Lecture Notes in Artificial Intelligence 2061

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
*Berlin*
*Heidelberg*
*New York*
*Barcelona*
*Hong Kong*
*London*
*Milan*
*Paris*
*Tokyo*

Jürgen Richter-Gebert   Dongming Wang (Eds.)

# Automated Deduction in Geometry

Third International Workshop, ADG 2000
Zurich, Switzerland, September 25-27, 2000
Revised Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jürgen Richter-Gebert
Technische Universität München, Zentrum Mathematik, SB4
80290 München, Germany
E-mail: richter@ma.tum.de

Dongming Wang
Université Pierre et Marie Curie – CNRS
Laboratoire d'Informatique de Paris 6
4 place Jussieu, 75252 Paris Cedex 05, France
E-mail: Dongming.Wang@lip6.fr

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Automated deduction in geometry : third international workshop ; revised papers / ADG 2000, Zurich, Switzerland, September 25 - 27, 2000. Jürgen Richter-Gebert ; Dongming Wang (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2061 : Lecture notes in artificial intelligence)
ISBN 3-540-42598-5

CR Subject Classification (1998): I.2.3, I.3.5, F.4.1, I.5, G.2

ISBN 3-540-42598-5 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN: 10781608 06/3142 5 4 3 2 1 0

# Preface

With a standard program committee and a pre-review process, the Third International Workshop on Automated Deduction in Geometry (ADG 2000) held in Zurich, Switzerland, September 25–27, 2000 was made more formal than the previous ADG '96 (Toulouse, September 1996) and ADG '98 (Beijing, August 1998). The workshop program featured two invited talks given by Christoph M. Hoffmann and Jürgen Bokowski, one open session talk by Wen-tsün Wu, 18 regular presentations, and 7 short communications, together with software demonstrations (see http://calfor.lip6.fr/~wang/ADG2000/). Some of the most recent and significant research developments on geometric deduction were reported and reviewed, and the workshop was well focused at a high scientific level.

Fifteen contributions (out of the 18 regular presentations selected by the program committee from 31 submissions) and 2 invited papers were chosen for publication in these proceedings. These papers were all formally refereed and most of them underwent a double review-revision process. We hope that this volume meets the usual standard of international conference proceedings, represents the current state of the art of ADG, and will become a valuable reference for researchers, practitioners, software engineers, educators, and students in many ADG-related areas from mathematics to CAGD and geometric modeling.

ADG 2000 was hosted by the Department of Computer Science, ETH Zurich. We thank all the individuals, in particular external referees and members of the program committee, for their help with the organization of ADG 2000 and the preparation of this volume. The next workshop ADG 2002 will take place in Linz, Austria in September 2002. The proceedings of ADG '96 and ADG '98 have been published as volumes 1360 and 1669 in the same series of Lecture Notes in Artificial Intelligence.

Jürgen Richter-Gebert
Dongming Wang

June 2001

## Invited Speakers

**Jürgen Bokowski** (Darmstadt University of Technology, Germany)
**Christoph M. Hoffmann** (Purdue University, USA)

## Open Session Speaker

**Wen-tsün Wu** (Chinese Academy of Sciences, China)

## Program Committee

**Shang-Ching Chou** (Wichita, USA)
**Andreas Dress** (Bielefeld, Germany)
**Luis Fariñas del Cerro** (Toulouse, France)
**Desmond Fearnley-Sander** (Hobart, Australia)
**Xiao-Shan Gao** (Beijing, China)
**Hoon Hong** (Raleigh, USA)
**Deepak Kapur** (Albuquerque, USA)
**Jürgen Richter-Gebert** (**Co-chair**, Zurich, Switzerland)
**Bernd Sturmfels** (Berkeley, USA)
**Dongming Wang** (**Co-chair**, Paris, France)
**Volker Weispfenning** (Passau, Germany)
**Neil White** (Gainesville, USA)
**Walter Whiteley** (Toronto, Canada)
**Franz Winkler** (Linz, Austria)
**Lu Yang** (Chengdu, China)

# Contents

# On Spatial Constraint Solving Approaches*

Christoph M. Hoffmann and Bo Yuan

Computer Science Department
Purdue University
West Lafayette, IN 47907-1398, USA
{cmh,yuan}@cs.purdue.edu

**Abstract.** Simultaneous spatial constraint problems can be approached algebraically, geometrically, or constructively. We examine how each approach performs, using several example problems, especially constraint problems involving lines. We also prove that there are at most 12 real tangents to four given spheres in $\mathbf{R}^3$.

## 1 Introduction

Spatial constraint solving involves decomposing the constraint schema into a collection of indecomposable subproblems, followed by a solution of those subproblems. Good algorithms for decomposing constraint problems have appeared recently, including [3,6]. The best of those algorithms are completely general, adopting a generic degree-of-freedom reasoning approach that extends the older approach of searching for characteristic constraint patterns from a fixed repertoire such as [7].

In the spatial setting, even small irreducible problems give rise to nontrivial algebraic equation systems and yield a rich set of challenging problems. Restricting to points and planes, prior work has succeeded in elucidating and solving with satisfactory results the class of octahedral problems. An octahedral problem is an indecomposable constraint schema on six geometric entities, points and/or planes, with the constraint topology of an octahedron; see [1,7,10]. Such problems have up to 16 real solutions.

When lines are added as geometric primitives, even sequential problems become nontrivial, such as placing a single line at prescribed distances from four fixed points. In [1] line problems have been investigated and solved using several homotopy continuation techniques in conjunction with algebraic simplification. In particular, the problem 3p3L was analyzed and solved in which three lines and three points are pairwise constrained in the topology of the complete graph $K_6$. In this paper, we consider the problems 4p1L and 5p1L of placing four or five points and one line by spatial constraints. We also contrast them to the 6p octahedral problem. Our main purpose is to learn how successful the different approaches to solving these problems are.

* Work supported in part by NSF Grant CCR 99-02025, by ARO Contract 39136-MA, and by the Purdue Visualization Center.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 1–15, 2001.

## 2 Three Ways to Solve Subproblems

Once a subproblem has been identified, it must be translated into a simultaneous system of nonlinear equations, usually expressed algebraically. The system is then solved. Due to application considerations, we are especially interested in solution strategies that can, in principle, identify all real solutions of such a system. Thus we exclude in particular the usual Newton iteration approach that, beginning with a particular initial configuration, numerically determines at most one solution of the system.

We are interested in three approaches to solving the algebraic equations that arise when evaluating a subproblem.

1. Simplify the equations using a systematic set of techniques that are appropriate for the problem. This is the approach taken in, e.g., [1].
2. Apply a pragmatic mixture of geometric reasoning that simplifies the equations, in conjunction with other algebraic manipulation. This approach has been taken in, e.g., [8,7].
3. Adopt a procedural approach in which basic geometric reasoning results in a tractable, numerical procedure. This approach is familiar from, e.g., [5,4].

In each case, the goal is to simplify the system so that it becomes tractable to evaluate all real solutions. Aside from the intrinsic repertoire of each of the three approaches, we note that the choice of a coordinate system in which to solve the system is of critical importance.

We will explore how each of these approaches performs by considering the constraint subproblem in which 5 points and one line are to be placed subject to constraints on them. In [1], it was argued that a good choice of the coordinate system seeks to place the lines in a fixed position, formulating the equations on the points and on lines that could not be placed. We have found this to be a good idea as well. However, in the sequential line placing problem, we will see that it is better to place the points.

In the following, we will consider three spatial irreducible constraint problems:

1. **The 6p Octahedral Problem:** Given six points in space and twelve prescribed distances between them in the topology of an octahedron, determine the six points relative to each other. This problem is related to the Stewart platform [8].
2. **The 4p1L Problem:** Given four known points, find a line that lies at prescribed distance from each of them. Equivalently, find the common tangents to four fixed spheres [1].
3. **The 5p1L Problem:** Given one line and five points, and thirteen constraints between them in the topology shown in Figure 2, determine their relative position.

We will see that the first problem yields to the systematic simplification approach. That it can be addressed with the second approach as well has been shown in [8], among others. An especially nice solution using the Cayley-Menger determinant was presented by Michelucci in [10].

The second problem is amenable to the algebraic approach as well, except that the coordinate system has to be chosen carefully. We will explain briefly two different choices and their consequences.

Finally, the third constraint problem has not yielded to the first two constraint solving approaches, and the only satisfactory approach we have found so far is the computational one.

## 3 The Spatial Constraint Problems

We explain each constraint problem we consider in turn, in increasing order of complexity.

### 3.1 The 6p Octahedral Problem

We are given six points and twelve distance constraints, as indicated in Figure 1. The position of the points relative to each other, or with respect to a global coordinate system, is not known. As noted in [7], this problem has several instances when replacing some of the points with planes and considering angle constraints between planes. In every case, the problem cannot be further decomposed and requires solving a simultaneous system of nonlinear equations. A solution is a coordinate assignment to the points that satisfies all twelve distance constraints. As we will explain, this problem yields to both the algebraic and to the reasoning approach.

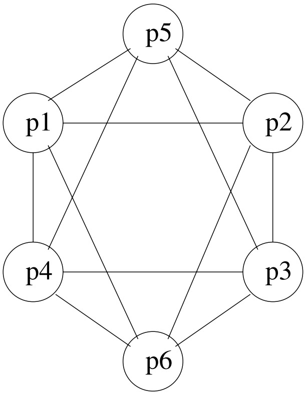


**Fig. 1.** The 6p Octahedral Problem: Graph vertices represent points, graph edges distance constraints

### 3.2 4p1L – Common Tangent to Four Spheres

We are given four points in fixed location in a global coordinate system. We are asked to find a line in space that lies at prescribed distance from each of the four points. Equivalently, we are given four fixed spheres in 3-space, not necessarily of the same radius, and are asked to find a line that is tangent to each sphere.

This problem is a sequential construction problem. A line has four independent coordinates, so four conditions such as the required distances determine its position. Suppose that we have a constraint problem in which each geometric element can be placed in a global coordinate system one-by-one in some order. If we admit as possible elements points, lines and planes, then this subproblem arises naturally. Note that geometric constraint problems that can be solved by such a sequential procedure are among the simplest problems.

We will discuss an algebraic approach to solving this problem that relies on a good choice of the coordinate system. Geometric reasoning approaches appear to fail to lead to more simplification.

### 3.3 The Problem 5p1L

Consider a configuration of five points and one line in 3-space that is constrained as shown in Figure 2. All constraints are distances. The subgraph of the five points has the topology of a square pyramid and is therefore not rigid. The point $p_5$ is the apex of the pyramid. In all, the configuration requires 19 generalized coordinates subject to 13 constraints, and is therefore generically a rigid body in 3-space.

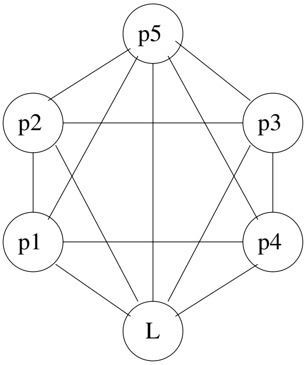


**Fig. 2.** The 5p1L Problem: Graph vertices represent points and a line, graph edges distances.

## 4 Solving Strategies

### 4.1 Algebraic Approach

In the algebraic approach, we choose a coordinate system and formulate a set of algebraic equations based on that choice. The equations are then simplified and brought into a form that gives greater insight into the number of distinct solutions and is sufficiently simple that root finding or reliable numerical techniques can be applied to solve the system. Ideally, the approach follows a systematic generic framework for manipulating the equations.

**The 6p – Octahedral Problem** The octahedral problem 6p has an elegant solution discovered by Michelucci [10] that is based on the Cayley-Menger determinant. Recall that the determinant relates the squared distances between five points in space. Consider the two unknown diagonal distances $d_{13} = d(p_1, p_3)$ and $d_{24} = d(p_2, p_4)$. Choosing the five points $\{p_1, p_2, p_3, p_4, p_5\}$, a quadratic relationship between $d_{13}^2$ and $d_{24}^2$ is obtained from the determinant. A similar relationship is obtained from the set $\{p_1, p_2, p_3, p_4, p_6\}$. Thus, we obtain two quartic equations in two unknowns, a system of total degree 16.

Michelucci's solution is independent of a coordinate system choice, a strong point, but it does not follow a systematic procedure. A systematic framework was developed by Durand in [1,2]. Choosing to place one point at the origin, one point on the $x$-axis, and one point in the positive quadrant of the $xy$-plane, the initial system consists of nine quadratic equations in nine unknowns. This system is then simplified by the following steps:

1. Gaussian elimination.
2. Solving univariate equations.
3. Parameterization of variables in bilinear and biquadratic equations.

The resulting system for 6p are three quartic equations in three variables, a system of total degree 64. By applying techniques from homotopy continuation, the final system required evaluating only 16 roots, of which, in the examples studied, 8 were real and 8 were complex.

**4p1L – Tangent to Four Spheres** The problem would appear to be classical, but we did not find much helpful literature on it. A systematic algebraic treatment of the problem was given by Durand in [1]. Durand found an equation system of degree 64 (the BKK bound) and experimentally determined that 40 of the 64 paths led to infinity. Thus, only 24 paths had to be explored. We improve this result now.

Placing three points at the origin, on the $x$-axis, and in the $xy$-plane, our initial equation system consists of six quadratic equations in six unknowns, (1–6). The unknowns are the point $(x, y, z)$ nearest to the origin on the sought line, and the unit length tangent $(u, v, w)$ of the line. Assume that $r_i$ is the distance of point $i$ from the line, and that the point coordinates are $(a_i, b_i, c_i)$. Then the initial equation system is

$$x^2 + y^2 + z^2 - r_1^2 = 0 \tag{1}$$
$$(a_2 - x)^2 + y^2 + z^2 - (a_2 u)^2 - r_2^2 = 0 \tag{2}$$
$$(a_3 - x)^2 + (b_3 - y)^2 + z^2 - (a_3 u + b_3 v)^2 - r_3^2 = 0 \tag{3}$$
$$(a_4 - x)^2 + (b_4 - y)^2 + (c_4 - z)^2 - (a_4 u + b_4 v + c_4 w)^2 - r_4^2 = 0 \tag{4}$$
$$xu + yv + zw = 0 \tag{5}$$
$$u^2 + v^2 + w^2 - 1 = 0 \tag{6}$$

We use equation (1) to eliminate the terms $x^2, y^2$ and $z^2$ from equations (2–4). Then those equations can be solved symbolically, yielding a solution that expresses the variables $x, y$ and $z$ as a quadratic expression in $u, v$ and $w$. This eliminates $x, y$ and $z$ from equations (5) and (6) and factors out a subsystem of three equations in $u, v, w$ of degree 2, 3 and 4, respectively. Thus, a degree reduction to 24 has been accomplished.

We note that for each solution $(x, y, z, u, v, w)$ of the system $(x, y, z, -u, -v, -w)$ is also a solution.[1] Geometrically, this says that the orientation of the lines is immaterial, which one expects. Therefore, the 24 solutions of the system, counted by Bezout's theorem, reduce to 12 geometric solutions. That this is the smallest number possible follows from the result by Theobald et al. [9]. They prove there are up to 12 distinct real tangents when all radii are equal, that is, when $r_1 = r_2 = r_3 = r_4 = r$.

It would seem that one could place the unknown line on the $x$-axis and seek equations to place the four points as a rigid structure subject to the distance constraints. Doing so yields equations with a high degree of symmetry and structure, but we have not found an attractive simplification of those equations.

