A decision algorithm for prenex normal form rational Presburger sentences based on combinatorial geometry

Naoki SHIBATA Kozo OKANO Teruo HIGASHINO Kenichi TANIGUCHI

Department of Informatics and Mathematical Science, Osaka University
Machikaneyama 1-3, Toyonaka, Osaka 560-8531, JAPAN
E-mail : { n-sibata, okano, higashino, taniguchi }@ics.es.osaka-u.ac.jp

Abstract In this paper, we propose a decision algorithm for theory of rationals with addition (the theory consisting of rational variables, rational constants, $+, -, =, <, \wedge, \vee, \neg, \forall$ and $\exists$) which runs in $r\alpha^d n^{\beta d(b+1)^a}$ time and it uses some techniques in combinatorial geometry, where α and β are proper constants, and r, n, d, a and b denote the maximum bit length of coefficients, the number of inequalities, the number of variable, the number of quantifier alternations and the maximum length of the consecutive same quantifiers in the input sentence, respectively. The known fastest decision algorithm was Ferrante and Rackoff's algorithm which runs in $r\gamma^d n^{\delta d(2b+1)^a}$ time, where γ and δ are proper constants.

1 Preface

Decision algorithms for prenex normal form sentences of the theory of rationals with addition, which we call PRP sentences, (the theory consisting of rational variables, rational constants, $+, -, =, <, \wedge, \vee, \neg, \forall, \exists$) are used in the tests of protocols, timing verification of hardware, and so on[1, 2]. There has been a lot of work on the problem of finding precise time and space complexity of this algorithm[3, 4, 5, 6, 7]. Ferrante and Rackoff have proposed a decision algorithm for PRP sentences which runs in $r\gamma^d n^{\delta d(2b+1)^a}$ time, where γ and δ are constants, and r, n, d, a, b denote the maximum bit length of coefficients, the number of inequalities, the number of variables, the number of quantifier alternations and the maximum length of the consecutive same quantifiers in the input sentence, respectively[8, 9]. This is the fastest decision algorithm until now as long as authors know. In this paper, we propose a faster algorithm which runs in $r\alpha^d n^{\beta d(b+1)^a}$ time (α and β are proper constants) and it uses some techniques in combinatorial geometry. The dominant part in the time complexity of Ferrante and Rackoff's is the term with double exponential, where our algorithm has an advantage.

The overview of our algorithm is as follows. From an input PRP sentence, we construct an arrangement (a set of all subspaces obtained by dividing a

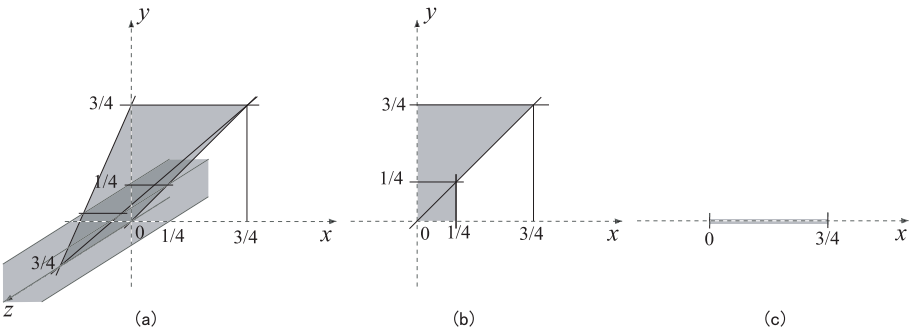

Fig.1 Transformation of an arrangement in execution of the algorithm.

whole d-dimensional space with hyperplanes corresponding to all inequalities in the sentence, where d is the number of variables in the sentence). In each of subspaces, any point has the same truth value when we evaluate the point in the matrix[1] of the input PRP sentence. Therefore, we assign each truth value of an adequate point in the matrix to the corresponding subspace. Next, we make a projection of those subspaces into $(d-s)$-dimensional subspace where s is the number of the innermost series of the same quantifiers. The value of the input PRP sentence is decided by the 0-dimensional arrangement obtained at last.

2 Algorithm

2.1 Outline

We give a brief look of a decision process for PRP sentence $F = \forall x \exists y \exists z \{x \geq 0 \wedge y \geq 0 \wedge [(y - x \geq 0 \wedge y + z \leq 3/4 \wedge z \geq 0) \vee (x \leq 1/4 \wedge y \leq 1/4)]\}$, giving some definitions relating to combinatorial geometry.

First, we divide a whole d-dimensional space into subspaces to make an arrangement from F where d denotes the number of variables in a given sentence. For F, we use a 3-dimensional space, because F has three variables. Here, we briefly introduce the notion of the arrangement. A plane divides a space into three subspaces, *i.e.* a subspace above the plane, a subspace below the plane and the plane itself. In general, a set of planes divides a space into points, segments without their ends, polygons without their periphery, and polyhedra without their surfaces. Each of these subspaces is called a face. Also, a point, a segment without ends, a polygon without periphery, and a polyhedron without surfaces are called a 0-face, a 1-face, a 2-face and a 3-face, respectively. A set of subspaces made by dividing a space with a set of planes is called an arrangement. A point, a line, a plane and a whole 3-dimensional space is called 0-flat, 1-flat, 2-flat and 3-flat, respectively.

[1] A matrix of PRP sentence is a logical expression which is obtained by taking all quantifiers off from the sentence.

