Annals of Pure and Applied Logic 59 (1993) 273–286
North-Holland

A decision algorithm for linear sentences on a PFM*

Lian Li

Department of Mathematics, Lanzhou University, Lanzhou 730000, People's Republic of China

Huilin Li

Department of Mathematics, Zhejiang University, Hangzhou 310027, People's Republic of China

Yixun Liu

Department of Mathematics, Lanzhou University, Lanzhou 730000, People's Republic of China

Communicated by A. Prestel
Received 2 June 1991
Revised 2 April 1992

Abstract

Li, L., H. Li and Y. Liu, A decision algorithm for linear sentences on a PFM, Annals of Pure and Applied Logic 59 (1993) 273–286.

By PFM, we mean a finitely generated module over a principal ideal domain; a linear sentence is a sentence that contains no disjunctive and negative symbols. In this paper, we present an algorithm which decides the truth for linear sentences on a given PFM, and we discuss its time complexity. In particular, when the principal ideal domain is the ring of integers or a univariate polynomial ring over the field of rationals, the algorithm is polynomial-time. Finally, we consider some applications to Abelian groups.

By PFM, we mean a finitely generated module over a principal ideal domain. A linear sentence on a PFM is a sentence which is a conjunction of atomic formulas preceded by an arbitrary finite string of quantifiers. In this paper, we discuss the problem how to decide if a linear sentence is true, and the complexity of the presented algorithm.

Our decision method for linear sentences is algebraic. In specific, it transforms a linear sentence into a linear system of equations; thus it converts the deciding problem for truth of sentences into the symbolic computation problem for linear systems.

In this paper we prove that there is an algorithm to decide, for any linear sentence $\Pi(\psi)$, whether or not $\Pi(\psi)$ holds on M when R is a computable

Correspondence to: Lian Li, Department of Mathematics, Lanzhou University, Lanzhou 730000, People's Republic of China.

* Project supported by National NSF of China.

principal ideal domain (see Definition 5), M is a finitely generated module over R. Further, the algorithm is polynomial-time when R is a ring such as the field of rationals, the ring of integers or a univariate polynomial ring over the field of rationals.

For undefined notions we refer to [4], [6].

1. Linear sentences and linear systems of equations

Let R be a principal ideal domain (PID for short), and let

$$A = \{a_1, a_2, \ldots, a_m\},$$

be a finite alphabet; a free module $M(A)$ generated by A over R is the module consisting of formal linear combinations

$$\sum_{i=1}^{m} d_i a_i.$$

Elements in $M(A)$ are denoted by m-dimensional vectors over R. We consider defining relations

$$\alpha_i = 0, \quad i = 1, 2, \ldots, k \tag{1.1}$$

where $\alpha_i = \sum_{j=1}^{m} d_{ij} a_j$, $i = 1, 2, \ldots, k$, are in $M(A)$. Let N be the submodule of $M(A)$ generated by these α_i. The quotient module $M(A)/N$ is called the module defined by defining relations (1.1).

Definition 1. Let R be a PID. We recall that a module M on R is given, if we are given a finite alphabet

$$A = \{a_1, a_2, \ldots, a_m\}$$

and defining relations G on A

$$\sum_{j=1}^{m} d_{ij} a_j = 0, \quad i = 1, 2, \ldots, k,$$

where $d_{ij} \in R$, $a_j \in A$, $j = 1, 2, \ldots, m$, such that M is, up to isomorphism, the module defined by A and G.

Every finitely generated module can be given in this way.

The matrix consisting of the coefficients d_{ij} of α_i

$$\begin{bmatrix} d_{11} & d_{21} & \cdots & d_{k1} \\ d_{12} & d_{22} & \cdots & d_{k2} \\ d_{13} & d_{23} & \cdots & d_{k3} \\ \cdot & \cdot & \cdots & \cdot \\ d_{1m} & d_{2m} & \cdots & d_{km} \end{bmatrix}$$

is called the defining matrix determined by G.

