

多项式等式型几何定理的可读证明

江建国^{1),2)} 张景中²⁾ 王晓京²⁾

¹⁾(辽宁师范大学数学学院 辽宁 大连 116029)

²⁾(中国科学院成都计算机应用研究所 成都 610041)

摘 要 目前的智能几何软件都使用基于搜索法的定理证明器作为推理引擎,其主要缺点是不能可读地证明涉及到几何量代数运算的几何定理,这极大地限制了智能几何软件的实际应用. 对一类结论为几何量多项式等式的几何定理,文中提出了一种能给出可读证明的启发式搜索算法. 该算法通过引入多项式的变形操作算子——标准项代换,把证明结论为多项式等式 $g=0$ 的几何定理转化为寻找从 g 到 0 的标准项代换序列的搜索问题. 采用 Lisp 语言实现了该算法,并做了 30 个结论为几何量等式的几何定理的推理实验. 实验结果表明算法具有较高的推理效率.

关键词 几何定理机器证明;搜索法;标准项代换;启发函数;可读证明

中图法分类号 TP181

Readable Proving for Geometric Theorems of Polynomial Equality Type

JIANG Jian-Guo^{1),2)} ZHANG Jing-Zhong²⁾ WANG Xiao-Jing²⁾

¹⁾(Mathematics School, Liaoning Normal University, Dalian, Liaoning 116029)

²⁾(Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041)

Abstract Currently the automated reasoning engineer of the intelligent geometry software is limited in the theorem prover based on search method. A drawback is that it can not give the readable proving for geometric theorems involving algebraic computation. A heuristic search algorithm using the standard item substitute is presented in this paper, which can give readable proving for a class of geometric theorems as long as its conclusions are polynomial equalities about geometric quantities, such as the length of segment, the degree of angle. The algorithm is implemented with Lisp and tested with 30 nontrivial geometry theorems. The experimental results show that it is more efficient than ever before.

Keywords automated geometry theorem proving; search method; item substitute; heuristic function; readable proof

1 引 言

几何定理机器证明一直是自动推理领域内的一个前沿基本课题. 近年来,其研究已经取得了突破进展^[1-4]. 通常几何定理机器证明的方法分为 3 大类:

代数法、向量法和人工智能法. 著名的吴法^[1-4]、Gröbner 基法^[5]、例证法^[6]、数值并行法^[2]等都是代数法;面积法^[3]、括号代数法^[7]、Clifford 代数法^[8]等都是典型的向量法;人工智能法又分为两类:搜索法和逻辑法. Gelernter 的后推法^[9]、Nevins 的前推法^[10]和周咸青等人的推理数据库法^[4]都属于搜索

法, Wos 等人的 OTTER 证明器^[11]以及 Balbiani 的项重写^[12]都是逻辑法. 有关这方面的研究工作可参考文献^[13].

一般地, 从证明效率上看, 代数法的效率最高, 向量法其次, 人工智能法的效率最低; 而从证明可读性上看, 次序却正好相反, 人工智能法的可读性最好, 向量法其次, 代数法的可读性最差. 特别是搜索法还能给出易于学生理解的传统风格的可读证明, 这使其适合于教育应用. 目前, 采用搜索法作为推理引擎的智能几何软件已经走进了教育领域^[14-16]. 本文主要研究基于搜索法的几何定理证明器.

搜索法使用受限的一阶逻辑来表示知识. 通常, 为了减小搜索空间和提高搜索效率, 搜索法限制使用许多有关代数运算和序关系的谓词^[10]. 这种限制带来了明显的缺点, 失去了知识表示法的充分性将会降低解题能力, 导致其不能证明无法表示的几何定理.

几何中的许多定理都会涉及到几何量的代数运算. 特别是有一类几何定理的结论是几何量的多项式等式, 例如平面几何中的斯图尔特定理、托勒密定理等, 本文将这类几何定理称为多项式等式型几何定理. 注意, 这里所说的多项式等式型几何定理类与代数法中通常所定义的“等式型”^[17]几何定理类不同, 它不是指使用坐标法可以化成代数等式的几何定理, 而是指结论是用几何量(如长度、角角度和面积)的多项式所表示的几何定理.

多项式等式型几何定理一直被视为很难使用搜索法给出可读证明的. 其主要原因在于, 如果添加有关几何量代数运算的谓词和搜索规则, 将会引起搜索空间的组合爆炸, 导致搜索效率急剧下降, 从而不能在人们所能容忍的时间内证明定理. 本文的工作力图增强基于搜索法的几何定理证明器的解题能力, 使其能自动地、可读地证明多项式等式型几何定理.