For F, we make an arrangement using seven planes corresponding to inequalities in E. Some parts of the arrangement and the domain in which E is true is shown in Fig.1 (a), where E is an expression $x \geq 0 \wedge y \geq 0 \wedge [(y - x \geq 0 \wedge y + z \leq 3/4 \wedge z \geq 0) \vee (x \leq 1/4 \wedge y \leq 1/4)]$, the matrix of F. This domain is a subset of the arrangement made by dividing the space with planes corresponding to inequalities in E.

We will extend these definitions mentioned above to general dimension ones. These definitions follows the literature [10].

DEFINITION 1 A flat perpendicular to the axis (parallel to an axis)

In a Cartesian coordinate system $\{v_1, ..., v_d\}$, if a flat fl contains a line perpendicular to an axis v_i, fl is perpendicular to an axis v_i. If a flat fl contains a line parallel to an axis v_i, fl is parallel to the axis v_i. □

DEFINITION 2 k-face

Without loss of generality, we introduce a hyperplane h which is not perpenticular to the axis v_d. As with an optional point $x = (x_1, x_2, ..., x_d)$ on h, there exists a unique set of rational numbers which meet $x_d = \eta_d + \sum_{i=1}^{d-1} \eta_i x_i$. For a point $p = (\pi_1, \cdots, \pi_d)$, we call that p is above, on and below h if $\eta_d + \sum_{i=1}^{d-1} \eta_i \pi_i$ is greater than π_d, equal to π_d, and less than π_d, respectively. Notations h^+ and h^- represent a set of points above h and that below h, respectively. Now, we assume no element of a given set of planes $H = \{h_1, ..., h_n\}$ is perpendicular to any axis v_i. For a given hyperplane h and a point p, we define $v_i(p)$ as follows.

$$v_i(p) = \begin{cases} +1 \ (p \in h_i^+) \\ 0 \ (p \in h_i) \\ -1 \ (p \in h_i^-) \end{cases}$$

A face is a set of points such that the values $v_1(p), ..., v_n(p)$ are all the same for any point p in the set. A face which can be contained in a k-flat and cannot be contained in a $(k-1)$-flat is called a k-face. Especially a 0-face is also called a vertex. □

DEFINITION 3 Arrangement

A finite set H of hyperplanes in a d-dimensional space divides the space into faces of various dimensions. We call this set of faces an arrangement which was made by dividing a whole d-dimensional Euclidian space. It is also simply called the arrangement of H. The elements of the arrangement A of H are called faces contained in A. If a k-flat fl contains k-face f which is contained in A, fl is called a flat contained in A. □

Next, we assign a boolean value u to each face f contained in the arrangement A obtained from E, where u is the value obtained by substituting the coordinate of some point in f to E. By this process, the value obtained by substituting the coordinate of any point p in the space to E should be equal to the value assigned to the face which contains p. The faces painted in grey in Fig. 1(a) are the faces to which true is assigned.

Then, we eliminate the quantifier $\exists z$. This is a process to make a projection of grey area in Fig.1(a) onto a (x, y)-plane. By this process, we obtain Fig 1(b).

To make this projection, we extract all 1-flats (lines) from Fig 1(a), then make their projections, then make a 2-dimensional arrangement from these projections, and finally assign true values to faces of the arrangement iff the faces are contained in the shadow of grey-painted domain of Fig 1(a).

DEFINITION 4 A projection of a point

A projection of a point $p = (V_1, V_2, ..., V_d)$ to $(v_1, v_2, ..., v_{d-s})$-space is $(V_1, V_2, ..., V_{d-s})$. □

DEFINITION 5 A projection of a face

A projection of a face f to $(v_1, v_2, ..., v_{d-s})$-space is a set of all projections of points contained in f. □

DEFINITION 6 A projection of a flat

A projection of a flat fl to $(v_1, v_2, ..., v_{d-s})$-space is a set of all projections of points contained in fl. □

DEFINITION 7 A projection of an arrangement

A projection of an arrangement B to $(v_1, v_2, ..., v_{d-s})$-space is an arrangement of a set of all projections of flats such that they are contained in B and their projections are $(d-1-s)$-flats. □

Generally, for any face f contained in an arrangement A, there exists a set $\mathcal{F}$ corresponding to f such that $\mathcal{F}$ satisfies the following two conditions : (1)$\mathcal{F}$ is a subset of $\mathbf{pr}A$ (projection of A) and (2) a set of points contained in the projection of f is the same as that contained in all elements of $\mathcal{F}$(refer LEMMA 1 in section 3.3).

DEFINITION 8 A projection of an assigned arrangement *w.r.t.* quantifier Q

Let A' be a projection of an arrangement A (*cf.* DEFINITION 7). A projection of an assigned arrangement *w.r.t.* quantifier Q to $(v_1, v_2, ..., v_{d-s})$-space is an arrangement A' which is assigned boolean values as follows :

If $Q =$ '$\exists$' : True is assigned to every f' contained in A' iff there exists a face f such that true is assigned to f and the projection of f contains f'.

If $Q =$ '$\forall$' : False is assigned to every f' contained in A' iff there exists a face f such that false is assigned to f and the projection of f contains f'. □

Because quantifiers of the input formula are $\forall x \exists y \exists z$, we eliminate $\exists y$ in the same way, and obtain a 1-dimensional assigned arrangement in the next step. This arrangement is represented in Fig.1(c). In fact, our algorithm eliminates a series of the same quantifiers simultaneously. In other words, we obtain Fig.1(c) from Fig.1(a) directly.