**5pL1** We can choose a coordinate system in which the line $L$ is on the $x$-axis and the point $p_5$ on the $z$-axis as shown in Figure 3. We denote the distance between $L$ and the point $p_i$ with $r_i$, $i = 1, \ldots, 5$. The distance between and point $p_5$ and $p_i$, $i = 1, \ldots, 4$, is denoted $d_i$, and the distance between points $p_i$ and $p_j$ with $d_{ij}$. This choice leads to a system consisting of 12 equations in 12 unknowns:

$$\begin{array}{ll} y_i^2 + z_i^2 = r_i^2 & i = 1, \ldots, 4 \\ x_i^2 + y_i^2 + (z_i - r_5)^2 = d_i^2 & i = 1, \ldots, 4 \\ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = d_{ij}^2 & ij = 12, 23, 34, 41 \end{array} \tag{7}$$

Naive counting of the number of possible solutions would yield 4096. Using the multi-homogeneous Bezout theorem of [11], a tighter bound of 512 finite solutions

[1] This is clearly true for the original system. Moreover, the expressions substituted for $x$, $y$ and $z$ also exhibit the sign symmetry; hence the claim is true for the resulting system of degree 24.

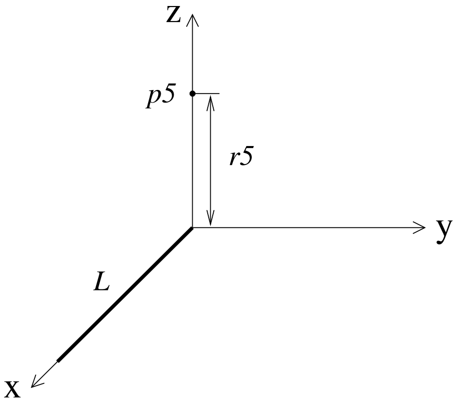


**Fig. 3.** Coordinate Assignment for the 5p1L Problem

is obtained. That bound does not make it practical to explore all solutions. Moreover, the system of equations resisted meaningful simplification, both ad-hoc manipulations as well as the systematic simplification steps developed before.

We could choose to place the coordinate system such that three points are put into a special position, say one point at the origin, one on the $x$-axis, and one in the $xy$-plane, but doing so did not lead to better equations.

### 4.2 Geometric Reasoning to Assist Simplification

In this approach we try to introduce auxiliary geometric structures, such as the curves described by a particular point when restricting to a subset of the constraints, especially if this can lead to a reasonable parameterization. Often, one can then introduce the additional constraints and derive a simpler equation system.

**6p – Octahedron** Geometric reasoning was used in [7] to yield a system of equations that, in conjunction with resultant techniques, succeeded in deriving a univariate polynomial of degree 16. It improves on the systematic approach by a factor of 4 and matches the Cayley-Menger solution.

**4p1L – Sphere Tangents** Presently, we do not have a good solution that exploits the geometry of the configuration. We believe that it should be possible to find one of total degree to 24 or less.

**5p1L** Placing the coordinate system as before, with the line on the $x$-axis and the point $p_5$ on the $z$-axis, we could proceed by parameterizing the locus of the point $p_1$ as function of the $z$-coordinate $Z$. From the distance constraints $r_1$ and

$d_1$ we obtain for the point $p_1$:

$$p_1 = \begin{cases} x_1(t) = \pm\sqrt{d_1^2 - r_1^2 - r_5^2 + 2r_5 t} \\ y_1(t) = \pm\sqrt{r_1^2 - t^2} \\ z_1(t) = t \end{cases} \tag{8}$$

We can then construct the remaining points whose coordinates are now a function of the parameter $t$, using the distance constraints for $r_2$, $d_2$, and $d_{12}$ for $p_2$, the distance constraints $r_4$, $d_4$ and $d_{41}$ for $p_4$. Finally, point $p_3$ is constructed using $r_3$, $d_3$ and $d_{23}$. This leaves the distance constraint $d_{34}$ to be used to determine the parameter $t$. The equations so derived have the following form:

$$\left\{\begin{array}{r}
-4d_2^2x_1(t)x_2 + 8x_1(t)y_1(t)x_2y_2 - 8r_5x_1(t)z_1(t)x_2 \\
-8r_5y_1(t)z_1(t)y_2 + 4x_1(t)^2x_2^2 + 4y_1(t)^2y_2^2 + 8r_5^2x_1(t)x_2 \\
-4d_1^2x_1(t)x_2 + 4d_{12}^2x_1(t)x_2 - 4d_2^2y_1(t)y_2 + 8r_5^2y_1(t)y_2 - 4d_1^2y_1(t)y_2 \\
+4d_{12}^2y_1(t)y_2 + 4d_2^2r_5z_1(t) - 4d_{12}^2r_5z_1(t) - 8r_5z_1(t)y_2^2 + 4r_5^2z_1(t)^2 \\
-8r_5^3z_1(t) + 4z_1(t)^2y_2^2 + 4r_5^2y_2^2 - 4r_2^2z_1(t)^2 + 8r_5r_2^2z_1(t) = D_1 \\
-z_1(t)x_2^2 + r_5x_2^2 - 2r_5x_1(t)x_2 - 2r_5y_1(t)y_2 + 2r_5^2z_1(t) \\
+d_2^2z_1(t) - r_5^2z_1(t) - r_2^2z_1(t) = D_2 \\
-4d_4^2x_1(t)x_4 + 8x_1(t)y_1(t)x_4y_4 - 8r_5x_1(t)z_1(t)x_4 - 8r_5y_1(t)z_1(t)y_4 \\
+4x_1(t)^2x_4^2 + 4y_1(t)^2y_4^2 + 8r_5^2x_1(t)x_4 - 4d_1^2x_1(t)x_4 + 4d_{41}^2x_1(t)x_4 \\
-4d_4^2y_1(t)y_4 + 8r_5^2y_1(t)y_4 - 4d_1^2y_1(t)y_4 + 4d_{41}^2y_1(t)y_4 + 4d_4^2r_5z_1(t) \\
-4d_{41}^2r_5z_1(t) - 8r_5z_1(t)y_4^2 + 4r_5^2z_1(t)^2 - 8r_5^3z_1(t) \\
+4z_1(t)^2y_4^2 + 4r_5^2y_4^2 - 4r_4^2z_1(t)^2 + 8r_5r_4^2z_1(t) = D_3 \\
-z_1(t)x_4^2 + r_5x_4^2 - 2r_5x_1(t)x_4 - 2r_5y_1(t)y_4 + 2r_5^2z_1(t) + d_4^2z_1(t) \\
-r_5^2z_1(t) - r_4^2z_1(t) = D_4 \\
-12z_1(t)x_2x_3 + 12r_5x_2x_3 - 4r_5x_1(t)x_2 - 4r_5y_1(t)y_2 + 4r_5^2z_1(t) \\
+4r_5z_1(t)z_3 - 4r_5^2z_3 + 2d_2^2z_1(t) + 2d_3^2z_1(t) \\
+2d_{23}^2z_1(t) - 4r_5^2z_1(t) = D_5 \\
4z_1(t)x_3x_4 - 4r_5x_3x_4 + 4z_1(t)y_3y_4 - 4r_5y_3y_4 - 4x_1(t)x_4z_3 \\
-4y_1(t)y_4z_3 + 2d_4^2z_3 + 2d_1^2z_3 - 2d_{41}^2z_3 + 4r_5x_1(t)x_4 + 4r_5y_1(t)y_4 \\
-4r_5z_1(t) - 2d_3^2z_1(t) - 2d_4^2z_1(t) + 2d_{34}^2z_1(t) + 4r_5^2z_1(t) = D_6 \\
y_3^2 + z_3^2 = D_7 \\
x_3^2 - 2r_5z_3 = D_8
\end{array}\right. \tag{9}$$

where $D_1, D_2, D_3, D_4, D_5, D_6$ are constants. The system is unattractive.

### 4.3 Construction by Computation

The closed-form algebraic expressions for the point coordinates of the 5p1L problem that were obtained by the geometric reasoning described before, do not seem to be simple enough to lead to further massive algebraic simplification. However, they are very easy to evaluate computationally, and can be used to define numerically a curve in a 2D coordinate space defined by the parameter and the distance $d_{34}$. When the curve is intersected with the nominal distance line, the real solutions are obtained. As illustrated in Figure 4, $p_{10}$ is on line $L$ and $\overline{p_{10}p_1} \perp L$, the angle between $\overline{p_{10}p_1}$ and the $xy$-plane is $\theta$. We use $\theta$ as parameter to calculate point $p_1$:

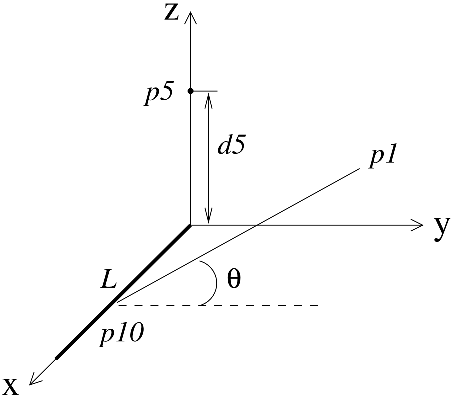


**Fig. 4.** Parameterization with $\theta$

$$p_1 = \begin{cases} x_1(\theta) = \pm\sqrt{d_1^2 - r_1^2 - r_5^2 + 2r_1r_5\sin(\theta)} \\ y_1(\theta) = r_1\cos(\theta) \\ z_1(\theta) = r_1\sin(\theta) \end{cases} \tag{10}$$

For practical purposes, the approach is satisfactory, since it gives a systematic, and sufficiently simple, procedure to find all real solutions. Moreover, the solutions so found can be further refined with other numerical processes, since they provide good starting points. From a theoretical perspective, the draw-back of the procedural approach is its inability to produce, with certainty, a bound on the number of solutions. Here are the details for our 5p1L problem, and several example solutions.

$p_2$ can be solved using the constraints $dist(L, p_2) = r_2$, $dist(p_5, p_2) = d_2$ and $dist(p_1, p_2) = d_{12}$. As illustrated in Figure 5, the point $p_s = (x_s, y_s, z_s)$ in triangle $\triangle p_1p_2p_5$ is on the line $\overline{p_1p_5}$ and $\overline{p_2p_s} \perp \overline{p_1p_5}$. So, we have

$$s = |p_sp_5| = \frac{(d_1^2 + d_2^2 - d_{12}^2)}{2d_1}$$

$$h = |p_sp_2| = \sqrt{d_2^2 - s^2}$$

We obtain

$$p_s = p_5 + \frac{s}{d_1}(p_1 - p_5)$$

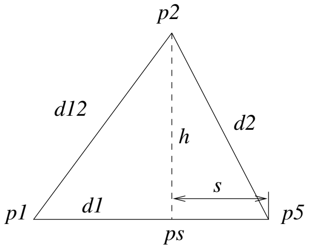


**Fig. 5.** Triangle $p_1p_2p_5$

Consider the vector $\boldsymbol{w} = \frac{p_1-p_5}{|p_1-p_5|} = \frac{p_1-p_5}{d_1}$. We define a plane $\Pi$ through the point $p_s$ perpendicular to $\boldsymbol{w}$. Since $dist(L, p_2) = r_2$, the point $p_2$ is on the cylinder $\Sigma : y^2 + z^2 = r_2^2$ whose axis is the line $L$ and whose radius is $r_2$. Let $p_c$ be the interaction point of line $L$ and plane $\Pi$, then $p_c = (x_c, y_c, z_c)$ where

$$\begin{aligned} x_c &= x_s + \tfrac{w_y}{w_x} y_s + \tfrac{w_z}{w_x} z_s \\ y_c &= 0 \\ z_c &= 0 \end{aligned}$$

Using the vectors

$$\begin{aligned} \boldsymbol{v} &= \tfrac{\boldsymbol{w}\times\boldsymbol{L}}{|\boldsymbol{w}\times\boldsymbol{L}|} \\ \boldsymbol{u} &= \boldsymbol{v} \times \boldsymbol{w} \end{aligned}$$

we set up a local coordinate system: $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$, where

$$\begin{aligned} o' &= p_c \\ \boldsymbol{x}' &= \boldsymbol{u} \\ \boldsymbol{y}' &= \boldsymbol{v} \\ \boldsymbol{z}' &= \boldsymbol{w} \end{aligned}$$

The matrix transform from the global coordinate system $(o, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ to the local system $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$ is

$$M = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Figure 6 illustrates the local coordinate system $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$ situated in the global system $(o, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.

From the construction process we know that the point $p_2$ lies on a circle in the plane $\Pi$ with radius $h$. Let $p_s' = Mp_s$, in the local coordinate system $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$. Then the equation of the circle is

$$(x' - x_s')^2 + (y' - y_s')^2 = h^2 \tag{12}$$

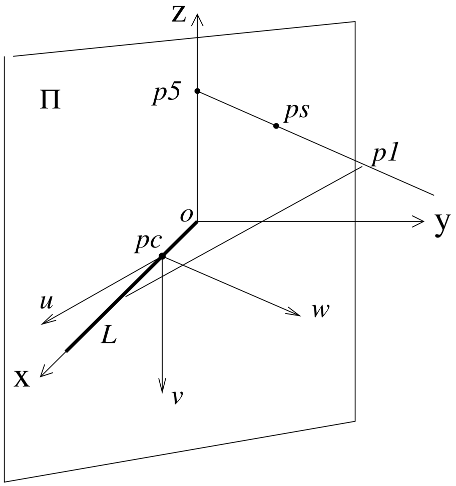


**Fig. 6.** Local Coordinate System $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$

Now the vector along line $L$ is $\boldsymbol{L} : (1, 0, 0)$, the angle between $\boldsymbol{L}$ and $\boldsymbol{w}$ is $\beta$, and the intersection of plane $\Pi$ and cylinder $\Sigma$ is an ellipse on plane $\Pi$. In the local system $(o', \boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}')$, the ellipse equation is

$$\frac{{x'}^2}{r_2^2 \csc^2(\beta)} + \frac{{y'}^2}{r_2^2} = 1 \tag{13}$$

Solving equations (12) and (13) simultaneously, we get $p_2'$ and from it, in turn, $p_2$:

$$p_2 = (x_2(\theta), y_2(\theta), z_2(\theta))$$

Note that there are up to 4 real solutions for $p_2$.

Similarly, we compute $p_4$ from its constraints with the line $L$ and the points $p_1$ and $p_5$. Finally, we compute $p_3$ from the constraints with line $L$ and points $p_2$ and $p_5$

$$p_3 = (x_3(\theta), y_3(\theta), z_3(\theta))$$
$$p_4 = (x_4(\theta), y_4(\theta), z_4(\theta))$$

$d_{34}(\theta)$ is a complicated curve in a coordinate space defined by the parameter $\theta$ and the distance $dist(p_3(\theta), p_4(\theta))$. The curve would be hard to express symbolically. However, we can trace it numerically.