本文第 2 节给出几个有关于多项式的基本概念; 在第 3 节中, 引入多项式的变形操作算子——标准项代换, 给出使用标准项代换可读证明多项式等式型几何定理的例子; 第 4 节给出证明多项式等式型几何定理的启发式搜索算法; 第 5 节使用 Lisp 语言实现本文提出的算法并给出实验结果; 第 6 节给出进一步的研究方向.

2 基本概念

本节给出几个有关多项式的基本概念, 相关的

基础知识可参考文献^[18].

2.1 多项式

定义 1. 由 n 个变元 x_1, x_2, \dots, x_n 组成的项就是变元幂的乘积 $x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$, 这里, 指数 $\alpha_1, \alpha_2, \dots, \alpha_n$ 都是非负整数.

为了记号简便, 使用 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbf{Z}_{\geq 0}^n$ 表示一个 n 元非负整数组, 用 \mathbf{x}^α 来表示项 $x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$. 特别地当 $\alpha = (0, \dots, 0)$ 时, 称 \mathbf{x}^α 为常数项, 此时有 $\mathbf{x}^\alpha = 1$.

定义 2. 一个实系数的 n 变元 x_1, x_2, \dots, x_n 的多项式 e 就是 n 变元 x_1, x_2, \dots, x_n 项的有限线性组合, 其组合的系数都是实数. 记为 $e = \sum_{\alpha} a_{\alpha} \mathbf{x}^{\alpha}$, $a_{\alpha} \in \mathbf{R}$, 这里, 求和符号 \sum 是对有限个 n 元非负整数组 α 进行求和.

为了能够让计算机识别和处理多项式, 需要添加表示多项式的谓词 pol, 多项式 e 在系统中表示为 (pol e). 例如 $2a^2b - ab^2$ 在系统中表示成

(pol((2((a 2)(b 1)))(-1((a 1)(b 2))))).

特别地, 0 多项式在系统中表示成 (pol((0 nil))). 多项式等式 $f = g$ 在系统中表示为 (eq (pol f) (pol g)), 其中“eq”是表示“相等”的谓词.

2.2 标准型

一个多项式通常可以写成多种不同的等价形式, 其原因在于加法和乘法运算满足交换律和结合律. 例如, 多项式 $ab + cd$ 还可以写成 $ab + dc, cd + ab, cd + ba$ 等 8 种不同形式. 为了便于计算机存储和处理多项式, 需要在多种等价表示形式中, 确定出一种能够唯一地表示多项式的形式. 为此, 需要引入“项序”的概念, 以便能够唯一地表示多项式, 本文主要使用一种称为“字典序”的项序.

定义 3. 设 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ 和 $\beta = (\beta_1, \beta_2, \dots, \beta_n) \in \mathbf{Z}_{\geq 0}^n$, 如果向量差 $\alpha - \beta \in \mathbf{Z}^n$ 的最左边的非零元是正的, 则称 $\alpha >_{\text{lex}} \beta$. 如果 $\alpha >_{\text{lex}} \beta$, 则称在字典序 $>_{\text{lex}}$ 下项 \mathbf{x}^α 大于项 \mathbf{x}^β , 记为 $\mathbf{x}^\alpha >_{\text{lex}} \mathbf{x}^\beta$.

定义 4. 按字典序 $>_{\text{lex}}$ 的降序对多项式 e 的项进行重新排序后, 所得到的多项式称为 e 的标准型.

字典序 $>_{\text{lex}}$ 要求首先对变元进行排序, 然后再对项进行排序. 在字典序 $>_{\text{lex}}$ 下, 任何一个多项式都可以唯一地写成标准型. 例如, 按字典序 $a >_{\text{lex}} b >_{\text{lex}} c >_{\text{lex}} d$, 多项式 $dc + ba$ 可以写成标准型 $ab + cd$.

3 标准项代换

一般地, 在证明多项式等式型几何定理时, 先要

证明或推导出一些几何量等式事实,然后再想方设法地从这些等式中推导出结论等式.在推导过程中,主要使用等量公理和一些简单的代数运算率.我们知道最经常使用的等量公理是“等量代换公理”,它是指“一个量总可以用它的等量去代换”,其它的等量公理都可以由等量反身公理和等量代换公理推导出来.下面,我们把多项式等式改写成一种类似于等量代换公理的特殊形式,作为多项式恒等变形的推理规则.

