第 26 卷 第 7 期
2003 年 7 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 26 No. 7
July 2003

一类构造性几何不等式的机器证明

杨 路[1),3)] 夏时洪[2)]

[1)](广州大学软件研究所 广州 510405)

[2)](中国科学院计算技术研究所 北京 100080)

[3)](中国科学院成都计算机应用研究所 成都 610041)

摘 要 阐述了一个基于胞腔分解的不等式证明算法.据此算法编制的 Maple 通用程序能有效地处理含有根式的不等式型定理,对于 Bottema 等所著《几何不等式》一书中的大部分不等式定理的验证尤其高效.

关键词 自动证明;几何不等式;胞腔分解;半代数系统

中图法分类号 TP301

Automated Proving for a Class of Constructive Geometric Inequalities

YANG Lu[1),3)] XIA Shi-Hong[2)]

[1)](*Software Institute*, *Guangzhou University*, *Guangzhou* 510405)

[2)](*Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100080)

[3)](*Chengdu Institute of Computer Applications*, *Chinese Academy of Sciences*, *Chengdu* 610041)

Abstract An automated inequality-proving algorithm is presented based on a mixed method including a so-called cell-decomposition. That is implemented by a Maple program named "BOTTEMA" which can prove or disprove propositions in an extensive class of geometric and algebraic inequalities involving radicals. Most of the theorems in "Geometric Inequalities" writed by Bottema *et al*., can be proven efficiently in this way.

Keywords automated proving; geometric inequality; cell-decomposition; semi-algebraic system

1 不等式机器证明与半代数系统的不相容性

过去的二十多年里,等式型定理的机器证明的研究取得了长足进展,在效率上有很大提高,远远超出了不等式型定理证明的效率.本文的工作力图提高不等式型定理证明的效率,缩短两种类型定理证明效率之间的差距.

不等式型定理的机器证明一直被视为自动推理领域中一个困难的课题,主要原因在于其相关算法本质上依赖实代数与实几何,其计算复杂度会随着维数(参数的个数)的增加而快速增长.从算法的角度看,除了节约内存销耗,效率的改进也是非常有意义的.譬如,当要求成批地验证非平凡的命题时,一个低效的算法是不能在人们所能忍受的时间内解决问题的.关于这方面的若干进展参见文献[1～8].

我们将下列由多项式方程组和多项式不等式组构成的系统

收稿日期:2001-10-24;修改稿收到日期:2003-01-07.本课题得到国家"九七三"重点基础研究发展规划项目(NKBRSF-G1998030602)和中国科学院知识创新工程基金资助. **杨 路**,男,1936 年生,研究员,博士生导师,主要研究方向为计算机自动推理、智能软件技术. E-mail: luyang@gzhu.edu.cn 或 cdluyang@mail.sc.cninfo.net. **夏时洪**,男,1974 年生,博士,助理研究员,主要研究方向为计算机自动推理、智能软件技术. E-mail: xsh@ict.ac.cn.

$$\begin{cases} p_1(x_1,x_2,\cdots,x_s)=0 \\ p_2(x_1,x_2,\cdots,x_s)=0 \\ \quad\cdots \\ p_n(x_1,x_2,\cdots,x_s)=0 \\ g_1(x_1,x_2,\cdots,x_s)\geq 0,\ \cdots,\ g_r(x_1,x_2,\cdots,x_s)\geq 0 \\ h_1(x_1,x_2,\cdots,x_s)>0,\ \cdots,\ h_t(x_1,x_2,\cdots,x_s)>0 \\ q_1(x_1,x_2,\cdots,x_s)\neq 0,\ \cdots,\ q_m(x_1,x_2,\cdots,x_s)\neq 0 \end{cases} \tag{1}$$

叫做“半代数系统”. 这里 p_i,g_j,h_k,q_l 都是整系数实多项式. 这样的系统定义的点集叫做一个“半代数集”.

当变量数多于方程数时，我们将一部分变量看作自由变量(参数)，记为 $u_1,u_2,\cdots,u_d$，其余的 $x_1,x_2,\cdots,x_n$ 看作约束变量. 系统(1)可以改写为

$$\begin{cases} p_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)=0 \\ p_2(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)=0 \\ \quad\cdots \\ p_n(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)=0 \\ g_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0,\ \cdots, \\ \quad g_r(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0 \\ h_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0,\ \cdots, \\ \quad h_t(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0 \\ q_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0,\ \cdots, \\ \quad q_m(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0 \end{cases} \tag{2}$$

设 Φ 是系统(2)表述的语句，Φ_0 是一个多项式等式或多项式不等式，那么

$$\Phi\Rightarrow\Phi_0$$

就是一个实代数或实几何的命题. 显而易见，这个命题是真的当且仅当如下半代数系统

$$\Phi\wedge\neg\Phi_0$$

是不相容的(也就是说，它没有任何实解). 这里 $\neg\Phi_0$ 表示 Φ_0 的否定命题.

当命题的题设部分含有多项式方程组时，一个自然的想法是消去某些变量以降低维数. 应用一个重要的经典结果，即吴文俊的“整序原理”(well-ordering principle)，系统(2)的零点可以分解为有限个如下的三角型系统的零点之并：

$$\begin{cases} f_1(u_1,u_2,\cdots,u_d,\ x_1)=0 \\ f_2(u_1,u_2,\cdots,u_d,\ x_1,x_2)=0 \\ \quad\cdots \\ f_n(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)=0 \\ g_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0,\ \cdots, \\ \quad g_r(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0 \\ h_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0,\ \cdots, \\ \quad h_t(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0 \\ q_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0,\ \cdots, \\ \quad q_m(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0 \end{cases} \tag{3}$$

对于一个三角型的方程组，如果里面的参数 $u_1,u_2,\cdots,u_d$ 取定了常数值，那么 $x_1,x_2,\cdots,x_n$ 就可以一个个地依次解出来. 也就是说，至少容易得到这个方程组的数值解. 否则，一般说来，我们无法求得该方程组在通常意义下的解. 不用说方程组了，就是对单个的(带文字系数的)方程，一般也是办不到的. 所以，对于一般情况下的实代数与实几何的定理机器证明，虽有通用的算法和程序，但复杂度高，实用性较差. 不过，如果这个三角型系统中的每个方程 f_i 关于其“导元”x_i 都是一次或二次的，则显然所有的约束变量 $x_1,x_2,\cdots,x_n$ 都可以解出来，而表为 $u_1,u_2,\cdots,u_d$ 的根式函数.

