: every algebraic set over a field can be described as the set of common roots of finitely many polynomial equations. Hilbert (1890) proved the theorem (for the special case of polynomial rings over a field) in the course of his proof of finite generation of rings of invariants.

Hilbert produced an innovative proof by contradiction using mathematical induction; his method does not give an algorithm to produce the finitely many basis polynomials for a given ideal: it only shows that they must exist. One can determine basis polynomials using the method of Gröbner bases.

2.4. Buchberger algorithm and its work.

Problems associated with ideals generated by finitesets F of polynomials in many variables, thereas a mathematical subtasks in different areas of theorytheory of systems, for example, see [1]. The method of Gröbner bases pre-seems a technique that provides algorithmic solutionsfor a variety of tasks such as, for example, finding the exactsolutions of F, considered as a system of algebraic equationstions, then, the task of performing calculations in a quotient ofmodulo the ideal generated by F; In addition, verification of variousproperties of the ideal generated by F; further, finding a poly-nomial solutions of homogeneous linear equationswith coefficients in F In addition, the equivalence problemwords in the rings modulo ideals, as well as the equivalencewords in commutative semigroups (in other words, reversiblePetri nets), and finally enumeration without repetitions of all poly-nomial ideals of integrity in the field, etc.For many years the work of Herman [2] gave a single-Izvestiyastny an algorithmic method to tackle the problemin the theory of polynomial ideals. Her work is stilla rich source of ideas, however, as stated in [3,4],solution to the basic question, whether this belongs to the polynomialmany variables, the ideal generated by F, does not realizableICDO solutions for the "simplification of the problem modulo>(In other words, the problem of choosing a single representativein the class modulo the ideal), as well as for the problemimplementation of effective computation in the quotient modulo the ideal.

The method of Gröbner bases as its primary goaloffers a solution to simplify the problem for polynomialideals, to provide a simple solution for a largenumber of other algorithmic tasks, including the original problemHerman of belonging. In addition, compared to the algo-Herman rhythms, our algorithm for constructing Gröbner basis, on-possesses the amazing simplicity and, depending on the consideredVai example, can result in intermediate calculationstions to the relatively small degree polynomials. On the otherhand, as shown in [5, 6], to clarify the question of the coincidencemodulo the ideal itself is intrinsically

difficult in-cottage. Therefore, in the worst case, the method of Gröbner bases andcan lead to unrealistic calculations. Stilldoing is a lot of work on the analysis and a priori pre-the prediction of these phenomena, as well as the expansion of application-of the method.The method of Gröbner bases was proposed in 1965 by the authorthis review in [7, 8], and since 1976 updated, generalizedPuppy, applied and analyzed in many studies [9-35].The main idea of this method is to transform this multi-set F of polynomials to a certain standard form G, forwhich in [9], the author coined the term "Grobner basis" asProfessor. W. Grebner, supervisor of the thesis [7], stimulation-stimulated the study of the subject, put the following question:how the multiplication table of an associative algebra,which is the quotient modulo the polynomialtion of the ideal,

Can be constructed algorithmically. He has also been given the first draft of such an algorithm, he proposed a "comple-thread "generating set by adding the differences of various govermental representatives of monomials (modulo).This, however, does not terminating the procedure. The main pre-attainment of the author could see and prove [7, 8], which is sufficient just add the difference (reduced form) of some "Critical pairs" (or, equivalently, the reduced form of "S-polynom" [7]), whose number is finite. In retrospect, it appears that the concept of "Grobner basis" under the name "standard basis" appearance moose is already a year earlier, in 1964, in a famous paper [36].However, Hironaka gave only non-constructive proof of the existence of these bases, while in [7]Together with the notion of such a basis, and describes the algorithm,but the algorithm can provide an effective solution briefthe above-mentioned various problems. Nonconstructive proof ,the proof of the existence of Gröbner bases can also be found[37]. It leads to the Hilbert basis theorem.Later in 1967 the two main ideas of our method, namely:critical pair completion, have been proposed as the KNU-if and Bendix [38] in a more general context of equality of termsthe first order. The algorithm of Knuth - Bendix has nowan important role in various areas of

computer science (eg,in the transformation of abstract data types in the prooftheorems on the properties of equations and applications in automatedNoah program verification).

Recently, an algorithm Knuth - Bendix algorithm, and the author for the construction of Gröbner bases were united in a common algorithmic structure [32] and independently in [39], see also [3] for a review, introduction to algorithms such as "completion of critical pairs." On the other hand,improve the algorithm's were made for the algorithm Knuth – Bendix , see [40]. Many require attention questions remain open in the near future may also have an impact on systems theory (eg, see[41] methods for the resolution of the Boolean algebra based on approach replenish critical pairs).This paper provides an overview of the method of Gröbner bases . The main emphasis in this paper is on the explicit formulation of algorithms (in a simple recording), as well as examples. Except for a few sketches, proofs are not given the underlying algorithm, however, are full references to original publications.

