Automated Discovering and Proving for Geometric Inequalities*

Lu Yang, Xiaorong Hou, Bican Xia

Chengdu Institute of Computer Applications
Academia Sinica, Chengdu 610041, China
luyang@guangztc.edu.cn, xhou@usa.net, bcxia@263.net

Abstract. Automated discovering and proving for geometric inequalities has been considered a difficult topic in the area of automated reasoning for many years. Some well-known algorithms are complete theoretically but inefficient in practice, which cannot verify non-trivial propositions in batches. In this paper, we present an efficient algorithm to discover and prove a class of inequality-type theorems automatically by combining discriminant sequence for polynomials with Wu's elimination and a partial cylindrical algebraic decomposition. Also this algorithm is applied to the classification of the real physical solutions of geometric constraint problems. Many geometric inequalities have been discovered by our program, DISCOVERER, which implements the algorithm in Maple.

1 Introduction

In the last 20 years, the efficiency of the computer algorithms for automated reasoning in both algebraic and logic approaches has greatly increased. One has reason to believe that computer will play a much more important role in reasoning sciences in the coming century. People will be able to prove theorems class by class instead of one by one. Since Tarski's [20] well-known work, *A Decision Method for Elementary Algebra and Geometry,* published in early 1950's, the algebraic approaches have made remarkable progress in automated theorem proving. Tarski's decision algorithm, which could not be used to verify any non-trivial algebraic or geometric propositions in practice because of its very high computational complexity, has only got theoretical significance. Some substantial progresses were made by Seidenberg, Collins [12] and others afterwards, but it was still far away from mechanically proving non-trivial theorems batch by batch, even class by class. The situation didn't change until Wen-tsün Wu [24, 25] proposed in 1978 a new decision procedure for proving geometry theorems of "equality type", i.e. the hypotheses and conclusions of the statements consist of polynomial equations only. This is a very efficient method for mechanically proving elementary geometry theorems (of equality type). S.C. Chou [8] has successfully implemented Wu's method for 512 examples which include almost all

* This work is supported in part by CAS and CNRS under a cooperative project between CICA and LEIBNIZ, and in part by the National "973" Project of China.

the well-known or historically interesting theorems in elementary geometry, and it was reported that for most of the examples the CPU time spent was only few seconds each, or less than 1 second!

The success of Wu's method has inspired in the world the advances of the algebraic approach [18, 21, 22] to automated theorem proving. In the past 20 years, some efficient provers have been developed based on different principles such as Gröbner Basis [5, 6], Parallel Numerical Method [31], and so on. Especially, J.Z. Zhang and his colleagues gave the algorithms and programs for automatically producing readable proofs of geometry theorems [11, 28]. The achievement has made the studies in automated proving enter a new stage that the proofs created by machines can compare with those by persons, while the decision problem was playing a leading role before. It has also important applications to mathematics mechanization and computer aided instruction(CAI).

Those methods mentioned above are mainly valid to equality-type theorem proving. As for automated inequality discovering and proving, the progress has been slow for many years. Chou, Gao et al. [9, 10] made helpful approaches in this aspect by combining Wu's method with CAD (Cylindrical Algebraic Decomposition) algorithm or others. Recently, a so-called "dimension-decreasing" algorithm has been introduced by L. Yang [26]. Based on this algorithm, a program called "BOTTEMA" was implemented on a PC computer. More than 1000 algebraic and geometric inequalities including hundreds open problems have been verified in this way. The total CPU time spent for proving 120 basic inequalities from Bottema's monograph, "Geometric Inequalities" on a Pentium/200, was 20-odd seconds only. To our knowledge, this is the first practical prover which can verify non-trivial inequality-type propositions in batches.

The reason for automated inequality discovering and proving to be a difficult topic in the area of automated reasoning is that the concerning algorithms depend on real algebra and real geometry, and some well-known algorithms are complete theoretically but inefficient in practice. In [29, 33, 30], Yang, Hou et al. introduced a powerful tool, a complete discrimination system (CDS) of polynomials, for inequality reasoning. By means of CDS, together with Wu's elimination and a partial CAD algorithm, we present an efficient algorithm to discover and prove a class of inequality-type theorems automatically. A program called "DISCOVERER" was implemented in Maple that is able to discover new inequalities automatically, without requiring us to put forward any conjectures beforehand. For example, by means of this program, we have re-discovered 37 inequalities in the first chapter of the famous monograph [19], "Recent Advances in Geometric Inequalities", and found three mistakes there.

2 The Problem

In general, the problem we study here is:

Give the necessary and sufficient condition which the parameters u must satisfy for following system TS to have exactly n distinct real solution(s) (or simply, to have real solution),

$$TS:\begin{cases} f_1(u,x_1)=0,\\ f_2(u,x_1,x_2)=0,\\ \cdots\cdots\\ f_s(u,x_1,x_2,\cdots,x_s)=0,\\ g_1(u,X)\ge 0, g_2(u,X)\ge 0,\cdots,g_t(u,X)\ge 0\end{cases} \tag{1}$$

where

$$u=(u_1,u_2,\cdots,u_d),\ \ X=(x_1,x_2,\cdots,x_s),$$

$$f_i\in Z(u)[x_1,\cdots,x_i],\quad 1\le i\le s,$$

$$g_j\in Z(u)[x_1,\cdots,x_s],\quad 1\le j\le t,$$

$\{f_1,f_2,\cdots,f_s\}$ is a "normal ascending chain" [32, 33] and is "simplicial" [32, 33] w.r.t. each g_j $(1\le j\le t)$ (also see section 3 in the present paper for the definitions of these two concepts). Some inequalities in (1) may be strict inequalities.