As mentioned above, existential quantifier elimination is to obtain a shadow of a true domain by casting ray parallel to the eliminating axis. On the other hand, eliminating a universal quantifier is to obtain a domain which is not in a shadow of false domain by casting ray parallel to the eliminating axis. Finally we eliminate the remaining $\forall x$ and obtain a 0-dimensional assigned arrangement. This arrangement is composed of a point assigned false. Therefore, the input formula is decided to be false.

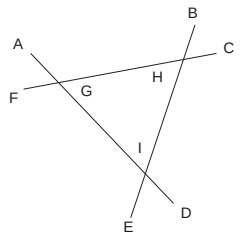

Fig.2 An example of an Arrangement.

2.2 Details of the algorithm

First, we describe the subroutines ARRANGE, ASSIGN, PROJECT, and then describe the main routine MAIN. ARRANGE converts a set of hyperplanes into the data structure of its arrangement (we call this operation "making an arrangement"). ARRANGE is used in MAIN and PROJECT. ASSIGN assigns truth values of PRP expression to an arrangement. PROJECT makes a projection of an assinged arrangement.

ARRANGE We use a set of coordinates of d affine independent points [2] (Let $P = \{\boldsymbol{p_0}, \cdots, \boldsymbol{p_k}\}$ *s.t.* k is finite. If $x = \Sigma_{i=0}^{k} \lambda_i \boldsymbol{p_i}$ and $\Sigma_{i=0}^{k} \lambda_i = 1$ hold, x is called an affine combination of P. If there is no $p_i \in P$ *s.t.* p_i is an affine combination of $P - \{p_i\}$, we call that P is affine independent).

Fig.3 shows the data structure of the arrangement in Fig.2. Here, each rectangle represents a face. Each face f has a set of edges connected to all superfaces of f, and a set of edges to all subfaces. A face also has additional information which contains the following information:

- The dimension of the face.
- The truth value assigned to the face.
- The coordinate of the point, if the face is 0-face.

If a face is unbounded, we use enough large values as its coordinate(*e.g.* We use the coordinate of a point which is on a ray AG and sufficiently far from a point G).

Algorithm ARRANGE

◇ INPUT: A set H of hyperplanes.

◇ OUTPUT: An arrangement of H.

See [10] for its algorithm.

ASSIGN The subroutine ASSIGN assignes the truth value of the input PRP expression to the corresponding face in the input arrangement. The function

[2] According to circumstances, we use more than d coordinates.

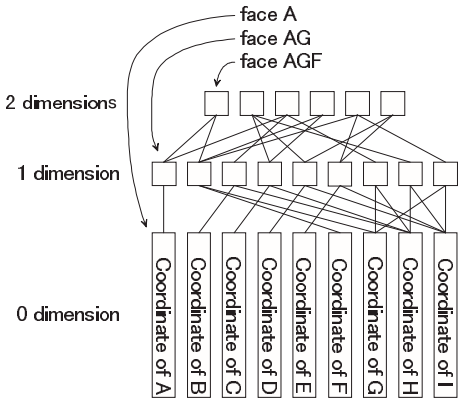

Fig.3 The Data structure of Fig. 2.

INNER returns a coordinate in the input face, such as a center of gravity of the face.

Algorithm ASSIGN

◇ INPUT: The matrix E of the input sentence, an arrangement A of a set of hyperplanes containing a set of hyperplanes corresponding to all inequalities in E.
◇ OUTPUT: An assigned arrangement which is assigned truth value of E.

▷ *Assign the truth value of E to faces contained in A.*

1 **for** each $f \in A$
2 Assign the truth value of E(INNER(f)) to f;
3 **next**
4 **return** A;
5 **end**

PROJECT The subroutine PROJECT is a subroutine which makes a projection of an arrangement. The function EXT used in this algorithm is a function making a flat which contains a face given as its argument and whose dimension is the same as the face.

Algorithm PROJECT

◇ INPUT: An integer value s, an quantifier q, an assigned arrangement A which meets the following condition: for all axes, there exists a hyperplane hp which is contained in A and perpendicular to the axis.
◇ OUTPUT: A projection of A.

1 $S := \{\text{EXT}(f) | f \in A, f \text{is}(d-1-s)\text{-face}\}$;
2 $A' := \text{ARRANGE}(S)$;
3 **if** $q =$ '$\exists$' **then** $u :=$ true **else** $u :=$ false;

▷ *Assign $\neg u$ to all $f' \in A'$.*

4 **for** each $f' \in A'$ **do** assign $\neg u$ to f' **next**

▷ *Assign truth value to faces contained in A'.*

5 **for** $f \in A$ *s.t.* u is assigned to f
6 **for** each $f' \in A'$
7 Let $\boldsymbol{x_1}, ..., \boldsymbol{x_k}$ be projections of vertices of f;

▷ *Check whether an optional point INNER(f') in f' is contained in sh f.*

8 **if** (INNER(f') $= \sum_{m=1}^{k} a_m \boldsymbol{x_m} \wedge 0 < a_m \wedge$ There exists $a_1, ..., a_k$ which satisfy $a_m < 1 \wedge \sum_{m=1}^{k} a_m = 1$)

9 **then** assign u to f'; **endif**

10 **next**

11 **next**

12 **return** A';

13 **end**

MAIN MAIN is the main routine of our algorithm and it decides the truth value of the input PRP sentence. The function AFFINE used in MAIN returns a set of points such that they are on a hyperplane h corresponding to the inequality in and the set is composed of enough many points so that they can represent h.

Algorithm MAIN

◇ INPUT: A PRP sentence F.