碰巧的是，所谓“构造性几何定理”的前提部分的方程组中，每个方程关于其导元都是一次或二次的. 这样所有约束变量 $x_1,x_2,\cdots,x_n$ 都可以解为 $u_1,u_2,\cdots,u_d$ 的根式函数. 将解得的结果代入要证的结论中，譬如说，要想证的不等式 $g(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\leq 0$，我们得到含有根式的不等式

$$G(u_1,u_2,\cdots,u_d)\leq 0 \tag{4}$$

现在只需证明这个仅含变量 $u_1,u_2,\cdots,u_d$ 的不等式就行了.

对题设不等式约束条件

$$\begin{cases} g_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0,\ \cdots, \\ \quad g_r(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\geq 0 \\ h_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0,\ \cdots, \\ \quad h_t(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)>0 \\ q_1(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0,\ \cdots, \\ \quad q_m(u_1,u_2,\cdots,u_d,\ x_1,x_2,\cdots,x_n)\neq 0 \end{cases} \tag{5}$$

中的变量 $x_1,x_2,\cdots,x_n$ 作同样的替换，所有的约束条件也都变成了只含变量 $u_1,u_2,\cdots,u_d$ 但可能含有根式的不等式.

根据这一思路提出了不等式机器证明的“降维算法”并编制了程序 BOTTEMA. 该程序对构造性几何定理特别有效.

例 1. 设给定实数 $x,y,z,u_1,u_2,u_3,u_4,u_5,u_6$ 满足下述 15 个条件，

$$\begin{cases} (xy+yz+xz)^2u_1{}^2-x^3(y+z)(xy+xz+4yz)=0 \\ (xy+yz+xz)^2u_2{}^2-y^3(x+z)(xy+yz+4xz)=0 \\ (xy+yz+xz)^2u_3{}^2-z^3(x+y)(yz+xz+4xy)=0 \\ (x+y+z)(u_4{}^2-x^2)-xyz=0 \\ (x+y+z)(u_5{}^2-y^2)-xyz=0 \\ (x+y+z)(u_6{}^2-z^2)-xyz=0 \\ x>0,\ y>0,\ z>0 \\ u_1>0,\ u_2>0,\ u_3>0,\ u_4>0,\ u_5>0,\ u_6>0 \end{cases} \tag{6}$$

求证 $u_1+u_2+u_3\le u_4+u_5+u_6$.

若将 x,y,z 看作自由变量，则系统(6)中的 6 个方程构成一个三角列，其中第 i 个方程关于其导元 u_i 的次数为 2. 可解出 $u_1,u_2,\cdots,u_6$，而将上述命题转换为如下含多个根式的不等式：

假定 $x>0$, $y>0$, $z>0$，求证

$$\frac{\sqrt{x^3(y+z)(xy+xz+4yz)}}{xy+yz+xz}+\frac{\sqrt{y^3(x+z)(xy+yz+4xz)}}{xy+yz+xz}+\frac{\sqrt{z^3(x+y)(yz+xz+4xy)}}{xy+yz+xz}\le\sqrt{x^2+\frac{xyz}{x+y+z}}+\sqrt{y^2+\frac{xyz}{x+y+z}}+\sqrt{z^2+\frac{xyz}{x+y+z}}\tag{7}$$

这个不等式包含 3 个变量和 6 个根式，而式(6)中却含有 9 个变量.

杨路[9~11]提出的降维算法能够有效处理含参根式，并能最大限度地缩减维数. 根据降维算法编制的 Maple 通用程序“BOTTEMA”已在 PC 上成功实现. 该程序验证了包含上百个公开问题在内的上千个代数与几何的不等式. 在 Pentium IV/2200 上证明 Bottema 等人的专著[12]“Geometric Inequalities”中的 100 个基本不等式①总共所用的 CPU 时间仅 2s 多. 从后文可见这个算法可以应用于非常广泛的一类不等式.

本文第 2 节用实例阐明一些必要的术语；第 3 节扼要地描述一个证明不等式的算法，该算法能够有效地处理根式；第 4 节引进一些变量替换公式. 这些公式在证明一类关于三角形的不等式时有助于降低多项式的次数，而这个类涵盖了文献[12]中的大部分不等式；第 5 节列举 BOTTEMA 中的主要指令及其用法；第 6 节用更多的例子展示 BOTTEMA 的性能；第 7 节为结束语.

2　基本定义

在描述所谓的降维算法之前，我们引进并举例阐明一些定义.

定义 1.　假定 $l(x,y,z,\cdots)$ 和 $r(x,y,z,\cdots)$ 是关于 $x,y,z,\cdots$ 的代数连续函数，我们称

$$l(x,y,z,\cdots)\le r(x,y,z,\cdots)\ \text{或}\ l(x,y,z,\cdots)<r(x,y,z,\cdots)$$

为关于 $x,y,z,\cdots$ 的代数不等式，称 $l(x,y,z,\cdots)=r(x,y,z,\cdots)$ 为代数等式.

定义 2.　假定 Φ 是关于 $x,y,z,\cdots$ 的代数不等式或等式，我们称 $L(T)$ 为 Φ 的一个左多项式，如果

①$L(T)$ 是 T 的多项式，其系数是关于 $x,y,z,\cdots$ 的有理多项式.

②Φ 的左端是 $L(T)$ 的一个零点.

下面还有一项要求不是必要的，但有助于降低计算复杂度：

③在所有满足上述两条的多项式中，$L(T)$ 关于 T 的次数是最低的.

根据定义，如果 Φ 的左端是一零多项式，则有 $L(T)=T$. 同样可以类似地定义 Φ 的右多项式 $R(T)$.

定义 3.　假定 Φ 是关于 $x,y,z,\cdots$ 的代数不等式或等式，$L(T)$ 和 $R(T)$ 分别是 Φ 的左、右多项式. 令 $P(x,y,\cdots)$ 表示 $L(T)$ 与 $R(T)$ 关于 T 的 Sylvester 结式，称其为 Φ 的边界多项式，并将由 $P(x,y,\cdots)=0$ 定义的曲面称为 Φ 的边界曲面.