2. **Gröbner bases**

Here K denotes a field, $K[x_1 ..., x_n]$ - The ring polynomials in n variables over K. We will use the following types of variables:

f, g, h, k, p, q — polynomial from $K[x_1 ..., x_n]$

F, G — finite subset of $K[x_1 ..., x_n]$

s, t, и — product of powers of the form $x_1^{i1},.....x_n^{in}$;

a, b, c, d — elements of K;

l , j , l , m — natural numbers,

Let F = $\{f_1, \ldots f_m\}$. designation Ideal (F) will be used for the ideal generated by F

(ie for a set of $\{\sum_{i<j<m} h_i f_i \mid h_i \in F, 1 \le i \le m\}$)

In addition, we write $f \equiv_f g$, If f is congruent to g modulo Ideal (F)

(т. e. f — g ∈ Ideal (F)).

Before proceeding to the definition of Grobner basis, to introduce the concept of "reduction". To do this, fix a linear ordering <r products of powers $x_1^{i1}, \ldots x_n^{in}$, For example, a linear ordering on the degree of

(т. e. $1 <_T x <_T y <_T x^2 <_T xy <_T y^2 <_T x^3 <_T x^2 y <_T x y^2 <_T y^3 <_T \ldots$ in the case of two variables), or "purely lexicographical ordering"

In fact, any suitable linear ordering with the following two properties:

(T1) $1 <_T t$ for all $t \neq 1$;

(T2) if $s <_T t$, то $su <_T tu$.

Linear ordering satisfying (T1), (T2), we called a "valid". Next, assume that a arbitrary admissible ordering <T fixed. Regarding the selected <T we use the following notation:

cf (g, t) — the coefficient of t in g;

lpp(f) — highest (with respect to <T) product of powers, included in f with nonzero coefficient;

lc(f) — coefficient of the product lpp(f) in f.

Definition 1 [7, 8]. Polynomial g is reduced to h no mod F (denoted by $g \to_F h$),

if there are f ∈ F, b, u,

that the following $g \rightarrow f, b, u$ and, in addition, $h = g - buf$;

polynomial g is reduced with the help of f, b, u (denoted by

$g \rightarrow f\ b\ u$), if $cf(g, u.lpp(f)) \neq 0$, and, moreover, $b = cf(g, u.lpp(f))/lc(f)$.

Thus, informally speaking, g is reduced to A by modulo F, if h is obtained from g by subtracting the appropriate works buf, while leading monomial of a polynomial buf coincides with a monomial of the polynomial g. In other words, reduction can be seen as one step of the generalized division.

Example 1.

Consider $F = \{f_1, f_2, f_3\}$ where

$f_1 = 3x^2y + 2xy + y + 9x^2 + 5x - 3,$

$f_2 = 2x^3y - xy - y + 6x^3 - 2x^2 - 3x + 3,$

$f_3 = x^3y + x^2y + 3x^2 + 2x^2.$

monomials of polynomials f_1, f_2, f_3 recorded in the order, corresponding to the purely lexicographic ordering. Seniors are the product of powers x^2y, x^3y, x^3y respectively, and the leading coefficients — 3, 2, 1 respectively. Consider the polynomial

$g = 5y^2 + 2x^2y + (5/2)\,xy + (3/2)\,y + 8x^2 + (3/2)\,x - 9/2.$

Polynomial g can be reduced modulo F, for example, to a polynomial

$h = 5y^2 + (7/6)\,xy + (5/6)\,y + 2x^2 - (11/6)\,x - 5/2.$

Just,

$g -_{f, b, u.}$ for $f = f_1$, $b = 2/3$, $u = 1$, since $cf(g, x^2y) \neq 0$ and

$b = cf(g, x^2y)/lc(f_1)$;

$h=g-(2/3)f_1$.

Definition 2. Polynomial h is given in normal form (or reduced form) modulo F, if there is no such polynomial h', which $h \to_F h'$.

The polynomial h is the normal form of g modulo F

(denoted by h = NF (F, g)), if there is a sequence of reductions

$g = k_0 \to_F k_1 \to_F k_2 \to_F \ldots \to_F k_m = h$ and, In addition, h is given in normal form modulo F. Algorithm S is called a normal form algorithm (or canonization) if for all F and g are polynomial in S (F, g) is normal form of g modulo F.

Lemma 1 [7,9]. The following algorithm is an algorithm normal form.

Alghoritm 1 (h = NF(F,g))

h:=g; while there are $f \in F$, b, u, which $h \to_{f,\, b\, ,\, u}$ do;

to choose such $f \in F$, b, u with the most significant with respect to <r the product of the degrees u lpp(f), which

$h \to_{f,\, b,\, u}$

h:=h — buf.