Given a step length $d\theta$, we calculate $d_{34}(\theta)$ for every step $\theta = \theta + d\theta$, and so obtain the curve $C_\theta : d_{34}(\theta) - \theta$ numerically. Let the absolute error of $d_{34}(\theta)$ and the nominal distance line $d_{34}$ be

$$\rho(\theta) = |d_{34}(\theta) - d_{34}|$$

Obviously, the smaller $\rho(\theta)$ is, the nearer $\theta$ is to a real solution of the 5pL1 problem. Call a point $(\theta, d_{34}(\theta))$ a *coarse solution* if $\theta$ satisfies

$$d_{34}(\theta) < \delta$$

**Table 1.** An Constraint Set of the 5p1L Problem

| | |
|---|---|
| $r_1$ | 5.12863551744133 |
| $r_2$ | 3.4797204504532 |
| $r_3$ | 5.12009033478805 |
| $r_4$ | 4.48866237372967 |
| $r_5$ | 0.854823450422681 |
| $d_1$ | 5.40391247291482 |
| $d_2$ | 4.92751853999451 |
| $d_3$ | 6.556901760918 |
| $d_4$ | 5.04776146732994 |
| $d_{12}$ | 2.49916074098941 |
| $d_{23}$ | 9.55687124240852 |
| $d_{34}$ | 9.15 |
| $d_{41}$ | 7.1858882412183 |

for a chosen tolerance $\delta$. The coarse solution set $S_\delta$ is then

$$S_\delta = \{q_\theta = (\theta, d_{34}(\theta)) | \rho(\theta) < \delta, q_\theta \in C_\theta\}$$

$\delta$ is the threshold of the coarse solutions, and the size of $|S_\delta|$ diminishes with $\delta$. The coarse solutions can be further refined with Newton-Raphson iteration since they provide good starting points.

### 4.4 An Example

Table 1 gives an example of constraint set of 5pL1 problem, by defining $d\theta = 1.0^o$, Figure 7 gives the discrete curve. In our example, if $\delta = 0.1$, $|S_c| = 108$, if $\delta = 0.2$, $|S_c| = 224$. When $\delta = 0.1$ we can get 20 refined real solutions; when $\delta = 0.2$ we can get 24 refined real solutions; when $\delta > 0.2$ we have more than 224 coarse solutions but the refined real solution number is still 24. Therefore, the maximum real solution number of the example is 24. The circles on the nominal distance line in Figure 7 represent the real solutions, Table 2 gives all the 24 real solutions of this example.

The computation was carried out using a tolerance-driven search for potential solutions followed by a Newton iteration refining the initial values. On a PC with a 500MHz Pentium 3 the initial search took 100 milliseconds with a tolerance of 0.2, and the subsequent refinement took an additional 233 ms. This contrasts favorably with the computation times obtained by Durand on a Sun SPARC 20 using homotopy continuation where 24 paths were evaluated in approximately 30 sec. The homotopy evaluation on the slower machine was a completely general implementation, while our computation of the solution was specifically designed for this particular problem. It would be interesting to test this problem on general multi-variate interval Newton solvers.

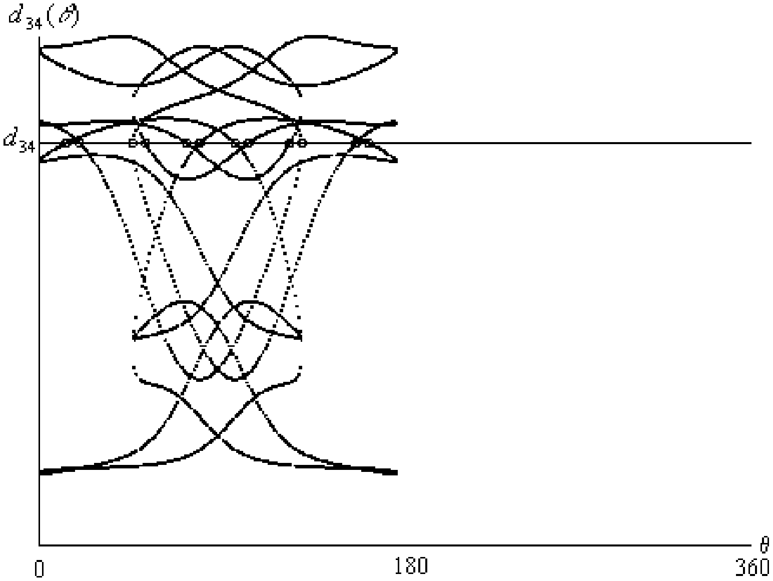


**Fig. 7.** $d_{34}(\theta) - \theta$ Curve

## 5 Further Discussion

The Construction by Computation approach can be used more generally. Let $F(X) = 0$ be a system of $n$ nonlinear equations $F = \{f_1, \ldots, f_n\}$ with $n$ unknowns $X = \{x_1, \ldots, x_n\}$. To find all real solutions of $F(X) = 0$, we can choose a real parameter set $T = \{t_1, \ldots, t_k\}_{k<n}$ such that $X$ can be solved as $X(T) = \{x_1(T), \ldots, x_n(T)\}$ by using $n - k$ equations

$$F_{n-k} = \{f^i | f^i \in F, 1 \leq i \leq n-k\} \subset F$$

Let

$$F_k = F - F_{n-k} = \{f^j | f^j \not\in F_{n-k}, f^j \in F, 1 \leq j \leq k\} \subset F$$

and define

$$\rho(T) = \begin{matrix} \max \\ \forall f^j \in F_k \end{matrix} (|f^j|)$$

Let domain of $T$ be $D_T = [t_{1min}, t_{1max}] \times \cdots \times [t_{kmin}, t_{kmax}]$, and for every $t_i \in T$ define a step size $dt_i$ such that we can calculate $\rho(T)$ on $D_T$ numerically for every $[t_1 = t_1 + dt_1] \times \cdots \times [t_k = t_k + dt_k]$. Obviously, $T \times \rho(T) \subset \Re^{k+1}$ is a hypersurface. Given a small positive real number $\delta$, we can get the Coarse Solution Set

$$S_c = \{q_T = (T, \rho(T)) | \rho(T) < \delta, q_T \in T \times \rho(T)\}$$

**Table 2.** Real Solution Set of the Example

| | $p_1$ | $p_2$ | $p_3$ | $p_4$ |
|---|---|---|---|---|
| 1 | (2.06,4.98,1.21) | (3.30,3.46,-0.34) | (-4.87,2.43,4.51) | (3.45, -1.36, 4.28) |
| 2 | (-2.06,4.98,1.21) | (-3.30,3.46,-0.34) | (4.87, 2.43, 4.51) | (-3.45, -1.36,4.28) |
| 3 | (2.28,4.81,1.77) | (3.85,2.88,1.96) | (-2.79,1.73,-4.82) | (-3.46,1.30,4.30) |
| 4 | (-2.28, 4.81,1.77) | (-3.85,2.88,1.96) | (2.79,1.73,-4.82) | (3.46,1.30,4.30) |
| 5 | (2.93,3.48,3.77) | (3.72,3.18,1.42) | (-4.95,1.29,4.95) | (-0.37,3.65,-2.61) |
| 6 | (-2.93,3.48,3.77) | (-3.72,3.18,1.42) | (4.95,1.29,4.95) | (0.37,3.65,-2.61) |
| 7 | (3.04,3.04,4.13) | (4.14,0.94,3.35) | (-4.93,-1.65,4.85) | (-1.58,4.32,-1.23) |
| 8 | (-3.04,3.04,4.13) | (-4.14,0.94,3.35) | (4.93,-1.65,4.85) | (1.58,4.32,-1.23) |
| 9 | (3.26,1.39,4.94) | (4.01,2.20,2.70) | (-3.22,3.90,-3.32) | (-3.47,-1.07,4.36) |
| 10 | (-3.26,1.39,4.94) | (-4.01,2.20,2.70) | (3.22,3.90,-3.32) | (3.47,-1.07,4.36) |
| 11 | (3.29,0.79,5.07) | (4.15,-0.84,3.38) | (-4.92,1.82,4.79) | (1.72,4.39,-0.95) |
| 12 | (-3.29,0.79,5.07) | (-4.15,-0.84,3.38) | (4.92,1.82,4.79) | (-1.72,4.39,-0.95) |
| 13 | (3.29,-0.79,5.07) | (4.15,0.84,3.38) | (4.92,-1.82,4.79) | (1.72,-4.39,-0.95) |
| 14 | (-3.29,-0.79,5.07) | (-4.15,0.84,3.38) | (4.92,-1.82,4.79) | (-1.72,-4.39,-0.95) |
| 15 | (3.26,-1.39,4.94) | (4.01,-2.20,2.70) | (-3.22,-3.90,-3.32) | (-3.47,1.07,4.36) |
| 16 | (-3.26,-1.39,4.94) | (-4.01,-2.20,2.70) | (3.22,-3.90,-3.32) | (3.47,1.07,4.36) |
| 17 | (3.04,-3.04,4.13) | (4.14, -0.94,3.35) | (-4.93,1.65,4.85) | (-1.58,-4.32,-1.23) |
| 18 | (-3.04, -3.04,4.13) | (-4.14,-0.94,3.35) | (4.93,1.65,4.85) | (1.58,-4.32,-1.23) |
| 19 | (2.93,-3.48,3.77) | (3.72,-3.18,1.42) | (-4.95,-1.29,4.95) | (-0.37,-3.65,-2.61) |
| 20 | (-2.93,-3.48,3.77) | (-3.72,-3.18,1.42) | (4.95,-1.29,4.95) | (0.37,-3.65,-2.61) |
| 21 | (2.06,-4.98,1.21) | (3.30,-3.46,-0.34) | (-4.87,-2.43,4.51) | (3.45,1.36,4.28) |
| 22 | (-2.06,-4.98,1.21) | (-3.30,-3.46,-0.34) | (4.87,-2.43,4.51) | (-3.45,1.36,4.28) |
| 23 | (2.28,-4.81,1.77) | (3.85,-2.88,1.96) | (-2.79,-1.73,-4.82) | (-3.46,-1.30,4.30) |
| 24 | (-2.28,-4.81,1.77) | (-3.85,-2.88,1.96) | (2.79,-1.73,-4.82) | (3.46,-1.30,4.30) |

For every $q_T \in S_c$ we can get an starting point $X^0$. Using Newton-Raphson iteration, we may refine the starting point to a real solution of $F(X) = 0$. After calculating all $q_T \in S_c$ we can get the real solution set $S_r$. If the step sizes $dt_i, i = 1, \ldots, k$, are small enough and $\delta$ is large enough, we can find all real solutions of $F(X) = 0$.

## References

1. C. Durand. *Symbolic and Numerical Techniques for Constraint Solving.* PhD thesis, Purdue University, Dept. of Comp. Sci., 1998.
2. C. Durand and C.M. Hoffmann. A systematic framework for solving geometric constraints analytically. *J. of Symbolic Computation*, 30:483–520, 2000.
3. I. Fudos and C.M. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Trans on Graphics*, 16:179–216, 1997.
4. X.S. Gao and C.C. Zhu. Geometric constraint solving with linkages and geometric method for polynomial equations-solving. In *MM Research Preprints 18*, pages 48–58. Academia Sincica, Institute of Systems Science, 1999.
5. J.X. Ge, S.C. Chou, and X.S. Gao. Geometric constraint satisfaction using optimization methods. *Computer Aided Design*, 31:867–879, 1999.

6. C.M. Hoffmann, A. Lomonosov, and M. Sitharam. Geometric constraint decomposition. In *Geometric Constraint Solving and Applications*, pages 170–195. Springer-Verlag, New York, 1998.
7. C.M. Hoffmann and P. Vermeer. Geometric constraint solving in $R^2$ and $R^3$. In *Computing in Euclidean Geometry*, pages 170–195. World Scientific Publishing, Singapore, 1995.
8. D. Lazard and J.P. Merlet. The (true) Stewart platform has 12 configurations. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2160–2165. IEEE, 1994.
9. I.G. Macdonald, J. Pach, and Th. Theobald. Common tangents to four unit balls in $\mathbf{R}^3$. *Discrete and Comp. Geometry*, to appear.
10. D. Michelucci. Using Cayley Menger determinants. Web document available from http://www.emse.fr/~micheluc/MENGER/index.html.
11. A. Morgan. A homotopy for solving general polynomial systems that respects $m$-homogeneous structures. *Applied Mathematics and Computation*, 24:101–113, 1987.

# A Hybrid Method for Solving Geometric Constraint Problems[⋆]

Xiao-Shan Gao, Lei-Dong Huang, and Kun Jiang

Institute of Systems Science
Academy of Mathematics and System Sciences
Academia Sinica
Beijing 100080, China
xgao@mmrc.iss.ac.cn

**Abstract.** We propose an algorithm for solving geometric constraint problems. The algorithm has linear complexity for constrained systems without loops, and is of quadratic complexity for constraint problems with loops. This algorithm is complete for constraint problems about simple polygons. The key of the algorithm is to combine the idea of graph based methods for geometric constraint solving and geometric transformations from rule-based methods.

## 1 Introduction

Geometric constraint solving (GCS) is the central topic in much of the current work of developing parametric and intelligent CAD systems. It also has applications in mechanical engineering, linkage design, computer vision and computer aided instruction. There are four main approaches to GCS: the graph analysis approach [6,2,16,13], the rule-based approach [1,3,14,11,17,9,18], the numerical computation approach [5,7,12,15], and the symbolic computation approach [4,10]. In practice, most people use a combination of these approaches to get the best result.

The basic idea of the graph analysis approach is to represent the geometric constraints with a graph and then use algorithms from graph theory to transform the drawing problem into certain constructive forms, like constructive forms with ruler and compass. In [16], graph decomposition algorithms are used to divide a constraint problem into three parts, to solve the parts separately, and then to assemble the three parts together to form the final diagram. In [2], geometric diagrams are divided into clusters and clusters are merged into larger clusters until a single cluster is obtained. In [13], graph methods are used to decide whether the the constraint problems are well-, under- or over-constrained. In particular, Hoffmann et al. presented an algorithm to decompose any constraint problem into constructible sub-graphs, called dense sub-graphs [8]. In most cases,

[⋆] This work is supported in part by a “973” project and by CNSF under an outstanding youth grant (NO. 69725002).

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 16–25, 2001.

this decomposition provides a computational procedure for the corresponding GCS problem.

However, the GCS problem is not solved perfectly. The basic steps of graph analysis methods are first decomposing a large constraint problem into smaller ones and then assembling them together. In this process, we need to solve systems of simultaneous equations. In the 2D case, if the maximal number of equations to be solved is equal to or less than two, then the problem is called *a constraint problem without loops*. Otherwise, it is called *a constraint problem with loops*. For constraint problems with loops, we need to solve a simultaneous equation system, which could be a difficult task. Therefore, it is still worth to find fast methods which may transfer constraint problems with loops to constraint problems without loops.

In this paper, we present a graph analysis method of GCS. For a constraint problem without loops, the method has linear complexity and can be used to solve problems involving geometric objects and geometric constraints of any degree of freedom (DOF). So the algorithm works for both 2D and 3D cases. Also, the method works for well and under constraint problems. For a constraint problem with loops, the method has quadratic complexity and is complete for constraint problems of simple polygons. Generally speaking the graph analysis methods have lower complexities and the rule-based approaches can solve more types of loops by using geometric transformations. The key idea of our method is to combine these two approaches to obtain a fast and powerful loop breaking method.

The rest of the paper is organized as follows. In Section 2, we will present the graph representation for the three geometric transformations. In Section 3, we will present the algorithm. In Section 4, we will present the conclusion.

## 2 Graph Representation of Geometric Transformations

### 2.1 Graph Representation of Constraint Problems

A constraint problem can be represented by a weighted graph [8]. The graph vertices represent the geometric objects, edges represent geometric constraints. The weight of a vertex is the DOF of the represented object. The weight of an edge represents the DOF eliminated by the represented constraint. For instance, Fig. 2 is the graph representation for the constraint problem in Fig. 1. Note that the weights for the vertices and edges are not marked explicitly. In 2D case, most constraints have one DOF and most objects have two DOFs.

For the convenience of complexity analysis, an *adjacent list* representation of graphs is used. Fig. 3 is the representation of the graph in Fig. 2, where the nodes in the first column are called head nodes.