Before we go further, we illustrate this class of problems by an example:

Give the necessary and sufficient condition for the existence of a triangle with elements s,r,R, where s,r,R are the half perimeter, in radius and circumradius, respectively.

Let a,b,c be the lengths of sides and, without loss of generality, let $s=1$, we have

$$PS':\begin{cases} p_1=a+b+c-2=0,\\ p_2=r^2-(1-a)(1-b)(1-c)=0,\\ p_3=4\,rR-abc=0,\\ 0<a<1,0<b<1,0<c<1,0<r,0<R.\end{cases}$$

It is easy to see that PS' is equivalent to the following system:

$$TS':\begin{cases} f_1(r,R,a)=a^3-2\,a^2+(r^2+4\,rR+1)a-4\,rR=0,\\ f_2(r,R,a,b)=ab^2+a(a-2)b+4\,rR=0,\\ f_3(r,R,a,b,c)=a+b+c-2=0,\\ 0<a<1,0<b<1,0<c<1,0<r,0<R.\end{cases}$$

where a,b,c are the variables, r,R the parameters. What we need to do is to find the condition satisfied by r,R, under which TS' has real solution(s). Obviously, this is a problem of the class defined above.

By the algorithm presented in this paper, we can easily find the condition is

$$s^4+2\,r^2s^2-4\,R^2s^2-20\,rRs^2+12\,r^3R+48\,r^2R^2+r^4+64\,rR^3\le 0,$$

which is called the "Fundamental Inequality" [19] for triangles.

3 Concepts and Notations

In this section, we review some concepts and notations in [32, 33][29, 7, 17].

Given a polynomial g and a triangular set $\{f_1, f_2, ..., f_s\}$, let

$$r_s := g, \quad r_{s-i} := \text{resultant}(r_{s-i+1}, f_{s-i+1}, x_{s-i+1}), \quad i = 1, 2, ..., s;$$

$$q_s := g, \quad q_{s-i} := \text{prem}(q_{s-i+1}, f_{s-i+1}, x_{s-i+1}), \quad i = 1, 2, ..., s.$$

and $\text{res}(g, f_s, ..., f_1)$, $\text{prem}(g, f_s, ..., f_1)$ denote r_0 and q_0, respectively.

Definition 3.1.[32, 33] (Normal Ascending Chain)
Given a triangular set $\{f_1, f_2, ..., f_s\}$, by I_i $(i = 1, 2, ..., s)$ denote the leading coefficient of f_i in x_i. Triangular set $\{f_1, f_2, ..., f_s\}$ is called a normal ascending chain if

$$I_1 \neq 0, \quad \text{res}(I_i, f_{i-1}, ..., f_1) \neq 0, \quad i = 2, ...s.$$

Definition 3.2.[32, 33] (Simplicial)
A normal ascending chain $\{f_1, f_2, ..., f_s\}$ is simplicial with respect to a polynomial g if either $\text{prem}(g, f_s, ..., f_1) = 0$ or $\text{res}(g, f_s, ..., f_1) \neq 0$.

Definition 3.3.[29] (Discrimination Matrix)
Given a polynomial with general symbolic coefficients,

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n,$$

the following $2n \times 2n$ matrix in terms of the coefficients,

$$\begin{bmatrix}
a_0 & a_1 & a_2 & \cdots & a_n & & & & \\
0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} & & & & \\
 & a_0 & a_1 & \cdots & a_{n-1} & a_n & & & \\
 & 0 & na_0 & \cdots & 2a_{n-2} & a_{n-1} & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & a_0 & a_1 & a_2 & \cdots & a_n & \\
 & & & 0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} &
\end{bmatrix}$$

is called the discrimination matrix of $f(x)$, and denoted by $\text{Discr}\,(f)$. By d_k or $d_k(f)$ denote the determinant of the submatrix of $\text{Discr}\,(f)$, formed by the first k rows and the first k columns for $k = 1, 2, \cdots, 2n$.

Definition 3.4.[29] (Discriminant Sequence)
Let $D_0 = 1, D_k = d_{2k}, k = 1, \cdots, n$. We call the $n+1$-tuple

$$D_0, D_1, D_2, \cdots, D_n$$

the discriminant sequence of $f(x)$, and denote it by $\text{DiscrList}\,(f)$.

The last term D_n is also called the discriminant of f with respect to x, and denoted by $\text{Discrim}(f, x)$. It should be noted that the definition of $\text{Discrim}(f, x)$ here is little different from the others which are D_n/a_0.

Definition 3.5.[29] (Sign List)
We call the list

$$[\text{sign}(A_0), \text{sign}(A_1), \text{sign}(A_2), \cdots, \text{sign}(A_n)]$$

the sign list of a given sequence $A_0, A_1, A_2, \cdots, A_n$, where

$$\text{sign}(x) = \begin{cases} 1, x > 0, \\ 0, x = 0, \\ -1, x < 0. \end{cases}$$

Definition 3.6.[29] (Revised Sign List)
Given a sign list $[s_1, s_2, \cdots, s_n]$, we construct a new list

$$[t_1, t_2, \cdots, t_n]$$

as follows: (which is called the revised sign list)

– If $[s_i, s_{i+1}, \cdots, s_{i+j}]$ is a section of the given list, where

$$s_i \neq 0, s_{i+1} = \cdots = s_{i+j-1} = 0, s_{i+j} \neq 0,$$

then, we replace the subsection

$$[s_{i+1}, \cdots, s_{i+j-1}]$$

by the first $j - 1$ terms of $[-s_i, -s_i, s_i, s_i, -s_i, -s_i, s_i, s_i, \cdots]$, that is, let

$$t_{i+r} = (-1)^{[(r+1)/2]} \cdot s_i, \quad r = 1, 2, \cdots, j - 1.$$

– Otherwise, let $t_k = s_k$, i.e. no changes for other terms.