A few words about the correctness of this algorithm rhythm. For the correctness of the choice of the older works degree is not necessary. However, this choice is very important for the efficiency of the algorithm. End of algorithm rate guaranteed by the following lemma:

Lemma 2 [7, 9]. For each F ratio $\to_F$ noetherian

(i.e.There is no infinite sequence $k_0 \to_F k_1 \to_F \ldots \to_F k_2 \to_F \ldots$)

Example 2. Polynomial h of Example 1 is in normal form modulo F: no product of powers, a member of h, not a divisor of the degrees of the older works of any polynomial of F. Thus, there can be no reduction.

another example of:

$x^3 y \rightarrow_{f1} - (2/3) x^2y - (1/3) xy - 3x^3 - (5/3) x^2 + x = g_1.$

further, g_1 can be reduced:

$g_1 \rightarrow_{f1} (1/9) xy + (2/9) y - 3x^3 + (1/3) x^2 + (19/9) x - 2/3 = g'_1.$

Polinomial g'_1 represented in normal form modulo F.

So g'_1 is the normal form of polynomial x^3y modulo F. In fact g'_1 may be the result of algorithm 1 for x^3y (depending on how the command is executed «choose an f ∈ F, что ...»). In the present example will be another reduction:

$x^3 y -_{f2}(1/2) xy + (1/2) y - 3x^3 + x^2 + (3/2) x - 3/2 = g_2.$

Polynom g_2 s already in normal form modulo F.

This example shows that, generally speaking, can be different normal forms $g_1 \neq g_2$ modulo F of the same polynomial g. Those of F, which indicated the situation is not possible to play a key role in our approach to the algorithmic solution of problems in the theory of polynomial ideals.

Definition 3 [7,9].

The set F is called a Gröbner basis (or set of Grebner) if for any polynomial g, h_1, h_2, such that h_1 and

h_2 — normal form of g modulo F, executed equality $h_1 = h_2$.

The main objective of this paper is to show that

a) for the sets F, are Gröbner bases, many important algorithmic problem (formulated in teorminah Ideal (F)) can be easily solved;

б) set F, which are not Gröbner bases can be converted to a set G, which are already Gröbner bases and generating the same ideal. Most of the algorithmic applications of Gröbner bases based on the following fundamental property of:Отменить изменения

Theorem 1 [7, 9, 22] (the theorem on the characterization of the bases Grebner). Let S - an arbitrary normal form algorithm. The following conditions are equivalent:

(GB 1) F — Grobner basis;

(GB 2) для all polynomials f, g compared $f \equiv_F g$ equivalent to S (F, f) = S (F, g);

(GB 3) ratio $\to_F$ satisfies the Church — Rossera.

Condition (BG 3) binds to the same Grobner bases concept for the equations of the first-order terms, as well as with the algorithm of Knuth - Bendix. Details can be found in [3]. Condition (BG 3) in this paper is not used. at establishing this connection is useful

Lemma 3 [22, 30] (relationship between the reduction and comparison). For all F, f, g comparison $f \equiv_F g$ equivalent to the relation $f \leftrightarrow_F g$ (here $\leftrightarrow_F$ denotes the reflexive, symmetric, transitive closure of the relation $\to_F$, i. e.

$f \leftrightarrow_F g$, if there exists a sequence $f = k_0 \leftrightarrow_F k_1 \leftrightarrow_F k_2 \leftrightarrow_F \ldots$

$\ldots \leftrightarrow_F k_m = g$, where $f \leftrightarrow_F g$, if either $f \to_F g$, if $g \to_F f$).

Condition (BG 2) shows directly that for the basis Gröbner problem of testing the relationship F «$f \equiv_F g$» algorithmically decidable (uniformly in F). Gröbner bases for other computing tasks have similar simple solutions.

3. Algorithmic construction of Gröbner bases

Before we give the algorithmic applications of Gröbner bases, we show how to check whether F is given Grobner basis, and how you can build Gröbner bases. For this purpose, the important role played by the concept "S-polynomial".

Definition 4 [7, 8, 9]. S-polynomial corresponding to polynomials f_1, f_2, is a polynomial $SP(f_1, f_2)= u_1f_1 —(c_1/c_2)u_2f_2$, where $c_i = lc(f_i)$, product of powers u_i such, that s_iu_i - coincides with the least common multiple of monomials s_1,s_2, where $s_i= lpp (f_i)$ (i=1,2).

Example 3. For polinomials f_1, f_2 like in example 1, polinom

$SP(f_1, f_2) = 2x^2y + (5/2)xy + (3/2)y + 8x^2 + (3/2)x -9/2.$

Note that the least common multiple of monomials s_1, s_2 coincides with the minimum product of the degrees, is reducible as a modulo f_1 and modulo f_2.

An algorithmic criterion for Gröbner bases formulated in the following theorem that constitutes the cornerstone of the method.

Theorema 2 [7, 8, 9, 22] (algorithmic characterization of Gröbner bases).

Let S - an arbitrary normal form algorithm. The following properties are equivalent:

(БГ 1) F — Grobner basis;

(БГ 4) for any $f_1 ,f_2 \in F$ performed $S (F, SP (f_1 ,f_2)) = 0$.

Condition (BG 4) actually allows you to check property «F-Gröbner basis": should be considered final the number of pairs $f_1 ,f_2 \in F$, Calculate the corresponding S-polynomials and determine whether they are reduced to zero by applying the normal form of the algorithm 5. In addition, Theorem 2 is the foundation of the basic algorithm 2, our to solve the following problem.