### 2.2 Geometric Transformations

Geometric transformations are used to solve constraint problems with loops in rule based methods [3,17,18]. Here, we will give the graph representation for three

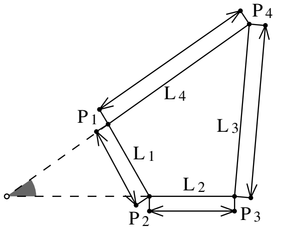


**Fig. 1.** Lengths of four edges and angle $(L_2, L_4)$ are given

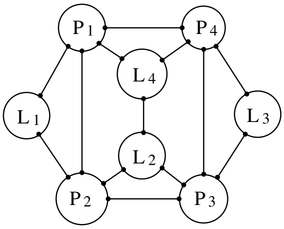
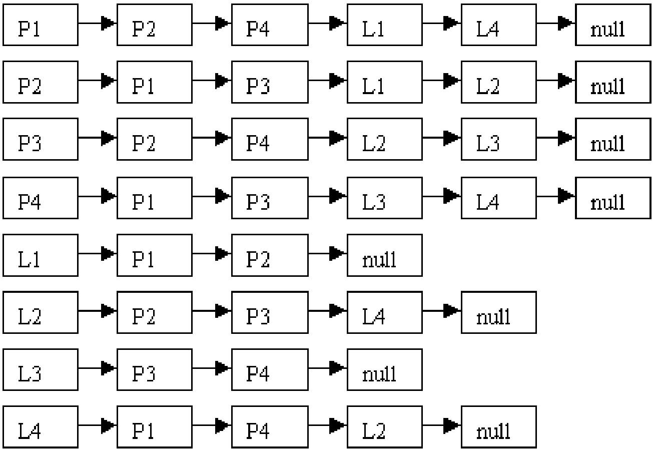


**Fig. 2.** The graph representation



P1 → P2 → P4 → L1 → L4 → null
P2 → P1 → P3 → L1 → L2 → null
P3 → P2 → P4 → L2 → L3 → null
P4 → P1 → P3 → L3 → L4 → null
L1 → P1 → P2 → null
L2 → P2 → P3 → L4 → null
L3 → P3 → P4 → null
L4 → P1 → P4 → L2 → null

**Fig. 3.** Adjacent list representation of graphs

geometric transformations, which will be used to solve all constraint problems about simple polygons.

- Rigid Body Transformation.
  If there exists a well-constrained sub-graph which can be solved sequentially by ruler and compass, then this sub-graph represents a rigid body in the original constrained diagram. In certain cases, we may represent this rigid body by a simpler geometric object.
  The following special form of this transformation is especially useful in solving constraint problems about polygons. Constraint sets like $\{|AB| = d_1, \angle ABC = \alpha, |BC| = d_2\}$, $\{|AB| = d_1, |BC| = d_2, |CA| = d_3\}$ and $\{\angle ABC = \alpha, |BC| = d_2, \angle ACB = \beta\}$ are called called *sas, sss, and asa* constraints respectively. A series of sas, sss or asa constraints may determine a rigid body which can be replaced by a segment if only the two ending points are connected to other geometric objects. Such an example is given in Fig. 6, which can be solved by using two sas transformations.
  We give a precise description for the sas transformation below. Descriptions for other transformations are omitted. For an object $v$, let LDEG$(v)$

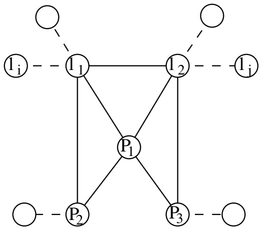


**Fig. 4.** Before the sas transformation

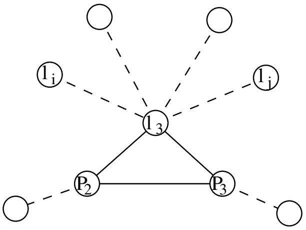


**Fig. 5.** After the sas transformation

and PDEG($v$) be the numbers of lines and points connected to $v$ in the constraint graph respectively. We need to find a sub-graph satisfying the following conditions (Fig. 4).

- $|P_0P_1|$, $|P_1P_2|$ and $\angle P_0P_1P_2$ are known.
- LDEG$(P_1) = 2$, PDEG$(P_1) = 2$.
- PDEG$(l_1) = 2$, PDEG$(l_2) = 2$.

Then we may delete $P_1, l_1, l_2$ and add a new line $l_3 = P_0P_2$ and a new constraint $|P_0P_2|$. Note that all other constraints about $l_1, l_2$ are angle constraints and can be converted to constraints about $l_3$.

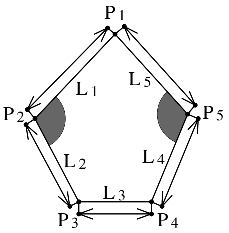


**Fig. 6.** A problem using the sas transformation

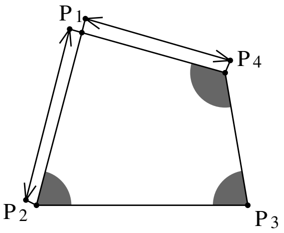


**Fig. 7.** A problem using the angle transformation

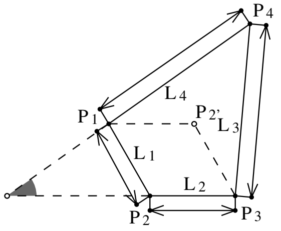


**Fig. 8.** A problem using the parallel transformation

– Angle Transformation. Angle transformations can be used to solve many constraint problems with loops. In Fig. 7, since three inner angles of the quadrilateral are known, its fourth inner angle at point $P_1$ is also known. With this angle known, the sas transformation can be used to solve the constraint problem.

In general, we may use the concept of *full-angles* [20] for angle transformation. We use $[L_i,L_j]$ to represent the *full-angle* from line $L_i$ to line $L_j$, which satisfies the following properties

$$[L_i, L_j] = -[L_j, L_i],$$
$$[L_i, L_j] = [L_i, L_k] + [L_k, L_j].$$

With the above property, we may easily divide the lines into equivalent classes such that the angle between any two lines in the same class is known or can be calculated. It is clear that we may find all such equivalent classes in linear time of the number of geometric objects and geometric constraints.

– Parallel Transformation. The problem in Fig. 1 cannot be solved with the above transformations, neither by Hoffmann and Owen's graph triangle decomposition methods. The methods reported in [8,14] may solve this problem. But in these methods, we need to solve equation systems with more than two equations. In other words, the loop in this problem is not broken. In [3,17], the loop in this problem can be broken by introducing auxiliary points. The solution is given in Fig. 8, where line segment $P_2P_3$ is translated to $P_1P_2'$. Since $\angle P_4P_1P_2'$ is the same as the angle formed by lines $P_1P_4$ and $P_2P_3$, we may solve the problem with an sas transformation.

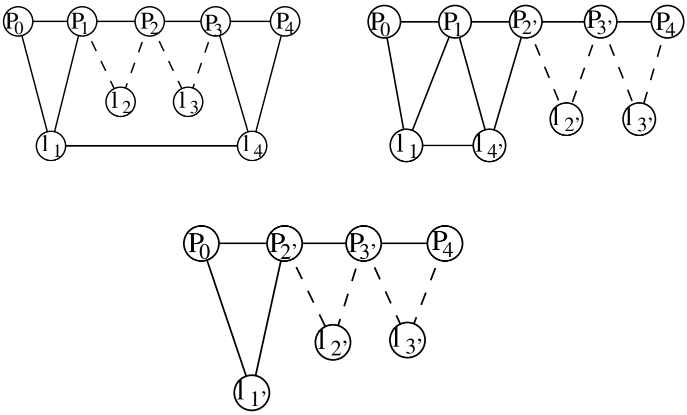


**Fig. 9.** Parallel transformation

In general, the parallel transformation works as follows. Let $P_0$, $P_1$, ..., $P_k$ be points in a constraint problem such that

1. Angle $[P_0P_1, P_{k-1}P_k]$ is known.
2. $|P_0P_1|, |P_1P_2|, \dots, |P_{k-1}P_k|$ are known.
3. The only geometric objects connecting $P_i, i = 1, \dots, k-1$ are points $P_{i-1}, P_{i+1}$ and lines $P_{i-1}P_i, P_iP_{i+1}$.
4. There are no points on lines $P_iP_{i+1}, i = 0, \dots, k-1$ besides $P_i$ and $P_{i+1}$.

Fig. 9 gives an example for $k = 4$, where the dotted lines mean that they may not exist. If the above conditions are met, we may make parallelograms $P_1P_2P_3'P_2'$, $P_2P_3P_4'P_3'$, ..., $P_{k-2}P_{k-1}P_kP_{k-1}'$. Now replacing $P_2, \dots, P_{k-1}$ by $P_2', \dots, P_{k-1'}$ in the original problem to obtain a new constraint problem (Fig. 9, middle). It is clear that

1. All the constraints about $P_2, \ldots, P_{k-1}$ can be converted to constraints about $P'_2, \ldots, P'_{k-1}$. In particular, $\angle P_0 P_1 P'_2$ is known.
2. If $P'_2, \ldots, P'_{k-1}$ are constructed, we may construct $P_2, \ldots, P_{k-1}$ by computing the intersections of lines.

Furthermore, we may do an sas transformation for $P_0, P_1, P'_2$ to obtain the final result of the parallel transformation (Fig. 9, right).
Suppose that the above $P_i$ have been found. Then we need $O(n + e)$ steps to do the parallel transformation.

## 3 The Algorithm and Applications

### 3.1 The Algorithm

For any vertex $v$ in the graph, let $\mathrm{DEG}(v)$ be the sum of the weights of all the edges connecting to this vertex. Then, a vertex can be determined explicitly by the constraints involving it if the following condition is satisfied:

$$\mathrm{DEG}(v) \leq \mathrm{DOF}(v). \tag{1}$$

The basic idea of the algorithm is to repeatedly remove those vertices satisfying this condition. The problem has no loops if and only if all the vertices of the graph can be removed in this way. If the problem has loops, we will use one of the three transformations to reduce the problem to a smaller one.

1. Generate the adjacent list representation for the constraint graph and $\mathrm{DEG}(v)$ for each head node $v$.
2. Do the angle transformation to find all the equivalent classes $L_1, \ldots, L_s$ such that the angle between two lines in the same class can be computed.
3. Let $v$ point to the first head node.
4. If $v$ is not an empty node then goto the next step. Otherwise we are at the end of the graph. There are two cases. If all objects are removed from the adjacent list, the problem can be solved by the algorithm, because each time we remove an object from the list, we actually give a construction method for it. If there are still nodes in the list, we cannot solve the problem.
5. Remove those nodes $v$ satisfying (1) from the graph and for each node $w$ in the adjacent liet of $v$, minus one from $\mathrm{DEG}(w)$. To ensure that the complexity of this step is linear, we will find the next node satisfying (1) from the neighboring nodes of $v$. If all nodes are removed in this way, goto Step 4. Otherwise, goto the next step.
6. If $v$ is a line satisfying
   - There are only two points $P_0$ and $P_1$ on it and
   - $|P_0 P_1|$ is known,

   do the next step. Otherwise set $v$ to be the next head node and goto Step 4.
7. Staring from $P_1$ (or $P_0$), find points $P_2, \ldots, P_k$ satisfying the conditions of the parallel transformation.
8. If $k = 1$, nothing need to be done. Let $v$ to be the next head node and go back to Step 4.

9. If $k = 2$, we may do an sas transformation to $P_0, P_1, P_2$. Put the new introduced line $P_0P_2$ into the equivalent class of $P_0P_1$ and also at the end of the adjacent list. Let $v$ be the next head node and go to Step 4.
10. Do a parallel transformation to $P_0, P_1, \ldots, P_k$ and update the constraint graph. Put the new introduced line $P_0P_2'$ into the equivalent class of $P_0P_1$ and also at the end of the adjacent list. Let $v$ to be the next head node and go to Step 4.

Let $n$ be the number of geometric objects and $e$ the number of constraints in the problem. The main loop of the algorithm starts from Step 4. There are four possibilities for each loop. (1) Do nothing. (2) Remove several nodes in Step 4. (3) Do an sas transformation. (4) Do a parallel transformation. In cases (2), (3) and (4), the number of geometric objects in the adjacent list will strictly decrease. Therefore, the maximal number of this loop is the number of the geometric object, $n$.

We will show that the complexity of each loop is $\mathrm{O}(n+e)$. Also, Steps 1 and 2 have complexity $\mathrm{O}(n+e)$. Therefore, the complexity of the algorithm would be $\mathrm{O}(n^2+ne)$. In Step 5, we may eliminate at most $n$ nodes. For each node eliminated, we need to search its adjacent list. Let $d_i, i = 1, \ldots, n$ be the steps needed to eliminate the $i$-th node. Then $d_i$ is actually propotional to the number of constraints involving the $i$-th node plus a constant. Then the complexity of this step is $\sum_{i=1}^{n} d_i = O(n+e)$. The complexity of Step 5 is $(e)$. In Step 6, we need to find a *path* $P_0P_1 \cdots P_k$ in the graph, each $P_i, 1 \le i \le k-1$ does not connect to other points. We may do this in $\mathrm{O}(n+e)$ steps. The sas and parallel transformations need $O(n+e)$ steps. Therefore, each loop will finish in $\mathrm{O}(n+e)$ steps.

From the above analysis, if a constraint problem has no loops, then it can be solved by the above algorithm in linear time. The quadratic complexity is mainly due to the fact that no backtracking is needed in Step 4 of the algorithm. We need to show that all possible sas and parallel transformations will be executed in the algorithm. This is because after each sas or parallel transformation, two equivalent classes for lines will not be merged. Therefore, if we cannot do sas and parallel transformations for a line before another transformation, we cannot do that after the transformation.

### 3.2 Constraint Problems for Simple Polygons

We will show that the algorithm reported in Section 3.1 can be used to solve all constraint problems for simple polygons. Let $P_0$, $P_1$, $\ldots$, $P_{n-1}$ be $n$ points in the plane. The constraint problem for *simple polygon* with $P_i$ as vertices consists of two types of constraints only:

1. Edge constraints, i.e., $|P_iP_{i+1}|$ is known, where the subscripts are understood to be mod $n$.
2. Angle constraints, i.e., the angle formed by $P_iP_{i+1}$ and $P_jP_{j+1}$ is known.

In [17], it is pointed out that all constraint problems about simple polygons can be solved by the parallelogram rules. This method is based on deductive rules and has no complexity analysis. Our algorithm is a graph based theory and is of quadratic complexity.

We first consider well-constrained problems for simple polygons which have $2n - 3$ constraints. We divide the problem into three cases.

1. The problem has $n$ edge constraints and $n - 3$ angle constraints. Since the lengths of all edges are known, we may construct the diagram with sas and parallel transformations.
2. The problem has $n - 1$ edge constrains and $n - 2$ angle constraints. Since the length of one edge is unknown, parallel transformations may be used to each pair of the edges. After a series of parallel and sas transformations, the problem becomes one of the following cases.
   (a) Both of the two unknown angles involving the edge whose length is unknown, Fig. 10(a). This is a problem without loops.
   (b) One of the two unknown angles involving the edge whose length is unknown, Fig. 10(b). This problem may be solved with one sas transformation.
   (c) The lengths for all the edges involving the two unknown angles are known, Fig. 10(c). This problem may be solved as follows. By an angle transformation, the angle formed by $l_1$ and $l_2$ is known. Now, this problem can be solved by a parallel transformation. Notice that this problem can also be solved by the methods in [2,16,3].

