DISCOVERER: A tool for solving problems involving polynomial inequalities *

Bican Xia
School of Mathematical Sciences, Peking University
Beijing, 100871, China
xiabican@sxx0.math.pku.edu.cn

Abstract DISCOVERER, developed under Maple, is a package aiming at discovering and proving inequality-type theorems. It can discover new inequalities automatically, without requiring us to put forward any conjectures beforehand. It is also applied to the classification of the real physical solutions of geometric constraint problems. Many inequalities with various backgrounds have been discovered or re-discovered by the package. Furthermore, DISCOVERER also has many other functions for solving problems related to real algebra and geometry such as, computing the number of distinct real solutions of a system of polynomial equations and inequalities and, determining whether or not a polynomial is semi-definite under some constraints of polynomial inequalities.

1 Introduction

Automated reasoning in real algebra and real geometry has become one of the most challenging topics in the area of automated reasoning and any new approach on those problems in this fields is welcome. In the past two decades, substantial efforts have been made on this aspect. Many algorithms such as those in [1, 2, 4, 5, 7, 9, 11, 14, 16, 18, 19, 20, 21], to name a few, have been developed. Among them, for example, there are several implemented quantifier elimination algorithms such as QEPCAD, QERRC and REDLOG [9]. There is also a very powerful tool for automatic theorem proving for problems involving inequalities, which is *dimension-decreasing* algorithm implemented in MAPLE-program BOTTEMA[18].

DISCOVERER is a package implemented under Maple. Although it aims at discovering and proving inequality-type theorems, it has many other functions. In this report, we will demonstrate how to use it for various purposes. You will find it is very easy to use and is very efficient for some kinds of problems. Maybe you can find some applications of DISCOVERER to the problems in your fields. For the detail of the algorithms covered by DISCOVERER, see references [7, 16, 17, 19, 20, 21, 22, 23, 24].

2 Functions: `tofind` and `Tofind`

These two functions aiming at discovering and proving inequality-type theorems are main functions in DISCOVERER. A description and a proof to the algorithm for these two functions can be found in [19, 20].

Example 1 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, inradius and circumradius, respectively.

*Project supported by Chinese National 973 Project (G1998030602) and the 95 Key Project of Fundamental Research of Chinese Academy of Sciences

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

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

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 PS' has real solution(s).

With our program DISCOVERER, we attack the problem by two steps. First of all, we call

$$\text{tofind}\,([p_1, p_2, p_3], [a, b, c, 1-a, 1-b, 1-c, r, R], [a, b, c, r, R], 1..n);$$

DISCOVERER runs 0.8 second on a Pentium/450 PC with Maple V.4, and outputs

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

$$[R_1 < 0]$$

where

$$R_1 = 1 + 2\,r^2 - 4\,R^2 - 20\,rR + 12\,r^3R + 48\,r^2R^2 + r^4 + 64\,rR^3.$$

Now, if parameters r, R are not on the boundary (that is, $R_1 = 0$), the condition obtained above is already a necessary and sufficient one (see the Remark in Section 5 of reference [19]). If we do want to know whether or not there exists a triangle with elements s, r, R when $R_1 = 0$, take the second step by typing in

$$\text{Tofind}\,([p_1, p_2, p_3, R_1], [a, b, c, 1-a, 1-b, 1-c, r, R], [a, b, c], [r, R], 1..n);$$

DISCOVERER runs 350 seconds and outputs

FINAL RESULT:
There is always required real solution(s).

Therefore, by homogenization, we conclude that there exists a triangle with elements s, r, R if and only if

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

which is called the "Fundamental Inequality" [13] for triangles.

Now, we give a detail description on the usage of `tofind` and `Tofind`. These two functions are applicable to all those problems which can be formulated into:

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

$$PS: \begin{cases} h_1(u, X) = 0, h_2(u, X) = 0, \cdots, h_s(u, X) = 0, \\ g_1(u, X) \ge 0, g_2(u, X) \ge 0, \cdots, g_t(u, X) \ge 0, \end{cases} \tag{1}$$

where u means $u_1, u_2, \cdots, u_d$, treated as parameters and X means $x_1, x_2, \cdots, x_s$, treated as variables, i.e.

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

and we always assume that $\{h_1(u, X) = 0, h_2(u, X) = 0, \cdots, h_s(u, X) = 0\}$ has only zero-dimensional solutions.