Example 3.1. The revision of the sign-list

$$[1, -1, 0, 0, 0, 0, 0, 1, 0, 0, -1, -1, 1, 0, 0, 0]$$

is

$$[1, -1, 1, 1, -1, -1, 1, 1, -1, -1, -1, -1, 1, 0, 0, 0].$$

Theorem 3.1.[29] Given a polynomial $f(x)$ with real coefficients,

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n,$$

if the number of the sign changes of the revised sign list of

$$D_0, D_1(f), D_2(f), \cdots, D_n(f)$$

is ν, then the number of the pairs of distinct conjugate imaginary roots of $f(x)$ equals ν. Furthermore, if the number of non-vanishing members of the revised sign list is l, then the number of the distinct real roots of $f(x)$ equals $l-1-2\nu$.

Definition 3.7. (Generalized Discrimination Matrix)
Given two polynomials $g(x)$ and

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n,$$

let

$$r(x) = \mathrm{rem}(f'g, f, x) = b_0x^{n-1} + b_1x^{n-2} + \cdots + b_{n-1}.$$

The following $2n \times 2n$ matrix

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_n & & & & \\ 0 & b_0 & b_1 & \cdots & b_{n-1} & & & & \\ & a_0 & a_1 & \cdots & a_{n-1} & a_n & & & \\ & 0 & b_0 & \cdots & b_{n-2} & b_{n-1} & & & \\ & & & \cdots & \cdots & & & & \\ & & & \cdots & \cdots & & & & \\ & & & a_0 & a_1 & a_2 & \cdots & a_n \\ & & & 0 & b_0 & b_1 & \cdots & b_{n-1} \end{bmatrix}$$

is called the generalized discrimination matrix of $f(x)$ with respect to $g(x)$, and denoted by Discr (f, g).

Definition 3.8.[7, 17] (Generalized Discriminant Sequence)
Given two polynomials $f(x)$ and $g(x)$, let the notations be as above, and $D_0 = 1$. Denote by

$$D_1(f, g), D_2(f, g), \cdots, D_n(f, g)$$

the even order principal minors of Discr (f, g). We call

$$D_0, D_1(f, g), D_2(f, g), \cdots, D_n(f, g)$$

the generalized discriminant sequence of $f(x)$ with respect to $g(x)$, and denote it by GDL(f, g). Clearly, GDL$(f, 1)$ = DiscrList (f).

Theorem 3.2.[7, 17] Given two polynomials $f(x)$ and $g(x)$, if the number of the sign changes of the revised sign list of GDL(f, g) is ν, and the number of non-vanishing members of the revised sign list is l, then

$$l - 1 - 2\nu = c(f, g_+) - c(f, g_-).$$

where

$$c(f, g_+) = \mathrm{card}(x \in R | f(x) = 0, g(x) > 0),$$

$$c(f, g_-) = \mathrm{card}(x \in R | f(x) = 0, g(x) < 0).$$

4 A Theoretical Algorithm

In this section, we give an algorithm which has theoretical significance. The practical algorithm is given in the next section.

Let

$$ps = \{p_i | 1 \leq j \leq n\}$$

be a nonempty, finite set of polynomials. We define

$$\text{mset}(ps) = \{1\} \cup \{p_{i_1} p_{i_2} \cdots p_{i_k} | 1 \leq k \leq n, 1 \leq i_1 < i_2 < \cdots < i_k \leq n\}.$$

For example, if $ps = \{p_1, p_2, p_3\}$, then

$$\text{mset}(ps) = \{1, p_1, p_2, p_3, p_1p_2, p_1p_3, p_2p_3, p_1p_2p_3\}.$$

Given system TS as above, we define

$$\begin{aligned} P_{s+1} &:= \{g_1, g_2, \cdots, g_t\}, \\ P_i &:= \{h(u, x_1, \cdots, x_{i-1}) | h \in \bigcup_{q \in \text{mset}(P_{i+1})} \text{GDL}(f_i, q)\}, \\ &\quad i = s, s-1 \cdots, 2. \\ P_1(g_1, g_2, \cdots, g_t) &:= \{h(u) | h \in \bigcup_{q \in \text{mset}(P_2)} \text{GDL}(f_1, q)\}. \end{aligned}$$

where $\bigcup_{q \in \text{mset}(P_{i+1})} \text{GDL}(f_i, q)$ means the set consisting of all the polynomials in each $\text{GDL}(f_i, q)$. Similarily, we can define $P_1(g_1, \cdots, g_j)\,(1 \leq j \leq t)$.

Then we have

Theorem 4.1. The necessary and sufficient condition for system TS to have a given number of distinct real solution(s) can be expressed in terms of the signs of the polynomials in $P_1(g_1, g_2, \cdots, g_t)$.