◇ OUTPUT: The truth value of F.

1 Let $Q_1, ..., Q_d$ be quantifiers of input PRP sentence. Let $v_1, ..., v_d$ be variables quantified by each quantifier.

2 $S :=$ (the set of all inequalities contained in F) $\bigcup$ $(\bigcup_{i=1}^{d} \{v_i = 0\})$;

3 $H := \emptyset$;

4 **for** each $h \in S$

5 $t :=$ AFFINE(h);

6 $H := H \bigcup \{t\}$;

7 **next**

8 $A :=$ ARRANGE(H);

9 $A :=$ ASSIGN(A,matrix of F);

▷ *i is the number of remaining quantifiers.*

10 $i := d$;

11 **while** $i \geq 1$ **do**

12

13 **if** $Q_i =$ '$\forall$' **then** $q :=$ '$\exists$' **else** $q :=$ '$\forall$'

14

▷ *Assign the number of the innermost sequance of same Q_i to s.*

15 **if** $q \in \{Q_1, ..., Q_i\}$ **then**

16 determine s, *s.t.* $Q_{i-s} = q \wedge$ for each $i - s + 1 \leq k \leq i$, $Q_k \neq q$;

17 **else**

18 $s := i$;

19 **endif**

20

21 $A :=$ PROJECT(A,s,Q_i);

22

23 $i := i - s$;

24 **endwhile**

25

▷ *A is composed of one point.*

26 **return** Truth value of the point of A;

27 **end**

3 Proof of correctness and analysis of time complexity of each routine

In this section, we describe each specification of every routine, and give the proofs of their correctness and analysis of time complexity of each routine.

Hereafter, we will use these notations.

n : the number of inequalities occurring in the input sentence
d : the number of variables occurring in the input sentence
l : the length of the input sentence
a : the number of quantifier alternation of the input sentence
b : the maximum length of the consecutive same quantifiers

3.1 ARRANGE

Specification:
INPUT: A set H of hyperplanes
OUTPUT: An arrangement of H

Proof of correctness: refer [10].

Time complexity: Let g be the maximum bit length of the denominator and the numerator of every coordinate points which represents hyperplanes in H. Thus the time complexity is $O(g \cdot \gamma^d |H|^d)$ and the size of output is also the same order, where γ is some constant[10].

Analysis of the time complexity: The number of faces contained in the arrangement is $O(|H|^d)$ (see [10]). The coordinates of the vertices in the output arrangement are solutions of simultaneous linear equations with d unknowns. Therfore, the bit lengths of the coordinates are $g \cdot \gamma^d$. Since the time complexity is a product of the number of faces and the maximum bit length of vertices in the output arrangement, the time complexity is $O(g \cdot \gamma^d |H|^d)$ and the size of the output is also the same order.

3.2 ASSIGN

Specification:
INPUT: A PRP expression E and an arrangement A of the set of hyperplanes which represents a set of hyperplane which is correspond to inequalities in E.
OUTPUT: An arrangement A' which is assigned truth values of E.

Proof of correctoness: It is clear by DEFINITION 3, and definitions of PRP expression and an assigned arrangement. □

Time complexity: The time complexity of ASSIGN is polynomial order of l, and the size of output arrangement is $O(l)$, where l is the size of A.

Analysis of time complexity: The loop from line 1 to line 3 runs in time of polynomial order of l. Therefore, the time complexity of this algorithm is polynomial order of l. The size of the output arrangement is equal to that of the input arrangement.

3.3 PROJECT

Specification:

INPUT: An assigned arrangement A which contains
hyperplanes perpendicular to every axis,
the number s of variables quantified by the innermost
series of same quantifiers, quantifier Q.

OUTPUT: A projection of A to $(d'-s)$-dimensional space
by quantifier Q, where d' is dimension of A.

Proof of correctness: By LEMMA 1 which is described later, the following condition holds: for any face $f \in A$ and $f' \in \mathbf{pr}\ A$, f' is contained in $\mathbf{sh}\ f$, iff any point in f' is contained in $\mathbf{sh}\ f$.

We now describe the correctness of line 8.

Let $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_k\}$ be a set of vectors of coordinates of vertices of a convex polyhedron C. If and only if any vector $\boldsymbol{p}$ of coordinates is contained in C, there exist $a_1, ..., a_k$ such that they satisfy the following expression (1). We can say the same thing if we substitute $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_k, \boldsymbol{y}_1, ..., \boldsymbol{y}_l\}$ for $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_k\}$($\boldsymbol{y}_1, ..., \boldsymbol{y}_m$ are all contained in C) (see LEMMA 4 in appendix). Therefore, the correctness of line 8 is proved.

$$\boldsymbol{p} = \sum_{m=1}^{k} a_m \boldsymbol{x_m} \wedge 0 < a_m < 1 \wedge \sum_{m=1}^{k} a_m = 1 \tag{1}$$

□

Time complexity: Let m and r be the number of faces in A and the maximum bit length of coordinates in A, respectively. Let α, η and θ be constants. The data size of the output A' is bounded by $r\alpha^{b+1}m^{b+1}$ and the total time complexity is bounded by $r\alpha^{\eta(b+1)}m^{\theta(b+1)}$.