3.1 项代换

定义 5. 非常数项 x^β 在多项式 $e = \sum_a a_a x^a$ 中的出现是指,存在 e 的某个能整除 x^β 的项 x^δ ,使得 $x^\delta = x^\gamma \cdot x^\beta$. 用符号 $e(x^\beta)$ 表示项 x^β 在 e 中的出现.

定义 6. 形如 $x^a = f$ 的等式称为项代换,这里 x^a 是非常数项, f 是标准型的多项式,且项 x^a 不在 f 中出现.本文中把项代换写成 $x^a \Leftarrow f$ 以区别于等式 $x^a = f$.

用符号 $e(x^a/f)$ 表示使用 f 替换掉 e 中所有出现的项 x^a 后而得到的表达式.如果 $x^a \Leftarrow f$ 是项代换,那么根据等词的替换公理可知,有 $e(x^a) = e(x^a/f)$. 换句话说,使用项代换可以把一个多项式重写成另一个和它相等的表达式.

每个多项式等式都可以改写成多个项代换.例如,在字典序 $a >_{\text{lex}} b >_{\text{lex}} c$ 下,多项式等式 $a+b=c$ 可改写成 3 个项代换 $c \Leftarrow a+b$, $b \Leftarrow -a+c$ 和 $a \Leftarrow -b+c$. 分别使用这 3 个项代换可将多项式 $ac+bc-c^2$ 重写成表达式 $a(a+b)+b(a+b)-(a+b)^2$, $ac+(-a+c)c-c^2$ 和 $(-b+c)c+bc-c^2$.

3.2 标准化

设 $e(x^a)$ 是一个标准型多项式, $x^a \Leftarrow f$ 是一个项代换.可以使用这个项代换把标准型多项式 $e(x^a)$ 重写成表达式 $e(x^a/f)$,再经过一些代数运算, $e(x^a/f)$ 还可进一步化简成一个新的标准型多项式.例如,表达式 $a(a+b)+b(a+b)-(a+b)^2$, $ac+(-a+c)c-c^2$ 和 $(-b+c)c+bc-c^2$ 都可以进一步化简成 0 多项式.我们把化简过程中使用到的一些代数运算写成推理规则,称为标准化规则.共有 6 条标准化规则:

- (1) 分配规则. 如果 $f = \sum_a a_a x^a$, 那么 $(a_\beta x^\beta) \cdot f = \sum_a (a_a \cdot a_\beta) \cdot (x^a \cdot x^\beta)$.
- (2) 结合规则. $x^a \cdot x^\beta = x^{a+\beta}$.
- (3) 交换规则. 如果 $x^\beta >_{\text{lex}} x^a$, 那么 $a_a x^a + a_\beta x^\beta = a_\beta x^\beta + a_a x^a$.

$$(4) \text{ 合并规则. } a_a x^a + a_\beta x^a = (a_a + a_\beta) x^a.$$

$$(5) \text{ 零化规则. } 0 \cdot x^a = 0.$$

$$(6) \text{ 消零规则. } a_a x^a + 0 = a_a x^a.$$

表达式 $e(x^a/f)$ 依次经过分配、结合、交换、合并、零化和消零等标准化规则作用,可以化简成标准型多项式,此过程称为标准化过程 std . 经过项代换 $x^a \Leftarrow f$ 和标准化过程 std 后,标准型多项式 $e(x^a)$ 将变换成另一个标准型多项式 $std(e(x^a/f))$,称 $std(e(x^a/f))$ 为 $e(x^a)$ 的子多项式,称 $e(x^a)$ 为 $std(e(x^a/f))$ 的父多项式.

定义 7. 把进行项代换后再进行标准化的多项式变形操作称为标准项代换.

3.3 算例

使用标准项代换可以给出多项式等式型几何定理的可读证明.下面以证明“欧拉定理”为例进行说明.

例 1(欧拉定理). 已知:如图 1 所示,一条直线上的 4 点顺次为 A, B, C 和 D .

求证: $AC \cdot BD = AB \cdot CD + BC \cdot AD$.