Task 1.

Given F, Find G, such that Ideal(F)= Ideal (G) и G — Grobner basis.

Alghoritm 2 [7, 8] (for task 1).

G:=F;

$B = \{(f_1, f_2) | f_1, f_2 \in G, f_1 \neq f_2\}$;

while $B \neq \varnothing$ do

(f_1, f_2) — pair from B;

$B := B \setminus \{(f_1, f_2)\}$;

$h := SP(f_1, f_2)$;

$h' := NF(G, h)$;

if $h' \neq 0$ then

$B := B \cup \{g, h') \mid g \in G\}$;

$G := G \cup \{h'\}$.

The correctness of this algorithm is partly determined by Theorem 2. The end of his work can be shown in two ways, see [8, 17]. We give a sketch of the first method [17].

Consider a sequence of ideals Ideal $(P_1) \subset$ Ideal $(P_2) \subset$ ...,

where P_i coincides with the set of products of powers of senior polynomials of G_i, where G_i is the value of G after perform i steps of expansion. It is easy to see that all the inclusions in the studied sequences - strict. so sequence is finite by the Hilbert basis theorem, See, eg, [42].

We also give a sketch of the second method [8]. From Lemma Dixon [43] applied to this situation, it follows that the sequence t_1, t_2, ... products of powers, having the property that for all $i < j$ the product t_j is not divisible by t_i, is the ultimate. Indeed, if the t_i — the older the product of powers of i-added to the polynomial G in the performance of the algorithm (i = 1, 2, ...), then the sequence t_1, t_2, ... has

formulated the property and, therefore, must be finite. In this way it was first proved in [8] the end of the algorithm using Lemma Dixon. Hilbert basis theorem can be deduced as a corollary of this approach, see [37].

Example 4. Beginning with set F of Example 1, we choose the first, for example, a pair of f_1 ,f_2 and compute

$SP(f_1, f_2) = 2x^2 y + (5,2)\, xy + (3/2)\, y + 8x^2 + (3,2)\, x - 9/2.$

The reduction of this polynomial to normal form gives a polynomial

$(7/6)\, xy + (5,6)\, y + 2x^2 - (11/6)\, x - 5/2.$

Add this to the polynomial G in the normalized form

$f_4 = xy + (5/7)\, y + (12/7)x^2 - (11/7)\, x - 15,7.$

This normalization is not necessary. However, as Swee- computing experience attests, it can lead to values of considerably simplify the calculations with rational numbers. Nevertheless, this phenomenon was not theoretically explained until equation. Studies of this type, done for the algorithm Euclid, important enough, see the review [44] on these issues. Next, we choose, for example, a pair of f_1 ,f_4:

$SF(f_1, f_4) = f_1 - 3xf_4 = -(1/7)\, xy + y - (36/7)\, x^3 + (96/7)\, x^2 + (80/7)\, x - 3.$

The reduction of this polynomial by subtracting —$(1/7)f_4$ (and normalization) leads to a new polynomial

$f_5 = y - (14,3)\, x^3 + (38,3)\, x^2 + (61/6)\, x - 3.$

Then we compute $SP(f_4, f_5) = f_4 - xf_5$ After subtracting the $(5/7)f_5$ and normalization, we obtain a polynomial

$f_6 = x^4 - 2x^3 - (15/4)\, x^3 - (5/4)\, x.$

Finally, the reduction of the polynomial $SP(f_1, f_3) = xf_1 — 3f_3$ results to a polynomial

$f_7 = x^3 — (5/2)\, x^2 – (5/2)\, x.$

Reduction of S-polynomials of all remaining pairs yields zero, and therefore does not require any further addition polynomials to construct a basis. For example, the

SP $(f_6, f_7) = (1/2)\, x^3 – (5/4)\, x^2 – (5,4)x$

is reduced to zero by subtracting $(1/2)f_7$. So $G = \{f_1, ..., f_7\}$— Grobner basis corresponding to F.

Definition 5 . F is called a reduced basis Grebner, if F - Grobner basis, and, moreover, for all $f \in F$ polynomial f is represented in normal form modulo $F\backslash\{f\}$ and $lc(f) = 1$.

Example 5. The set G of Example 4 is a reduced Grobner basis. for example, f_1 reducible to zero modulo $\{f_2,...,f_7\}$. Consistently redutsiruya all poly- noma of the Grobner basis modulo the rest of the polynomials from the basis and normalizing the leading coefficients, we can always converted to a reduced Grobner basis basis Greb- the inequality of the same ideal. We do not give a formal de- Sania this procedure, since it will automatically be included- Chen in an improved version of the algorithm below. In this example f_2, f_3, f_4, f_6 also be reduced to zero and f_5 reducible to the $f_3' = y + x^2 - 3/2x - 3$. Therefore, the reduced basis Gröbner corresponding F, is

$G' = \{f_5', f_7'\} = \{y + x^2 - (3,2)x - 3,\ x^3 - (5/2)x^2 - (5/2)x\}.$

Theorem 3 (uniqueness of the reduced basis Grobner). If Ideal (F) = Ideal (F'), where F and F' — reduced Grobner bases, then F = F '.