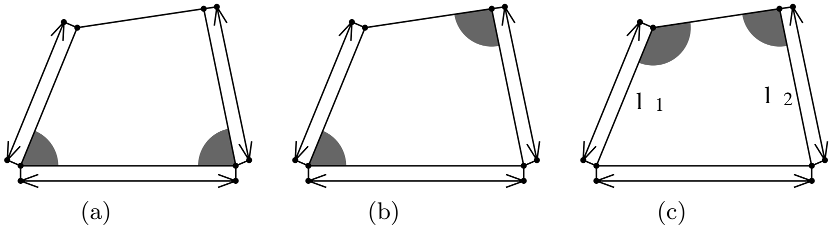


**Fig. 10.** Constraint problems for simple polygons

3. The problem has $n - 2$ edge constraints and $n - 1$ angle constraints. Using angle transformations, angles formed by any two edges are known. The lengths of two edges are unknown. These two edges divide the vertices into two sets such that two vertices in the same set can be connected with a series of edges with known lengths. Therefore, we may use parallel transformations to the two sets. By a series of parallel and sas transformations, the problem becomes one of the following cases.
   (a) The two edges with unknown lengths have a common end point, Fig. 11(a). This is a problem without loops.
   (b) The two edges with unknown lengths have no common end point, Fig. 11(b). This problem can be solved similar to Fig. 9 by a parallel transformation about $l_1$ and $l_2$ since the angle between them are known. Note

that in the current case, $|P_1P_2|$ in Fig. 9 is not known. But we may still use the parallel transformation to solve it. We may add this diagram as a special case to our general algorithm. This problem can also be solved by the methods in [2,16,3].

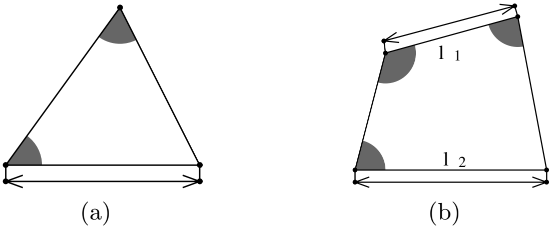


**Fig. 11.** Constraint problems for simple polygons

For an under-constrained problem, we may first transform it into a well-constrained problem by adding a proper number of constraints [13], and then solve it with the method presented in this paper.

## 4 Conclusion

The algorithm proposed in this paper uses a hybrid method by combining the graph analysis approach and the rule-based approach. As a result, it has the advantage of both approaches: it is fast and capable of solving a large number of constraint problems with loops. Actually, it is complete for all constraint problems of simple polygons. We believe that this approach has more potential due to the fact that there should exist many techniques of geometric constructions in the classical geometry, which are useful to GCS. One possible extension is to introduce more transformations such that the algorithm may solve all the problems within the scope of Hoffmann-Owen's triangle decomposition approaches and is still of quadratic complexity.

## References

1. Dufourd, J.-F., Mathis, P., and Schreck, P.: Geometric Construction by Assembling Solved Subfigures. *Artificial Intelligence*, **99**(1998), 73–119.
2. Fudos, I. and Hoffmann, C. M.: A Graph-Constructive Approach to Solving Systems of Geometric Constraints. *ACM Transactions on Graphics*, **16**(1997), 179–216.
3. Gao, X.-S. and Chou, S.-C.: Solving Geometric Constraint Systems I. A Global Propagation Approach. *Computer Aided Design*, **30**(1998), 47–54.
4. Gao, X.-S. and Chou, S.-C.: Solving Geometric Constraint Systems II. A Symbolic Approach and Decision of Rc-constructibility. *Computer Aided Design*, **30**(1998), 115–122.

5. Ge, J.-X., Chou, S.-C., and Gao, X.-S.: Geometric Constraint Satisfaction Using Optimization Methods. *Computer Aided Design*, **31**(1999), 867–879.
6. Hendrickson, B.: Conditions for Unique Realizations. *SIAM J. Computing*, **21** (1992), 65–84.
7. Heydon, A. and Nelson, G.: The Juno-2 Constraint-Based Drawing Editor. *SRC Research Report 131a* (1994).
8. Hoffmann, C. M., Lomonosov, A., and Sitharam, M.: Finding Solvable Subsets of Constraint Graphs. *LNCS*, vol. 1330, Springer-Verlag, Berlin Heidelberg (1997), 163–197.
9. Joan-Arinyo, T. and Soto-Riera, A.: Combining Constructive and Educational Geometric Constraint-Solving Techniques. *ACM Trans. on Graphics*, **18**(1999), 35–55.
10. Kondo, K.: Algebraic Method for Manipulation of Dimensional Relationships in Geometric Models. *Computer Aided Design*, **24**(1992), 141–147.
11. Kramer, G.: *Solving Geometric Constraint Systems.* MIT Press, Cambridge (1992).
12. Lamure, H. and Michelucci, D.: Solving Geometric Constraints by Homotopy. *IEEE Trans on Visualization and Computer Graphics*, **2**(1996), 28–34.
13. Latheam, R. S. and Middleditch, A. E.: Connectivity Analysis: A Tool for Processing Geometric Constraints. *Computer Aided Design*, **28**(1994), 917–928.
14. Lee, J. Y.: A 2D Geometric Constraint Solver for Parametric Design Using Graph Reduction and Analysis. *Automated Deduction in Geometry*, *LNAI*, vol. 1669, Springer-Verlag, Berlin Heidelberg (1999), 258–274.
15. Light, R. and Gossard, D.: Modification of Geometric Models Through Variational Geometry. *Geometric Aided Design*, **14**(1982), 208–214.
16. Owen, J.-C.: Algebraic Solution for Geometry from Dimensional Constraints. *Proc. 1st Symp. Solid Modeling Foundations & CAD/CAM Applications*, ACM Press, New York (1991), 379–407.
17. Verroust, A., Schonek, F., and Roller, D.: Rule-Oriented Method for Parameterized Computer-Aided Design. *Computer Aided Design*, **24**(1992), 531–540.
18. Sunde, G.: Specification of Shape by Dimensions and Other Geometric Constraints. *Geometric Modeling for CAD Applications*, M. J. Wozny et al., eds., North Holland (1988), 199–213.
19. Wang, D.: Reasoning about Geometric Problems Using an Elimination Method. *Automated Practical Reasoning: Algebraic Approaches*, Pfalzgraf, J. and Wang, D., eds., Springer, Wien New York (1995), 147–185.
20. Wu,W.-T.: *Mechanical Theorem Proving in Geometries:Basic Principles*. Springer-Verlag, Wien New York (1994).

# Solving the Birkhoff Interpolation Problem via the Critical Point Method: An Experimental Study

Fabrice Rouillier[1], Mohab Safey El Din[2], and Éric Schost[3]

[1] LORIA, INRIA-Lorraine, Nancy, France
Fabrice.Rouillier@loria.fr
[2] CALFOR, LIP6, Université Paris VI, Paris, France
Mohab.Safey@lip6.fr
[3] Laboratoire GAGE, École Polytechnique, Palaiseau, France
schost@gage.polytechnique.fr

**Abstract.** Following the work of Gonzalez-Vega, this paper is devoted to showing how to use recent algorithmic tools of computational real algebraic geometry to solve the *Birkhoff Interpolation Problem.* We recall and partly improve two algorithms to find at least one point in each connected component of a real algebraic set defined by a single equation or a system of polynomial equations, both based on the computation of the critical points of a distance function.

These algorithms are used to solve the Birkhoff Interpolation Problem in a case which was known as an open problem. The solution is available at the U.R.L.: `http://www-calfor.lip6.fr/~safey/applications.html`.

## 1 Introduction

The problem of interpolating a function $f : \mathbb{R} \longrightarrow \mathbb{R}$ by a univariate polynomial from the values of $f$ and some of its derivatives on a set of sample points is one of the main questions in Numerical Analysis and Approximation Theory.

Let $\chi = \{x_1, \ldots, x_n\}$ be a set of real numbers such that $x_1 < \cdots < x_n$, $r$ an integer, and let $\mathcal{I} \subset \{1, \ldots, n\} \times \{0, \ldots, r\}$ be the set of pairs $(i, j)$ such that the value $f_{i,j} = f^{(j)}(x_i)$ is known. The problem of determining the existence and uniqueness of a polynomial $Q$ in $\mathbb{R}[X]$ of degree bounded by $r$ such that

$$\forall (i, j) \in \mathcal{I}, \quad Q^{(j)}(x_i) = f_{i,j}$$

is called the *Birkhoff Interpolation Problem.*

In [17], Gonzalez-Vega focuses on determining, for fixed integers $n$ and $r$, the family of $\mathcal{I}$'s for which this question is solvable for any choice of $\chi$ and the values $f_{i,j}$. To this end, he shows that the problem can be reduced to deciding if some hypersurfaces contain real points with non-zero coordinates. In [17] the cases $n = 2$, $r \in \{1, \ldots, 6\}$, $n = 3$, $r \in \{1, 2, 3\}$ and $n = 4$, $r \in \{1, \ldots, 4\}$ are solved, using techniques adapted from the Cylindrical Algebraic Decomposition.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 26–40, 2001.

In 1998, the case $n = 5$ and $r = 4$ was presented as an open problem in [18]. The aim of the paper is to show how we have solved this case.

The most popular algorithm deciding the emptiness of semi-algebraic sets — as a particular case of deciding the truth of a first order formula — is Collins' Cylindrical Algebraic Decomposition (CAD) [12,11] whose complexity is doubly exponential in the number of variables in terms of basic arithmetic operations and size of output.

In Grigoriev and Vorobjov's paper [16] appeared new algorithms, based on the critical point method. These algorithms have a single exponential complexity in the number of variables, in terms of basic arithmetic operations and size of output. Still, in [20], Hong shows that the algorithms proposed in the papers [24] and [16] are not usable in practice. According to the experiments in [31], the same conclusions apply for more recent methods given in [9,30,19]. These algorithms adopt strategies of the following kind:

- In the first place, solve the problem in favorable cases, such as a compact and smooth variety.
- Get back from general situations to the favorable cases using various tricks, such as infinitesimal deformations or sums of squares.

The papers of the TERA group [7,8] treat the case of a smooth complete intersection variety, and propose an algorithm based on evaluation techniques, whose complexity is polynomial in terms of intrinsic real degrees. Still, these favorable situations, in particular compactness, are not easily detectable, and systematically applying the tricks above makes the computations difficult in practice.

In the papers [28,6], two algorithms inspired by the ideas in [16,19,24,9,30] are proposed. Both of them are based on the computation of the critical points of a distance function (thus avoiding the hypothesis of compactness) and improve the aforementioned algorithms. The first algorithm [28] computes at least one point in each connected component of a real algebraic set defined by a single equation. The second [6] applies to a real algebraic set defined by a polynomial system, in the spirit of [10,13]. The experiments in [28,6] show that these strategies are competitive with the CAD on small examples and allow to deal with more significant ones, unreachable with the CAD.

In this paper, we pursue the investigations of [20] by analyzing the practical behavior of these two recent algorithms on the Birkhoff Interpolation Problem. In sections 3, 4 and 5, we recall the algorithm given in [28], our contribution being a new way to solve a system with infinitesimal parameters arising in its course, and then the algorithm given in [6]. We conclude this paper with our experimental results, which solve the case $n = 5$, $r = 4$ of Birkhoff's problem. This gives us the opportunity to compare the size of the outputs and computational times of both algorithms.

Throughout this paper, the base field is the rational field $\mathbb{Q}$. The algorithms presented here generalize to the case of an ordered field $K$, replacing the field $\mathbb{R}$ by the real closure of $K$ and the complex field $\mathbb{C}$ by its algebraic closure.

## 2 The Birkhoff Interpolation Problem

### 2.1 Formulation

We want to determine the sets $\mathcal{I}$ for which the Birkhoff Problem Interpolation admits a unique solution for all choices of $\chi$ and of the $f_{i,j}$. To this end, we follow closely [17], and adopt its convenient matricial formulation.

Consider the matrix $\mathcal{E} = (e_{i,j})$ with $n$ rows and $r+1$ columns [17], filled with 0's and 1's, such that $e_{i,j} = 1$ if and only if $(i,j) \in \mathcal{I}$. The problem admits a solution only if $\mathcal{E}$ has as many 1's as columns. This amounts to saying that the coefficients of the interpolating polynomial $Q$ are solution of a linear *square* system, with associated matrix $\mathcal{M}_{\mathcal{E}}$. This matrix is parametrized by $\chi$ and its shape depends on $\mathcal{E}$. We are interested in determining the matrices $\mathcal{E}$ for which the determinant of $\mathcal{M}_{\mathcal{E}}$ is non-zero for all $\chi$, in which case the matrix $\mathcal{E}$ is said to be *poised.*

*Example 1.* Let $n = 4$ and $r = 3$ and consider the matrix:

$$\mathcal{E} = \begin{pmatrix} 1\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,1\,0\,0 \end{pmatrix}$$

Let $Q(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ be the generic polynomial of degree 3. Writing $Q^{(j)}(x_i) = f_{i,j}$ if and only if $e_{i,j} = 1$, we obtain the following linear system:

$$\begin{cases} a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 = f_{1,1} \\ a_1 + 2a_2 x_1 + 3a_3 x_1^2 = f_{1,2} \\ a_1 + 2a_2 x_3 + 3a_3 x_3^2 = f_{3,2} \\ 2a_2 + 6a_3 x_2 = f_{3,2} \end{cases}$$

whose matrix is:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_3 & 3x_3^2 \\ 0 & 0 & 2 & 6x_2 \end{pmatrix}$$

The interpolation problem is solvable if and only if

$$12x_3x_2 + 6x_1^2 - 12x_2x_1 - 6x_3^2$$

does not vanish for all values $x_1, x_2, x_3$ satisfying $x_1 < x_2 < x_3$.

In [17], Gonzalez-Vega shows that the question can be reduced to testing if a particular factor of the determinant of the matrix $\mathcal{M}_{\mathcal{E}}$ has real roots with non-zero coordinates. Replacing $(x_1, \ldots, x_n)$ by $(x_1, x_1 + t_1^2, \ldots, x_1 + t_1^2 + \ldots + t_{n-1}^2)$ yields a homogeneous polynomial in $(t_1, \ldots, t_{n-1})$. Letting $t_1 = 1$, we are brought to test if a hypersurface defined by a polynomial $P \in \mathbb{R}[t_2, \ldots, t_{n-1}]$ has real roots with non-zero coordinates.

### 2.2 Sketch of the Resolution

In order to determine all the poised matrices in the case $n = 5$ and $r = 4$, we have to study quasi-algebraic sets defined by a unique equation in $\mathbb{R}[t_2, t_3, t_4]$ and several inequations. While algorithms deciding the emptiness of real algebraic sets have known recent significant progress [13,28,6], more algorithmic work is necessary in the semi-algebraic case [9].

We thus treat this question using and adapting the algorithms for the algebraic case described in [28] and [6]. The algorithm described in [28] will be named **Algorithm 1**. It takes as input a single polynomial equation and returns at least one point in each connected component of the real algebraic variety defined by the equation. The algorithm described in [6] will be named **Algorithm 2**. It takes as input a polynomial system of equations and returns at least one point in each connected component of the real algebraic variety defined by the system.

To solve our problem, given a polynomial $P$ in $\mathbb{Q}[t_2, t_3, t_4]$, we adopt the following scheme.

*First Step.* We study the hypersurface defined by $P = 0$, using either Algorithm 1 or Algorithm 2. If this subroutine returns no real point, or a real point with non-zero coordinates, we can give a positive (resp. negative) answer to the Interpolation Problem.