Usually, for system PS, we call `tofind` first to find a satisfactory condition, which means the condition is already a necessary and sufficient one if parameters are not on the "boundaries" (also see the Remark in Section 5 of reference [19]), and then, if necessary, call `Tofind` to find further results when parameters are on boundaries.

The calling sequence in DISCOVERER for system PS has following three forms:

1) tofind $([h_1,\cdots,h_s],[g_1,\cdots,g_t],[g_r,\cdots,g_t],[x_1,\cdots,x_s,u_1,\cdots,u_d],\alpha)$;
2) tofind $([h_1,\cdots,h_s],[g_1,\cdots,g_t],[x_1,\cdots,x_s,u_1,\cdots,u_d],\alpha)$;
3) tofind $([h_1,\cdots,h_s],[x_1,\cdots,x_s,u_1,\cdots,u_d],\alpha)$; .

These three forms correspond respectively to systems

$$PS1:\begin{cases} h_1(u,X)=0,h_2(u,X)=0,\cdots,h_s(u,X)=0,\\ g_1(u,X)\ge 0,\cdots,g_{r-1}(u,X)\ge 0,g_r(u,X)>0,\cdots,g_t(u,X)>0,\end{cases} \quad (2)$$

$$PS2:\begin{cases} h_1(u,X)=0,h_2(u,X)=0,\cdots,h_s(u,X)=0,\\ g_1(u,X)>0,\cdots,g_t(u,X)>0,\end{cases} \quad (3)$$

and

$$PS3:\begin{cases} h_1(u,X)=0,h_2(u,X)=0,\cdots,h_s(u,X)=0,\\ x_1>0,\cdots,x_s>0,u_1>0,\cdots,u_d>0,\end{cases} \quad (4)$$

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.

In all three cases, α has following three kind of choices:

- a non-negative integer b which means to get the condition for PS to have exactly b distinct real solution(s);
- a range $b..c$ (b,c are non-negative integers, $b<c$) which means to get 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 to get the condition for PS to have more than or equal to b distinct real solutions.

Similarly, `Tofind` also has following three calling sequences:

1) Tofind $([h_1,\cdots,h_s,R_1,\cdots,R_m],[g_1,\cdots,g_t],[g_r,\cdots,g_t],[x_1,\cdots,x_s],[u_1,\cdots,u_d],\alpha)$;
2) Tofind $([h_1,\cdots,h_s,R_1,\cdots,R_m],[g_1,\cdots,g_t],[x_1,\cdots,x_s],[u_1,\cdots,u_d],\alpha)$;
3) Tofind $([h_1,\cdots,h_s,R_1,\cdots,R_m],[x_1,\cdots,x_s],[u_1,\cdots,u_d],\alpha)$;

where each R_i is a "boundary" which is a polynomial in parameters obtained by `tofind` or some constraint polynomial (g_j) in parameters.

Remark In the calling sequences of `tofind` and `Tofind`, there is a little difference between the input lists of variables and parameters. In the calling sequences of `tofind`, we input $[x_1,\cdots,x_s,u_1,\cdots,u_d]$ where the first s elements are regarded as variables (in system PS, we have as many equations as variables) and the others parameters. On the other hand, when using `Tofind`, we input $[x_1,\cdots,x_s]$ and $[u_1,\cdots,u_d]$ where the elements in the first list are regarded as variables and that in the second list parameters. Certainly, the order of variables (or parameters) may have effects on the computational complexity but not on the results.

Example 2[10] Which triangles can occur as sections of a regular tetrahedron by planes which separate one vertex from the other three?

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, then, the problem is reduced to finding the necessary and sufficient condition that a,b should satisfy for the following system to have real solution(s),

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

With our program DISCOVERER, first of all, we type in:

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

DISCOVERER runs 13.4 seconds on a Pentium/450 PC with Maple V.4, and outputs

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

$$[0<R_1,0<R_2]$$
or
$$[0<R_1,R_2<0,0<R_3]$$

where

$$R_1 = a^2+a+1-b^2$$

$$R_2 = a^2-1+b-b^2$$

$$\begin{aligned} R_3 = \quad & 1-\frac{8}{3}a^2-\frac{8}{3}b^2+\frac{16}{9}a^8-\frac{68}{27}b^6a^2+\frac{241}{81}b^4a^4-\frac{68}{27}b^2a^6 \\ & -\frac{68}{27}b^4a^2-\frac{68}{27}b^2a^4-\frac{2}{9}b^6+\frac{16}{9}b^8-\frac{2}{9}a^6+\frac{46}{9}b^2a^2 \\ & +\frac{16}{9}b^4+\frac{16}{9}a^4+\frac{46}{9}b^2a^8+\frac{46}{9}b^8a^2-\frac{68}{27}b^6a^4-\frac{68}{27}b^4a^6 \\ & +\frac{16}{9}b^4a^8-\frac{8}{3}b^{10}a^2+\frac{16}{9}b^8a^4-\frac{2}{9}b^6a^6-\frac{8}{3}b^2a^{10}-\frac{8}{3}b^{10} \\ & +b^{12}-\frac{8}{3}a^{10}+a^{12}. \end{aligned}$$

Reference [10] gave a sufficient condition that any triangle with two angles $> 60^\circ$ is a possible section. It is easy to see that this condition is equivalent to $[R_1>0,\ R_2>0\,]$.

Now, if parameters a,b are not on the boundaries (that is, $R_1=0,\ R_2=0,\ R_3=0,\ a-1=0,\ b-a=0$), the condition obtained above is already a necessary and sufficient one (see the Remark in Section 5 of reference [19]). But, strictly speaking, to get a necessary and sufficient condition, we have to give the result when a,b are on the boundaries. Thus, we take the second step. If we want to know the result when a,b are on a certain boundary, say R_1, we need only to type in

$$\text{Tofind}\,([h_1,h_2,h_3,R_1],[x,y,z,a-1,b-a,a+1-b],$$

$$[x,y,z,a+1-b],[x,y,z],[a,b],1..n);$$

DISCOVERER outputs that

FINAL RESULT:
There is no required solution(s)!

By this way together with some interactive computations, we finally get the condition for the system to have real solution(s):

$$[0<R_1,0<R_2,R_3\le 0,0<a-1,0\le b-a,0<a+1-b]$$
or
$$[0<R_1,0\le R_3,0\le a-1,0\le b-a,0<a+1-b].$$

Actually, by DISCOVERER, we can do more than the request to this problem. If we type in respectively

$$\text{tofind}\,([h_1,h_2,h_3],[x,y,z,a-1,b-a,a+1-b],[x,y,z,a+1-b],[x,y,z,a,b],1);$$
$$\text{tofind}\,([h_1,h_2,h_3],[x,y,z,a-1,b-a,a+1-b],[x,y,z,a+1-b],[x,y,z,a,b],2);$$
$$\text{tofind}\,([h_1,h_2,h_3],[x,y,z,a-1,b-a,a+1-b],[x,y,z,a+1-b],[x,y,z,a,b],3);$$

we will get the condition for above system to have exactly 1 or 2 or 3 real solution(s) respectively. By this way, we get the so-called complete solution classification of this problem.

It was reported [12] that Vincent Cloffari had given an answer to this problem. But his result has not been published formally and from the short description in reference [12], we cannot decide whether his result is correct and equivalent to ours or not. But one thing is clear that our method has at least two advantages: first, our method is applicable to a class of problems rather than a single problem; second, our method is applicable to the so-called solution classification problems.

3 Function: `nearsolve`

Suppose we are given a system

$$PSC: \begin{cases} p_1(x_1, x_2, ..., x_s) = 0, \\ p_2(x_1, x_2, ..., x_s) = 0, \\ \cdots\cdots, \\ p_s(x_1, x_2, ..., x_s) = 0, \\ g_1(x_1, x_2, ..., x_s) \geq 0, ..., g_r(x_1, x_2, ..., x_s) \geq 0, \\ g_{r+1}(x_1, x_2, ..., x_s) > 0, ..., g_t(x_1, x_2, ..., x_s) > 0, \\ h_1(x_1, x_2, ..., x_s) \neq 0, ..., h_m(x_1, x_2, ..., x_s) \neq 0, \end{cases}$$

where $p_i(1 \leq i \leq s)$, $g_j(1 \leq j \leq t)$, $h_k(1 \leq k \leq m)$ are all polynomials with integer coefficients and equations $\{p_1 = 0, p_2 = 0, \cdots, p_s = 0\}$ has a finite number of common solutions. The question is how many distinct real solutions the system has.