Proof. First of all, we regard f_s and every g_i as polynomials in x_s. By theorems 3.1 and 3.2 we know that under constraints $g_i \geq 0, 1 \leq i \leq t$, the number of distinct real solutions of $f_s = 0$ can be determined by the signs of polynomials in P_s; let $h_j (1 \leq j \leq l)$ be the polynomials in P_s, then we regard every h_j and f_{s-1} as polynomials in x_{s-1}, repeat the same discussion as what we do for f_s and g_is, so we get that, under constraints $g_i \geq 0, 1 \leq i \leq t$, the number of distinct real solutions of $f_s = 0, f_{s-1} = 0$ can be determined by the signs of polynomials in P_{s-1}; do the same discussion until $P_1(g_1, g_2, \cdots, g_t)$ is used. Because the conditions obtained in each step are necessary and sufficient, the theorem holds.

Now, theoretically speaking, we can get the necessary and sufficient condition for system TS to have (exactly n distinct) real solution(s) as follows:

Step 1 Compute $P_1(g_1, g_2, \cdots, g_t)$, the set of polynomials in parameters, which is defined above for TS.

Step 2 By the algorithm of PCAD [2, 3], we can get a P_1-invariant cad D of parameter space R^d and its cylindrical algebraic sample (cas) S [23].

Step 3 For every cell c in D and its sample point $s_c \in S$, substitute s_c into TS (denote it by $TS(s_c)$). Compute the number of distinct real solutions of system $TS(s_c)$, in which polynomials all have constant coefficients now. At the same time, computer the signs of polynomials in $P_1(g_1, g_2, \cdots, g_t)$ on this cell by substituting s_c into them respectively. Record the signs of polynomials in $P_1(g_1, g_2, \cdots, g_t)$ when the number of distinct real solutions of system $TS(s_c)$ equals to the required number n (or when the number > 0, if we are asked to find the condition for TS to have real solution). Obviously, the signs of polynomials in $P_1(g_1, g_2, \cdots, g_t)$ on cell c form a first order formula, denoted by Φ_c.

Step 4 If, in step 3, all we have recorded are $\Phi_{c_1}, \cdots, \Phi_{c_k}$, then $\Phi = \Phi_{c_1} \vee \cdots \vee \Phi_{c_k}$ is what we want.

5 The practical algorithm

The algorithm given in section 4 is not practical in many cases since $P_1(g_1, g_2, \cdots, g_t)$ usually has too many polynomials. Because not all those polynomials are necessary for expressing the condition we want, so we have to give an efficient algorithm to select those necessary polynomials from $P_1(g_1, g_2, \cdots, g_t)$.

Theorem 5.1. Given system TS as above. If $PolySet$ is a finite set of polynomials in parameters u, e.g.

$$PolySet = \{q_i(u) \in Z[u_1, \cdots, u_d] | 1 \leq i \leq k\},$$

then, by the algorithm of PCAD, we can get a $PolySet$-invariant cad D of parameter space R^d and its cas. If $PolySet$ satisfies that

1. the number of distinct real solutions of system TS is invariant in the same cell and
2. the number of distinct real solutions of system TS in two distinct cells C_1, C_2 is the same if $PolySet$ has the same sign in C_1, C_2,

then the necessary and sufficient conditions for system TS to have exactly n distinct real solution(s) can be expressed by the signs of the polynomials in $PolySet$. If $PolySet$ satisfies (1) only, then the necessary condition for system TS to have exactly n distinct real solution(s) can be expressed by the signs of the polynomials in $PolySet$.

Proof. We replace parameters u in TS with each sample point respectively. Because D is $PolySet$-invariant and $PolySet$ satisfies (1), we can record the signs of polynomials in $PolySet$ and the number of distinct real solutions of TS on each cell respectively. Choose all those cells on which TS has n distinct real solution(s). The signs of polynomials in $PolySet$ on those cells form a first order formula, e.g.

$$\Phi = \Phi_1 \vee \Phi_2 \vee \cdots \Phi_l.$$

where each Φ_i represents the signs of polynomials in $PolySet$ on a certain cell on which TS has n distinct real solution(s). We will say that Φ is the condition we want.

Given a parameter $a = (a_1, \cdots, a_d)$, which must fall into a certain cell, if $TS(a)$ has n distinct real solution(s), then a must belong to a cell on which TS has n distinct real solution(s), i.e. a must satisfy a certain formula Φ_i; on the contrary, if a satisfies a certain formula Φ_i, then, because TS has n distinct real solution(s) on the cell represented by Φ_i and $PolySet$ satisfies (2), we thus know that TS must have n distinct real solution(s) on the cell which a belongs to. Thus, the theorem holds.

Given system TS. For every f_i, let

$$R_1 = \mathrm{Discrim}(f_1, x_1),$$
$$R_i = \mathrm{res}\,(\mathrm{Discrim}(f_i, x_i), f_{i-1}, f_{i-2}, \cdots, f_1), i \geq 2,$$

where the definition of $\mathrm{res}\,(\mathrm{Discrim}(f_i, x_i), f_{i-1}, f_{i-2}, \cdots, f_1)$ can be found in section 3. For every g_j, let

$$Rg_j = \mathrm{res}\,(g_j, f_s, f_{s-1}, f_{s-2}, \cdots, f_1).$$

We define

$$BPs = \{R_i | 1 \leq i \leq s\} \bigcup \{Rg_j | 1 \leq j \leq t\}.$$

Clearly, $BPs \subseteq P_1(g_1, g_2, \cdots, g_t)$.