为了更有效地计算边界曲面，左、右多项式的定义是必需的. 譬如在例 1 中，设

$$f_1=(xy+yz+xz)^2u_1{}^2-x^3(y+z)(xy+xz+4yz),$$
$$f_2=(xy+yz+xz)^2u_2{}^2-y^3(x+z)(xy+yz+4xz),$$
$$f_3=(xy+yz+xz)^2u_3{}^2-z^3(x+y)(yz+xz+4xy),$$
$$f_4=(x+y+z)(u_4{}^2-x^2)-xyz,$$
$$f_5=(x+y+z)(u_5{}^2-y^2)-xyz,$$
$$f_6=(x+y+z)(u_6{}^2-z^2)-xyz,$$

则不等式(7)的左、右多项式可以通过逐次作结式计算求得：

$$\text{resultant}(\text{resultant}(\text{resultant}(u_1+u_2+u_3-T,f_1,u_1),f_2,u_2),f_3,u_3),$$
$$\text{resultant}(\text{resultant}(\text{resultant}(u_4+u_5+u_6-T,f_4,u_4),f_5,u_5),f_6,u_6).$$

去掉不含 T 的因子，我们有

$$L(T)=(xy+xz+yz)^8T^8-4(x^4y^2+2x^4yz+x^4z^2+4x^3y^2z+4x^3yz^2+x^2y^4+4x^2y^3z+4x^2yz^3+x^2z^4+2xy^4z+4xy^3z^2+4xy^2z^3+2xyz^4+y^4z^2+y^2z^4)(xy+xz+yz)^6T^6+\cdots,$$

$$R(T)=(x+y+z)^4T^8-4(x^3+x^2y+x^2z+xy^2+3xyz+xz^2+y^3+y^2z+yz^2+z^3)(x+y+z)^3\cdot T^6+2(16xyz^4+14xy^2z^3+14xy^3z^2+16xy^4z+$$

① 包含一些经典不等式，如 Euler 不等式，Finsler-Hadwiger 不等式，Gerretsen 不等式等等.

$14x^2yz^3+14x^2y^3z+14x^3yz^2+14x^3y^2z+16x^4yz+3x^6+5x^4y^2+5x^4z^2+5x^2y^4+5x^2z^4+5y^4z^2+5y^2z^4+21x^2y^2z^2+3y^6+3z^6+6x^5y+6x^5z+4x^3y^3+4x^3z^3+6xy^5+6xz^5+6y^5z+4y^3z^3+6yz^5)(x+y+z)^2T^4-4(x+y+z)(x^6-x^4y^2-x^4z^2+2x^3y^2z+2x^3yz^2-x^2y^4+2x^2y^3z+7x^2y^2z^2+2x^2yz^3-x^2z^4+2xy^3z^2+2xy^2z^3+y^6-y^4z^2-y^2z^4+z^6)(x^3+3x^2y+3x^2z+3xy^2+7xyz+3xz^2+y^3+3y^2z+3yz^2+z^3)T^2+(-6xy^2z^3-6xy^3z^2-6x^2yz^3-6x^2y^3z-6x^3yz^2-6x^3y^2z+x^6-x^4y^2-x^4z^2-x^2y^4-x^2z^4-y^4z^2-y^2z^4-9x^2y^2z^2+y^6+z^6+2x^5y+2x^5z-4x^3y^3-4x^3z^3+2xy^5+2xz^5+2y^5z-4y^3z^3+2yz^5)^2.$

在 Pentium IV/2200 上用 Maple 8 作逐次结式计算求得 $L(T)$ 与 $R(T)$ 的时间分别为 0.14s 和 0.03s. 另费时 38.31s 后求得边界多项式，去掉非零因子和重因子后，其次数为 100，且含有 2691 项.

当然，我们可以通过移项将式(7)等价变形为

$$\frac{\sqrt{x^3(y+z)(xy+xz+4yz)}}{xy+yz+xz}+\frac{\sqrt{y^3(x+z)(xy+yz+4xz)}}{xy+yz+xz}+\frac{\sqrt{z^3(x+y)(yz+xz+4xy)}}{xy+yz+xz}-\sqrt{x^2+\frac{xyz}{x+y+z}}-\sqrt{y^2+\frac{xyz}{x+y+z}}\leq\sqrt{z^2+\frac{xyz}{x+y+z}} \tag{8}$$

在同一台机器上(内存为 256Mb)使用类似的 Maple 程序

```
f := u1 + u2 + u3 - u4 - u5 - T;
for i to 5 do f := resultant(f, f.i, u.i) od;
```

却未能求得式(8)的左多项式. 运行 5h 后未出结果，只好打断.

也可以使用下述程序，不用左、右多项式的概念而直接计算边界多项式，

```
f := u1 + u2 + u3 - u4 - u5 - u6;
for i to 6 do f := resultant(f, f.i, u.i) od;
```

但情况并无改善. 运行 5h 后未出结果，只好打断. 实践表明，从左、右多项式计算边界多项式的这种“分治法”(devide-and-conquer)要比直接计算边界多项式的效率高出许多倍.

例 2. 给定一个关于 x,y,z 的代数不等式

$$m_a+m_b+m_c\leq 2s \tag{9}$$

其中

$$m_a=\frac{1}{2}\sqrt{2(x+y)^2+2(x+z)^2-(y+z)^2},$$

$$m_b=\frac{1}{2}\sqrt{2(y+z)^2+2(x+y)^2-(x+z)^2},$$

$$m_c=\frac{1}{2}\sqrt{2(x+z)^2+2(y+z)^2-(x+y)^2},$$

$$s=x+y+z,$$

并且 $x>0$，$y>0$，$z>0$，计算其左多项式、右多项式及边界多项式.

设

$$f_1=4m_a{}^2+(y+z)^2-2(x+y)^2-2(x+z)^2,$$

$$f_2=4m_b{}^2+(x+z)^2-2(y+z)^2-2(x+y)^2,$$

$$f_3=4m_c{}^2+(x+y)^2-2(x+z)^2-2(y+z)^2$$

并作逐次结式计算

$$\text{resultant}(\text{resultant}(\text{resultant}(m_a+m_b+m_c-T, f_1, m_a), f_2, m_b), f_3, m_c),$$

于是得到式(9)的左多项式

$$T^8-6(x^2+y^2+z^2+xy+yz+zx)T^6+9(x^4+2xyz^2+y^4+2xz^3+2x^3y+z^4+3y^2z^2+2y^2zx+2y^3z+2yz^3+3x^2z^2+2x^3z+2x^2yz+2xy^3+3x^2y^2)T^4-(72x^4yz+78x^3yz^2+4x^6+4y^6+4z^6+12xy^5-3x^4y^2-3x^2z^4-3x^2y^4-3y^4z^2-3y^2z^4-3x^4z^2-26x^3y^3-26x^3z^3-26y^3z^3+12xz^5+12y^5z+12yz^5+12x^5z+12x^5y+84x^2y^2z^2+72xyz^4+72xy^4z+78xy^3z^2+78xy^2z^3+78x^2yz^3+78x^3y^2z+78x^2y^3z)T^2+81x^2y^2z^2(x+y+z)^2 \tag{10}$$