Analysis of time complexity: Let A be the arrangement of H. Then the number of faces in A is $O(|H|^d)$. Let m and r be the number of faces in A and the maximum bit length of coordinates in A, respectively. The number of $(d-s-1)$-flats contained in A is less than or equal to $r\alpha^{s+1}m^{s+1}$. Since $s \leq b$ holds, the number of $(d-s-1)$-flats contained in A is less than $r\alpha^{\eta(b+1)}m^{\theta(b+1)}$. The projection $\mathbf{pr}\ A$ is an arrangement of a set of hyperplanes whose number is the same as the number of hyperplanes contained in A. Thus, $\mathbf{pr}\ A$ contains $O(|H|^d)^{b+1}$ faces. Thus, the number of faces in the output arrangement is less than m^{b+1}.

Let r be the maximum bit length of coordinates in A. The maximum bit length of coordinates in $\mathbf{pr}\ A$ is less than $r\alpha^{b+1}$.

We now describe the time complexity of line 8. Since the number of vertices is less than m^{b+1}, the number of inequalities is less than $d + 2m^{b+1} + 1$ and the number of variables is less than m^{b+1}. Since this is a linear programing, the time complexity of this decision is $O((m^{b+1})(d + 2m^{b+1} + 1)^3 r)$[11].

The time complexity between line 5 and line 11 is as follows: since the number of faces in A is m and that of faces in A' is m^{b+1}, it is polynomial order of m^{b+2}.

Therefore, this subroutine runs in $r\alpha^{\eta(b+1)}m^{\theta(b+1)}$ time.

LEMMA 1 Let S be a set of faces in **pr** A. If there exist hyperplanes such that one of the hyperplanes is not parallel to each of all axes [3], there exists a subset S' in S for all face f in A such that set of all points in S' is equal to the projection of f to $(v_1, ..., v_{d-s})$ space.

Proof: See Appendix.

3.4 MAIN

Specification:

INPUT: A PRP sentence F.

OUTPUT: The truth value of F.

Proof of correctness: The proof of the correctness of MAIN is achieved if all four propositions below are proved.

DEFINITION 9 A function $\mathcal{VAL}$

The arguments of a function $\mathcal{VAL}$ are an assigned arrangement A and a coordinate p. $\mathcal{VAL}$ returns the truth value assigned to the face which contains p. □

DEFINITION 10 A function $\mathcal{FML}$

Let A be an assigned arrangement and $Q_1, ..., Q_i$ be quantifiers. We introduce a new function $\mathcal{FML}$ as follows :

$$\mathcal{FML}(A, (Q_1, ..., Q_i)) = Q_1 v_1, ..., Q_i v_i \mathcal{VAL}(A, (v_1, ..., v_i))$$

□

PROPOSITION 1 When the control reaches line 12 of MAIN for the first time, the truth value of $\mathcal{FML}(A, (Q_1, ..., Q_i))$ is equal to the truth value of F and A contains hyperplanes perpendicular to each of all axes.

PROPOSITION 2 On line 12 of MAIN, the truth value of $\mathcal{FML}(A, (Q_1, ..., Q_i))$ is equal to the truth value of F and A contains hyperplanes perpendicular to each of that axes.

PROPOSITION 3 When the control reaches line 25 of MAIN for the first time, the following three conditions hold: (1)A is a 0-dimensional arrangement, (2)$i = 0$ and (3)t is the truth value of the input PRP sentence.

PROPOSITION 4 The control goes out of the while loop at some time.

Proof of PROPOSITION 1 : It is clear, because in each face the truth value of E doesn't change, and A containes hyperplane perpendicular to all axes line 2. □

Proof of PROPOSITION 2 Let a_1 and a_2 be the value of $\mathcal{FML}(A, (Q_1, ..., Q_i))$ on line 12 and line 22, respectively. From the correctness of subroutines PROJECT and LEMMA 2, we can say $a_1 = a_2$.

[3] Here we describe the reason why this condition is needed. There is an arrangement of plane $x = 0$ in a 3-dimensional space. We want to make projection of this arrangement to (x, y)- plane which contains the line $x = 0$. We can make it if we add a plane $z = 0$ which is not parallel to z axis to the 3-dimensional arrangement.

From LEMMA 1, a projection of an arrangement which contains hyperplanes perpendicular to all axes contains hyperplanes perpendicular to all axes. Thus, the correctness of PROPOSITION 2 is proved.

□

LEMMA 2 Let A be an arrangemet which contains hyperplanes perpendicular to all axes, and **pr** A projection of A to $(v_1, ..., v_{i-s})$-space by quantifier Q. For all $V_1, ..., V_{i-s}$, we can say :

$$\begin{aligned}&\mathcal{VAL}(\mathbf{pr}\ A, (V_1, ..., V_{i-s}))\\ &= Q\ v_{i-s+1}, ..., Q\ v_d \mathcal{VAL}(A, (V_1, ..., V_{i-s}, v_{i-s+1}, ..., v_d))\end{aligned}$$

Proof Let f be a face contained in A. Let $\mathcal{F}$ be a set of all faces in **pr** A such that **sh** f contains them. By LEMMA 1, the set of points contained in elements of $\mathcal{F}$ is equal to **sh** f. We describe the case $Q =$ '$\exists$'. By the algorithm, if true is assigned to f, true is assigned to all faces in $\mathcal{F}$. Thus, for each point p in the face f to which true is assigned, **sh** p is contained in a face to which true is assigned. If false is assigned to f, the assigned values of faces in $\mathcal{F}$ can be either true or false. If false is assigned to a face f' in $\mathcal{F}$, false is assigned to every face f'' such that $f' \subseteq$ **sh** f''. Therefore, if **sh** p' is contained in a face to which false is assigned, all points p'' such that **sh** $p' =$ **sh** p'' are contained in face to which false is assigned. We can prove the case $Q =$'$\forall$' in the similar way. Thus, the correctness of this lemma is proved. □

Proof of PROPOSITION 3 It is clear. □

Proof of PROPOSITION 4 On line 22, s is more than zero. Thus, i decreases in each loop. Therefore, the correctness of this lemma is proved. □

Time complexity: The time complexity of the MAIN routine, which is the time complexity of whole algorithm, is $r \cdot \alpha^{\beta d} n^{\gamma d (b+1)^a}$, where α, β and γ are proper constants.