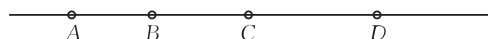


图 1 欧拉定理

在这条直线上一共有 6 条线段,分别是 AB, AC, AD, BC, BD 和 CD ,不妨设线段的字典序为 $AB >_{\text{lex}} AC >_{\text{lex}} AD >_{\text{lex}} BC >_{\text{lex}} BD >_{\text{lex}} CD$. 从图 1 中可知 6 条线段之间满足一组等式关系:

$$AC = AB + BC \quad (1)$$

$$AD = AB + BD \quad (2)$$

$$AD = AC + CD \quad (3)$$

$$AD = AB + BC + CD \quad (4)$$

$$BD = BC + CD \quad (5)$$

把这些等式都改写成项代换,可得到一组项代换:

$$AC \Leftarrow AB + BC \quad (1)$$

$$AB \Leftarrow AC - BC \quad (2)$$

$$BC \Leftarrow -AB + AC \quad (3)$$

$$AD \Leftarrow AB + BD \quad (4)$$

$$AB \Leftarrow AD - BD \quad (5)$$

$$BD \Leftarrow -AB + AD \quad (6)$$

$$AD \Leftarrow AC + CD \quad (7)$$

$$AC \Leftarrow AD - CD \quad (8)$$

$$CD \Leftarrow -AC + AD \quad (9)$$

$$AD \Leftarrow AB + BC + CD \quad (10)$$

$$AB \Leftarrow AD - BC - CD \quad (11)$$

$$BC \Leftarrow -AB + AD - CD \quad (12)$$

$$CD \Leftarrow -AB + AD - BC \quad (13)$$

$$BD \Leftarrow BC + CD \quad (14)$$

$$BC \Leftarrow BD - CD \quad (15)$$

$$CD \Leftarrow -BC + BD \quad (16)$$

要使用这些项代换证明结论等式:

$$AC \cdot BD = AB \cdot CD + BC \cdot AD,$$

只需证明:

$$AB \cdot CD - AC \cdot BD + AD \cdot BC = 0$$

即可. 下面给出把 $AB \cdot CD - AC \cdot BD + AD \cdot BC$ 重写成 0 的项代换序列, 等式后面带括号的数字是使用项代换的标号.

$$\begin{aligned} & AB \cdot CD - AC \cdot BD + AD \cdot BC \\ &= -AB \cdot BD + AB \cdot CD + AD \cdot BC - BC \cdot BD \end{aligned} \quad (1)$$

$$= AB \cdot BC - AB \cdot BD + AB \cdot CD \quad (4)$$

$$= 0 \quad (14)$$

显然, 有很多个项代换序列都可以把多项式 $AB \cdot CD - AC \cdot BD + AD \cdot BC$ 重写成 0, 比如项代换序列 (1)(4)(15), (2)(4)(16), (5)(7)(16) 等等.

4 启发式搜索算法

基于搜索法的几何定理证明器做前向搜索, 当达到推理不动点时^[4,19], 能搜索出可由推理规则找到的图形中的全部几何量等式, 比如等角、等长、等比、内角和、勾股和等等, 这些等式构成了推导结论等式 $g=0$ 的已知条件集 E . 问题的关键是, 怎样才能让计算机从 E 中推导出结论等式 $g=0$ 呢?

从 3.3 节的算例可以看出, 把 E 中的等式都改写成项代换, 可以得到一个有限的项代换集 I . 如果能使用这些项代换对多项式 g 做恒等变形, 最终把 g 恒等变形为 0 多项式, 即找到一条从 g 到 0 的项代换序列, 那么, 也就给出了结论等式 $g=0$ 的可读证明.

基于上述想法, 可设计一个能给出多项式等式型几何定理可读证明的搜索算法. 问题的初始节点是多项式 g , 目标节点是 0 多项式, 节点扩展算是 I 中的项代换, $g, 0$ 和 I 确定了一个搜索空间. 从初始节点到目标节点的路径, 就是把 g 重写成 0 所进行的项代换序列.

一般情况下, I 中有很多个标准项代换. 如果盲目地使用项代换去扩展节点, 会导致搜索空间的爆炸, 使算法根本就不能在合理的时间范围内搜索到目标节点. 如何才能使算法有效地搜索目标节点呢?

若能找到一种方法排列待扩展的节点, 使算法能选择“最有希望”的节点加以扩展, 那么将会显著地提高算法的搜索效率. 为此, 需要一个能测量待扩展多项式 p 的“希望”的启发函数 $h(p)$, 用启发值 $h(p)$ 来刻画把 p 重写成 0 的“难易”程度.

本文定义启发函数为 $h(p) = i(p) \times d(p) \times v(p)$, 其中 $i(p)$, $d(p)$ 和 $v(p)$ 分别为多项式 p 的项数、最高次数和变元个数. 例如, 如果 $p = AB \cdot CD + BC \cdot DA - AC \cdot BD$, 那么 $h(p) = 3 \times 2 \times 6 = 36$. $h(p)$ 的值越大, 表明 p 越难以重写成 0; 反之, $h(p)$ 的值越小, 表明 p 越易于重写成 0. 算法使用 $h(p)$ 来选择最有希望的节点进行扩展. 下面给出能证明结论为多项式等式的几何定理的启发式搜索算法.