In DISCOVERER, function **nearsolve** which implements an algorithm in reference [16, 17] is designed to solve the problem. For system PSC, the calling sequence of **nearsolve** is

$$\text{nearsolve}([p_1, \cdots, p_s], [g_1, \cdots, g_r], [g_{r+1}, \cdots, g_t], [h_1, \cdots, h_m], [x_1, \cdots, x_s]);$$

Example 3 Given system

$$\begin{cases} f_1(x) = -5000000x^6 + 1875000x^4 - 68125x^2 + 8 = 0, \\ f_2(x,y) = (-1000 - 8000x^3)y^2 + (-8000x^4 + 4000x^2 + 8)y - 117x \\ \qquad -8000x^5 + 3000x^3 = 0, \\ f_3(x,y,z) = (8x + 8y)z + 8xy - 1 = 0, \\ x + y > 0, y + z > 0, z + x > 0, \end{cases}$$

we call

$$\texttt{nearsolve}([f_1, f_2, f_3], [\,], [x + y, y + z, z + x], [\,], [x, y, z]);$$

the output of DISCOVERER is 5.

Example 4 How many distinct real solutions does following system have?

$$\begin{cases} f_1(b) = 0, \\ f_2(b,c) = 0, \\ f_3(b,c,d) = 0, \\ f_4(b,c,d,e) = 0, \\ b > 0, c > 0, d > 0, e > 0, c - d \neq 0, \end{cases}$$

where

$$\begin{aligned} f_1 = {} & 241538508382138075462768483549507937558926051383237186598921\backslash \\ & 351434775082998337615592652313777086354071766371461311711285097 62\backslash \\ & 9761b^{32} + 635066713778840598710749498577504496793070850884097947974\backslash \\ & 380292191777777224247909356698829052300189668676627063462218165 26\backslash \end{aligned}$$

$$
\begin{array}{l}
252732166640b^{30} - 617516729685594231347246877282308919087789340602 36\backslash \\
333460795119633799976734997949468942620276039633337231215472829 57\backslash \\
28824956115726848000b^{28} + 27390034646753639766624212069599001290967\backslash \\
5944831268619419963947375736698335046092233994317817055192976247 0\backslash \\
251477314187497028082105057280b^{26} - 1437145166237554579477639351890\backslash \\
5979491561814339246077914862723414402467431025889585529684328202 6\backslash \\
273568967644536703423955174325414264832 0b^{24} - 2986092587283398359 15\backslash \\
1187320928040065994286379338588944475146452773892605985950218440 1\backslash \\
24363918776508369053084089437022884472546257797 12b^{22} + 435447287511\backslash \\
2985246215589621601392927034444281183521252549255181277184403348 5\backslash \\
662489132077458388407791801673830767006425164301268418560b^{20} - 3414\backslash \\
8807436745609347300495600312270857833357366729397393592991014187 8\backslash \\
354056591935239593924781478529672997249005700302610906831283814 4b^{18} + \\
8223756555269861165757056665815244377194121394959540192025015505 4\backslash \\
5049085149044464549229420580826884899873578176474995576252160540 6\backslash \\
72b^{16} - 18465863911534614222771407254727218540553077903981494060726\backslash \\
3349347328428027423682270256946111377666109617855536427391671109 6\backslash \\
2872320b^{14} + 21045839815430151526439387212875775018342624349983019 2\backslash \\
2594581580196154317969812721378800827355158137115610536502092524 5\backslash \\
8008412160b^{12} - 13978667557446331767642193782855396004749356953998 5\backslash \\
1825067720609793412281015523288310487856480352735638714141311722 8\backslash \\
188455403352b^{10} + 585188215302425253433702248414515313183366444530 00\backslash \\
0749744203303733447689121059491379312443203975837106226735111603 9\backslash \\
560626176b^{8} - 155390183378463952221186520858978074062380250509979 3\backslash \\
4777821414922700387593995537411137322766733076998082737318834930 1\backslash \\
88288b^{6} + 313996054016507120446473671329184542290007796627774567 47\backslash \\
42263224178629629604686504273402365034150253387778953172536524 8b^{4} - \\
2214798152846620823775109546914376369749948855722620121351416670 2\backslash \\
188991127101416805749416908807763189989750987555481 6b^{2} + 6072087665\backslash \\
3402761142507664131495320256148247367176990429610550229613067763 9\backslash \\
952549181438379528451116769583982182 4
\end{array}
$$