Every atomic formula on a module M has the form

$$\sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{m} e_j a_j = 0,$$

where $c_i, e_j \in R$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Each variable x_i occurs only once. In the sequel, formulas and sentences are always written in prenex normal form.

$$\omega = Q_1 x_1 Q_2 x_2 \cdots Q_s x_s(\psi),$$

where Q_i's are quantifiers, $i = 1, 2, \ldots, n$, ψ is a quantifier-free formula. ψ is called the matrix of ω, and

$$\Pi = Q_1 x_1 Q_2 x_2 \cdots Q_s x_s$$

is called the quantifier prefix of ω. For simplicity, we write the successive universal or existential variables occurring in Π in vector form like $(x_{i1}, x_{i2}, \ldots, x_{ir})^{\mathrm{T}}$. Thus Π is rewritten as

$$\Pi = Q_1 X_1 Q_2 X_2 \cdots Q_n X_n, \tag{1.2}$$

where Q_i and Q_{i+1} are distinct quantifiers, $i = 1, 2, \ldots, n-1$. There is no difference between the ordinary form and the vector form but the writing pattern. The number n of quantifiers occurring in the vector form of Π is called the depth of Π, and is denoted by $\mathrm{dep}(\Pi)$. Without loss of generality, we suppose that in the vector form of (1.2), the first quantifier is a universal quantifier and the last one is an existential quantifier. The reason is that we can always form a new sentence $\forall x\, \Pi\, \exists y\, (x = x \wedge y = y \wedge \psi)$ which is equivalent to $\Pi(\psi)$.

Definition 2. A sentence $\omega = Q_1 X_1 Q_2 X_2 \cdots Q_n X_n(\psi)$ is called *linear*, if ψ is a conjunction of atomic formulas.

Example 1. Let $\mathbb{Z}$ be the ring of integers, $A = \{a, b, c\}$, so the sentences $\forall x\, (2x = 0)$, $\exists x\, (4x - 2c = 0)$ are linear sentences. M_1 given by the defining relations

$$a + b - c = 0, \qquad 2a = 0, \qquad 2b = 0,$$

is the Klein 4-group; M_2 given by the defining relations

$$8a = 0, \qquad 4a - b = 0, \qquad 2a - c = 0,$$

is the cyclic group of order 8. It is obvious that the sentence $\forall x\, (2x = 0)$ holds in M_1, but does not hold in M_2. And $\exists x\, (4x - 2c = 0)$ holds both in M_1 and M_2.

Let $\alpha_i = \sum_{j=1}^{m} d_{ij} a_j$, $i = 1, 2, \ldots, k$, be defining relations of a module M. For any atomic formular $\beta = 0$ where

$$\beta = \sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{m} e_j a_j. \tag{1.3}$$

$\beta = 0$ is true in M for some assignment of x_i iff there exist values of y_{ij}, z_i in R such that the following equality

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_i y_{ij} a_j + \sum_{j=1}^{m} e_j a_j + \sum_{i=1}^{k}\sum_{j=1}^{m} (d_{ij} z_i) a_j = 0$$

holds in M. It is equivalent to say that the following system of linear equations

$$\begin{aligned}
&\sum_{i=1}^{n} c_i y_{i1} + \sum_{i=1}^{k} d_{i1} z_i = -e_1,\\
&\sum_{i=1}^{n} c_i y_{i2} + \sum_{i=1}^{k} d_{i2} z_i = -e_2,\\
&\cdots,\\
&\sum_{i=1}^{n} c_i y_{im} + \sum_{i=1}^{k} d_{im} z_i = -e_m
\end{aligned}$$

has solutions. We write the system as $AX + DZ = E$, where D is called the defining matrix which depends only on the module M. This linear system is called the system corresponding to the atomic formula $\beta = 0$.

For a linear sentence ω,

$$\omega = \Pi\Big(\bigwedge_{i=1}^{t} (\beta_i = 0)\Big), \tag{1.4}$$

let $A_i X_i + DZ_i = E_i$ be the corresponding linear systems to $\beta_i = 0$, $i = 1, 2, \ldots, t$. The system

$$\begin{aligned}
&A_1 X_1 + DZ_1 = E_1,\\
&A_2 X_2 + DZ_2 = E_2,\\
&\cdots,\\
&A_t X_t + DZ_t = E_t
\end{aligned} \tag{1.5}$$

is called the system corresponding to the linear sentence ω. We write this system as $AX = E$. Note that all unknowns in different Z_i are mutually disjoint, but there may be the same unknowns in different X_i since different β_i may include the same variables.

According to the quantifiers (1.2), we write $AX = E$ as

$$A_1 X_1 + A_2 X_2 + \cdots + A_n X_n = E,$$

where $(A_1, A_2, \ldots, A_n) = A$. The system $AX = E$ has $m * n + t * k$ unknowns where m is the number of elements in the generating set A, n is the number of variables in ω, t is the number of atomic formulas in ω, and k is the number of relations in the defining relations of M. In such a system, the m unknowns y_{ij} are determined by the variable x_i in ω. Motivated by this point, we define the expansion form for the quantifiers Π as follows.