由于该不等式的右端没有根式，因此很容易求出右多项式，即

$$T-2(x+y+z) \tag{11}$$

计算式(10)，(11)关于 T 的结式，得到

$$(144x^5y+144x^5z+780x^4y^2+1056x^4yz+780x^4z^2+1288x^3y^3+3048x^3y^2z+3048x^3yz^2+1288x^3z^3+780x^2y^4+3048x^2y^3z+5073x^2y^2z^2+3048x^2yz^3+780x^2z^4+144xy^5+1056xy^4z+3048xy^3z^2+3048xy^2z^3+1056xyz^4+144xz^5+144y^5z+780y^4z^2+1288y^3z^3+780y^2z^4+144yz^5)(x+y+z)^2.$$

去掉一个非零因子 $(x+y+z)^2$，得到边界曲面

$$144x^5y+144x^5z+780x^4y^2+1056x^4yz+780x^4z^2+1288x^3y^3+3048x^3y^2z+3048x^3yz^2+1288x^3z^3+780x^2y^4+3048x^2y^3z+5073x^2y^2z^2+3048x^2yz^3+780x^2z^4+144xy^5+1056xy^4z+3048xy^3z^2+3048xy^2z^3+1056xyz^4+144xz^5+144y^5z+780y^4z^2+1288y^3z^3+780y^2z^4+144yz^5=0 \tag{12}$$

3　降维算法概述

虽然降维算法对更广泛的一类不等式都适用，但本文要处理的是具下述形式的命题，

$$\Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_s \Rightarrow \Phi_0,$$

其中 $\Phi_0, \Phi_1, \cdots, \Phi_s$ 是关于 $x, y, z, \cdots$ 的代数不等式，题设 $\Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_s$ 要么定义一个开集①，要么定义一个开集及其部分或全部边界.

例 1 可以写成 $(x>0) \wedge (y>0) \wedge (z>0) \Rightarrow (7)$，其中题设 $(x>0) \wedge (y>0) \wedge (z>0)$ 确定了参数空间 R^3 中的一个开集因此例 1 属于我们描述的这个类. 例 2 也属此类. 该类覆盖了 Bottema 等的著作[12] 以及 Mitrinovic 等的著作[13]"Recent Advances in Geometric Inequalities"中的大部分不等式. 事实上，例 2 是一个由变元 x, y, z 表述的几何不等式[12].

当结论 Φ_0 是一个非严格的不等式($\leq$型)时，我们采用下述的步骤来处理.(如果 Φ_0 是一严格的不等式($<$型)，我们还需要验证方程 $l_0(x, y, \cdots) - r_0(x, y, \cdots) = 0$ 在题设条件下是否有实解，其中 $l_0(x, y, \cdots)$ 及 $r_0(x, y, \cdots)$ 分别为 Φ_0 的左端和右端).

1. 找出不等式 $\Phi_0, \Phi_1, \cdots, \Phi_s$ 的边界曲面.

2. 边界曲面将参数空间划分为有限个胞腔，我们从中选出所有的连通开集 $D_1, D_2, \cdots, D_k$，而忽略低维的胞腔. 在选出的每一个连通开集中至少选取一个测试点，记为 $(x_v, y_v, \cdots) \in D_v$, $v=0,1,\cdots,k$. 这一步骤可以采用不完全的"柱形代数剖分"(忽略低维胞腔). 因为是在开集中选取，可以保证所选的每个测试点都是有理点.

3. 逐个对每一个测试点 $(x_1, y_1, \cdots)$, $\cdots$, $(x_k, y_k, \cdots)$ 验证命题的正确性. 命题成立当且仅当对所有的测试点命题皆成立.

下面来证明这个方法的正确性.

令 $l_\mu(x, y, \cdots)$, $r_\mu(x, y, \cdots)$ 及 $P_\mu(x, y, \cdots) = 0$ 分别表示不等式 Φ_μ 的左端、右端及边界曲面，并令

$$\delta_\mu(x, y, \cdots) \stackrel{\text{def}}{=} l_\mu(x, y, \cdots) - r_\mu(x, y, \cdots),$$

其中 $\mu = 0, 1, \cdots, s$.

因为所有 $\delta_\mu(x, y, \cdots)$ 的零点的集合是一个闭集，故其补集 Δ 是一个开集. 另一方面，集合

$$D \stackrel{\text{def}}{=} D_1 \cup D_2 \cup \cdots \cup D_k$$

恰好是所有 $P_\mu(x, y, \cdots)$ 的零点的集合的补集.

因为 $\delta_\mu(x, y, \cdots)$ 的零点都是 $P_\mu(x, y, \cdots)$ 的零点，所以 $D \subset \Delta$. 令 $\Delta_1, \Delta_2, \cdots, \Delta_t$ 表示 Δ 的所有连通支，其中每一个都是连通开集. 每一个 Δ_λ 必然包含 D 中的一点，因为开集不可能全部由某些 $P_\mu(x, y, \cdots)$ 的零点构成. 假定 Δ_λ 包含 D 的某个连通支 D_i 的一点，则有 $D_i \subset \Delta_\lambda$，因为 Δ 的两个不连通的分支不可能同时与 D_i 相交. 由上述算法的第二步知 D_i 包含一个测试点 $(x_i, y_i, \cdots)$，因此每个 Δ_λ 至少包含一个测试点.