算法 1.

输入: 一个结论为多项式等式 $g=0$ 的几何定理.

输出: 如果能搜索到从 g 到 0 多项式的项代换序列, 则输出几何定理的可读证明; 否则, 输出指定的无法证明信息.

1. 基于搜索法的几何定理证明器做前向搜索, 当达到推理不动点时, 证明器搜索出一个几何量等式集 E .

2. 把 E 中的几何量等式都改写成项代换, 得到一个项代换集 I .

3. 把 $open$ 表和 $closed$ 表都初始化为空. $open$ 表用于存储未扩展节点, $closed$ 表用于存储已扩展节点.

4. 计算 $h(g)$, 生成初始节点 $i = (g, nil, h(g), nil, I)$, 将 i 放入 $open$ 表中.

5. 如果 $open$ 表为空, 则输出指定的无法证明信息, 算法结束.

6. 取出 $open$ 表中的第 1 个节点 n (即具有最小 h 值的节点), 并把 n 放入 $closed$ 表中.

7. 如果节点 n 中的子多项式 q 是 0 多项式, 即节点 n 是目标节点 j , 则输出几何定理的可读证明, 算法结束.

8. 令 T 为节点 n 的项代换集 S_n , 即 $T = S_n$.

9. 如果 T 为空, 则转步 5.

10. 令 $T = T - \{\varphi\}$, φ 是 T 中的第 1 个项代换.

11. 如果 φ 的项在 p 中不出现, 则将 φ 丢弃, 转步 9.

12. 用 φ 对 p 做恒等变形, 生成 p 的子标准型多项式 q .

13. 如果 q 在 $open$ 表或 $closed$ 表中的节点的子多项式中出现过, 即 q 是一个已经搜索到的多项式, 则将 q 丢弃, 转步 9.

14. 计算 $h(q)$, 生成 n 的子节点 $m = (q, p, h(q), \varphi, S_n - \{\varphi\})$.

15. 按照 $h(q)$ 的大小, 将子节点 m 插入到有序表 $open$ 中, 转步 9.

在上面的算法 1 中, 搜索空间中的每个节点 n 都是一个由子多项式 q 、父多项式 p 、子多项式的启发值 $h(q)$ (该值也是节点 n 的启发值, 即 $h(n) = h(q)$)、从 p 到 q 所使用的项代换 φ 以及可用于子多

项式 q 的项代换集 S 所组成的 5 元组, 即 $n=(q, p, h(q), \varphi, S)$. 特别地, 初始节点 $i=(g, nil, h(g), nil, I)$, 这里的两个“ nil ”分别表示初始节点 i “没有父多项式 p ”和“没有从父多项式到 g 所使用的项代换 φ ”, i 的子多项式 q 是初始多项式 g , 即结论等式中的左端多项式, I 是可用于 g 的初始项代换集; 而在目标节点 $j=(0, p, 0, \varphi, S)$ 中, j 的子多项式 q 一定是目标多项式 0 , 即结论等式中的右端多项式, 并且 q 的启发值 $h(q)$ 一定为 0 .

算法 1 的搜索空间是一颗树 T , 树的根节点是初始节点 i . $open$ 表中的节点都是待扩展的节点, 而 $closed$ 表中的节点都是已扩展的节点, 有些是树 T 的叶子节点, 有些是树 T 的分支节点. 算法 1 的步 13 确保了树 T 的所有节点中的子多项式 q 都是互不相同的, 都是初始多项式 g 的不同等价形式. 因此, 树 T 中不存在循环的节点序列.

树 T 中的每个节点都附有一个可用的项代换集. 从算法 1 中可见, 如果使用项代换 φ 将节点 n 扩展到节点 m , 即节点 m 是节点 n 的子节点, 那么节点 m 的项代换集 S_m 与节点 n 的项代换集 S_n 满足: $S_m=S_n-\{\varphi\}$, 也就是说当算法向下扩展树 T 时, 节点的项代换集是越来越小的. 这说明树 T 的最长路径的长度不会大于 $|I|$ (I 是根节点的项代换集, 是个有限集), 即树 T 的高度小于或等于 $|I|$, 从而可知树 T 是一颗有限树. 因此, 算法 1 一定是可终止的.