Let $\Pi = Q_1x_1Q_2x_2 \cdots Q_nx_n$ be a quantifier prefix. For every quantifier Q_ix_i in Π, we substitute a group of quantifiers

$$Q_iY_i = Q_iy_{i1}Q_iy_{i2} \cdots Q_iy_{im}.$$

Q_iY_i is called the expansion of Q_ix_i (in terms of generating set A and defining relations G). The quantifier prefix

$$Q_1Y_1Q_2Y_2 \cdots Q_nY_n \exists Z_1 \cdots \exists Z_t$$

is called the expansion form of Π (in terms of generating set A, defining relations G and sentence ω) and is denoted as $E(\Pi)$.

Definition 3. A linear system $AX = E$ is said to satisfy $E(\Pi)$, if the sentence $E(\Pi)(AX = E)$ is true in R.

In the sequel, $E(\Pi)$ is always written in vector form; the number of quantifiers occurring in the vector form of $E(\Pi)$ is called the depth of $E(\Pi)$. Thus $E(\Pi)$ has the same depth as Π by our agreement that the last quantifier of Π is an existential quantifier.

Theorem 1. *A linear sentence $\omega = \Pi(\bigwedge_{i=1}^{t} \beta)$ is true in M iff the corresponding system of linear equations $AX = E$ satisfies $E(\Pi)$.*

Proof. This is obvious from the discussion above. □

When a linear sentence has n variables and t atomic formulas, the corresponding system $AX = E$ contains not more than $n*m + t*k$ unknowns and $t*m$ equations. Thus the transformation from a linear sentence to a system of linear equations is very practical.

Theorem 2. *When the system $AX = E$ has solutions, $AX = E$ satisfies $E(\Pi)$ iff the homogeneous system $AX = 0$ satisfies $E(\Pi)$.*

Proof. We note that the geometric sense of Theorem 2 is obvious. For any nonhomogeneous system $AX = E$, its solutions are a parallel translation of the solutions of the homogeneous system $AX = 0$. With a parallel translation, however, there is no change of satisfiability.

The detailed proof can be done by induction on the quantifier depth $\text{dep}(\Pi)$, and is omitted. □

2. A sufficient and necessary condition for satisfiability

Let $AX = E$ be a system of linear equations. When does the system satisfy quantifiers $E(\Pi)$? The first thing at all is that the system must be solvable. For

otherwise it is impossible to satisfy $E(\Pi)$. Furthermore, $AX = E$ satisfies $E(\Pi)$ iff the homogeneous system $AX = 0$ satisfies $E(\Pi)$ by Theorem 2. So we can restrict our discussion to the homogeneous case.

Let Ω be the set of solutions of equations $AX = 0$ and let

$$E(\Pi) = Q_1X_1Q_2X_2 \cdots Q_pX_p$$

be the expansion form of quantifier prefix Π. We denote the projection of an element $\alpha \in \Omega$ on X_i by $\alpha(X_i)$, i.e., $\alpha(X_i)$ are those components of α corresponding to the unknowns

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{ir})^{\mathrm{T}}.$$

Let $\Omega(X_i) = \{\alpha(X_i) \mid \alpha \in \Omega\}$.

Definition 4. The set of Ω of solutions of $AX = 0$ is said to be *ergodic* on the unknowns

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{ir})^{\mathrm{T}},$$

if $\Omega(X_i) = R^{(r)}$. Ω is said to be ergodic on $E(\Pi)$, if Ω is ergodic on all universal variables X_{2i-1} of $E(\Pi)$, $i = 1, 2, \ldots, p/2$.

For a given X_i, let

$$\Delta_i = \{\alpha \in \Omega \mid \alpha(X_i) = 0\}, \quad i = 1, 2, \ldots, p.$$

Δ_i is a submodule of Ω, $i = 1, 2, \ldots, p$, and $\bigcap_{i=1}^{p} \Delta_i = \{0\}$.