Theorem 5.2. Given system TS, BPs is defined as above. If we consider only those cells which are homeomorphic to R^d and do not consider those cells which are homeomorphic to $R^k\,(k < d)$ when use PCAD, then BPs satisfies (1) in theorem 5.1, so the necessary conditions (if we omit the parameters on those cells homeomorphic to $R^k\,(k < d)$) for system TS to have n distinct real solution(s) can be expressed by the signs of the polynomials in BPs.

Proof. By PCAD, we can get a BPs-invariant cad of R^d and its cas. Because we consider only those cells which are homeomorphic to R^d, given a cell C, the signs of each R_i and Rg_j on C are invariant and do not equal to 0.

First of all, by the definition of R_1, the sign of R_1 on C is invariant means that the number of real solutions of $f_1(u, x_1)$ is invariant on C; then we regard $f_2(u, x_1, x_2)$ as a polynomial in x_2, because on C,

$$f_1(u, x_1) = 0, R_2 = \mathrm{res}\,(\mathrm{Discrim}(f_2, x_2), f_1, x_1) \neq 0,$$

thus $\mathrm{Discrim}(f_2, x_2) \neq 0$ on C, i.e. if we replace x_1 in f_2 with the roots of f_1, the number of real solutions of f_2 is invariant. That is to say, the signs of R_1 and R_2 are invariant on C means the number of real solutions of $f_1 = 0, f_2 = 0$ is invariant on C; now, it's easy to see that the signs of $R_1, \cdots, R_s$ are invariant on C means the number of real solutions of $f_1 = 0, \cdots, f_s = 0$ is invariant on C.

Secondly, by the definition of Rg_j, we know that $Rg_j \neq 0$ means, on C, if we replace $x_1, \cdots, x_s$ in g_j with the roots of $f_1 = 0, \cdots, f_s = 0$, then the sign of g_j is invariant. That ends the proof.

By theorem 5.2, we can start our algorithm from BPs as follows:

Step 1 Let $PolySet = BPs$, $i = 1$.

Step 2 By the algorithm of PCAD [2, 3], compute a $PolySet$-invariant cad D of parameter space R^d and its cylindrical algebraic sample (cas) S [23]. In this step, we consider only those cells homeomorphic to R^d and do not consider those homeomorphic to R^k $(k < d)$, i.e., all those cells in D are homeomorphic to R^d and all sample points in S are taken from cells in D.

Step 3 For every cell c in D and its sample point $s_c \in S$, substitute s_c into TS (denote it by $TS(s_c)$). Compute the number of distinct real solutions of system $TS(s_c)$, in which polynomials all have constant coefficients now. At the same time, computer the signs of polynomials in $PolySet$ on this cell by substituting s_c into them respectively. Obviously, the signs of polynomials in $PolySet$ on cell c form a first order formula, denoted by Φ_c. When all the $TS(s_c)$s are computed, let

$$set1 = \{\Phi_c | \ TS \text{ has } n \text{ distinct real solution(s) on } c\},$$

$$set0 = \{\Phi_c | \ TS \text{ has not } n \text{ distinct real solution(s) on } c\}.$$

Step 4 Decide whether all the recorded Φ_cs can form a necessary and sufficient condition or not by verifying whether $set1 \cap set0$ is empty or not (because of Theorem 5.1 and Theorem 5.2). If $set1 \cap set0 = \emptyset$, go to Step 5; If $set1 \cap set0 \neq \emptyset$, let

$$PolySet = Polyset \cup P_1(g_1, \cdots, g_i), \ i = i + 1,$$

and back to Step 2.

Step 5 If $set1 = \{\Phi_{c_1}, \cdots, \Phi_{c_m}\}$, then $\Phi = \Phi_{c_1} \vee \cdots \vee \Phi_{c_m}$ is what we want.

Remark: In order to make our algorithm practical, we do not consider the "boundaries" when use PCAD. So, the condition obtained by this algorithm is a necessary and sufficient one if we omit the parameters on the "boundaries". In another word, all the conditions obtained by our program DISCOVERER, which is implemented in Maple according to this algorithm, should be understood as follows: if the strict inequalities hold, the conditions hold; if equalities hold, more discussions needed.

Now, we consider following problem:

Give the necessary and sufficient condition which the parameters u must satisfy for the following system PS to have (exactly n distinct) real solution(s),

$$PS : \begin{cases} h_1(u, X) = 0, h_2(u, X) = 0, \cdots, h_s(u, X) = 0 \\ g_1(u, X) \geq 0, g_2(u, X) \geq 0, \cdots, g_t(u, X) \geq 0. \end{cases} \tag{2}$$

where u means $u_1, u_2, \cdots, u_d$, treated as parameters; X means x_1, x_2, $\cdots$, x_s, treated as variables. That is to say,

$$h_i, g_j \in Z(u_1, \cdots, u_d)[x_1, \cdots, x_s], 1 \leq i \leq s, 1 \leq j \leq t.$$

First of all, by Wu's elimination [24, 25], we can reduce $h_1(u, X) = 0, h_2(u, X) = 0, \cdots, h_s(u, X) = 0$ to some "triangular sets". Then, if necessary, by WR algorithm [32, 33], make these triangular sets into normal ascending chains in which every chain is simplicial w.r.t. each g_j $(1 \le j \le t)$. So, under some nondegenerate conditions, we can reduce PS to some TSs. These nondegenerate conditions, however, do not bring any new limitations on our algorithm because we do not consider "boundaries" when use PCAD and all those parameters which make degenerate conditions true are contained in "boundaries". Another situation we do have to handle is some TSs reduced from PS may have real solutions with dimension greater than 0. In our present algorithm and program, we do not deal with this situation and if it occurs, DISCOVERER outputs a message and does nothing else.