如果树 T 中存在从初始节点 i 到目标节点 j 的路径, 那么算法 1 会在步 7 搜索到目标节点 j , 然后根据节点间的父-子关系, 在 $closed$ 表中回溯到初始节点 i , 找到从初始多项式 g 到目标多项式 0 的项代换序列, 输出几何定理的可读证明; 否则, 算法 1 会搜遍树 T 的所有节点, 直到没有可扩展的节点时, 即 $open$ 为空时, 算法才会在步 5 结束搜索, 输出指定的无法证明信息.

5 实现与实验

为了验证算法 1 的可行性和推理效率, 我们采用 Lisp 语言实现了该算法, 并在 CPU 主频为 Pentium IV 1GHz, 内存为 256MB 的 PC 机上进行了实验. 表 1 是 30 个运行实例的实验结果表. 表 1 中的第 1 列是测试题序号; 第 2 列是证明器所找到图形中等式的个数; 第 3 列是初始项代换集 I 中项代换的个数; 第 4 列是搜索空间中的已经展开但还

没有搜索的节点数; 第 5 列是已经搜索的节点数; 第 6 列是在找到证明时搜索到的层数; 第 7 列是完成证明所运行的时间. 限于篇幅, 在此只列举 4 个实验用例(见表 1 中的黑体行).

表 1 30 个多项式等式型几何定理的运行实验结果						
题号	等式	项代换	未搜索节点	已搜索节点	搜索深度	运行时间/s
1	60	190	87	43	3	2.19
2	27	83	48	4	3	0.43
3	20	60	37	5	4	0.50
4	44	120	79	6	5	1.48
5	54	166	80	9	5	1.13
6	52	154	750	43	6	50.62
7	45	86	6	2	1	0.19
8	49	161	73	4	3	1.74
9	25	75	54	6	5	0.87
10	24	76	44	4	3	0.68
11	31	103	48	8	5	0.79
12	3	9	5	5	3	0.39
13	14	37	61	19	6	0.51
14	4	11	12	4	3	0.24
15	6	18	13	5	4	0.25
16	7	19	17	5	3	0.19
17	21	65	48	9	6	0.59
18	10	28	42	30	4	0.29
19	13	33	26	9	6	2.70
20	11	31	63	19	5	0.40
21	24	59	20	4	3	0.36
22	10	28	45	11	8	1.00
23	23	54	17	13	5	4.07
24	15	39	67	17	3	0.40
25	11	30	21	4	3	0.41
26	7	21	180	125	6	2.43
27	26	61	70	24	5	1.41
28	47	124	89	7	5	23.48
29	6	18	13	5	4	0.30
30	71	155	41	282	6	25.01

例 2(五角星内角和). 任意五角星的 5 个内角之和为 180° (图 2).

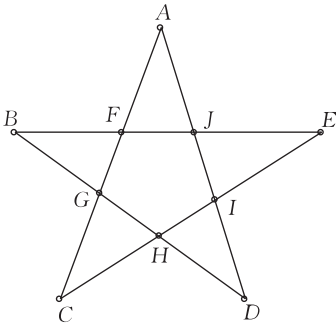


图 2 五角星内角和

此题是表 1 中的第 1 题. 已知条件是 5 条两两相交的直线 AC, AD, BE, BD 和 CE 构成的五角星, 要证的结论是 $\angle CAD+\angle DBE+\angle ECA+\angle ADB+\angle BEC=180^\circ$. 证明器达到推理不动点时发现了 60 个角和等式, 改写后得到 190 个项代换. 在搜索过程

中展开了 130 个节点, 搜索 43 个节点后, 在第 3 层就达到目标节点. 运行 2.19s 后输出可读证明.

例 3. 已知: 设在角 A 为直角的 $\triangle ABC$ 的各边上向外侧作正方形 $BADE$, $CBFG$ 和 $ACHI$, 连结 EF , GH (图 3). 求证: $EF^2 + GH^2 = 5BC^2$.

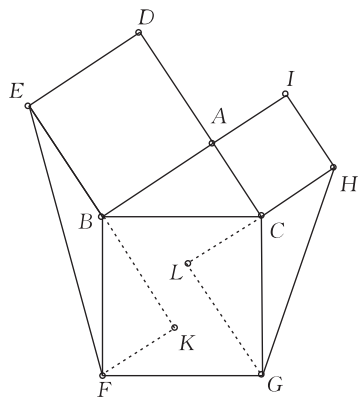


图 3

此题是表 1 中的第 30 题. 需要添加辅助线, 过点 F 作 EB 的垂线, 与 EB 的延长线相交于点 K ; 过点 G 作 HC 的垂线, 与 HC 的延长线相交于点 L . 证明器需要运行 19.61s 后才能达到推理不动点, 找到 71 个线段等式, 改写后得到 155 个项代换. 在搜索过程中展开了 323 个节点, 搜索 41 个节点后, 在第 6 层达到目标节点. 总共运行 25.01s 后输出可读证明.