Lemma. *Given a residue class* $\alpha + \bigcap_{i=1}^{s} \Delta_i$ *of submodule* $\bigcap_{i=1}^{s} \Delta_i$ *in* Ω, *where* $\alpha \in \Omega$, *then for all residue classes* $\beta + \Delta_j$ *of submodule* Δ_j *in* Ω,

$$\left(\alpha + \bigcap_{i=1}^{s} \Delta_i\right) \cap (\beta + \Delta_j) \neq \emptyset,$$

iff $\bigcap_{i=1}^{s} \Delta_i + \Delta_j = \Omega$.

Proof. For arbitrary $u \in \Omega$, from

$$\left(\alpha + \bigcap_{i=1}^{s} \Delta_i\right) \cap (\beta + \Delta_j) \neq \emptyset$$

holds for all $\beta \in \Omega$, we have $\alpha + \delta_1 = \alpha - u + \delta_2$ for some δ_1 in $\bigcap_{i=1}^{s} \Delta_i$, and δ_2 in Δ_j. So

$$u = \delta_2 - \delta_1 \in \bigcap_{i=1}^{s} \Delta_i + \Delta_j.$$

Conversely, for any $\beta + \Delta_j$, from

$$\bigcap_{i=1}^{s} \Delta_i + \Delta_j = \Omega.$$

we have $\delta_1 - \delta_2 = \beta - \alpha$ for some $\delta_1 \in \bigcap_{i=1}^{s} \Delta_i$, and $\delta_2 \in \Delta_j$. So $\alpha + \delta_1 = \beta + \delta_2$. That is $(\alpha + \bigcap_{i=1}^{s} \Delta_i) \cap (\beta + \Delta_j) \neq \emptyset$. □

Theorem 3. *Let $E(\Pi) = Q_1 X_1 Q_2 X_2 \cdots Q_p X_p$. A system of linear equations $AX = 0$ satisfies $E(\Pi)$, iff the set Ω of solutions of $AX = 0$ is ergodic on $E(\Pi)$, and*

$$\bigcap_{i=1}^{2r} \Delta_i + \Delta_{2r+1} = \Omega, \quad i = 1, 2, \ldots, p - 2/2.$$

Proof. Here we give the brief proof of this theorem. For the details we refer to [11].

We write the system $AX = 0$ as

$$AX = A_1 X_1 + A_2 X_2 + \cdots + A_p X_p = 0.$$

Let $\forall X_1 \exists X_2 \cdots \exists X_{2r} \forall X_{2r+1}$ be a prefix of $E(\Pi)$. Since $AX = 0$ satisfies $E(\Pi)$, there are valuations of $X_1, X_2, \ldots, X_{2r}$, say $\alpha_1, \alpha_2, \ldots, \alpha_{2r}$, such that for any valuation α_{2r+1} of X_{2r+1}, the system

$$A_1 \alpha_1 + \cdots + A_{2r+1} \alpha_{2r+1} + A_{2r+2} X_{2r+2} + \cdots + A_p X_p = 0$$

is solvable. Let ν be a solution of the system, then

$$\beta = (\alpha_1, \alpha_2, \ldots, \alpha_{2r}, \alpha_{2r+1}, \nu)^{\mathrm{T}}$$

is a solution of $AX = 0$. The projection of β on X_i is α_i, $i = 1, 2, \ldots, 2r$, respectively. So all the solutions which have projection α_i on X_i, $i = 1, 2, \ldots, 2r$, are exactly the elements in $\beta + \bigcap_{i=1}^{2r} \Delta_i$.

Let γ be a solution of $AX = 0$ which has projection α_{2r+1} on X_{2r+1}. Then all solutions which have projection α_{2r+1} on X_{2r+1}, are exactly the elements in $\gamma + \Delta_{2r+1}$. When γ runs through all solutions of $AX = 0$, the projections of γ on X_{2r+1} run through all vectors in $R^{(s)}$ (where s is the dimension of X_{2r+1}) because of the ergodicity on X_{2r+1}. Thus we have, for any $\gamma \in \Omega$,

$$\left(\beta + \bigcap_{i=1}^{2r} \Delta_i\right) \cap (\gamma + \Delta_{2r+1}) \neq \emptyset,$$

and $\bigcap_{i=1}^{2r} \Delta_i + \Delta_{2r+1} = \Omega$ by the lemma.