*Second Step.* We are in the case when all the real points we obtained have at least one coordinate equal to zero. Using Algorithm 1, we study the hypersurface defined by $P^2 + (Tt_1t_2t_3 - 1)^2 = 0$; when Algorithm 2 we study the polynomial system $P = 0$ and $Tt_1t_2t_3 - 1 = 0$.

Note that in some situations, we can avoid such extra computations, using for example the following result:

**Lemma 1.** *Let $P \in \mathbb{R}[t_1, \ldots, t_n]$ and $\mathcal{V}(P)$ the hypersurface defined by $P = 0$. Let $M = (\mu_1, \ldots, \mu_n) \in \mathcal{V}(P) \cap \mathbb{R}^n$ such that $\mathbf{grad_M}(P) \neq \mathbf{0}$, and $I \subset \{1, \ldots, n\}$ the set of indexes for which $\mu_i$ is zero. If $\mathbf{grad_M}(P)$ is collinear to none of the axes $(t_i)_{i \in I}$, then there exists a point $M'$ in $\mathcal{V}(P) \cap \mathbb{R}^n$ with non-zero coordinates.*

## 3 Preliminary Results

This section describes the basics of the critical point method, valid for both Algorithm 1 and Algorithm 2.

Let $P$ be a square-free polynomial in $\mathbb{Q}[x_1, \ldots, x_n]$, and $\mathcal{V}(P) \subset \mathbb{C}^n$ the complex variety defined by $P = 0$. Our strategy to compute at least one point in each connected component of $\mathcal{V}(P) \cap \mathbb{R}^n$ relies on the computation of the critical points on $\mathcal{V}(P) \cap \mathbb{R}^n$ of a "distance function". Given a point $\mathbf{A} = (a_1, \ldots, a_n)$ in $\mathbb{R}^n$, the function $d_{\mathbf{A}} : \mathbf{M} \mapsto ||\mathbf{AM}||^2$ admits a minimum on each connected component of $\mathcal{V}(P) \cap \mathbb{R}^n$. These minima are solutions of the system

$$S(P, \mathbf{A}) = \big\{P(\mathbf{M}) = 0,\ \mathbf{grad_M}(P) // \mathbf{AM}\big\},$$

where the condition $\mathbf{grad_M}(P)//\mathbf{AM}$ is expressed by setting the determinants to zero. The set of all complex roots of this system is denoted by $\mathcal{C}(P, A)$.

Two cases will be distinguished, according to the dimension of the singular locus of $\mathcal{V}(P)$. The following result from [28] deals with the first case, where there is a finite number of singularities:

**Theorem 1.** *Let $P$ be a square-free polynomial in $\mathbb{Q}[x_1, \ldots, x_n]$. If $\mathcal{V}(P)$ contains a finite number of singular points, there exists a Zariski-dense set $\mathcal{F}' \subset \mathbb{C}^n$ such that for $\mathbf{A}$ in $\mathcal{F}'$, the system $S(P, \mathbf{A})$ is zero-dimensional.*

*Moreover, if $\mathcal{V}(P)$ is a smooth hypersurface, there exists a Zariski-dense set $\mathcal{F} \subset \mathbb{C}^n$ such that for $\mathbf{A}$ in $\mathcal{F}$, the system $S(P, \mathbf{A})$ is zero-dimensional and radical.*

Hence, if $\mathcal{V}(P)$ contains a finite number of singular points, one can choose a point $\mathbf{A}$ such that $S(P, \mathbf{A})$ is zero-dimensional. Since $S(P, \mathbf{A})$ intersects each semi-algebraically connected component of $\mathcal{V}(P) \bigcap \mathbb{R}^n$, the problem is reduced to isolate the real roots of $S(P, \mathbf{A})$.

Throughout this paper, we will represent the solutions of a zero-dimensional system with coefficients in a field $k$ using primitive element techniques, in a way that can be traced back to Kronecker [22]. Such a representation consists in:

- a linear form $u = \sum u_i x_i$ which separates the zeros of the system[1];
- its minimal polynomial $q$ in $k[t]$ and the parameterizations $(v_1, \ldots, v_n)$ in $k[t]^n$, where $\deg v_i < \deg q$, such that the zeros of the system are described by
$$q(t) = 0, \quad \begin{cases} \widetilde{q}(t) x_1 = v_1(t), \\ \qquad \vdots \\ \widetilde{q}(t) x_n = v_n(t), \end{cases}$$
where $\widetilde{q}(t)$ is the derivative of the square-free part of $q(t)$. We will denote this kind of representation $(u, \mathcal{R})$, where $\mathcal{R}$ is the vector $[q, v_1, \ldots, v_n]$.

Modern presentations and algorithms include the Rational Univariate Representation [26,25] or the Geometric Resolution [14,15], which coincide in the case of radical ideals of dimension zero. Such representations are useful, in that they allow to count and isolate the real roots of the system using for instance Sturm-Habicht sequences [30].

When the singular locus of $\mathcal{V}(P)$ is of positive dimension, difficulties arise, as the system $S(P, \mathbf{A})$ is also of positive dimension for any choice of the point $\mathbf{A}$. In the following sections, we focus on this case, and present in turn the strategies used in Algorithms 1 and 2.

- The algorithm described in [28] performs an infinitesimal deformation on the hypersurface $\mathcal{V}(P)$ to get back to a smooth situation; the required information is extracted from the solution of the deformed problem.
- The approach from [6] is based on the iterated study of singular loci, as varieties of lower dimension.

[1] The image of two distinct zeros by $u$ are distinct.

## 4 Algorithm 1: Using Infinitesimal Deformations

In the first subsection, we recall the main steps of the algorithm in [28], to which we refer for further details. We then present our solution to the specific subproblem of computing a univariate representation depending on the deformation parameter.

### 4.1 Overview of the Algorithm

Let $\varepsilon$ be an infinitesimal; we denote by $\mathbb{C}\langle\varepsilon\rangle$ the algebraically closed field of algebraic Puiseux series with coefficients in $\mathbb{C}$. The sub-ring of elements of non-negative valuation (called "bounded elements") is naturally equipped with the operation denoted $\lim_{\varepsilon\to 0}$. If $\mathcal{C}$ is a set of elements of $\mathbb{C}\langle\varepsilon\rangle$, $\lim_{\varepsilon\to 0}\mathcal{C}$ denotes the set of the limits of the bounded of elements of $\mathcal{C}$. Finally, if $x$ is $\sum_{i\geq i_0} a_i\varepsilon^{i/q} \in \mathbb{C}\langle\varepsilon\rangle$, where $i_0 \in \mathbb{Z}, q \in \mathbb{Q}, a_i \in \mathbb{C}$, we denote by $o(x)$ the rational number $i_0/q$.

The following result [28] shows that the study of $S(P-\varepsilon, \mathbf{A})$ enables to solve the original problem.

**Proposition 1.** *The set* $(\lim_{\varepsilon\to 0}\mathcal{C}(P-\varepsilon, \mathbf{A})) \cap \mathbb{R}^n$ *intersects each connected component of* $\mathcal{V}(P)\cap\mathbb{R}^n$.

Since $\mathcal{V}(P-\varepsilon) \subset \mathbb{C}\langle\varepsilon\rangle^n$ is smooth, Theorem 1 implies that for a generic choice of the point $\mathbf{A}$, the system $S(P-\varepsilon, \mathbf{A})$ is radical of dimension zero. In this case, its solutions can be described by a univariate representation $(u, \mathcal{R})$, with coefficients in $\mathbb{Q}(\varepsilon)$. The paper [28] then gives an algorithm to compute the limits of the bounded solutions it describes, when $u$ is a *well separating element.* This is the case when:

- for all $\alpha \in \mathcal{C}(P-\varepsilon, \mathbf{A})$, $o(u(\alpha)) = \min(o(X_i(\alpha)), i = 1, \ldots, n)$,
- for all $(\alpha, \beta) \in \mathcal{C}(P-\varepsilon, \mathbf{A})^2$, $u(\alpha)$ and $u(\beta)$ are infinitesimally close if and only if $\alpha$ and $\beta$ are infinitesimally close.

Indeed, from a univariate representation associated to a well separating element

$$q(\varepsilon, t) = 0, \begin{cases} \widetilde{q}(\varepsilon, t)x_1 = v_1(\varepsilon, t), \\ \qquad\vdots \\ \widetilde{q}(\varepsilon, t)x_n = v_n(\varepsilon, t), \end{cases}$$

if $q_1 q_2^2 \ldots q_m^m$ is the square-free decomposition of $q(0,t)$, then, $\forall j \in \{1, \ldots, m\}$, one can compute polynomials $\widetilde{q}^{(j)}$ and $v_i^{(j)}$ such that the limits of the bounded solutions are represented by

$$\left( q_j(t) = 0, \begin{cases} \widetilde{q}^{(j)}(0, t)x_1 = v_1^{(j)}(0, t), \\ \qquad\vdots \\ \widetilde{q}^{(j)}(0, t)x_n = v_n^{(j)}(0, t). \end{cases} \right)_{j\in\{1,\ldots,m\}}$$

We can now give the first algorithm, which is the synthesis of all the previous points:

**Algorithm 1**

```
Input: A squarefree polynomial $P$
Output: At least one point on each connected component of $\mathcal{V}(P)$

1. Find by trial and error a point $\mathbf{A}$ such that $S(P-\varepsilon,\mathbf{A})$ is zero-dimensional and radical.
2. Compute a parametric resolution $(u,\mathcal{R})$ of its roots, with coefficients in $\mathbb{Q}(\varepsilon)$.
3. If $u$ is a well-separating element for $S(P-\varepsilon,\mathbf{A})$, compute the limits of the bounded solutions described by $\mathcal{R}$ as $\varepsilon \to 0$.
4. else change $u$, check if it is a separating element for $S(P-\varepsilon,\mathbf{A})$ and return to step 3.
```

We detail a solution to point 1 above. In [28] the authors have proved the following results:

**Lemma 2.** *If $G$ is a Gröbner basis of the system $S(P-\varepsilon,\mathbf{A})$ in $\mathbb{Q}[\varepsilon,x_1,\ldots,x_n]$, for a block-ordering such that $[\varepsilon]<[x_1,\ldots,x_n]$, then $G$ is a non-reduced Gröbner basis of $S(P-\varepsilon,\mathbf{A})$ in $\mathbb{Q}\langle\varepsilon\rangle[x_1,\ldots,x_n]$.*

*If there exists a value $\varepsilon_0$ in $\mathbb{Z}$ which doesn't cancel any of the leading coefficients of the polynomials in $G$, and such that the system $S(P-\varepsilon_0,\mathbf{A})$ is radical of dimension zero, then $S(P-\varepsilon,\mathbf{A})$ is radical of dimension zero. Moreover, for such $\varepsilon_0$ and $\mathbf{A}$, if $u$ is a separating element for $S(P-\varepsilon_0,\mathbf{A})$ then $u$ is a separating element for $S(P-\varepsilon,\mathbf{A})$.*

Given any point $A$ that fits the hypothesis of Theorem 1, one can show that only a finite number of $\varepsilon_0$ will not fit the conditions of Lemma 2. Hence, a simultaneous search by trial and error of $\mathbf{A}\in\mathbb{Z}^n$ and $\varepsilon_0\in\mathbb{Z}$ can be performed to obtain a point $\mathbf{A}$ such that $S(P-\varepsilon,\mathbf{A})$ is radical. For a given $\mathbf{A}$, this requires to compute the basis $G$; testing that $S(P-\varepsilon_0,\mathbf{A})$ is radical can be done using Hermite's quadratic form (see [26] for example).

### 4.2 Computing a Parametric Resolution

We now turn to the second of the tasks mentioned above, namely computing a resolution parametrized by $\varepsilon$. In [33], Schost proposes a probabilistic algorithm to do so, based on the work of the TERA group [14,3,15]. The algorithm relies on a formal Newton approximation process, which is an analog of numerical root-finding techniques, and reminiscent of Hensel lifting methods. In the sequel, we first recall the main steps of this algorithm, then provide solutions to certify its output in our case (radical and zero-dimensional ideal).

*Overview of the Algorithm.* For a random choice of $\varepsilon_0$ in $\mathbb{Q}$, given any resolution $(u,\mathcal{R}_0)$ of the system $S(P-\varepsilon_0,\mathbf{A})$, there exists a resolution $(u,\mathcal{R})$ of the system $S(P-\varepsilon,\mathbf{A})$ with coefficients in $\mathbb{Q}(\varepsilon)$ whose specialization at $\varepsilon_0$ is $(u,\mathcal{R}_0)$. The strategy presented in [33] consists in approximating the solution $(u,\mathcal{R})$ starting from $(u,\mathcal{R}_0)$.

The output $\mathcal{R}$ can be rewritten in terms of $\varepsilon' = \varepsilon - \varepsilon_0$, as a vector $\mathcal{R}'$ of polynomials in $\mathbb{Q}(\varepsilon')[t]$, where none of the denominators of the expressions in $\varepsilon'$ vanishes at zero. Denote by $\mathcal{R}'_i$ the vector of polynomials of $\mathbb{Q}[[\varepsilon']][t]$, where all coefficients are expanded at precision $2^i$. The initial value is obtained by solving the system $S(P - \varepsilon_0, \mathbf{A})$; the formal Newton operator, denoted $\mathtt{Lift}(\mathcal{R}'_i)$ computes $\mathcal{R}'_{i+1}$ from the argument $\mathcal{R}'_i$.

The whole algorithm is organized around a while loop. Each pass begins by computing the resolution $\mathcal{R}'_{i+1}$, of precision $2^{i+1}$. The `RationalReconstruction` subroutine then computes Padé approximants [35] of all the coefficients, as rational functions in $\varepsilon'$, with numerators and denominators of degree at most $2^i$; a boolean value $b$ indicates success. If the reconstruction is possible, the subroutine `StopCriterion`, detailed below, tests if the resolution, rewritten in terms of $\varepsilon$, is correct. If this is not the case, we go trough another loop.

*A Certified Result.* The probabilistic aspect of the algorithm in [33] is twofold: it lies in the choice of the specialization value $\varepsilon_0$, and in the test `StopCriterion`. We provide here certified versions of these subroutines.

The value $\varepsilon_0$ must satisfy some genericity conditions: the system $S(P - \varepsilon_0, \mathbf{A})$ must be zero-dimensional, its roots must be simple and in maximal number. The algorithm in [33] is based on the fact that all choices of $\varepsilon_0$ but a finite number fulfill these conditions, and that a bound on the number of bad values is readily available. Instead, we use the following result: if $\varepsilon_0$ cancels none of the leading coefficients of the polynomials in the basis $G$ computed above, and if the system $S(P - \varepsilon_0, \mathbf{A})$ is radical, then $\varepsilon_0$ satisfies the genericity conditions.

Finally, to check that a solution $(u = \sum u_i x_i, \mathcal{R})$ is the correct solution, it is enough to check that the minimal polynomial and the parameterizations in $\mathcal{R}$, with $t$ evaluated at $\sum u_i x_i$, reduce to zero modulo the basis $G$. This implies that the resolution $\mathcal{R}$ describes a set of points containing $\mathcal{C}(P - \varepsilon, \mathbf{A})$. As these two sets have the same cardinality, we are done.

The output is a list of polynomials in $\mathbb{Q}(\varepsilon)[t]$. The degree in $t$, the degrees in $\varepsilon$ of the numerators and the denominators of the coefficients are bounded by the Bézout number $d^n$, where $d$ is the degree of the polynomial $P$ [33]. The "size" of the output is thus $O(n(d+1)^{2n})$ elements of the base-field $\mathbb{Q}$.