Definition 6. Denote by BG the function F that assigns to every reduced Grobner basis G, so that Ideal(F)= Ideal(G).

From Theorems 2, 3, 2, and the comments of the algorithm to Example 5, we Finally, we obtain the following fundamental theorem, which formulated algorithmic results on Gröbner bases.

The main theorem 4 [7, 8, 9]. Public function BG're chickens sienna and has the following properties for all FG:

(СБГ 1) Ideal (F) = Ideal (БГ(F));

(СБГ 2) if Ideal (F) = Ideal(G), то ЬГ(F) = BT(G);

(СБГ 3) BT(F) is the reduced Grobner basis.

4. An improved version of the algorithm

For a review of practical examples it is important to improve algorithm. Here are three possibilities to accelerate the computation

1. The order of selection pairs (f_1,f_2) Are constructed for S-polinomy, though logically irrelevant to a large extent affects the complexity of the algorithm. Typically, couples with a minimum relative ordering <r the least common multiple of senior products of powers should be chosen in the first queue. In conjunction with paragraph 2, this can significantly reduce computation time.

2. Every time another is added to the basis of a polynomial, all the other polynomials can be reduced by using also a new polynomial. Thus, we can again remove from G several polynomials. Such a reduction causes a cascade of reductions and deletions. Also, if this procedure is performed systematically during the algorithm, the final result of the algorithm is automatically reduced Gröbner basis. The reduction of polynomials modulo other polynomial basis should be carried out also in the beginning of the algorithm.

3. While the strategies 1 and 2 do not require a new theoretical basis, the following approach is based on a thin theoretical result [19], which proved to be useful also in the general context of algorithms "critical recharge pairs ", in particular for the Knuth-Bendix algorithm. Operations require the algorithm of the large amount of computation - a reduction of h 'modulo G in the unit while. we offer "Criterion", which will, roughly speaking, to recognize of the S-polynomial h

of its reducibility to zero without actually performing the reduction. This can lead to significant reductions in computation. Using this criterion, in favorable situations, it suffices to consider the 5-polynomials in the number of 0 (l) instead of O (l2), where l is the number of polynomials in the basis. (Of course, in general, l change the course of the implementation of the algorithm, and therefore very difficult to consider theoretically the effect of this criterion.)

The strategy part 1 is already used in [7, 8]. Correctness TEXT nicks reductions and deletions, briefly presented in Section 2, was already established in [7,8]. The criterion described in clause 3, was introduced [19], and in the same installed it correctly, the details of the proof of correctness can be found in [20]. Before turning to a detailed description of the improved version of the algorithm is based on paragraph 1-3, will present a sketch of it.

In addition to the G and B, we use a set of R and R. The set R contains the polynomials of G, which can be reduced modulo the other polynomials in G. While R is not empty, we reduce the polynomials in R and accumulate obtained reduced polynomials in the set R. Only when R is empty, we add to G reduced polynomials in P and define a new pair of B, which must be considered S-polynomials. If the S-polynomial for the pair of B is reduced to a non-zero polynomial h ', then h' is entered in P, and again sought polynomials in G. reducible with respect to h '. such polynomials recorded in R, and then continue the systematic reduction in R. We now describe in detail an improved version of the algorithm.

Task 2.

Given: F. Find: G, so that Ideal (F) = Ideal (G) и G — reduced Grobner basis.

Alghoritm 3 |19] (for task 2).

R:=F; P:=∅; G:= ∅; B := ∅;

reduce all (R, P, G, B);

The new basis (P, G, B);

while B:= ∅ do

(f_1,f_2):= pair from B with a minimum relative

<тНОК(lpp(f_1), lppt(f_2));

B:=B\{(f_1 ,f_2)};

if (not criterion 1 (f_1 ,f_2, G, B) and not criterion 2 (f_1 ,f_2))

then h:=NF(G, SP(f_1 ,f_2));

if h≠0 then

G_0:={g∈G | lpp(h)≤$_M$lpp(g)};

R:=G_0; P:={h}; G:=G\G_0;

B:=B\ {(f_1 ,f_2) | f_1 ∈ Go или f_2 ∈ G_0};

reduce all (R, P, G, B);

The new basis (P, G, B).

subalgorithms reduce all (выход R, P, G, B):

while R≠0 do

h:= element from R; R:=R \{h};

h :=NF (GUP, h)

if h≠0 then

G_0:={ g ∈ G | lpp (h)≤$_M$lpp(g)};

P_0:={ p ∈ P | lpp (h)≤$_M$lpp(p)};

G:=G\G_0;

$P:=P\backslash P_0$;

$R:=R\cup G_0\cup P_0$;

$B:=B\backslash \{(f_1 ,f_2)\in B \mid f_1 \in G_0$ или $f_2 \in G_0\}$;

Subalgorithms new basis (выход: P, G, B)

$G:=G\cup P$;

$B:=B\cup\{(g,p) \mid g\in G, p\in P, g\neq p\}$;