By inverting this procedure we have a proof for the other direction. □

3. Computable PID, Smith canonical form

A ring R is called *discrete,* if every element in R can be specified in a finite form. From the viewpoint of computations, a computer can only work on a discrete ring. But there is no impairment for our discussion for real fields, polynomial rings, etc., even if these rings are non-discrete. When a sentence over a PFM is given, the objects we need to handle are always concrete and finitely representable. So we work within a discrete subring of the non-discrete ring in practice. This becomes clearer when we inspect the work of theorem-proving on elementary geometry in [15], which works in the field of real numbers, but we never bump into the 'genuine' real numbers in the computing process, so the elementary geometry is working, in fact, in a discrete subfield of the field of real numbers.

Definition 5. A PID R is called *computable,* if

(1) for arbitrary a and b, it is decidable if $a = b$;

(2) the addition and multiplication are computable;

(3) the factorization is computable;

(4) for relatively prime elements x, y, elements a, b such that $ax + by = 1$ are computable;

If R is an Euclidean ring then the conditions (3), (4) are replaced by:

(3)′ the division with remainder is computable.

For further discussion of the conditions of Definition 5 we refer to [13]. Almost all PIDs practical in research are computable PIDs. In the following discussion, PID R is always assumed computable.

Let R be a PID; we denote the unit matrix by I, and the matrix which has 1 at the ith row and jth column and 0 elsewhere by e_{ij}. Let

$$T_{ij}(b) = I + be_{ij}, \quad b \in R,$$

$$D_i(u) = I + (u-1)e_{ii}, \quad u \textit{ is an invertible element of } R,$$

$$P_{ij} = I - e_{ii} - e_{jj} + e_{ij} + e_{ji}.$$

Furthermore, we introduce a class of invertible matrices:

$$\begin{bmatrix} x & y & 0 \\ s & t & 0 \\ 0 & 0 & I \end{bmatrix}, \qquad \text{where } \det\begin{bmatrix} x & y \\ s & t \end{bmatrix} = 1.$$

By multiplying a matrix A from the left resp. the right by one of the four class matrices above leads to a row resp. column transformation of A.

An $m * n$ matrix H is called upper (lower) Hermite, if for all $j < i$, $H(i, j) = 0$ (for all $j > i$, $H(i, j) = 0$), where $H(i, j)$ denotes the element at the ith row and jth column of H.

An $m * n$ matrix S is called Smith if $S = (B, 0)$ or $S = (B, 0)^{\mathrm{T}}$ where B is a square diagonal matrix

$$\mathrm{diag}(d_1, d_2, \ldots, d_r, 0, \ldots, 0).$$

with $d_i \neq 0$, $i = 1, 2, \ldots, r$ (the divisibility of d_{i+1} by d_i is not required).

Reference [13, pp. 344, Theorem 2] provides a method to construct a base of solutions for a homogeneous system of linear equations on an Euclidean ring. It is easy to prove that the method is appropriate for PIDs; so we have

Theorem 4 [13]. *Let $AX = 0$ be a system of linear equations on a* PID *R with m equations and n unknowns and the coefficient matrix A has rank r, $S = PAQ$ is the Smith canonical form, where P, Q are invertible matrices with $m * m$ and $n * n$ elements respectively. Then the last $n - r$ columns of Q are a base of solutions of $AX = 0$.*

By Theorem 4, the crucial step to find a base of solutions is to convert the coefficient matrix into Smith canonical form. At this point most lectures talk about how to convert a matrix on a Euclidean ring or PID into Smith canonical form or Hermite canonical form, (e.g. [6]), but it was unknown whether or not there is a polynomial algorithm to solve a linear system in the ring of integers until 1976. In that year, a polynomial algorithm was found for the solution of linear systems in $\mathbb{Z}$ (the ring of integers, [14]). Later, a polynomial algorithm for Smith or Hermite canonical forms was obtained for integer matrices in 1979, and for matrices over the ring of univariate polynomials with rational coefficients in 1985 (see [9, 10]). Here we provide a rough method which converts a matrix on a PID into Smith canonical form. We can prove that the cost for the arithmetic operations is not more than a polynomial of the product of the primary length of a longest minor in A and the order of A.

Definition 6. The *primary length* of an element a in R is its number of prime factors plus 1. The primary length of element 0 is 1.

In a PID, the primary length is well-defined.

A minor of A of longest primary length is called the longest minor of A; the longest minor is, in general, not unique. The greatest common divisor of all minors of order r is called the r-factor of A which is invariant under the transformations mentioned before.

Definition 7. The *order* of a matrix A is the maximum of its row number and column number.

Definition 8. The *size* of a matrix A is the product of the primary length of a longest minor and the order of A.