6 Examples

Many problems with various background can be formulated into system PS and can be solved by DISCOVERER automatically.

The calling sequence of DISCOVERER for system PS is:

$$\text{tofind}\,([h_1, \cdots, h_s], [g_1, \cdots, g_t], [x_1, \cdots, x_s, u_1, \cdots, u_d], \alpha);$$

where α has following three kind of choices:

- a non-negative integer b which means the condition for PS to have b distinct real solution(s);
- a range $b..c$ (b, c are non-negative integers, $b < c$) which means the condition for PS to have b or $b + 1$ or $\cdots$ or c distinct real solutions;
- a range $b..n$ (b is a non-negative integer, n a name) which means the condition for PS to have more than or equal to b distinct real solutions.

Example 6.1. Which triangles can occur as sections of a regular tetrahedron by planes which separate one vertex from the other three?

This example appeared as an unsolved problem in the American Mathematical Monthly (Oct., 1994)[15]. In fact, it is a special case of the well-known Perspective-three-Point (P3P) problem (see following Example 5.2).

If we let $1, a, b$ (assume $b \ge a \ge 1$) be the lengths of three sides of the triangle, and x, y, z the distances from the vertex to the three vertexes of the triangle, respectively (see Figure 2), then, what we need is to find the necessary and sufficient condition that a, b should satisfy for the following system to have real solution(s),

$$\begin{cases} h_1 = x^2 + y^2 - xy - 1 = 0, \\ h_2 = y^2 + z^2 - yz - a^2 = 0, \\ h_3 = z^2 + x^2 - zx - b^2 = 0, \\ x > 0, y > 0, z > 0, a - 1 \ge 0, b - a \ge 0. \end{cases}$$

With our program DISCOVERER, we need only to type in

$$\text{tofind}\,([h_1,h_2,h_3],[x,y,z,a-1,b-a],[x,y,z,a,b],1..n);$$

DISCOVERER runs 26 seconds on a k6/233 PC with MAPLE5.3, and outputs

FINAL RESULT :
The system has required real solution(s) IF AND ONLY IF

$$[0 \le R1, 0 \le R2]$$
or
$$[0 \le R1, R2 \le 0, 0 \le R3]$$

where

$$R1 = a^2 + a + 1 - b^2$$
$$R2 = a^2 - 1 + b - b^2$$
$$\begin{aligned} R3 = 1 &- \frac{8}{3}a^2 - \frac{8}{3}b^2 + \frac{16}{9}a^8 - \frac{68}{27}b^6a^2 + \frac{241}{81}b^4a^4 - \frac{68}{27}b^2a^6 \\ &- \frac{68}{27}b^4a^2 - \frac{68}{27}b^2a^4 - \frac{2}{9}b^6 + \frac{16}{9}b^8 - \frac{2}{9}a^6 + \frac{46}{9}b^2a^2 \\ &+ \frac{16}{9}b^4 + \frac{16}{9}a^4 + \frac{46}{9}b^2a^8 + \frac{46}{9}b^8a^2 - \frac{68}{27}b^6a^4 - \frac{68}{27}b^4a^6 \\ &+ \frac{16}{9}b^4a^8 - \frac{8}{3}b^{10}a^2 + \frac{16}{9}b^8a^4 - \frac{2}{9}b^6a^6 - \frac{8}{3}b^2a^{10} - \frac{8}{3}b^{10} \\ &+ b^{12} - \frac{8}{3}a^{10} + a^{12} \end{aligned}$$

The article [15] has given a sufficient condition that any triangle with two angles $> 60°$ is a possible section. It is easy to see that this condition is equivalent to $[R1 > 0,\ R2 > 0\,]$.

Example 6.2. Given the distance between every pair of 3 control points, and given the angle to every pair of the control points from an additional point called the centre of perspectivity (say P), find the lengths of the segments joining P and each of the control points. This problem originates from camera calibration and is called perspective-three-point (P3P) problem. The corresponding algebraic equation system is called the P3P equation system.

So-called a solution classification of P3P equation system is to give explicit conditions under which the system has none, one, two, $\cdots$, real physical solutions, respectively. This problem had been open for many years [16], until [27] appeared recently.

This example is about a special case of P3P problem and was studied in different way by Gao and Cheng [16]. Suppose two angles are the same and the three distances between the control points A, B, C are the same(see Figure 1), give the solution classification. The corresponding equation system is

$$\begin{cases} h_1 = y^2 + z^2 - 2yzp - 1 = 0, \\ h_2 = z^2 + x^2 - 2zxq - 1 = 0, \\ h_3 = x^2 + y^2 - 2zxq - 1 = 0, \\ x > 0, y > 0, z > 0, 1 - p^2 > 0, 1 - q^2 > 0, p + 1 - 2q^2 \ge 0. \end{cases}$$

where parameters p, q denote the cosines of the angles

$$\angle BPC, \quad \angle CPA \ (= \angle APB),$$

and x, y, z denote the lengths of the segments PA, PB, PC, respectively.
By our program DISCOVERER, we need only type in