因此 $\delta_\mu(x, y, \cdots)$ 在整个 Δ_λ 上保持与 $\delta_\mu(x_{i_\lambda}, y_{i_\lambda}, \cdots)$ 相同的符号，其中 $(x_{i_\lambda}, y_{i_\lambda} \cdots)$ 是 Δ_λ 中的一个测试点，$\lambda = 1, 2, \cdots, t$; $\mu = 0, 1, \cdots, s$. 否则，如果存在一点 $(x', y', \cdots) \in \Delta_\lambda$ 使得 $\delta_\mu(x', y', \cdots)$ 的符号与 $\delta_\mu(x_{i_\lambda}, y_{i_\lambda}, \cdots)$ 的符号相反，作一连接两点 $(x', y', \cdots)$ 和 $(x_{i_\lambda}, y_{i_\lambda} \cdots)$ 的路径 Γ 使得 $\Gamma \subset \Delta_\lambda$，那么必然有一点 $(\bar{x}, \bar{y}, \cdots) \in \Gamma$ 使得 $\delta_\mu(\bar{x}, \bar{y}, \cdots) = 0$，得出了矛盾结果！

令 $A \cup B$ 表示题设所定义的集合，其中 A 是由

$$(\delta_1(x, y, \cdots) < 0) \wedge \cdots \wedge (\delta_s(x, y, \cdots) < 0),$$

定义的开集，它由 Δ 的某些连通支及 $\delta_0(x, y, \cdots)$ 的某些实零点组成，比方说，$A = Q \cup S$，其中 $Q = \Delta_1 \cup \Delta_2 \cup \cdots \cup \Delta_j$, S 是 $\delta_0(x, y, \cdots)$ 的某些实零点集合. B 是 A 的全部或部分边界，由 $\delta_\mu(x, y, \cdots)$ 的某些实零点组成，$\mu = 1, 2, \cdots, s$.

最后逐个对 A 中的所有测试点验证不等式 $\delta_0 < 0$ 是否成立. 如果有一个测试点使得 $\delta_0 > 0$，则命题不成立；否则，$\delta_0 < 0$ 在 Q 上成立. 因为 Q 的每一个连通支都含有一个测试点，又 δ_0 在 Δ_λ 保持符号不变，因此由连续性可知 $\delta_0 \leq 0$ 在 A 上成立，同样在 $A \cup B$ 上成立，从而命题为真.

上述步骤有时可以简化. 当结论 Φ_0 属于所谓"CGR"类，在做第三步时，我们只需要比较 Φ_0 的左多项式与右多项式在测试点处的值.

定义 4.　我们称一个代数不等式属于 CGR 类，如果它的左端是其左多项式的最大根，而其右端是其右多项式的最大根.

显然，在例 1 中，因为所有根式前的符号均为正，所以不等式(7)的左右两端分别是其左多项式 $L(T)$ 右多项式 $R(T)$ 的最大根. 因而不等式(7)属于 CGR 类，我们只需验证 $L(T)$ 的最大根是否小于或等于 $R(T)$ 的最大根. 从精确计算的角度看，这比判断两个复杂根式的大小容易一些.

如果不等式只含有单层根式，通过移项总是能转换为等价的属于 CGR 类的不等式. 事实上，文献[12,13]中的大部分不等式，包含本文的例子，都属

① 不一定连通.

于 CGR 类. 详情请参见文献[9].

4 关于三角形的不等式

文献[12]中讨论的数百个不等式绝大部分是关于三角形的. 此后,在各类出版物中又出现了上千个三角形不等式.

在讨论关于单个三角形的几何不等式时,通常采用几何不变量而不是笛卡儿坐标作为全局变量. 令 a,b,c 表示三角形各边的长度,s 表示半周长,即 $\frac{1}{2}(a+b+c)$. 又 x,y,z 分别表示 $s-a,s-b,s-c$. 此外,令 A,B,C 表示三角形的三内角,S 表示面积,R 表示外接圆半径,r 表示内切圆半径. r_a,r_b,r_c 表示旁切圆半径. h_a,h_b,h_c 表示高,m_a,m_b,m_c 表示中线的长度,w_a,w_b,w_c 表示内角平分线的长度等等.

人们常常选择 x,y,z 作为独立变量,而将其余变量视为约束变量. 从降低多项式次数的角度来考虑,或许存在着更好的选择.

一个代数不等式 $\Phi(x,y,z)$ 可以看成一个关于三角形的几何不等式,如果

①$x>0$, $y>0$, $z>0$;

②Φ 的左端 $l(x,y,z)$ 和右端 $r(x,y,z)$ 都是齐次的;

③$l(x,y,z)$ 与 $r(x,y,z)$ 的次数相同.

①意味着三角形的两边之和大于第三边;②③意味着相似变换不改变命题的真伪. 例如不等式(9)的左端 $m_a+m_b+m_c$ 和右端 $2s$ 都是关于 x, y, z 的一次齐次函数.

更进一步,假定 $\Phi(x,y,z)$ 的左端 $l(x,y,z)$ 和右端 $r(x,y,z)$ 是关于 x, y, z 的对称函数. 则将 $l(x,y,z)$ 与 $r(x,y,z)$ 中的 x, y, z 替换为如下的 x',y',z' 不会改变命题的真伪:$x'=\rho x, y'=\rho y, z'=\rho z$ 且 $\rho>0$.

显然 $\Phi(x',y',z')$ 的左多项式 $L(T, x',y',z')$ 和右多项式 $R(T, x',y',z')$ 关于 x',y',z' 都是对称的,因此它们都可以用 x',y',z' 的初等对称函数来表达,即

$$H_l(T, \sigma_1,\sigma_2,\sigma_3) = L(T, x',y',z'),$$
$$H_r(T, \sigma_1,\sigma_2,\sigma_3) = R(T, x',y',z'),$$

其中 $\sigma_1=x'+y'+z'$, $\sigma_2=x'y'+y'z'+z'x'$, $\sigma_3=x'y'z'$.

令 $\rho=\sqrt{\frac{x+y+z}{xyz}}$,我们有 $x'y'z'=x'+y'+z'$,即 $\sigma_3=\sigma_1$. 进而令

$$s=\sigma_1(=\sigma_3), \quad p=\sigma_2-9,$$

我们可以将 $L(T, x',y',z')$ 和 $R(T, x',y',z')$ 转换为 T, p, s 的多项式,记之为 $F(T,p,s)$ 与 $G(T,p,s)$. 特别地,如果 F 和 G 均只含有 s 的偶次项,那么它们还可以进一步转换为 T, p 及 q 的多项式,这里设 $q=s^2-4p-27$. 通常后者的次数与项数都比 $L(T,x,y,z)$ 和 $R(T,x,y,z)$ 中的次数与项数低. 因此我们用 p,s 或 p,q 表达的边界曲面来剖分 (p,s)-平面或 (p,q)-平面,而不是对 R^3 作这种剖分. 对于很大一部分几何不等式,这样的变换都会极大地降低计算复杂度. 如下一例选自文献[12].