The following theorem is about how to convert a matrix into Hermite or Smith form. Note that in our convention all algebraic operations including the computation of g.c.d. can be completed in one step.

Theorem 5. *There is a polynomial algorithm with respect to the size of a matrix A to convert A into Smith form.*

Proof. Let A be an $m \times n$ matrix. Without loss of generality, we assume $m \leqslant n$ and $\text{rank}(A) = m$ (i.e., A is row full rank). Let ζ be the primary length of a longest minor of A; a_{11} is the element at the 1st row and 1st column and assume $a_{11} \neq 0$. We can get a matrix which has 0 in its 1st column except a_{11} by use of the four class transformations. So we arrive at a matrix

$$\begin{bmatrix} d & * \\ 0 & B \end{bmatrix}$$

We continue this procedure for B; thus we convert A into an upper Hermite matrix A_1. In turn, we convert A_1 into a lower Hermite matrix A_2 by the same procedure above for columns. This converting from A into A_1 and then into A_2 is called a UL transformation. Now A_2 is a lower triangle matrix and the product c of diagonal elements is the unique nonzero m-minor, so c is the m-factor of A which is invariant under transformations. Therefore primary length of c is less than or equal to ζ. Especially, the primary length of every element on the diagonal is not more than ζ. For each UL transformation the primary length of a_{11} is reduced by at least 1 until a_{11} divides all elements in the first row and first column. By at most ζ UL transformations we get a matrix which has elements 0 in its 1st row and 1st column except a_{11}. The problem is reduced with respect to the order of the matrix. Note that the primary length of diagonal elements is never more than ζ, so we get a Smith matrix by at most $m * \zeta$ UL transformations. It is obvious that a UL transformation is polynomial-time with respect to the size of A. Thus the whole algorithm is also polynomial-time. □

Many PIDs such as the ring of integers, the polynomial ring over rationals, the ring of Gauss integers, etc., possess the property that the input length of any element does not exceed a polynomial of its primary length, so the size of a matrix over these rings does not exceed a polynomial of its input length. For these rings, we have in fact an algorithm which converts a matrix into Smith form in polynomial-time with respect to the input length of the matrix by Theorem 5.

The main defect in this conversion is that we ignore the cost of arithmetic operations (all of these operations are completed in one step). But in practice, the entries of a matrix could become very large during the converting procedure and the cost of arithmetic operations could be very high. So the algorithm of Theorem 5 is not very efficacious. The practical complexity of converting a matrix into Smith form has not been obtained yet for many PIDs. However, when the

ring is the ring of integers or a univariate polynomial ring over a rational field, there is a converting algorithm which is polynomial-time with respect to the input length [8–10].

4. The decision for satisfiability and the analysis of complexity

In terms of the discussion above, the procedure to decide if $AX = 0$ satisfies a quantifier prefix $E(\Pi)$ consists of the following computations and decisions:

(1) computation of a base of solutions;
(2) decision for ergodicity;
(3) computation of the submodule $\bigcap_{i=1}^{r} \Delta_i$ of Ω;
(4) decision whether or not $\bigcap_{i=1}^{r} \Delta_i + \Delta_{r+1} = \Omega$.

In this section, we give an algorithm to do them.

Problem (1). Computation of a base of solutions for a system of linear equations, as discussed in Theorems 4 and 5.

Problem (2). Decision for the ergodicity on $E(\Pi)$

Let Ω be the set of solutions of $AX = 0$, and let $\alpha_1, \alpha_2, \ldots, \alpha_t$, be a base of Ω. To decide whether or not Ω is ergodic on universal variables X_j with dimension s, we denote the projection of α_i on X_j as β_i, $i = 1, 2, \ldots, t$. Thus Ω is ergodic on X_j iff for any s-dimensional standard vector δ_k, i.e.

$$\delta_k = (0, \ldots, 0, \underset{k}{1}, 0, \ldots, 0), \quad k = 1, 2, \ldots, r,$$

the system

$$(\beta_1, \beta_2, \ldots, \beta_t)Y = \delta_k, \tag{4.1}$$

has a solution. In this case, for any $\delta \in R^{(r)}$, let $(y_1, y_2, \ldots, y_r)^{\mathrm{T}}$ be a solution of $(\beta_1, \beta_2, \ldots, \beta_t)Y = \delta$. Then $y_1\alpha_1 + y_2\alpha_2 + \cdots + y_r\alpha_r$ is a solution of $AX = 0$, and has projection δ on X_j. So Ω is ergodic on X_j.