Analysis of time complexity The time complexity of line 5 is polynomial order of l. The time complexity is $O(r \cdot \gamma^d n^d)$ and the size of A is the same order. On line 9, the size of the returned value A from ASSIGN is $O(l \cdot \gamma^d n^d)$. Here we describe the time complexity of the loop between line 11 and line 24. On line 12, let m be the number of faces which is contained in A and r' be the maximum bit length of coordinates of A. The time complexity of line 21 is $O(mr)$, where α, η and θ are constants. The size of A is $r'\alpha^{b+1} m^{b+1}$. Let m_s be the number of faces in A before the control reaches the loop and r_s the maximum bit length of coordinates of A before the control reaches the loop. Since the number of iteration is a, the time complexity of line 21 of a-th loop is $r_s \alpha^{\eta a b} m_s^{\gamma d (b+1)^a}$. Since $d = O(ab)$ holds, the total time complexity is $r\alpha^d n^{\beta d (b+1)^a}$, where α and β are constants. □

4 A brief explanation of Ferrante and Rackoff's algorithm

In this section, we briefly explain the time complexity of algorithm of Ferrante and Rackoff.

Let F be the PRP sentence to be decided, and E be the matrix of F. Inequalities contained in E can be represented in the form v_d **op** t_i, where **op** is $<, =$, or $>$, and t_i is of the form $t_i = \sum_{j=1}^{d-1} c_j v_j$, where c_i's are rationals.

The following lemma holds.

LEMMA 3

$$\exists v_m E(v_1, ..., v_m) = \bigvee_{\substack{v_m = t_i \text{ or} \\ v_m = (1/2)(t_i + t_j) \\ \text{or } v_m = +\infty \text{ or } v_m = -\infty}} E(v_1, ..., v_m)$$

$$\forall v_m E(v_1, ..., v_m) = \bigwedge_{\substack{v_m = t_i \text{ or} \\ v_m = (1/2)(t_i + t_j) \\ \text{or } v_m = +\infty \text{ or } v_m = -\infty}} E(v_1, ..., v_m)$$

□

The principle of Ferrante and Rackoff's algorithm is to eliminate all quantifiers in the input sentence in turn using LEMMA 3 and then to decide truth from the obtained expression. The righthand side of both equation in LEMMA 3 has $(n^2 + 2) = O(n^2)$ subexpressions where n is the number of inequalities in $E(v_1, ..., v_m)$. Thus, If one quantifier is eliminated using LEMMA 3, the number of inequation increases to the third power. If the same type of this quantifier is succeeded, the second quantifier can be distributed to each term. Thus eliminating two successing quantifiers makes the number of inequation to the fifth power.

Generally, we can obtain $n^{d(2b+1)^a}$ as the upperbound of the number of inequalities contained in the expession obtained by eliminating all quantifiers by the method above. The maximum bit length of coefficients increases by five times whenever one quantifier is eliminated. Therefore, the upperbound of the maximum bit length of coefficients in the expession is $r5^d$. The time complexity of their algorithm is polynomial of $r5^d \cdot n^{d(2b+1)^a}$. Thus, the time complexity of their algorithm is $r\gamma^d n^{\delta d(2b+1)^a}$.

5 Conclusion

In this paper, we proposed a decision algorithm for PRP sentences which runs in $r\alpha^d n^{\beta d(b+1)^a}$ and it uses some techniques in combinatorial geometry, where α and β are proper constants, and r, n, d, a and b denote the maximum bit length of coefficients, the number of inequalities, the number of variable, the number of quantifier alternations and the maximum length of the consecutive same quantifiers in the input sentence, respectively.

In the future, we plan to improve the average time complexity and make it applicable to larger and practical problems.

References

[1] T. Higashino, J. Kitamichi and K. Taniguchi : "Presburger Arithmetic and its Application to Program Developments," Journal of Japan Society for Software Science and Technology, Vol.9,**6**, pp.31-39, 1992. (In Japanese).
[2] A. Nakata, T. Higashino and K. Taniguchi : "Time-Action Alternating Model for Timed LOTOS and its Symbolic Verification of Bisimulation Equivalence", Proceedings of Joint International Conference on 9th Formal Description Techniques and 16th Protocol Specification, Testing, and Verification (FORTE/PSTV'96), pp.279-294, 1996.
[3] A.R.Bruss and A.Meyer : "On time-space classes and their relation to the theory of real addition," Thoret. Comput. Sci. **11** pp.59-69, 1980.
[4] L. Berman : "The complexity of logical theories," Theoret. Comput. Sci. **11** pp.71-77, 1980.
[5] V. Weispfenning : "The complexity of linear problems in fields," J. Symbolic Computation **5** pp.3-27, 1988.
[6] C.Hosono and Y.Ikeda : "A formal derivation of the decidability of the theory SA," Thoret. Comput. Sci. **127** pp.1-23, 1994.
[7] M.J.Fischer and M.O.Rabin : "Super exponential complexity of Presburger Arithmetic," SIAM-AMS Proc. **VII**(AMS,Providence,RI), 1974.
[8] J.Ferrante and C.Rackoff : "A decision procedure for the first order theory of real addition with order," SIAM J. Comput. **4**, pp.69-76, 1975.
[9] J.E. Hopcroft and J.D. Ullmann : "Introduction to automata theory, languages and computation," Addison-Wesley, pp.355-357, 1979.
[10] H. Edelsbrunner : "Algorithms in Combinatorial Geometry," Springer-Verlag, 1987.
[11] L.G. Khachiyan : "Polynomial algorithms for linear programming," Dokl. Akad. Nauk SSSR **244**, pp.1093-1096, 1979.