例 3. 令 w_a,w_b,w_c 与 s 分别表示三角形的内角平分线与半周长,求证

$$w_bw_c+w_cw_a+w_aw_b \le s^2.$$

易知

$$w_a = 2\frac{\sqrt{x(x+y)(x+z)(x+y+z)}}{2x+y+z},$$
$$w_b = 2\frac{\sqrt{y(x+y)(y+z)(x+y+z)}}{2y+x+z},$$
$$w_c = 2\frac{\sqrt{z(x+z)(y+z)(x+y+z)}}{2z+x+y},$$

且 $s=x+y+z$. 按照前述的连续结式计算,我们得到一个次数为 20 的含 557 项的左多项式,右多项式为 $T-(x+y+z)^2$,边界多项式 $P(x, y, z)$ 的次数为 15 且含 136 项.

然而,如果我们将左、右多项式均用 p, q 来表达,可得

$$\begin{aligned}&(9p+2q+64)^4T^4-32(4p+q+27)(p+8)(4p^2+\\&pq+69p+10q+288)(9p+2q+64)^2T^2-\\&512(4p+q+27)^2(p+8)^2(9p+2q+64)^2T+\\&256(4p+q+27)^3(p+8)^2(-1024-64p+\\&39p^2-128q-12pq-4q^2+4p^3+p^2q)\end{aligned}$$

及 $T-4p-q-27$,边界多项式为

$$\begin{aligned}Q(p,q)=&5600256p^2q+50331648p+33554432q+\\&5532160p^3+27246592p^2+3604480q^2+\\&22872064pq+499291p^4+16900p^5+2480q^4+\\&16q^5+143360q^3+1628160pq^2+22945p^4q+\\&591704p^3q+11944p^3q^2+2968p^2q^3+\\&242568p^2q^2+41312pq^3+352pq^4,\end{aligned}$$

其次数为 5 且只含 20 项.

5 指令及用法

作为一个证明器,整个程序都是用 Maple 实现

的，而不需额外的软件包.

用 BOTTEMA 验证不等式，我们只需要键入证明指令，机器就会自动证明而无需人工干预. 如果命题为真，则输出“The inequality holds”，反之输出“The inequality does not hold”并给出反例. 有三种基本的证明指令：prove，xprove 及 yprove.

prove：证明一个关于三角形的几何不等式或与之等价的代数不等式.

调用方式：

prove(ineq)；

prove(ineq, ineqs)；

参数：

ineq——要证明的几何不等式，须用后面表中列出的几何不变量来表达.

ineqs——一组表示条件①的不等式，也须用后面表中列出的几何不变量来表达.

描述：

①命令‘prove’对于待证结论‘ineq’为‘≤’或‘≥’类型的关于三角形的几何不等式有效，条件部分‘ineqs’定义一个开集或一个开集及其全部或部分边界；并且‘ineq’和‘ineqs’必须用后表所列几何不变量的有理函数或根式表达.

②命令‘prove’对于题设和结论(即‘ineqs’和‘ineq’)都是关于 x,y,z 的齐次代数不等式，它们只含有理函数与根式，而且 $x>0$，$y>0$，$z>0$ 的命题有效. 如前所述，这样的命题等价于关于一个三角形的几何不等式.

③下面关于三角形几何不变量的列表是可以扩充的.

三角形的几何不变量(可扩充)：

a,b,c：	三角形 ABC 的三边长；
s：	$s:=(a+b+c)/2$，半周长；
x,y,z：	$x:=s-a$，$y:=s-b$，$z:=s-c$；
S：	三角形的面积；
R：	外接圆半径；
r：	内切圆半径；
ra,rb,rc：	旁切圆半径；
ha,hb,hc：	高；
ma,mb,mc：	中线；
wa,wb,wc：	内角平分线；
p：	$p:=4*r*(R-2*r)$；
q：	$q:=s\hat{}2-16*R*r+5*r\hat{}2$；
HA,HB,HC：	垂心到三顶点的距离；
IA,IB,IC：	内心到三顶点的距离；
A,B,C：	三角形的三个内角；
$\sin(A),\sin(B),\sin(C)$：	内角的正弦值；
$\cos(A),\cos(B),\cos(C)$：	内角的余弦值；
$\tan(A),\tan(B),\tan(C)$：	内角的正切值；
$\cot(A),\cot(B),\cot(C)$：	内角的余切值；
$\sec(A),\sec(B),\sec(C)$：	内角的正割值；
$\csc(A),\csc(B),\csc(C)$：	内角的余割值；

$\sin(A/2),\sin(B/2),\sin(C/2)$；

$\cos(A/2),\cos(B/2),\cos(C/2)$；

$\tan(A/2),\tan(B/2),\tan(C/2)$；

$\cot(A/2),\cot(B/2),\cot(C/2)$；

$\sec(A/2),\sec(B/2),\sec(C/2)$；

$\csc(A/2),\csc(B/2),\csc(C/2)$；

例子.

```
>read bottema;
>prove(a^2+b^2+c^2>=4*sqrt(3)*S+
  (b-c)^2+(c-a)^2+(a-b)^2);
            The inequality holds
>prove(A>=B, [a>=b]);
            The inequality holds.
```

xprove：证明一个代数不等式，要求变量取正值.

调用方式：

xprove(ineq)；

xprove(ineq, ineqs)；

参数：

ineq——待证的代数不等式，其变量取正值.

ineqs——一组代数不等式条件，变量取正值.

描述：

①命令‘xprove’要求待证的代数不等式‘ineq’具有形式‘≤’或‘≥’，代数不等式条件‘ineqs’定义一个开集或一个开集及其全部或部分边界.

②条件和结论都必须用有理函数和根式来表达.

③‘ineq’中的所有变量都假定为正，而不需要在条件中重新输入.

例子.

```
>read bottema;
>xprove(sqrt(u^2+v^2)+sqrt((1-u)^2+
  (1-v)^2)>=sqrt(2),[u<1,v<1]);
            The inequality holds
```

① 用方括号，括号内各条件之间用逗号隔开.

```
>f:=(x+1)^1/3)+sqrt(y-1)+
  x*y+1/x+1/y^2:
>xprove(f>=42496/10000,[y>1]);
            The inequality holds
>xprove(f>=42497/10000,[y>1]);
            with a counter example
```

$$\left[x=\frac{29}{32},\ y=\frac{294117648}{294117647}\right],$$

```
        The inequality does not hold.
```

yprove:证明一般的代数不等式.

调用方式:

yprove(ineq);

yprove(ineq, ineqs);

参数:

ineq——待证的代数不等式.

ineqs——一组代数不等式表示的条件.

描述:

①命令'yprove'对形如'≤'或'≥'的代数不等式'ineq'有效,'ineqs'定义一个开集或一个开集及其全部或部分边界.