$$
\begin{aligned}
f_2 = {} & 2075\, b^{16}\, c^{12} + 284580\, b^{14}\, c^{10} + 357840\, b^{12}\, c^{9} + 10185588\, b^{12}\, c^{8} + 20167488\, b^{10}\, c^{7} \\
& + (21285312\, b^{8} - 62355744\, b^{10})\, c^{6} - 99610560\, b^{8}\, c^{5} \\
& + (-4855244976\, b^{8} - 361573632\, b^{6})\, c^{4} + (-37158912\, b^{4} - 3758980608\, b^{6})\, c^{3} \\
& + (54181472832\, b^{6} + 429235200\, b^{4})\, c^{2} + (4897760256\, b^{4} + 488374272\, b^{2})\, c \\
& - 123974556480\, b^{4} + 18874368 + 9432723456\, b^{2}
\end{aligned}
$$

$$
f_3 = 9\, b^4\, d^3 + 45\, b^4\, c\, d^2 + (35\, b^4\, c^2 - 486\, b^2)\, d - 108\, b^2\, c - 264 + 10\, b^4\, c^3
$$

$$
\begin{aligned}
f_4 = {} & (36\, d^2\, b^2 - 8\, c^2\, b^2 - 28\, d\, b^2\, c)\, e + 15\, b^4\, c\, d^2 + 6\, b^4\, c^3 + 21\, b^4\, c^2\, d - 144\, b^2\, c \\
& + 9\, b^4\, d^3 - 120 - 648\, b^2\, d.
\end{aligned}
$$

We call

$$\texttt{nearsolve}([f_1, f_2, f_3, f_4], [\,], [b, c, d, e], [c - d], [b, c, d, e]);$$

and get the number of distinct real solutions is 6.

Example 5 System

$$\begin{cases} f_1(r) = r^4 + 24r^3 + 194r^2 + 472r - 15 = 0, \\ f_2(r,a) = a^3 - 2a^2 + (r^2 + 8r + 1)a - 8r = 0, \\ f_3(r,a,b) = ab^2 + (a^2 - 2a)b + 8r = 0, \\ f_4(r,a,b,c) = a + b + c - 2 = 0, \\ r > 0, a > 0, b > 0, c > 0, 1 - a > 0, 1 - b > 0, 1 - c > 0. \end{cases}$$

has 4 distinct real solutions.

The time spent for above three examples on a Pentium/450 PC with Maple V.4 are 0.6 second, 134 seconds and 98 seconds respectively.

4 More

There are many other tools in DISCOVERER such as `tell`, `discr`, `nrd`, `wrd` and so on. Their functions are listed as follows,

`tell`: to determine, under some polynomial inequalities, whether a polynomial is positive (negative) semi-definite or not;

`discr`: to get the discriminant sequence [21] of a given polynomial;

`nrd`: to get the negative discriminant sequence [22] of a given polynomial;

`wrd`: to get the WR decomposition [24] of a triangular set AS w.r.t. a polynomial g;

`PCAD`: to get a *Cylindrical Algebraic Sample* [15] of a polynomial set by algorithm PCAD [7];

... ...

Example 6 The calling sequence for `tell` is

$$\text{tell}(f, [g_1, \cdots, g_m], [x_1, ..., x_n], 'r');$$

where f, g_i $(1 \le i \le m)$ are polynomials in $x_1, ..., x_n$ with integer coefficients. The output is one of *positive semi-definite, negative semi-definite* and *indefinite* which means f is positive (negative) semi-definite or indefinite under $g_1 > 0, ..., g_m > 0$. The last parameter $'r'$ is optional. If f is indefinite, r will be two sample points at which f has opposite signs.

We call

$$\text{tell}(adc - bd - bc, [a, b, c, d, 4b - a^2, b + cd - ac - ad, a - d - c], [a, b, c, d]);$$

the answer is "negative semi-definite". Thus, under constraints

$$\{a > 0, b > 0, c > 0, d > 0, 4b - a^2 > 0, b + cd - ac - ad > 0, a - d - c > 0\},$$

we have $adc - bd - bc \le 0$.