例 4 (斯图尔特定理). 已知: D 是 $\triangle ABC$ 的边 BC 上的一点 (图 4).

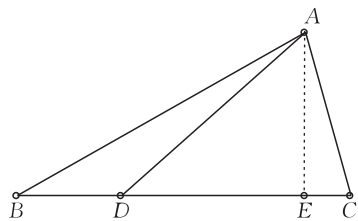


图 4 斯图尔特定理

求证: $AB^2 \cdot DC + AC^2 \cdot BD - AD^2 \cdot BC = BC \cdot DC \cdot BD$.

此题是表 1 中的第 26 题. 需要添加一条辅助线, 过点 A 作 BC 的垂线, 垂足为点 E . 证明器达到推理不动点时发现了 7 个线段等式, 改写后得到 21 个项代换. 在搜索过程中展开了 305 个节点, 搜索了 125 个节点, 在第 6 层到达目标节点. 运行 2.43s 后输出可读证明.

例 5 (托勒密定理). 圆内接四边形 $ABCD$ 的两组对边乘积的和 $AB \cdot CD + BC \cdot DA$ 等于它的对角线的乘积 $AC \cdot BD$ (图 5).

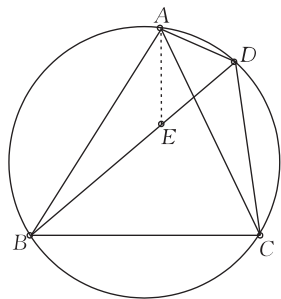


图 5 托勒密定理

此题是表 1 中的第 17 题. 需要如图作一条辅助线 AE , 使得 $\angle BAE = \angle CAD$. 证明过程中需要用到比例等式. 可以通过引入新变元将比例等式改写成项代换. 例如, $a/b = c/d = k$, k 是引入的新变元, 写成项代换是 $a \leftarrow bk$, $bk \leftarrow a$, $c \leftarrow dk$ 和 $dk \leftarrow c$. 如果 k_1 和 k_2 是两个新引入的变元, 并且还互为倒数, 那么还需要添加项代换 $k_1 k_2 \leftarrow 1$. 证明器达到推理不动点时, 发现 1 条线段等式和 20 条比例等式, 改写后得到了 65 个项代换. 在搜索过程中展开了 57 个节点, 搜索 9 个节点后, 在第 6 层达到目标节点. 共运行 0.59s 后输出可读证明.

6 结束语

本文提出的算法 1 能可读地证明结论为几何量多项式等式的几何定理. 由于该算法是一种启发式搜索算法, 故具有较高的推理效率. 但是, 算法还存在着一定的局限性, 对那些需要通过求解方程才能推导出结论等式的几何定理, 算法还不能给出可读证明. 从这方面可以看出, 算法 1 并不是一个完全算法. 另外, 由于基于搜索法的几何定理证明器目前还不能自动添加辅助线, 在证明需要添加辅助线的几何定理时, 还须人工添加辅助线, 这极大地降低了推理的智能性. 寻找添加辅助线的智能算法或策略将是更加具有挑战性的研究方向.

致 谢 审稿人的宝贵建议极大地提高了本文的质量!

参 考 文 献

- [1] Wu W-T. On the decision problem and the mechanization of theorem proving in elementary geometry. *Journal of Systems Science and Mathematical Science*, 1978, 21(16): 157-179
- [2] Zhang J, Yang L, Deng M. The parallel numerical method of mechanical theorem proving. *Theoretical Computer Science*, 1990, 74(3): 253-271