$H:=G$; $K:=\varnothing$;

while $H \neq 0$ do

h := element from H; $H := H \setminus \{h\}$;

$k:=NF(G \setminus\{h\}, h)$; $K:=K\cup\{k\}\backslash$

G:=K.

subalgorithms criterion 1 (f_1, f_2, G, B) (выход: «да» или «нет»):

There is a $p\in G$, so $f_1\neq p$, $f_2\neq p$, $lpp(p)\leq м$

$\leq_M НОК(lpp(f_1),lpp(f_2)$. $(f_1 ,p)\in B$, $(p,f_2)\notin B$.

subalgorithms criterion 2 (f_1, f_2) (выход: «да» или «нет»):

НОК (Ipp(f_1), lpp(f_2)) = Ipp (f_1) lpp (f_2) .

Abbreviations:

Ipp(f) —" the older the product of powers in the f;

НОК (s, t)— least common divisor s и f;

$s \leq_M t$ — s is a divisor t.

The proof of the correctness of this algorithm is an improved version based on the following lemma and theorem.

Lemma 4 [7, 8]. for arbitrary F, f, g, if HOK (lpp(f_1),

lpp(f_2)) = lpp(f_1)lpp (f_2), then the polynomial SP(f_1,f_2) reduced to zero modulo F.

Theorem 5 [19] (the establishment of unnecessary reductions of S-polynomials). Let S - an arbitrary normal form algorithm.

The following conditions are equivalent:

(БГ 1) F — Grobner basis;

(БГ 5) for every f, g ∈ F there are h_1, ..., h_k ∈ F.

so f = h_1 ,g = h_k, HOK (lpp(h_1, ... lpp(h_k)≤$_M$ HOK(lpp(f),

lpp(g)) и S{F, SP (h_i, h_{i+1})) = 0 for 1≤i<k.

Lemma 4 ensures that there is no need to consider S-polynomial of two polynomials f_1 and f_2. Senior product of powers of which satisfy the condition formulated in Lemma (see Criterion 2). Iteration Criterion 1 in algorithm 3 ensures that at the end of the algorithm, condition (BG 5) holds for G, and hence, G is a Gröbner basis.

Example 6. Let f ={f_1,f_2,f_3}. where

$f_1 = x^3yz — xz^2$, $f_2 = xy^2z — xyz$, $f_3 = x^2y^2 — z^2$.

In this example, we use the ordering of works powers to the full extent: first we order the full degree penalties, and then within a given degree-lexicographic ordering of cally. We took the example of a very simple structure of private polynomials to simplify the process of reduction and stress the following key point: Unlike a straight version of the algorithm from improved, which is expressed in different sets of pairs of polynomials

(f_1, f_2), which are considered S-polynomials.

Minutes of the line version of the algorithm can be the following (if you use the strategy of choice for couples part 1 polynomials, we write $f_i, f_j \rightarrow f_k$ in the record to indicate that that the reduction of S-polynomials of the polynomial f_i, f_j leads to a polynomial f_k):

$f_2, f_3 \rightarrow f_1 = x^2yz - z^3$;

$f_1, f_4 \rightarrow f_5 = xz^3 - xz^2$

$f_2, f_4 \rightarrow f_6 = yz^3 - z^3$

$f_3, f_4 \rightarrow 0$;

$f_5, f_6 \rightarrow f_7 = xyz^2 - xz^2$

$f_4, f_7 \rightarrow f_8 = z^4 - x^2z^2$

$f_2, f_7 \rightarrow 0$;

$f_5, f_7 \rightarrow 0$;

$f_6, f_7 \rightarrow 0$;

$f_5, f_8 \rightarrow f_9 = x^3z^3 - xz^2$

$f_6, f_8 \rightarrow 0$.

S-polynomials of all the other couples be reduced to zero. only need to reduce the 36 S-polynomials.

In the first application of an improved version of subalgorithms "Reduce all" polynomials f_1, f_2, f_3 reduced relative to each other. In this example, this process reduction leaves the original basis of the same. Further, by subalgorithms 'new basis' polynomials f_1, f_2, f_3 recorded in G. At the same time generating set in pairs, for which need to consider their S-polynomials. The first pair again $f_2, f_3 \rightarrow f_4$

At this stage, a subroutine is called again, "reduce all. "The algorithm determines that a polynomial f_1 reduced f_5 modulo (f_2 ,f_3, f_4)- Consequently, the f_1 can be removed from G and respectively, a pair of (f_1 ,f_2) and (f_1 ,f_3) can be removed from B. The use of the subprogram "new basis" adds to the polynomials G f_4 and f_5 and accordingly changes the set B. Consideration of the following pairs of B leads to a reduction f_2 ,$f_4 \rightarrow f_5$

Sub-program "reduce all" in this case does not produce virtually no effect. Thus, the f_6 directly connected to the base and the correspondingly changing the set B. To consider the following pair (f_3 ,f_4) $\in$ B turn to sub-program "Criterion 1", which gives the answer "yes":

lpp(f_2) = xy^2z divide HOK (lpp(f_3), lpp(f_4) = x^2y^2z and pairs (f_3 ,f_2)

и (f_2, f_4) does not belong to B, because they are already subject to review. Appeal to the following pairs leads to reductions

f_5 ,$f_6 \rightarrow f_7$;

f_4 ,$f_7 \rightarrow f_8$

mutatis mutandis, G and B (no reduction and removal of the polynomials from G impossible!). S-polynomials of the following pairs are reduced to zero:

f_2 ,$f_7 \rightarrow 0$;

f_5 ,$f_7 \rightarrow 0$.