We call

$$\text{tell}(-adc + bd + bc, [a, b, c, d, 4b - a^2, b + cd - ac - ad, -a + d + c], [a, b, c, d], 'r');$$

the answer is "indefinite". Because $r = [[1, 3/10, 2, 2], -1], [[1, 2, 3, 1], 1]$, we know $-adc + bd + bc < 0$ at $(a, b, c, d) = (1, 3/10, 2, 2)$ and $-adc + bd + bc > 0$ at $(a, b, c, d) = (1, 2, 3, 1)$ (Note that these two sample points verify all the constraints).

It's not necessary to demonstrate the usage of all the other functions here. For the description of the usage of a certain function (say `func`) in DISCOVERER, just type in howtouse(`func`);.

Acknowledgment

The author wishes to thank Prof. L. Yang and Prof. X. R. Hou for their support and help.

References

[1] 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.

[2] 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.

[3] Bottema, O, et al, *Geometric Inequalities*, Wolters-Noordhoff Publishing, Groningen, Netherland, 1969.

[4] 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.

[5] Chou, S C, Gao, X S & McPhee, N, A combination of Ritt-Wu's method and Collins' method, *Proc. of CADE-12,* Springer-Verlag, 1994. pp.401-415.

[6] 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.

[7] Collins, G E & Hong, H, Partial cylindrical algebraic decomposition for quantifier elimination, *J. Symbolic Computation,* **12**, 299-328 (1991).

[8] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, Springer-Verlag, New York (1998).

[9] Andreas Dolzmann, Solving Geometric Problems with Real Quantifier Elimination, in: *Automated Deduction in Geometry*, X.-S. Gao, D. Wang & L. Yang (eds.), *LNAI 1669*, Springer-verlag, 1999. pp.14-29.

[10] Folke, E, Which triangles are plane sections of regular tetrahedra? *American Mathematics Monthly,* Oct., 1994, 788-789.

[11] L. Gonzalez-Vega, F. Rouillier, M.-F. Roy and G. Trujillo, Symbolic Recipes for Real Solutions, In *Some Tapas of Computer Algebra*, (Edited by A.M. Cohen *et al.*), pp. 121-167, Springer-Verlag, New York, (1999).

[12] Guy, R K & Nowakowski, R J, Monthly Unsolved Problems, 1969-1997, *American Mathematical Monthly,* Dec., 1997, 967-973.

[13] Mitrinovic, D S, Pecaric, J E, & Volenec, V, *Recent Advances in Geometric Inequalities,* Kluwer Academic Publishers, 1989.

[14] Volker Weispfenning, A new approach to quantifier elimination for real algebra, in: *Quantifier Elimination and Cylindrical Algebraic Decomposition*, B. F. Caviness & J. R. Johnson (eds.), Springer, Wien, New York, 1998, pp.376-392.

[15] Winkler, F, *Polynomial Algorithms in Computer Algebra,* Springer-Verlag/Wien, 1996.

[16] Xia, B C & Hou, X R, A Complete Algorithm for Counting Real Solutions of Polynomial Systems of Equations and Inequalities, Preprint.

[17] Xia, B C & Hou, X R, Automated determining the number of real solutions to triangular system with integral coefficients and its applications, in: *Proceedings of the Fourth Asian Technology Conference in Mathematics (Chinese Volume)*, L. Yang & S.X. Liang (eds.), Guangzhou, 1999, pp.6-13.

[18] Yang, L, Practical automated reasoning on inequalities: Generic programs for inequality proving and discovering, *Proceedings of the Third Asian Technology Conference in Mathematics,* W.-C. Yang et al.(eds), Springer-Verlag 1998, pp.24-35.

[19] Yang, L, Hou, X R & Xia, B C, Automated Discovering and Proving for Geometric Inequalities, in: *Automated Deduction in Geometry*, X.-S. Gao, D. Wang & L. Yang (eds.), *LNAI 1669*, Springer-verlag, 1999. pp.30-46.

[20] Yang, L, Hou, X R & Xia, B C, A complete algorithm for automated discovering of a class of inequality-type theorems, *Science in China*, to appear (Dec., 2000).

[21] Yang, L, Hou, X R & Zeng, Z B, A Complete Discrimination System for Polynomials, *Science in China,* Series **E39**:6(1996), 628-646.

[22] 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.

[23] Yang, L & Xia, B C, Explicit criterion to determine the number of roots in an interval of a polynomial, *Progress in Natural Science*, to appear (Nov., 2000).

[24] 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.