$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 0);$$
$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 1);$$
$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 2);$$
$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 3);$$
$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 4);$$
$$\text{tofind}\,([h_1, h_2, h_3], [x, y, z, 1-p^2, 1-q^2, p+1-2q^2], [x, y, z, p, q], 5..n);$$

respectively. After 64 seconds, on a PII/266 PC with MAPLE5.4, we get the solution classification except some boundaries:

The system has 0 real solution IF AND ONLY IF

$$[0 < R1, 0 < R2, R3 < 0, 0 < R4]$$

or

$$[R1 < 0, 0 < R2, R3 < 0]$$

The system has 1 real solution IF AND ONLY IF

$$[R2 < 0, R3 < 0, 0 < R5]$$

or

$$[R2 < 0, R5 < 0]$$

The system has 2 real solutions IF AND ONLY IF

$$[0 < R1, 0 < R2, R3 < 0, R4 < 0, 0 < R5]$$

or

$$[0 < R1, 0 < R2, 0 < R3, 0 < R4, 0 < R5]$$

or

$$[0 < R1, 0 < R2, R4 < 0, R5 < 0]$$

The system has 3 real solutions IF AND ONLY IF

$$[R2 < 0, 0 < R3, R4 < 0, 0 < R5]$$

The system has 4 real solutions IF AND ONLY IF

$$[0 < R1, 0 < R2, 0 < R3, R4 < 0, R5 < 0]$$

The system has $5..n$ real solutions IF AND ONLY IF

There are not $5..n$ real solutions in this branch.

where

$$R1 = q,$$
$$R2 = 2p - 1,$$
$$R3 = 2q - 1,$$
$$R4 = 2p - 1 - q^2,$$
$$R5 = 2pq^2 - 3q^2 + 1.$$

Note that $p + 1 - 2q^2 \geq 0$, the given condition, should hold in all cases. And if you want to know the situation when parameters are on a boundary, say $R2$, you need only to type in

tofind $([h_1, h_2, h_3, R2], [x, y, z, 1 - p^2, 1 - q^2, p + 1 - 2q^2], [x, y, z, p, q], 0);$
tofind $([h_1, h_2, h_3, R2], [x, y, z, 1 - p^2, 1 - q^2, p + 1 - 2q^2], [x, y, z, p, q], 1);$
tofind $([h_1, h_2, h_3, R2], [x, y, z, 1 - p^2, 1 - q^2, p + 1 - 2q^2], [x, y, z, p, q], 2);$
tofind $([h_1, h_2, h_3, R2], [x, y, z, 1 - p^2, 1 - q^2, p + 1 - 2q^2], [x, y, z, p, q], 3);$
tofind $([h_1, h_2, h_3, R2], [x, y, z, 1 - p^2, 1 - q^2, p + 1 - 2q^2], [x, y, z, p, q], 4);$

By this way, we get the complete solution classification, as indicated in Figure 2.

Example 6.3. Solving geometric constraints is the central topic in many current work of developing intelligent Computer Aided Design systems and interactive constraint-based graphic systems. The existing algorithms do not consider whether the given system has real physical solution(s) or not. Our algorithm and program can give the necessary and sufficient condition for the system to have real physical solution(s). For example:

Give the necessary and sufficient condition for the existence of a triangle with elements a, h_a, R, where a, h_a, R means the side-length, altitude, and circumradius, respectively.

Clearly, we need to find the necessary and sufficient condition for the following system to have real solution(s),

$$\begin{cases} f_1 = a^2h_a^2 - 4s(s-a)(s-b)(s-c) = 0, \\ f_2 = 2Rh_a - bc = 0, \\ f_3 = 2s - a - b - c = 0, \\ a > 0, b > 0, c > 0, a + b - c > 0, b + c - a > 0, \\ c + a - b > 0, R > 0, h_a > 0. \end{cases}$$

We type in

tofind $([f_1, f_2, f_3], [a, b, c, a + b - c, b + c - a,$
$c + a - b, R, h_a], [s, b, c, a, R, h_a], 1..n);$

DISCOVERER runs 4.5 seconds on a PII/266 PC with MAPLE5.4, and outputs

FINAL RESULT :
The system has required real solution(s) IF AND ONLY IF

$$[0 \leq R1, 0 \leq R3]$$
or
$$[0 \leq R1, R2 \leq 0, R3 \leq 0]$$

where

$$R1 = R - \frac{1}{2}a$$

$$R2 = Rh_a - \frac{1}{4}a^2$$

$$R3 = -\frac{1}{2}h_a^2 + Rh_a - \frac{1}{8}a^2$$

In [19], the condition they gave is $R1 \geq 0 \wedge R3 \geq 0$. Now, we know they are wrong and that is only a sufficient condition.

Our program, DISCOVERER, is very efficient for solving this kind of problems. By DISCOVERER, we have discovered or rediscovered about 70 such conditions for the existence of a triangle, and found three mistakes in [19].