②前提和结论都必须用有理函数和根式来表达.

例子.

```
>read bottema;
>f:=x^6*y^6+6*x^6*y^5-6*x^5*y^6+
  15*x^6*y^4-36*x^5*y^5+15*x^4*
  y^6+20*x^6*y^3-90*x^5*y^4+90*
  x^4*y^5-20*x^3*y^6+15*x^6*y^2-
  120*x^5*y^3+225*x^4*y^4-120*
  x^3*y^5+15*x^2*y^6+6*x^6*y-90*
  x^5*y^2+300*x^4*y^3-300*x^3*
  y^4+90*x^2*y^5-6*x*y^6+x^6-
  36*x^5*y+225*x^4*y^2-400*x^3*
  y^3+225*x^2*y^4-36*x*y^5+y^6-
  6*x^5+90*x^4*y-300*x^3*y^2+
  300*x^2*y^3-90*x*y^4+6*y^5+
  15*x^4-120*x^3*y+225*x^2*y^2-
  120*x*y^3+15*y^4-20*x^3+
  90*x^2*y-90*x*y^2+20*y^3+16*
  x^2-36*x*y+16*y^2-6*x+6*y+1:
>yprove(f>=0);
            The inequality holds
```

6 若干实例

著名的 Janous 不等式[14]在 1986 年作为一个公开问题提出,1988 年获解.

例 4. 设 m_a, m_b, m_c 和 $2s$ 分别表示三角形三边上的中线长及半周长,证明

$$\frac{1}{m_a}+\frac{1}{m_b}+\frac{1}{m_c}\geq\frac{5}{s}.$$

这个不等式难在它的左端隐含了三个根式,BOTTEMA 在证明这个命题前会自动将几何命题转换为与之等价的代数命题. 在 Pentium IV/2200 上证明这一命题①用时 3.94s.

下一例子是 America Mathematical Monthly 1986 年 93 期 299 页上的一个公开问题 E.3146*.

例 5. 令 a,b,c 和 s 分别表示三角形的三边长和半周长,判定下述命题的真伪.

$$2s(\sqrt{s-a}+\sqrt{s-b}+\sqrt{s-c})\leq$$
$$3(\sqrt{bc(s-a)}+\sqrt{ca(s-b)}+\sqrt{ab(s-c)}).$$

在同一机器上验证这一命题费时 11.36s.

下一例子是数学通讯上的一个公开问题,编号 169.

例 6. 设 r_a,r_b,r_c 和 w_a,w_b,w_c 分别表示三角形的内切圆半径及内角平分线长,判定下述命题的真伪.

$$\sqrt[3]{r_ar_br_c}\leq\frac{1}{3}(w_a+w_b+w_c).$$

也就是说, r_a,r_b,r_c 的几何平均值是否小于或等于 w_a,w_b,w_c 的算术平均值.

上述不等式的右端也隐含了三个根式,BOTTEMA 验证这一猜想的时间为 11.95s. 而验证文献[10]中的下述猜想用时 63.55s.

例 7. 令 a,b,c,m_a,m_b,m_c 及 w_a,w_b,w_c 分别表示三角形的边长、中线长及内角平分线长,判定下述命题的真伪.

$$am_a+bm_b+cm_c\leq\frac{2}{\sqrt{3}}(w_a^2+w_b^2+w_c^2).$$

1985 年 Garfunkel 在 Crux Math 上首次提出下述猜想,随后再次作为公开问题出现在文献[13,15]中.

例 8. 令 A,B,C 为三角形的三内角,判定下述命题的真伪.

$$\cos\left(\frac{B-C}{2}\right)+\cos\left(\frac{C-A}{2}\right)+\cos\left(\frac{A-B}{2}\right)\leq$$
$$\frac{1}{\sqrt{3}}\left(\cos\left(\frac{A}{2}\right)+\cos\left(\frac{B}{2}\right)+\cos\left(\frac{C}{2}\right)+\sin A+\sin B+\sin C\right).$$

① 本节各例所用时间均为在 Pentium IV/2200 机上使用 Maple 8 的运行数据.

判定这一命题用时 24.98s.

为了回答 Erdös 提出的问题，Oppenheim 研究了下述不等式[13].

例 9.　设 a，b，c 和 m_a，m_b，m_c 分别为三角形的三边长和中线长，如果 $c=\min\{a, b, c\}$，那么

$$2m_a+2m_b+2m_c \le 2a+2b+(3\sqrt{3}-4)c.$$

由于题设部分含有条件 $c=\min\{a, b, c\}$，因此我们键入指令

prove(2 * ma + 2 * mb + 2 * mc <= 2 * a + 2 * b + (3 * sqrt(3) − 4) * c,[c <= a,c <= b]);

验证用时 211.24s. 如果我们键入的指令仅为

prove(2 * ma + 2 * mb + 2 * mc <= 2 * a + 2 * b + (3 * sqrt(3) − 4) * c);

则输出“The inequality does not hold”，并给出反例 $[a=203, b=706, c=505]$.

下面这个正半定判定问题源自文献[8].

例 10.　假定 $x>0$，$y>0$，$z>0$，证明

$$\begin{aligned}&2187(y^4z^4(y+z)^4(2x+y+z)^8+\\&x^4z^4(x+z)^4(x+2y+z)^8+\\&x^4y^4(x+y)^4(x+y+2z)^8)-\\&256(x+y+z)^8(x+y)^4(x+z)^4(y+z)^4\ge 0.\end{aligned}$$

这个多项式展开后共有 201 项，系数的绝对值最大为 181394432. 通常判定一个多项式是否半正定不是一件容易的事，由于我们利用了这个多项式的对称性与齐次性，降低了次数与维数，故验证这一命题只需 0.59s.

著名的 Euler 不等式 $R\ge 2r$ 和不等式 $m_a\ge w_a$ 常被用来阐述不等式证明的各种不同算法. 这里我们比较 $R-2r$ 与 m_a-w_a 的大小.

例 11.　设 R，r 表示三角形的外接圆半径与内切圆半径，m_a，w_a 表示三角形一边上的中线与内角平分线，求证

$$m_a-w_a\le R-2r.$$

用时 3.11s.

当然我们的程序不仅仅适用于关于三角形的不等式，譬如可以用来证明所谓的“Ptolemy Inequality”. 这次我们使用笛卡儿坐标而不是几何不变量.