Problem (3). Computation of the submodule $\bigcap_{i=1}^{r} \Delta_i$ of Ω.

It is obvious that $\bigcap_{i=1}^{r} \Delta_i$ is the set consisting of those elements in Ω which take value 0 on X_i, $i = 1, 2, \ldots, r$. Again let $\alpha_1, \alpha_2, \ldots, \alpha_t$ be a base of Ω, and let β_i be the projection of α_i on variables $(X_1, X_2, \ldots, X_r)$, $i = 1, 2, \ldots, t$; i.e., β_i are those components of α_i corresponding to the first r variables $X_1, X_2, \ldots, X_r$ of $E(\Pi)$. Consider the following system

$$(\beta_1, \beta_2, \ldots, \beta_t)Y = 0. \tag{4.2}$$

If $Y_0 = (c_1, c_2, \ldots, c_t)^T$ is a solution of (4.2), then

$$\alpha = c_1\alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t$$

is in $\bigcap_{i=1}^{r} \Delta_i$ and vice versa. Furthermore, if $Y_1, Y_2, \ldots, Y_q$ is a base of solutions of (4.2), then the vectors $\gamma_1, \gamma_2, \ldots, \gamma_q$ form a base of $\bigcap_{i=1}^{r} \Delta_i$ where

$$\gamma_i = (\alpha_1, \alpha_2, \ldots, \alpha_t)Y_i, \quad i = 1, 2, \ldots, q.$$

So we can get a basis of $\bigcap_{i=1}^{r} \Delta_i$ from a base of solutions of (4.2).

Problem (4). Decision whether or not $\bigcap_{i=1}^{r} \Delta_i + \Delta_{i+1} = \Omega$.

Let Ω have dimension t, and let $\alpha_1, \alpha_2, \ldots, \alpha_q \in \bigcap_{i=1}^{r} \Delta_i$ be a basis of $\bigcap_{i=1}^{r} \Delta_i$, let $\beta_1, \ldots, \beta_s \in \Delta_{r+1}$ be a basis of Δ_{r+1}. Then $\bigcap_{i=1}^{r} \Delta_i + \Delta_{r+1} = \Omega$ iff for every standard vector δ_i of dimension t, $i = 1, 2, \ldots, t$, the system

$$(\alpha_1, \alpha_2, \ldots, \alpha_q)X + (\beta_1, \beta_2, \ldots, \beta_s)Y = \delta_i$$

is solvable.

Now we give an algorithm to decide the truth of a linear sentence.

Algorithm Truth ($\Pi(\psi)$)
(To decide if a given linear sentence $\Pi(\psi)$ is true)

Input. A linear sentence $\Pi(\psi)$ with first quantifier universal and last quantifier existential.

Output. Y, if $\Pi(\psi)$ is true; N, else.

Step 1. Convert ψ into the corresponding system of linear equations $AX = E$.

Step 2. Rewrite Π in its expansive form $E(\Pi)$ about ψ.

Step 3. If $AX = E$ has no solution then halt with output 'N'; else compute a base of the set Ω of solutions of $AX = 0$.

Step 4. Decide ergodicity of Ω; if not then halt, output 'N'.

Step 5. Compute submodules Δ_i, $i = 1, 2, \ldots, p$, where p is the quantifier depth of $E(\Pi)$.

Step 6. Decide if

$$\bigcap_{i=1}^{2r} \Delta_i + \Delta_{2r+1} = \Omega, \quad r = 1, 2, \ldots, p - 2/2.$$

The core of the algorithm is to find a base of solutions of the system. In addition, some components of elements in a base from preceding systems will often be coefficients of later systems, so the size of the components of elements in a base is important for the running time of the algorithm. Unfortunately, it is comparatively difficult to estimate such bounds. We hope this difficulty will be overcome by developing new algorithms.

However, for the following three rings:
(1) the field of rationals,
(2) the ring of integers,
(3) the univariate polynomial ring over the field of rationals,
it has been proved that both the running time and the size of components of a base are polynomial bounded with respect to input length, where (1) is a special case of (3); and for (2), (3) we refer to [9, 10].

Theorem 6. *For a given generating set A and a defining relation G, when R is one of the three kind of rings above, then the algorithm* Truth($\Pi(\psi)$) *is polynomial time with respect to the input length of* $\Pi(\psi)$.