Criterion can not establish this fact in advance. However, for the consideration of the following pairs (f_6, f_7) can again involve "a criterion": the polynomial f_5 , p plays a role in the criteria. We then consider the following pairs:

f_5 ,$f_8 \rightarrow f_9$,

f_6 ,$f_8 \rightarrow 0$,

f_4 ,$f_9 \rightarrow 0$.

To consider the following pair (f_7, f_9) You can re-use "criterion 1". Finally, the f_5 ,$f_9 \rightarrow 0$.

Further, the application of the "Criterion 1" indicates a priori without actually performing the reductions, all the remaining pairs can be ignored. Therefore, instead of 36 reductions improved algorithm should perform only 11. pair (f_3 ,f_8) is example of a pair, for which the "Criterion 2" gives the answer "yes." The gain from the use of criteria, in particular, "Criterion 1", it becomes more significant with increasing complexity of the case studies, when considered in terms of the number of variables, the degrees of polynomials and their number.

Buchberger algorithm

Representation and operations with polynomials:

```
type
    TPolynom = record
        List : array [0..mxst,0..mxst] of Double;
    end;
    TMonom = record
        k : Double;
        x, y : integer;
    end;
procedure  TPolynome.Add(m:TMonom);
begin
```

```
        list[m.x, m.y] := list[m.x, m.y]+m.k;
end;
```

Ranking members of the polynomial:

```
function TPolynom.LT : TMonom;
var I, j:integer;
begin
      for i := mxst downto 0 do begin
        for j := mxst downto 0 do begin
            if (abs(list[i,j])<eps) the
              result.k := list[i,j];
              result.x := i;
              result.y := j;
              exit;
end;
end;
```

S-polynom

```
function SPolynom(f1, f2:TPolynom):TPolynom
var m1, m2, k1, k2 :TMonome;
        p2 : TPolynome;
begin
     m1 := f1.LT;
```

```
    m2 := f2.LT;
    k1.k := m2.k;
    k1.x := max(m2.x-m2.x, 0);
    k1.y := max(m2.y-m1.y, 0);
    k2.k := m1.k;
    k2.x := max(m1.x-m2.x, 0);
   k2.y := max(m1.y-m2.y, 0);
   p2 := f2;
   p2.mul(k2);
   result := f1;
   result.mul(k1);
   result.sub(p2);
end;
```

Buchberger algorithm

```
class function TGBasis.Create(f1, f2:TPolynom):TGBasis;
var G:TGBasis;
    k, i : integer;
    s : TPolynom;
 begin
  G := TGBasis.Create;
  G.NewPolynom(f1);
```

```
  G.NewPolynom(f2);
k := 1;
while (k< G.count) and (G.count <= maxPolynoms) do
begin
f1 := Glist[k];
i :=0;
while i<= G.count-1 do begin
   if i = k then begin
      inc(i);
      continue;
    end;
f2 := G.list[i];
s : G.Reduce(SPolynom(f1, f2));
if not s.isConst then begin
   G.NewPolynom(s);
   If G.count > maxPolynoms then break;
end;
    inc(i);
  end;
  inc(k);
end;
result := G;
end;
```

Conclusion

In this paper we have shown the basic concepts of algorithm Buchberger Gröbner basis have shown examples of how they work. The main problem, which is rather difficult to handle is in the process of computing the explosive growth begins integer coefficients. This leads to a very high cost both in CPU time and memory in the calculation bases for real systems. Deal with this problem in two ways: to improve the algorithm for constructing the bases, or create a parallel version of the well-known algorithms. In the first path at this stage the main advance made by the introduction of new heuristics that appear as a result of rich experience in building bases. The second area studied is not enough. Fortunately, these two paths do not contradict each other: most of the improvements achieved in the sequential algorithms can be somehow transferred to the parallel version. At the moment we are on the threshold, when the increase in computer power on the one hand and the improvement of the algorithms on the other hand can afford to handle the actual system of algebraic equations.

Literature

[1] N. K. Bose. Applied multidimensional system theory.—Van Nostrand, N. Y., 1982.

[2] G. Hermann. Die Frage der endlichen vielen Schritte in der Theorie der Polynomideale. — Math. Ann., 1926, B. 95, S. 736—788.

[3] B. Buchberger, R. Loos. Algebraic simplification. — In «Computer algebra. — Symbolic and algebraic computation», Springer, 1982, p. 11—34 (см. перевод в настоящем сборнике).