Appendix

In this paper, we assume the properties below without proofs. Here, we also give LEMMA 4 and the proof of LEMMA 1.

PROPERTY 1

In a d-dimensional space, if and only if a crossing of $m(\leq d)$ hyperplanes is a $(d-m)$-flat, the normal vectors of these hyperplanes are linearly independent. □

PROPERTY 2

Let fl be a k-flat in $(v_1, ..., v_d)$ space. Let S be a set of axes which are parallel to fl. If a projection of fl to $(v_1, ..., v_{d-s})$ is k'-flat, $k - k' = |S|$ holds. □

PROPERTY 3

In an arrangement A, for every $k(\leq d-1)$-flat fl in A, there exists $d-k$ hyperplanes whose normal vectors are linearly independent, and fl is contained in all of them. □

PROPERTY 4

If the same dimensional flats fl and fl' meet $fl \subseteq fl'$, then $fl = fl'$ holds. □

PROPERTY 5

Let fl be a flat in $(v_1, ..., v_d)$ space. Let S be a set of axes such that each of which is in $v_{d-s+1}, ..., v_d$ and parallel to fl. Let fl' be a crossing of fl and a hyperplane hp such that hp is perpendicular to an optional axis v_i in S. fl' is equal to a projection of fl to $(v_1, ..., v_{d-s})$ space. □

PROPERTY 6

Let fl a $k(< d)$-flat in A. If and only if fl is parallel to an axis j, all hyperplanes which contain fl and contained in A are parallel to the axis j. □

PROPERTY 7

Let fl_1 and fl_2 be the same dimensional flat in $(v_1, ..., v_d)$ space. Let **sh** fl_1 and **sh** fl_2 be projections of fl_1 and fl_2 to $(v_1, ..., v_{d-s})$ space, respectively. If $fl_1 \neq fl_2$ and **sh** fl_1 = **sh** fl_2 hold, a flat which contains both fl_1 and fl_2 is parallel to one or more axes in v_{d-s+1}axis, ..., $v_d$axis. □

PROPERTY 8

If and only if an arrangement A is equal to an arrangement B, a set of hyperplanes contained in A is equal to that of B. □

LEMMA 4 Let S be a set of hyperplanes which meets $|S| \leq d$. The following statements (A) and (B) are equivalent.

(A) Normal vectors of hyperplanes in S are linearly independent.

(B) Let S_1 and S_2 be sets of hyperplanes such that $S_1 \subseteq S, S_2 \subseteq S$ and $S_1 \neq S_2$ hold. The crossing of all hyperplanes in S_1 is not equivalent to the crossing of all hyperplanes in S_2.

Proof

(A) ⇒ (B): If the number of hyperplanes in S_1 and S_2 differs, the dimensions of their crossings are different and it contradicts (B). Thus, we only think the case that the number of hyperplanes in S_1 and that of S_2 are the same. We use contradiction. We assume that the crossings are the same. Let fl be the crossing. The crossing of hyperplanes in $S_1 \bigcup S_2$ is also the same as fl. From

the fact $S_1 \neq S_2$, S_1 contains a hyperplane which is not contained in S_2. Thus, $S_1 \bigcup S_2$ contains more hyperplanes than S_1, and the crossings of hyperplanes in $S_1 \bigcup S_2$ should be flat in less dimension than that of S_1. This contradicts the assumption.

(A) $\Leftarrow$ (B): We prove (B) $\wedge\neg$ (A) doesn't hold. Let S be $\{h_1, h_2, ..., h_{|S|}\}$. We substitute h_1 for S_1 and $h_1 \bigcap h_2$ for S_2, and $h_1 \neq h_1 \bigcap h_2$ holds. Otherwise, let fl_i be $h_1 \bigcap ... \bigcap h_i$ and we can say $fl_i \neq fl_{i+1}$ for $1 \leq i \leq |S| - 1$. From PROPERTY 1, the dimension of fl_i is not equal to that of fl_{i+1}. From the definition of fl_i and the assumption $fl_{i+1} \neq fl_i$, fl_{i+1} is a subset of fl_i. Thus, the dimension of fl_{i+1} is less than that of fl_i. Therefore, fl_k is a $(d-k)$-flat where $1 \leq k \leq |S|$. From PROPERTY 1, the normal vectors of hyperplanes in S should be linearly independent and that contradicts $\neg$ (A). □

Proof of LEMMA 1

The proof consists of five steps, (S1) to (S5).

(S1) Let fl be an optional flat contained in A. Let $i(\leq d-1-s)$ be the dimension of sh fl. We prove that there exists an i-flat c such that c is contained in A and meets sh fl = sh c.