例 12.　给定平面上的 4 点 A,B,C,D，用 AB，AC，AD，BC，BD，CD 表示各点两两之间的距离，求证

$$AB\cdot CD+BC\cdot AD\ge AC\cdot BD \tag{13}$$

令 $A=\left(-\frac{1}{2},0\right)$，$B=(x,y)$，$C=\left(\frac{1}{2},0\right)$，$D=(u,v)$，将式(13)化为

$$\begin{aligned}&\sqrt{\left(-\frac{1}{2}-x\right)^2+y^2}\sqrt{\left(\frac{1}{2}-u\right)^2+v^2}+\\&\sqrt{\left(x-\frac{1}{2}\right)^2+y^2}\sqrt{\left(-\frac{1}{2}-u\right)^2+v^2}\ge\\&\sqrt{(x-u)^2+(y-v)^2}\end{aligned} \tag{14}$$

我们只需键入指令“yprove(%)”，其中%代表式(14)，运行 3.67s 后输出“The inequality holds”.

根据我们的记录，在 Pentium IV/2200 上使用 Maple 8 运行上述各例所用 CPU 时间以及测试点的数目如表 1 所示.

表 1　各例所用 CPU 时间以及测试点数目

例	CPU 时间(s)	测试点个数
例 1	108.58	23
例 2	0.02	1
例 3	0.03	1
例 4	3.94	12
例 5	11.36	135
例 6	11.95	4
例 7	63.55	3
例 8	24.98	121
例 9	211.24	287
例 10	0.59	2
例 11	3.11	22
例 12	3.67	48

表 1 所列的时间包括求左、右多项式、边界多项式、胞腔分解及逐个验证所有样本点(即测试点)所花的时间.

7　结束语

(1)这个程序对题设与结论都是由有理函数与根式表达的不等式型定理有效，并且要求结论部分是“≤”或“≥”类型的不等式，而题设部分定义一个开集或一个开集及其全部或部分边界.

(2)目前的程序不适用于除有理函数及根式以外的其他代数函数.

(3)证明过程完全自动化，无需人工干预.

(4)本程序对关于三角形的不等式尤其高效，如果用几何不变量来表达最佳.

(5)关于本程序在全局最优化问题中的应用，参见文献[9,10,16]

我们在第 1 节中已经提到，本项工作的目的是给出不等式型定理机器证明的实用高效算法和实现该算法的通用程序. 像 BOTTEMA 这样专用于不等式机器证明的高效能通用软件此前尚未见报道. 前面提到的相近工作[1~4]等只是举少数例子说明他们的方法怎样用于证明几何不等式. 所关注的是能

不能做的问题,尚未及效率的高低,而且文中所举例子都比较容易.所以无法与 BOTTEMA 作恰当的比较.

参 考 文 献

1 Chou S C, Gao X S, Arnon D S. On the mechanical proof of geometry theorems involving inequalities. In: Hoffmann C M ed. Advances in Computing Research, 6. Greenwith: JAI Press, 1992. 139～181

2 Chou S C, Gao X S, McPhee N. A combination of Ritt-Wu's method and Collins' method. In: Bundy A ed. Proceedings of CADE-12, Berlin: Springer-Verlag, 1994. 401～415

3 Dolzmann A, Sturm T, Weispfenning V. A new approach for automatic theorem proving in real geometry. Journal of Automated Reasoning, 1998, 21(3): 357～380

4 Dolzmann A, Sturm T, Weispfenning V. Real quantifier elimination in practice. In: Matzat B H, Greuel G-M, Hiss G eds. Algorithmic Algebra and Number Theory. Berlin: Springer-Verlag, 1998. 221～247

5 Wu W-T. On a finiteness theorem about problem involving inequalities. System Science and Mathematical Science, 1994, 7(2): 193～200

6 Wu W-T. On global-optimization problems. In: Proceedings of ASCM'98, Lanzhou, 1998. 135～138

7 Yang L, Hou X R, Xia B C. Automated discovering and proving for geometric inequalities. In: Gao X S, Wang D, Yang L eds. Automated Deduction in Geometry. LNAI 1669. Berlin: Springer-Verlag, 1999. 30～46

8 Yang L, Hou X R, Xia B C. A complete algorithm for automated discovering of a class of inequality-type theorems. Science in China Series F, 2001, 44(1): 33～49

9 Yang L. Recent advances in automated theorem proving on inequalities. Journal of Computer Science & Technology, 1999, 14(5): 434～446

10 Yang L, Xia S H. An inequality-proving program applied to global optimization. In: Yang W C *et al* eds. Proceedings of the Asian Technology Conference in Mathematics. Blacksburg: ATCM, Inc., 2000. 40～51

11 Yang L, Zhang J. A Practical program of automated proving for a class of geometric inequalities. In: Richter-Gebert J, Wand D eds. Automated Deduction in Geometry, LNAI 2061. Berlin: Springer-Verlag, 2001. 41～57

12 Bottema O *et al*. Geometric Inequalities. Groningen, The Netherlands: Wolters-Noordhoff Publishing, 1969

13 Mitrinovic D S, Pecaric J E, Volenec V. Recent Advances in Geometric Inequalities. Dordrent: Kluwer Academic Publishers, 1989

14 Janous W, Problem 1137. Crux Mathematicorm, 1986, 12(2～3): 79, 177

15 Kuang J C. Applied Inequalities, 2nd ed. Changsha: Hunan Educational Publishing House, 1993(in Chinese)
(匡继昌.常用不等式.长沙:湖南教育出版社,1993)

16 Yang L. A symbolic algorithm for global optimization and finiteness principle. In: Lin D D *et al* eds. Mathematics and Mathematics Mechanization. Jinan: Shandong Educational Publishing House, 2001, 210～220(in Chinese)
(杨 路.全局优化的符号算法与有限核原理.见:林东岱等主编.数学与数学机械化.济南:山东教育出版社,2001.210～220)

17 Shan Z. Geometric Inequality in China. Jiangsu Educational Publishing House, 1996 (in Chinese)
(单 墫.几何不等式在中国.南京:江苏教育出版社,1996)

YANG Lu, born in 1936, professor in computer science at Chengdu Institute of Computer Applications, Chinese Academy of Sciences and at Guangzhou University concurrently. His major research interests include automated theorem proving, symbolic computation and intelligent software technology.

XIA Shi-Hong, born in 1974, received Ph. D. degree in computer software and theory from Graduate School of the Chinese Academy of Sciences in 2002. Now He is an assistant researcher in computer science at Institute of Computing Technology, Chinese Academy of Sciences. His major research interests include automated theorem proving, symbolic computation and computer simulation.