[4] A. Blass, Yu. Sh. Gurevich Equivalence relations, invariants and normal forms. — Techn. Rep. Dept. Math, and Dept. Сотр. Commun. Sci. Univ. of Michigan, Ann. Arbor, 1983.

[5] E. Cardoza, R. Lipton, A. R. Meyer. Exponential space complete problems for Petri nets and commutative semigroups. — Conf. Record 8th ACM Symp. Th. Comput., 1976, p. 50—54.

[6] E. W. Mayr, A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. — Adv. Math., 1982, v. 46, p. 305—329.

[7] B. Buchberger. An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (in German). — Ph. D. Thesis, Univ. Innsbruck, 1965.

[8] B. Buchberger. An algorithmical criterion for the solvability of algebraic

systems of equations (in German). — Aequationes Math., 1970, v. 4, N 3, p. 374—383.

[9] B. Buchberger. A theoretical basis for the reduction of polynomial to canonical form. —ACM SIGSAM Bull., 1976, v. 10, N 3, p. 19—29.

[10] B. Buchberger. Some properties of Grobner bases for polynomial ideals.— ACE SIGSAM Bull., 1976, v. 10, N 4, p. 19—24.

[11] M. Lauer. Canonical representatives for residue classes of a polynomial ideal (in German). — Diploma Thesis, Univ., Kaiserlautern, 1976.

[12] M. Lauer. Canonical representatives for residue classes of a polynomial ideal. — Proc. ACM Symp. Algebr. Comput., N. Y., 1976, p. 339—345.

[13] R. Schrader. Contributions to constructive ideal theory (in German).— Diploma Thesis, Univ. Karlsruhe, 1976.

[14] D. Spear. A constructive approach to commutative rign theory. — Proc. MACSYMA Users' Conf., MIT, 1977, p. 369—376.

[15] W. Trinks. On Buchberger's method for solving systems of algebraic equations. — J. Number Theory, 1978, v. 10, N4, p. 475—488.

[16] F. Winkler. Implementation of an algorithm for constructing Grobner bases (in German). — Diploma Thesis, Univ. Linz, 1970.

[17] G. M. Bergman The diamond lemma for ring theory. — Adv. Math, 1978 v. 29, p. 178—218.

[18] G. Zacharias. Generalized Grobner bases in commutative polynomial rings. — Bachelor Thesis, MIT, Dept. Comput. Sci., 1978.

[19] B. Buchberger. A criterion for detecting unnecessary reductions in the construction of Grobner bases. — Lect. Notes Comput. Sci., 1979 v. 72, p. 3—21.

[20] B. Buchberger, F. Winkler. — Miscellaneous results on the construction of Grobner bases for polynomial ideals I. — Techn. Rep. N137, Univ. Linz, 1979.

[21] S. Schaller. Algorithmic aspects of polynomial residue class rings.— Ph. D. Thesis, Techn. Rep. N 370, Univ. Wiskonsin-Madison, Dept. Comput. Sci., 1979.

[22] L. Bachmair, B. Buchberger. A simplified proof of the characterization theorem for Grobner bases. — ACM SIGSAM Bull., 1980, v. 14, N4, p. 29—34.

[23] A. M. Ballantyne, D. S. Lankford. New decision algorithms for finitely presented commutative semigroups. — Computers and Math, with Appl., 1981, v. 7, p. 159—165.

[24] G. Bauer. The representation of monoids by confluental rule systems. — Ph. D. Thesis, Univ. Kaiserlautern, 1981.

[25] F. Mora. An algorithm to compute the equations of tangent cones. — Lect.

Notes Comput. Sci., 1982, v. 144, p. 158—165.

[26] M. Pohst, D. Y. Yun. On solving systems of algebraic equations via ideal bases and elimination theory. — Proc. ACM Symp. Symb. Algebr. Comput., ACM, 1981, p. 206—211.

[27] J. P. Guiver. Contributions to two-dimensional systems theory. — Ph. D. Thesis, Univ. Pittsburgh, 1982.

[28] D. Bayer. The division algorithm and the Hilbert scheme. — Ph. D. Thesis, Harvard Univ., 1982.

[29] B. Buchberger. A note on the complexity of constructing Grobner bases. — Lect. Notes Comput. Sci., 1983, v. 162, p. 137—145.

[30] B. Buchberger. A critical-pair/completion algorithm for finitely generated ideals in rings. — Lect. Notes Comput. Sci., 1984, v. 171.

[31] D. Lazard. Grobner bases, Gaussian elimination and resolution of systems of algebraic equations. — Lect. Notes Comput. Sci., 1983, v. 162, p. 146— 156.

[32] R. Llopis de Trias. Canonical forms for residue classes of polynomial ideals and term rewriting systems. — Prepr. Univ. Aut. Madrid, 1983.

[33] F. Mora, H. M. Moeller. The computation of the Hilbert function. — Lect. Notes Comput. Sci., 1983, v. 162, p. 157—167.

[34] F. Mora, H. M. Moeller. New constructive methods in classical ideal theory. — Prepr. Univ. Genova, 1983.