Proof. Step 1 in algorithm Truth($\Pi(\psi)$) can be completed in polynomial time. Step 2 is routine. Step 3 is the computation of a Smith canonical form; it is polynomial time by the discussion above. In Step 4, we need to solve $p/2$ systems of linear equations, which can be completed in polynomial time by [9, 10]. By the same reasons, Step 5 and Step 6 can also be completed in polynomial time with respect to the length of $\Pi(\psi)$. □

From these discussion we can see that, for a concrete ring R, we have a polynomial time algorithm to decide the truth of a linear sentence whenever there is a polynomial time algorithm to compute a base of solutions of a system $AX = 0$ on R.

5. Application and conclusion

A finitely generated Abelian group is a finitely generated module over the ring of integers. From the results above, we have a polynomial time algorithm to decide the truth of a linear sentence over a finitely generated Abelian group.

Theorem 7. *Given a generating set and definining relations of a finitely generated Abelian group G, we can decide the truth of linear sentences on G in polynomial time.*

Remark 1. There are deep and close relations between theorem-proving for algebraic systems and algebraic symbolic computations. In fact, a purely algebraic system is a system which has no predicate symbols (except equality); then an atomic formula corresponds to an algebraic equation, and a quantifier-free formula corresponds to a group of systems of equations and inequations, so the truth of a sentence ψ is naturally dependent on the distribution of the solutions of these systems.

Remark 2. Most results about theorem-proving work on a class of models not a single model. For example, the theorem-proving on finitely generated Abelian groups checks those sentences which hold at all Abelian groups[5]; this leads often to very high complexity and makes it hard to apply these theoretical results. In this paper we consider the truth of a linear sentence on a concrete model; this reduces the complexity of decision considerably. Of course, another reason for the reduced complexity is the restriction to linear sentences. In [5], the complexity to decide the truth of a sentence on a class of models of finitely generated Abelian groups is at least NTIME($2^{2^{cn}}$), and the complexity of decision of a linear sentence in a concrete model is merely polynomial time. This indicates partially that there are great differences in complexity although the truth decision on a class of models sometimes implies the decision on a single model. Indeed, in some fields of algebra, we only need to know the truth of the sentences on a concrete model; the lower complexity is favorable to practical applications on a computer.

Acknowledgements

We would like to thank Professor Zhao Ying for his helpful discussions, and the referee for many useful suggestions.

References

[1] R.V. Book, Decidable sentences of Church–Rosser congruences, Theoret. Comput. Sci. 24 (1983) 301–312.
[2] R.V. Book, True systems as rewriting systems, J. Symbolic Computation 3 (1987) 39–68.
[3] V.J. Digricoli and M.C. Harrison, Equality-based binary resolution, J. ACM 33 (2) (1986) 253–289.
[4] H.B. Enderton, A Mathematical Introduction to Logic (Academic Press, New York, 1972).
[5] J. Ferrante and C.W. Rackoff, The Computational Complexity of Logical Theories, Lecture Notes in Math. 718 (Springer, Berlin, 1979).
[6] N. Jacobson, Basic Algebra I (Freeman, San Francisco, CA, 1974).
[7] N. Jacobson, Basic Algebra II (Freeman, San Francisco, CA, 1980).
[8] E. Kaltofen, M. Krishnamoorthy and B. Saunders, Fast parallel computation of Hermite and Smith forms of polynomial matrices, SIAM J. Algebraic Discrete Methods 8 (4) (1987) 683–690.
[9] R. Kannan, Solving systems of linear equations over polynomials, Theoret. Comput. Sci. 39 (1985) 69–88.
[10] R. Kannan and A. Bachem, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix, SIAM J. Comput. 8 (4) (1979) 499–507.
[11] Li Lian, Li Huilin and Liu Yixun, On the satisfiability of a system of linear equations for quantifiers, Scientia Sinica, to appear.
[12] D.W. Loveland, Automated theorem-proving: a quarter-century review, in: Contemporary Mathematics, Vol. 29: Automated Theorem Proving: After 25 Years, 1–46.
[13] C.C. Sims, Abstract Algebra: A Computational Approach (Wiley, New York, 1984).
[14] E. Specker and V. Strassen, Komplexität von Entscheidungs-problemen (Springer, Berlin, 1976).
[15] Wenjun Wu, The Basic Principles of Theorem Proving in Elementary Geometry (Academia Sinica, 1984).