- [3] Chou S, Gao X, Zhang J. Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems. Singapore: World Scientific, 1994
- [4] Chou S, Gao X, Zhang J. A deductive database approach to automated geometry theorem proving and discovering. *Journal of Automated Reasoning*, 2000, 25(3): 219-246
- [5] Buchberger B, Collins G, Kutzler B. Algebraic methods for geometric reasoning. *Annual Review of Computer Sciences*, 1985, 3(19): 85-119
- [6] Hong J. Can geometry be proved by an example? *Scientia Sinica*, 1986, 29(8): 824-834
- [7] Richter-Gebert J. Mechanical theorem proving in projective geometry. *Annul of Mathematics and Artificial Intelligence*, 1995, 13(1-2): 139-172
- [8] Li H. Some applications of Clifford algebra to geometries// *Proceedings of the ADG'98*. LNAI 1669. Berlin: Springer-Verlag, 1998: 156-179
- [9] Gelernter H, Hansen J, Loveland D. Empirical explorations of the geometry theorem proving machine//Feigenbaum E, Feldman J. *Computers and Thought*. New York: McGraw-Hill, 1963: 153-163
- [10] Nevins A. Plane geometry theorem proving using forward chaining. *Artificial Intelligence*, 1975, 6(1): 1-23
- [11] McCharen J, Overbeek R, Wos L. Problems and experiments for and with automated theorem-proving programs. *IEEE Transactions on Computers*, 1976, C-25(8): 773-782
- [12] Balbiani P, Del Cerro L F. Affine geometry of collinearity and conditional term rewriting//*Proceedings of the Term Re-*
writing. LNCS 909. Berlin: Springer-Verlag, 1995: 196-213
- [13] Chou S, Gao X. Automated reasoning in geometry//Alan A, Voronkov A. *Miland: Handbook of Automated Reasoning*. Elsevier Science Publishers, 2001: 708-749
- [14] Gao Xiao-Shan, Zhang Jing-Zhong, Chou S C. *Geometry Expert*. Beijing: China Children's Press, 1998(in Chinese)
(高小山, 张景中, 周咸青. 几何专家. 北京: 中国少年儿童出版社, 1998)
- [15] Li Chun-Zhong, Zhang Jing-Zhong. *Super Sketchpad*. Beijing: Beijing Normal University Press, 2004(in Chinese)
(李传中, 张景中. 超级画板. 北京: 北京师范大学出版社, 2004)
- [16] Noboru M, Kurt V. GRAMY: A geometry theorem prover capable of construction. *Journal of Automated Reasoning*, 2004, 32(1): 3-33
- [17] Wu Wen-Jun. *Basic Principles of Mechanical Theorem Proving in Geometries*. Beijing: Science Press, 1984(in Chinese)
(吴文俊. 几何定理机器证明的基本原理. 北京: 科学出版社, 1984)
- [18] Cox D, Little J, O' Shea D. *Ideals, Varieties and Algorithms*. New York: Springer-Verlag, 1992
- [19] Jiang Jian-Guo, Zhang Jing-Zhong. The automated geometry reasoning network based on equivalent class reasoning. *Pattern Recognition and Artificial Intelligence*, 2006, 19(5): 617-622(in Chinese)
(江建国, 张景中. 基于等价类推理的几何自动推理网. 模式识别与人工智能, 2006, 19(5): 617-622)



JIANG Jian-Guo, born in 1969, Ph. D., lecturer. His major research interests include automated reasoning and intelligent software technology.

ZHANG Jing-Zhong, born in 1938, professor, Ph. D. supervisor, member of the Chinese Academy of Sciences. His major research interests include automated reasoning and intelligent software technology.

WANG Xiao-Jing, born in 1955, professor, Ph. D. supervisor. His major research interests include information security and intelligent software technology.

Background

The project is supported by the National Basic Research 973 Program of China under grant No. 2004CB318000. The long-term goal is to build intelligent geometry software for school, which can be helpful to teachers and students. Any proofs produced by the software must be stated with the common ontology of Euclidean geometry — The axiom geometry taught in schools.

There are mainly two approaches to proving geometry theorems using computers: the artificial intelligence (AI) approach, like search methods, point elimination methods, and the algebraic computation (AC) approach, like Wu's methods, Gröber base methods. Generally speaking, the AC approaches are decision procedures and are more powerful. The AI approaches are not decision procedures. But proofs produced by the AI approaches are more readable than the AC

approaches. Thus the AI approaches are more suitable for intelligent geometry software.

In fact, two pieces of intelligent geometry software, named Geometry Experts and Super Sketchpad, are developed in recent years. Now both are used in a lot of middle schools in China, and were warmly welcomed by teachers and students. The automated reasoning engineer of the intelligent geometry software is usually limited in the theorem prover based on search method. A drawback is that it can not give the readable proving for geometric theorems involving algebraic computation. This paper presents a high efficient algorithm for the search method that can give readable proving for a class of geometric theorems as long as its conclusions are polynomial equalities about geometric quantities, such as the length of segment, the degree of angle.