The details of the computations done in Newton's operator are given in [15,33]. Following the usual numeric Newton operator, they rely on the evaluation of the vector $\mathbf{Jac}(S(P - \varepsilon, \mathbf{A}))^{-1} S(P - \varepsilon, \mathbf{A})$ in suitable quotient rings, where $\mathbf{Jac}(S)$ denotes the jacobian matrix of the system $S$.

**Computing the parametric resolution**

```
Input: a point $\mathbf{A}$ such that the roots of the system $S(P-\varepsilon,\mathbf{A})$ are simple.
       a Gröbner basis $G$ of $S(P-\varepsilon,\mathbf{A})$ in $\mathbb{Q}\langle\varepsilon\rangle[x_1,\ldots,x_n]$.
Output: a parametric resolution $(u,\mathcal{R})$ of $S(P-\varepsilon,\mathbf{A})$

1. Find by trial and error a random rational number $\varepsilon_0$ such that the system
   $S(P-\varepsilon_0,\mathbf{A})$ is zero-dimensional and has simple roots, in maximal number.
2. Compute a univariate representation $\mathcal{R}'_0$ of the roots of $S(P-\varepsilon_0,\mathbf{A})$
3. Use the Newton-Hensel lifting process to compute the successive approximations $\mathcal{R}'_i$:

   $i \leftarrow 0$; finished $\leftarrow$ false
   while not finished do
       $\mathcal{R}'_{i+1} \leftarrow$ Lift($\mathcal{R}'_i$)
       $b, \mathcal{R}' \leftarrow$ RationalReconstruction($\mathcal{R}'_{i+1}$)
       if $b$ then
           $\mathcal{R} \leftarrow$ Substitute $\varepsilon'$ by $\varepsilon+\varepsilon_0$ in $\mathcal{R}'$
           finished $\leftarrow$ StopCriterion($\mathcal{R}$)
       end if
       $i \leftarrow i+1$
   end while

4. return $(u,\mathcal{R})$
```

As in [7,15,8], we use the *Straight-Line Program* model to encode the input polynomial $P$. In this model, the complexity of the whole lifting process is polynomial in the size of the output, in the number of variables $n$, and the number of operations $L$ necessary to evaluate the polynomial $P$ [33]. Still, the whole algorithm requires the precomputation of the Gröbner basis $G$, and the subroutine `StopCriterion` relies on normal form computations. This part dominates the whole cost of the algorithm.

## 5 Algorithm 2: Iterated Study of Singular Loci

The second approach generalizes the critical point methods used here to the case of polynomial systems. In the presence of infinitely many singular points, the infinitesimal deformation is avoided by studying the singular locus as a variety of lower dimension.

Let $\mathcal{V} \subset \mathbb{C}^n$ be an equidimensional algebraic variety of dimension $d$ and $P = \{P_1,\ldots,P_s\}$ polynomials in $\mathbb{Q}[x_1,\ldots,x_n]$ such that $\mathcal{I}(\mathcal{V}) = (P_1,\ldots,P_s)$. Following the notation of the two previous sections, given any point $\mathbf{A}$ in $\mathbb{C}^n$, we define the polynomial system:

$$\begin{aligned} S(P,\mathbf{A}) = \big\{ & P_1(\mathbf{M}) = \ldots = P_s(\mathbf{M}) = 0, \\ & \operatorname{rank}(\mathbf{grad}_{\mathbf{M}}(P_1),\ldots,\mathbf{grad}_{\mathbf{M}}(P_s),\mathbf{AM}) \le n-d\big\}, \end{aligned}$$

where the rank condition is expressed by setting to zero the $(n-d+1)\times(n-d+1)$ minors of the matrix $[\mathbf{Jac}(P), \mathbf{AM}]$. The roots of this system form a set denoted $\mathcal{C}(\mathcal{V}, \mathbf{A})$.

The algorithm proposed in [6] is based on the following theorem:

**Theorem 2.** *The set* $\mathcal{C}(\mathcal{V}, \mathbf{A})$ *meets every connected component of* $\mathcal{V}\cap\mathbb{R}^n$*. Moreover, there exists a Zariski-dense subset* $\mathcal{F}$ *of* $\mathbb{C}^n$ *such that, for all* $\mathbf{A}$ *in* $\mathcal{F}$*,* $\mathcal{C}(\mathcal{V}, \mathbf{A})$ *can be written* $\mathrm{Sing}(\mathcal{V}) \cup \mathcal{V}_0$*, where*

- $\mathcal{V}_0$ *is a finite set of points in* $\mathbb{C}^n$*,*
- $\mathrm{Sing}(\mathcal{V}) = \{\mathbf{M} \in \mathcal{V} \mid \mathrm{rank}(\mathbf{grad}_{\mathbf{M}}(P_1), \ldots, \mathbf{grad}_{\mathbf{M}}(P_s)) < n-d\}$.

*In particular, in this case,* $\dim(\mathcal{C}(\mathcal{V}, \mathbf{A})) < \dim(\mathcal{V})$*.*

Suppose that $\mathcal{V} \subset \mathbb{C}^n$ and $\mathbf{A} \in \mathbb{R}^n$ satisfy the conditions of Theorem 2, and that $\mathcal{V}_1$ is a finite set of points that meets each connected component of $\mathrm{Sing}(\mathcal{V}) \cap \mathbb{R}^n$. Theorem 2 implies that $\mathcal{V}_1 \cup \mathcal{V}_0$ meets each connected component of $\mathcal{V} \cap \mathbb{R}^n$. The set $\mathcal{V}_1$ can in turn be obtained by applying Theorem 2 to each equidimensional component of $\mathrm{Sing}(\mathcal{V})$. The algorithm in [6] consists in applying inductively the above process, performing at each step equidimensional decompositions of intermediate varieties $\mathcal{C}(\mathcal{V}_i, \mathbf{A}_i)$. In the end, we obtain a family of zero-dimensional sets which meets each connected component of $\mathcal{V} \cap \mathbb{R}^n$.

At each step, we need to apply a subroutine taking as input a polynomial system $S$ and returning a set of generators of radical equidimensional ideals whose intersection is $\sqrt{S}$. This can be done using the algorithms mentioned in [4,31] or by performing a decomposition into regular and separable triangular sets [5,23,21,34] and computing a Gröbner basis of the saturated ideal of each triangular set. We denote by `EquiDimDecomposition` such a radical equidimensional decomposition algorithm.

**Algorithm 2**

**Input**: A polynomial system $P$
**Output**: At least one point on each connected component of $\mathcal{V}(P)$

```
1. list ← EquiDimDecomposition($P$), result ← []
2. while list /= [] do
     – $\widetilde{P}$ ← first(list) and remove $\widetilde{P}$ from list,
     – if $\dim(\widetilde{P}) = 0$ then result ← result ∪ $\widetilde{P}$
     – else find by trial and error a point $\mathbf{A}$ such that $\dim(\mathcal{C}(\mathcal{V}(\widetilde{P}), \mathbf{A})) < \dim(\widetilde{P})$
         and list ← list ∪ EquiDimDecomposition($S(\widetilde{P}, \mathbf{A})$)
3. count and isolate the real roots of all the polynomial systems in result.
```

## 6 Experimental Results

### 6.1 Methodology and Basic Algorithms

Both algorithms presented above have been implemented, using the following software tools:

- Gb/AGb: implemented in C++ by J.-C. Faugère [4] and devoted to Gröbner basis computations;
- RS: implemented in C by F. Rouillier, devoted to computing Rational Univariate Representations, and to counting and isolating real roots of univariate polynomials;
- Kronecker: implemented in Magma by G. Lecerf [15], devoted to computing Geometric Resolutions, from which we borrowed the formal Newton iterator.

The subroutine `EquiDimDecomposition` was implemented using Maple and a file connection with Gb, following the algorithm described in [31] and based on the results in [5,23].

All the computations were done on the computers of the UMS MEDICIS [2], on a PC Pentium II 400 MHz with 512 MB of RAM.

### 6.2 Solution of the Problem

The case $n = 5$, $r = 4$ of Birkhoff's problem generates 53130 matrices, which produces as many hypersurfaces of $\mathbb{C}^3$ to study.

- 42925 of these hypersurfaces are defined by constant polynomials.
- For the non-constant polynomials, to avoid unnecessary computations, we specialized all variables but one at random non-zero values and applied Uspensky's algorithm on the univariate polynomials we obtained, looking for non-zero real roots. At the end of this preprocessing, about one thousand hypersurfaces remained to study.
- About 900 of these hypersurfaces had zero or a finite number of singularities. In all these cases, the first step given in section 2.2 was sufficient to conclude: the situation where $\mathcal{C}(P, \mathbf{A})$ had exclusively real roots with a coordinate equal to zero was never encountered.

On a PC bi-Pentium 400 MHz with 512 MB of RAM, 4 hours are necessary to perform these 3 steps for all hypersurfaces.

- There remained 102 hypersurfaces containing an infinity of singularities. We will see below that our implementation of Algorithm 1 can not solve all of them, whereas Algorithm 2 succeeded in all cases. In 60 out of these 102 cases, we had to go through the second step given in section 2.2.
  On the same machine, 2 additional hours are necessary to perform this final step with Algorithm 2.

As a conclusion, 19092 out of the 53130 matrices are poised. Their complete list can be found in Maple format at the web page [1].

### 6.3 Comparing Algorithm 1 and Algorithm 2

We give a more detailed account of the behavior of Algorithms 1 and 2 on a sample of the family of hypersurfaces with infinitely many singularities. These hypersurfaces are denoted Birk.3-1,...,Birk3-15. All of them have degree less than 8; the whole list can be found at the web page [1].

Table 1 summarizes our results on this family. The sign $\infty$ indicates that the computations were stopped after 24 hours.

- **Algorithm 1:** The first column gives the number of bounded roots we obtain, which is a measure of the size of the output. The second and third columns give the degrees in $t$ and $\varepsilon$ of the generic resolution, which is a measure of the size of intermediate data. The last column gives the time necessary to perform all computations, in seconds.
- **Algorithm 2:** The first column gives the sum of the degrees of the zero-dimensional systems produced in the course of the computations; the second indicates the total time of computations, given in seconds.

**Table 1.** Algorithms 1/2: Size of the output and computation times

| **Hypersurface** | **Algorithm 1** | | | | **Algorithm 2** | |
|---|---|---|---|---|---|---|
| Birk.3-1 | 12 | 16 | 3 | 5.6 | 12 | 0,08 |
| Birk.3-2 | 7 | 16 | 3 | 5.2 | 7 | 0,13 |
| Birk.3-3 | 25 | 34 | 5 | 32 | 25 | 0,37 |
| Birk.3-4 | 16 | 36 | 5 | 46 | 16 | 0,18 |
| Birk.3-5 | 31 | 40 | 5 | 116 | 31 | 0,46 |
| Birk.3-6 | 37 | 52 | 7 | 149 | 37 | 0,86 |
| Birk.3-7 | 38 | 52 | 7 | 115 | 38 | 0,72 |
| Birk.3-8 | 45 | 130 | 19 | 3927 | 45 | 7,11 |
| Birk.3-9 | 47 | 132 | 19 | 2945 | 47 | 7,88 |
| Birk.3-10 | 48 | 136 | 31 | 18843 | 48 | 8,04 |
| Birk.3-11 | 50 | 138 | 31 | 26536 | 50 | 8,88 |
| Birk.3-12 | 50 | 138 | 31 | 17508 | 50 | 10,01 |
| Birk.3-13 | 32 | 252 | 29 | $\infty$ | 32 | 9,26 |
| Birk.3-14 | 60 | 264 | 31 | $\infty$ | 60 | 67 |
| Birk.3-15 | 60 | 272 | 31 | $\infty$ | 60 | 83 |

For the last examples, the time spent in Algorithm 1 in the checking phase becomes largely predominant. Other strategies to certify this output, based on a sufficient number of sample checks, could lower this time. Even *without* this certification phase, the computation is longer than with Algorithm 2. Still, with regard to the degrees in $t$ and $\varepsilon$ of the parametric resolution, we consider that our implementation of Algorithm 1 shows good performance.

A relevant criterion to analyze the algorithms based on the critical point method is the degrees of the zero-dimensional systems they produce. For Algorithm 1, this is the cardinality of the set of bounded roots $\lim_{\varepsilon \to 0} \mathcal{C}(P - \varepsilon, \mathbf{A})$. To this regard, the outputs of Algorithm 1 and Algorithm 2 are of similar size. The size of the intermediate data in Algorithm 1, such as the degree of the parametric resolution, is bigger, as several points of $\mathcal{C}(P - \varepsilon, \mathbf{A})$ collapse on a same point when $\varepsilon \to 0$.

Nevertheless, the degrees of the output and of the intermediate bivariate polynomials in Algorithm 1 are bounded by $d^n$, while we have no similar bound for Algorithm 2. An open problem is to precise such a bound for Algorithm 2. In all these examples, the dimension of the singular locus was 1, so that there was at most one recursive call in Algorithm 2. Experiments with more intricate singular loci should tell us more about this question.

## 7 Conclusions

The case $n = 5$ and $r = 4$ of the Birkhoff Interpolation Problem is now automatically solved. The case $n = 6$ and $r = 5$ requires to study 1947792 hypersurfaces in $\mathbb{C}^4$; this combinatorial number is now the limiting factor. More research on qualitative nature should be devised to have a better control on this number; in this sense, the conclusions and suggestions in [17] are still a topical question.

This problem gave us the opportunity to compare two recent algorithms of computational real algebraic geometry and illustrate their practical use. It appears that the algorithms based on the critical point method can now solve application problems.

In particular, we have implemented computations with an infinitesimal, considering it as a parameter. Another approach consists in implementing an infinitesimal arithmetic; we refer to [27] for such a realization in Axiom. Nevertheless, obtaining good performance in practice using this type of arithmetic is still a computer science challenge.

Besides, the use of infinitesimals in computational real algebraic geometry is not exclusive to the desingularization of hypersurfaces: they are required in several algorithms to decide the emptiness of semi-algebraic sets, such as [19,9,30].

## References

1. `http://www-calfor.lip6.fr/~safey/applications.html`
2. `http://www.medicis.polytechnique.fr`
3. `http://www.tera.medicis.polytechnique.fr`
4. `http://www-calfor.lip6.fr/~jcf`
5. P. Aubry, *Ensembles Triangulaires de Polynômes et Résolution de Systèmes Algébriques, Implantations en Axiom*, PhD thesis, Université Paris VI, 1999.
6. P. Aubry, F. Rouillier, M. Safey El Din, *Real Solving for Positive Dimensional Systems,* Research Report, Laboratoire d'Informatique de Paris VI, March 2000.
7. B. Bank, M. Giusti, J. Heintz, M. Mbakop, *Polar Varieties and Efficient Real Equation Solving*, Journal of Complexity, Vol. 13, pages 5–27, 1997; best paper award 1997.
8. B. Bank, M. Giusti, J. Heintz, M. Mbakop, *Polar Varieties and Efficient Real Elimination*, to appear in Mathematische Zeitschrift (2000).
9. S. Basu, R. Pollack, M.-F. Roy, *On the Combinatorial and Algebraic Complexity of Quantifier Elimination.* Journal of the Association for Computing Machinery, Vol. 43, pages 1002–1045, 1996.