Let S be a set of axes which are parallel to fl and S is a subset of $\{v_{d-s+1}\text{axis}, ..., v_d\text{axis}\}$. From PROPERTY 2, fl is an $i + |S|$-flat. Because A contains hyperplanes each of which is perpendicular to every axis, fl is crossing to each hyperplane $j = 0$ for every element j of S. Let P be a set of such hyperplanes. From PROPERTY 6, any $(d-1)$-flat α such that α is contained in fl and contained in A is parallel to each element in S. Thus, the normal vectors of $(d-1)$-flat contained in P or α are linearly independent. There exists a crossing c of all hyperplanes in P and fl. By PROPERTY 1, c is an i-flat. Let α' be a set of $(d-1)$-flats which are contained in A and contain c. $\alpha' = \alpha \bigcup P$ holds. There is no axis which is parallel to c in $v_{d-s+1}\text{axis}, ..., v_d\text{axis}$. By PROPERTY 5, **sh** fl is equal to **sh** c.

(S2) We prove that there exist s $(d-1)$-flats such that they are contained in A and contain c and their crossing is not parallel to any of v_{d-s+1}axis, ..., $v_d$axis.

Because c is not parallel to any of $v_{d-s+1}\text{axis}, ..., v_d\text{axis}$, we can say the following statements from PROPERTY 6. There exists a hyperplane hp for each axis in $\{v_{d-s+1}\text{axis}, ..., v_d\text{axis}\}$ such that hp is contained in A and hp contains c and hp is not parallel to j. Let β' be a set of such hyperplanes. Because the number of $v_{d-s+1}\text{axis}, ..., v_d\text{axis}$ is s, the number of elements of β' is less than $s+1$. Because the flat c contains i-flats and PROPERTY 3 holds, there are $(d-i)$ $(d-1)$-flats which contain c. By assumption, $d - i > s$ holds. Therefore, there exists a combination of s $(d-1)$-flats which contain c and any of their crossing c' is not parallel to any of $v_{d-s+1}\text{axis}, ..., v_d\text{axis}$. Let β be a set of these s $(d-1)$-flats. c' is the flat we were trying to prove existence.

(S3) We prove that there exists a set of $(d-1-s)$-flats such that every $(d-1-s)$-flat contains c and each $(d-1-s)$-flat is contained in A and the set is equal to the set of $(d-i-s)$ $(d-1-s)$-flats such that

their normal vectors are linearly independent and these are contained in pr A

We can find $(d-i-s)$ $(d-1-s)$-flats by the following way. By definition, $\beta \subset \alpha'$ holds. Let γ be a set $\alpha' - \beta$. Because α' and β are sets of $d-i$ and s hyperplanes, respectively. γ contains $(d-i-s)$ hyperplanes.

Every crossing of each hyperplane in γ and all hyperplanes in β is a $(d-1-s)$-flat which is not parallel to any of v_{d-s+1}axis, ..., $v_d$axis. Let δ be a set of these $(d-1-s)$-flat. Because the crossing of all hyperplanes in β is a $(d-s)$-flat and the normal vectors of all hyperplanes in γ and β are linearly independent, δ contains $d-i-s$ flats by PROPERTY 2. Because the normal vectors of hyperplanes in α' are linearly independent, the elements in δ are all different. We can say the normal vectors of hyperplanes in δ are linearly independent as follows: because the normal vectors of hyperplanes in α' are linearly independent, crossings of all two combinations of flats in δ are different. All projections of crossings of flats in δ are also different. The reasons are the following. If and only if we let fl_1 and fl_2 be crossings of some flats in δ and assume **sh** fl_1 = **sh** fl_2 holds, fl_1 and fl_2 are the same dimensional flat. From $fl_1 \neq fl_2$ and PROPERTY 7, all hyperplanes in β, which contain both fl_1 and fl_2, should be parallel to one or more of v_{d-s+1}axis, ..., $v_d$axis. This is a contradiction. Therefore, from LEMMA 4, the normal vectors of projections of flats in δ are linearly independent.

(S4) We prove that there exists a flat in pr A which is equal to projection sh fl of each flat in A.

All elements in H' are $(d-1-s)$-flats. Because each element of δ is a $(d-1-s)$-flat contained in A, and all, projections of the elements in δ are $(d-1-s)$-flats, projections of all elements in δ are elements of H'. Because a set of all hyperplanes in β and γ, used in definition of δ, is α', and the crossings of all hyperplanes in α' is c, a crossing of $(d-1-s)$-flats in δ is the i-flat c. Because the normal vectors of projections of flats in δ are linearly independent, a crossing c' of projections of flats in δ is an i-flat. Because **sh** c is contained in projections of flats in δ, **sh** $c \subseteq c'$ holds, as well as PROPERTY 4 holds, $c = c'$ holds. Thus, **sh** c is equal to a crossing of projections of all flats in δ. Therefore, we can say **sh** c is contained in **pr** A. Thus (S4) is proved.

(S5) We prove that there exist sets of faces which are contained in pr A and equal to projections sh f of each face f in A.

Let B be an arrangement which has the minimal number of faces and holds the following condition: there exist sets of faces which are equal to a projection of each face f in A. Note that B uniquely corresponds to A. There exist sets of faces which is equal to a projection of each flat in A. From PROPERTY 8, B = **pr** A holds. Therefore, (S5) is proved. □

The intuitive meaning of some symbols used in proof of LEMMA 1 is described in the following table.

c	: An i-flat which meets **sh** fl = **sh** c
α'	: A set of $(d-1)$-flats which contain c and is contained in A
β	: A set of $(d-1)$-flats which contain c' and is contained in A
δ	: Projections of $(d-1-s)$-flat which is contained in δ and contain **sh** c and contained in **pr** A