References

1. Arnon, D S, Geometric Reasoning with Logic and Algebra, *Artificial Intelligence,* **37**(1988), 37-60.
2. Arnon, D S, Collins, G E & McCallum, S, Cylindrical Algebraic Decomposition I: The Basic Algorithm, *SIAM J.Comput.*, Vol. **13**, No. 4 (1984), 865-877.
3. Arnon, D S, Collins, G E & McCallum, S, Cylindrical Algebraic Decomposition II: An Adjacency Algorithm for the Plane, *SIAM J.Comput.*, Vol. **13**, No. 4 (1984), 878-889.
4. Bottema, O, et al, *Geometric Inequalities*, Wolters-Noordhoff Publishing, Groningen, Netherland, 1969.
5. Buchberger, B, Applications of Gröbner bases in non-linear computational geometry, in *Geometric Reasoning,* pp.413-446. MIT Press, Cambridge, MA, 1989.
6. Buchberger, B, Collins, G E & Kutzler, B, Algebraic methods for geometric reasoning, *Annual Review of Computing Science* **3**:85-119(1988).
7. Chen, M, Generalization of Discrimination System for Polynomials and its Applications, Ph.D dissertation, Sichuan Union University, 1998. (in Chinese)
8. Chou, S C, *Mechanical Geometry Theorem Proving,* D. Reidel Publishing Company, (Amsterdam) 1988.
9. Chou, S C, Gao, X S & Arnon, D S, On the mechanical proof of geometry theorems involving inequalities, in *Advances in Computing Research,* Vol.**6**, pp.139-181. JAI Press Inc., 1992.
10. Chou, S C, Gao, X S & McPhee, N, A combination of Ritt-Wu's method and Collins's method, *Proc. of CADE-12,* Springer-Verlag, 1994. pp.401-415.
11. Chou, S C, Gao, X S & Zhang J Z, *Machine Proofs in Geometry,* World Scientific Press, 1994.
12. Collins, G E, Quantifier elimination for real closed fields by cylindrical algebraic decomposition, in *Lecture Notes in Computer Science,* **33**:134-183. Springer-Verlag, Berlin, 1975.
13. Collins, G E & Hong, H, Partial cylindical algebraic decomposition for quantifier elimination, *J. Symbolic Computation,* **12**, 299-328 (1991).
14. Cox, D, Little, J & O'Shea, D, *Ideals, Varieties, and Algorithms,* Springer-Verlag, 1992.

15. Folke, E, Which triangles are plane sections of regular tetrahedra? *American Mathematics Monthly,* Oct., 1994, 788-789.
16. Gao, X S & Cheng, H F, On the Solution Classification of the "P3P" Problem, *Proc. of ASCM'98,* Z. Li (ed.) Lanzhou University Press, 1998. pp.185-200.
17. González-Vega, L., Lombardi, H., Recio, T., Roy, M.-F., Sturm-Habicht sequence. *Proc. of ISSAC'89*. ACM Press. pp. 136-146, 1989.
18. Kapur, D, Automated geometric reasoning: Dixon resultants, Gröbner bases, and characteristic sets, Springer's LNAI **1360**, 1997. pp.1-36.
19. Mitrinovic, D S, Pecaric, J E, & Volenec, V, *Recent Advances in Geometric Inequalities,* Kluwer Academic Publishers, 1989.
20. Tarski, A, *A Decision Method for Elementary Algebra and Geometry,* University of California Press, Berkeley, 1951.
21. Wang, D M, Elimination procedures for mechanical theorem proving in geometry, *Ann. Math. Artif. Intell.* **13**: 1-24 (1995).
22. Wang, D M, A Decision Method for Definite Polynomial, MM Research Preprints, No. **2**, 1987, Beijing, pp. 68-74.
23. Winkler, F, *Polynomial Algorithms in Computer Algebra,* Springer-Verlag/Wien, 1996.
24. Wu, W T, On the decision problem and the mechanization of theorem-proving in elementary geometry, *Sci. Sinica* **21**: 159-172 (1978).
25. Wu, W T, *Mechanical theorem proving in geometries: Basic principles* (translated from the Chinese by X. Jin and D. Wang), Springer, 1994.
26. Yang, L, Practical automated reasoning on inequalities: Generic programs for inequality proving and discovering, *Proceedings of the Third Asian Technology Conference in Mathmatics,* W.-C. Yang et al.(eds), Springer-Verlag 1998, pp.24-35.
27. Yang, L, A Simplified Algorithm for Solution Classification of the Perspective-three-Point Problem, MM Research Preprints, No. **17**, 1998, Beijing, pp. 135-145.
28. Yang, L, Gao, X S, Chou, S C & Zhang, J Z, Automated production of readable proofs for theorems in non-Euclidean geometries. In: Springer's LNAI **1360**, pp. 171-188 (1997).
29. Yang, L, Hou, X R & Zeng, Z B, A Complete Discrimination System for Polynomials, *Science in China,* Series **E39**:6(1996), 628-646.
30. Yang, L & Xia, B C, Explicit Criterion to Determine the Number of Positive Roots of a Polynomial, MM Research Preprints, No. **15**, 1997, Beijing, pp. 134-145.
31. Yang, L, Zhang, J Z & Hou, X R, A Criterion of Dependency between Algebraic Equations and Its applications, *Proceedings of the 1992 International Workshop on Mathematics Mechanization,* International Acdemic Publishers 1992, Beijing. pp.110-134.
32. Yang, L, Zhang, J Z & Hou, X R, An Efficient Decomposition Algorithm for Geometry Theorem Proving Without Factorization, *Proceedings of Asian Symposium on Computer Mathematics 1995*, H. Shi & H. Kobayashi (eds), Scientists Incorporated 1995, Japan. pp.33-41.
33. Yang, L, Zhang, J Z & Hou, X R, *Nonlinear Algebraic Equation System and Automated Theorem Proving,* Shanghai Scientific and Technological Education Publishing House, Shanghai, 1996. (in Chinese)
34. Zhang, J Z, Yang, L & Hou, X R, A criterion for dependency of algebraic equations with applications to automated theorem proving, *Sci. China* (Ser. **A**) **37**: 547-554 (1994).