10. E. Becker, R. Neuhaus, *Computation of Real Radicals for Polynomial Ideals*, Computational Algebraic Geometry, Progress in Math., Vol. 109, pages 1–20, Birkhäuser, 1993.
11. G. E. Collins, H. Hong, *Partial Cylindrical Algebraic Decomposition,* Journal of Symbolic Computation, Vol. 12, No. 3, pages 299–328, 1991.
12. G. E. Collins, *Quantifier Elimination for Real Closed Field by Cylindrical Algebraic Decomposition,* Lectures Notes in Computer Science, Vol. 33, pages 515–532, 1975.
13. P. Conti, C. Traverso, *Algorithms for the Real Radical,* Unpublished manuscript.
14. M. Giusti, J. Heintz, *La Détermination des Points Isolés et de la Dimension d'une Variété Algébrique Réelle peut se faire en Temps Polynomial*, Computational Algebraic Geometry and Commutative Algebra, Symposia Matematica, Vol. 34, D. Eisenbud and L. Robbiano (eds.), pages 216–256, Cambridge University Press, 1993.
15. M. Giusti, G. Lecerf, B. Salvy, *A Gröbner Free Alternative for Solving Polynomial Systems*, Journal of Complexity, Vol. 17, No. 1, pages 154–211, 2001.
16. D. Grigor'ev, N. Vorobjov, *Solving Systems of Polynomial Inequalities in Subexponential Time,* Journal of Symbolic Computation, Vol. 5, No. 1–2, pages 37–64, 1988.
17. L. Gonzalez-Vega, *Applying Quantifier Elimination to the Birkhoff Interpolation Problem,* Journal of Symbolic Computation Vol. 22, No. 1, pages 83–103, 1996.
18. M.-J. Gonzalez-Lopez, L. Gonzalez-Vega, *Project 2: The Birkhoff Interpolation Problem,* Some Tapas of Computer Algebra, A. Cohen (ed.), pages 297–310, Springer, 1999.
19. J. Heintz, M.-F. Roy, P. Solerno, *On the Theoretical and Practical Complexity of the Existential Theory of the Reals,* The Computer Journal, Vol. 36, No. 5, pages 427–431, 1993.
20. H. Hong, *Comparison of Several Decision Algorithms for the Existential Theory of the Reals*, Research Report, RISC-Linz, Johannes Kepler University, 1991.
21. M. Kalkbrener, *Three Contributions to Elimination Theory*, PhD thesis, RISC-Linz, Johannes Kepler University, 1991.
22. L. Kronecker, *Grundzüge einer arithmetischen Theorie der algebraischen Größen*, Journal Reine Angew. Mathematik, Vol. 92, pages 1–122, 1882.
23. M. Moreno Maza, *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Equations Algébriques,* PhD thesis, Université Paris VI, 1997.
24. J. Renegar, *On the Computational Complexity and Geometry of the First Order Theory of the Reals,* Journal of Symbolic Computation, Vol. 13, No. 3, pages 255–352, 1992.
25. F. Rouillier, *Algorithmes Efficaces pour l'Étude des Zéros Réels des Systèmes Polynomiaux,* PhD thesis, Université de Rennes I, 1996.
26. F. Rouillier, *Solving Zero-Dimensional Systems through the Rational Univariate Representation,* Applicable Algebra in Engineering Communications and Computing, Vol. 9, No. 5, pages 433–461, 1999.
27. R. Rioboo, *Computing with Infinitesimals*, Manuscript.
28. F. Rouillier, M.-F. Roy, M. Safey El Din, *Finding at Least One Point in Each Connected Component of a Real Algebraic Set Defined by a Single Equation,* Journal of Complexity, Vol. 16, No. 4, pages 716–750, 2000.
29. F. Rouillier, P. Zimmermann, *Efficient Isolation of a Polynomial Real Roots,* Research Report, INRIA, No. RR-4113, 2001.

30. M.-F. Roy, *Basic Algorithms in Real Algebraic Geometry: From Sturm Theorem to the Existential Theory of Reals,* Lectures on Real Geometry in memoriam of Mario Raimondo, Expositions in Mathematics, Vol. 23, pages 1–67, Berlin, 1996.
31. M. Safey El Din, *Résolution Réelle des Systèmes Polynomiaux en Dimension Positive*, PhD thesis, Université Paris VI, 2001.
32. É. Schost, *Computing Parametric Geometric Resolutions*, Preprint, École Polytechnique, 2000.
33. É. Schost, *Sur la Résolution des Systèmes Polynomiaux à Paramètres*, PhD thesis, École Polytechnique, 2000.
34. D. Wang, *Computing Triangular Systems and Regular Systems,* Journal of Symbolic Computation, Vol. 30, No. 2, pages 221–236, 2000.
35. J. Von Zur Gathen, J. Gerhardt, *Modern Computer Algebra,* Cambridge University Press, 1999.

# A Practical Program of Automated Proving for a Class of Geometric Inequalities*

Lu Yang[1] and Ju Zhang[2]

[1] Chengdu Institute of Computer Applications, Chinese Academy of Sciences
[2] GCTECH Info. Tech. Ltd., Beijing 100013, China

**Abstract.** An inequality-proving algorithm based on cell decomposition and a practical program written in Maple are presented, which can efficiently treat inequality-type theorems involving radicals, especially, a class of geometric inequalities including most of the theorems in a well-known book on the subject.

## 1 Introduction

In the last 20 years, the efficiency of automated theorem proving for equality-type theorems has increased greatly leaving behind inequality-type theorems, especially, for geometric theorems. This work is an effort to shorten the distance.

Automated theorem proving on inequalities is always considered a difficult topic in the area of automated reasoning. Relevant algorithms depend fundamentally on real algebra and real geometry, and the computational complexity increases very quickly with the dimension, that is, the number of parameters. Besides the memory saving, the speed improvement in theorem proving sometimes is also of importance. When a problem requires verification of a batch of non-trivial propositions, an inefficient algorithm cannot handle it within the time allowed by human patience. For recent progress made in this aspect, see [3,4,2,8,11,12,14,15].

When the hypotheses contain polynomial equations, one may think about eliminating some variables to make the dimension lower. However, we usually have to deal with irrational algebraic functions, such as parametric radicals.

**Example 1*a*.** Given real numbers $x, y, z, u_1, u_2, u_3, u_4, u_5, u_6$ satisfying the following 15 conditions

$$\begin{cases} (xy+yz+xz)^2u_1^2 - x^3(y+z)(xy+xz+4\,yz) = 0, \\ (xy+yz+xz)^2u_2^2 - y^3(x+z)(xy+yz+4\,xz) = 0, \\ (xy+yz+xz)^2u_3^2 - z^3(x+y)(yz+xz+4\,xy) = 0, \\ (x+y+z)(u_4^2 - x^2) - xyz = 0, \\ (x+y+z)(u_5^2 - y^2) - xyz = 0, \\ (x+y+z)(u_6^2 - z^2) - xyz = 0, \\ x > 0,\ y > 0,\ z > 0, \\ u_1 > 0,\ u_2 > 0,\ u_3 > 0,\ u_4 > 0,\ u_5 > 0,\ u_6 > 0, \end{cases} \tag{1}$$

prove that $u_1 + u_2 + u_3 \le u_4 + u_5 + u_6$.

* The work is supported in part by NKBRSF-(G1998030602). Lu Yang is concurrently at Guangzhou University, Guangzhou 510405, China.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 41–57, 2001.

Eliminating $u_1, \ldots, u_6$ from (1) by solving the 6 equations, we convert the proposition to the following inequality which appeared as a conjecture in reference [10].

**Example 1.** Show that

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

where $x > 0$, $y > 0$, $z > 0$.

This includes 3 variables but 6 radicals, while (1) includes 9 variables.

A dimension-decreasing algorithm introduced by the authors can efficiently treat parametric radicals and maximize reduction of the dimensions. Based on this algorithm, a generic program called "BOTTEMA" was implemented on a PC computer. More than 1000 algebraic and geometric inequalities including hundreds of open problems have been verified in this way. The total CPU time spent for proving 100 basic inequalities[1] from Bottema et al.'s monograph [1] "*Geometric Inequalities*" on a Pentium III/550 was 10-odd seconds only. It can be seen later that the inequality class, to which our algorithm is applicable, is very inclusive.

The paper is organized as follows. Section 2: illustrate fundamental notions with examples; Section 3: sketch an inequality-proving algorithm which can treat radicals efficiently; Section 4: introduce a transformation of variable which helps reduce the degrees of some polynomials concerned in proving a class of inequalities on triangles, that class contains most of the inequalities in book [1]; Section 5: commands and syntax for running the program BOTTEMA; Section 6: show the performance of the program with a series of examples; Section 7: conclusion.

## 2 Fundamental Definitions

Before we sketch the so-called dimension-decreasing algorithm, some definitions should be introduced and illustrated.

**Definition 1.** Assume that $l(x, y, z, \ldots)$ and $r(x, y, z, \ldots)$ are continuous algebraic functions of $x, y, z, \ldots$. We call

$$l(x, y, z, \ldots) \le r(x, y, z, \ldots) \quad \text{or} \quad l(x, y, z, \ldots) < r(x, y, z, \ldots)$$

[1] which include some classical results such as Euler's Inequality, Finsler-Hadwiger's Inequality, and Gerretsen's Inequality.

an algebraic inequality in $x, y, z \ldots$, and $l(x, y, z, \ldots) = r(x, y, z, \ldots)$ an algebraic equality in $x, y, z, \ldots$.

**Definition 2.** Assume that $\Phi$ is an algebraic inequality (or equality) in $x, y, z, \ldots$. $L(T)$ is called a *left polynomial* of $\Phi$, provided that

- $L(T)$ is a polynomial in $T$, its coefficients are polynomials in $x, y, z, \ldots$ with rational coefficients;
- the left-hand side of $\Phi$ is a zero of $L(T)$.

The following item is unnecessary for this definition, but it helps to reduce the computational complexity in the process later.

- Amongst all the polynomials satisfying the two items above, $L(T)$ is what has the lowest degree in $T$.

According to this definition, $L(T) = T$ if the left-hand side is 0, a zero polynomial. The *right polynomial* of $\Phi$, namely, $R(T)$, can be defined analogously.

**Definition 3.** Assume that $\Phi$ is an algebraic inequality (or equality) in $x, y, \ldots$ etc., $L(T)$ and $R(T)$ are the left and right polynomials of $\Phi$, respectively. By $P(x, y, \ldots)$ denote the resultant of $L(T)$ and $R(T)$ with respect to $T$, and call it the border polynomial of $\Phi$, and the surface defined by $P(x, y, \ldots) = 0$ *the border surface of* $\Phi$, respectively.

The notions of left and right polynomials are needed in practice for computing the border surface more efficiently. In Example 1, we set

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

then the left and right polynomials of (2) can be found by successive resultant computation:

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

Removing the factors which do not involve $T$, we have

$$\begin{aligned}
L(T) &= (x\,y + x\,z + y\,z)^8\,T^8 - 4(x^4\,y^2 + 2\,x^4\,y\,z + x^4\,z^2 + 4\,x^3\,y^2\,z + 4\,x^3\,y\,z^2\\
&\quad + x^2\,y^4 + 4\,x^2\,y^3\,z + 4\,x^2\,y\,z^3 + x^2\,z^4 + 2\,x\,y^4\,z + 4\,x\,y^3\,z^2 + 4\,x\,y^2\,z^3\\
&\quad + 2\,x\,y\,z^4 + y^4\,z^2 + y^2\,z^4)(x\,y + x\,z + y\,z)^6\,T^6 + \cdots,
\end{aligned}$$

$$\begin{aligned}
R(T) = {} & (x+y+z)^4\,T^8 - 4(x^3 + x^2\,y + x^2\,z + x\,y^2 + 3\,x\,y\,z + x\,z^2 + y^3 + y^2\,z \\
& + y\,z^2 + z^3)\,(x+y+z)^3\,T^6 + 2(16\,x\,y\,z^4 + 14\,x\,y^2\,z^3 + 14\,x\,y^3\,z^2 + 16\,x\,y^4\,z \\
& + 14\,x^2\,y\,z^3 + 14\,x^2\,y^3\,z + 14\,x^3\,y\,z^2 + 14\,x^3\,y^2\,z + 16\,x^4\,y\,z + 3\,x^6 + 5\,x^4\,y^2 \\
& + 5\,x^4\,z^2 + 5\,x^2\,y^4 + 5\,x^2\,z^4 + 5\,y^4\,z^2 + 5\,y^2\,z^4 + 21\,x^2\,y^2\,z^2 + 3\,y^6 + 3\,z^6 \\
& + 6\,x^5\,y + 6\,x^5\,z + 4\,x^3\,y^3 + 4\,x^3\,z^3 + 6\,x\,y^5 + 6\,x\,z^5 + 6\,y^5\,z + 4\,y^3\,z^3 + 6\,y\,z^5) \\
& (x+y+z)^2\,T^4 \\
& - 4(x+y+z)(x^6 - x^4\,y^2 - x^4\,z^2 + 2\,x^3\,y^2\,z + 2\,x^3\,y\,z^2 - x^2\,y^4 + 2\,x^2\,y^3\,z \\
& + 7\,x^2\,y^2\,z^2 + 2\,x^2\,y\,z^3 - x^2\,z^4 + 2\,x\,y^3\,z^2 + 2\,x\,y^2\,z^3 + y^6 - y^4\,z^2 - y^2\,z^4 + z^6) \\
& (x^3 + 3\,x^2\,y + 3\,x^2\,z + 3\,x\,y^2 + 7\,x\,y\,z + 3\,x\,z^2 + y^3 + 3\,y^2\,z + 3\,y\,z^2 + z^3)\,T^2 \\
& + (-6\,x\,y^2\,z^3 - 6\,x\,y^3\,z^2 - 6\,x^2\,y\,z^3 - 6\,x^2\,y^3\,z - 6\,x^3\,y\,z^2 - 6\,x^3\,y^2\,z + x^6 \\
& - x^4\,y^2 - x^4\,z^2 - x^2\,y^4 - x^2\,z^4 - y^4\,z^2 - y^2\,z^4 - 9\,x^2\,y^2\,z^2 + y^6 + z^6 + 2\,x^5\,y \\
& + 2\,x^5\,z - 4\,x^3\,y^3 - 4\,x^3\,z^3 + 2\,x\,y^5 + 2\,x\,z^5 + 2\,y^5\,z - 4\,y^3\,z^3 + 2\,y\,z^5)^2.
\end{aligned}$$

The successive resultant computation for $L(T)$ and $R(T)$ spent CPU time 0.51s and 0.08s, respectively, on a Pentium III/550 with Maple V.5.1. And then, It took us 111.21s to obtain the border polynomial of degree 100 with 2691 terms.

We may of course reform (2) to the equivalent one by transposition of terms, e.g.

$$\begin{aligned}
& \frac{\sqrt{x^3(y+z)(xy+xz+4\,yz)}}{xy+yz+xz} + \frac{\sqrt{y^3(x+z)(xy+yz+4\,xz)}}{xy+yz+xz} + \\
& \frac{\sqrt{z^3(x+y)(yz+xz+4\,xy)}}{xy+yz+xz} - \sqrt{x^2 + \frac{xyz}{x+y+z}} - \sqrt{y^2 + \frac{xyz}{x+y+z}} \\
& \leq \sqrt{z^2 + \frac{xyz}{x+y+z}}. \qquad (3)
\end{aligned}$$

However, the left polynomial of (3) cannot be found on the same computer (with memory 256 Mb) by a Maple procedure as we did for (2),

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

after running 1066.52s, the screen shows “Error, object too large” at last.

One might try to compute the border polynomial directly without employing left and right polynomials, that is, using the procedure

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

but the result is not better. After running 1453.67s, the screen shows “Error, object too large” once again.

**Example 2.** Given an algebraic inequality in $x, y, z$,

$$m_a + m_b + m_c \leq 2\,s \qquad (4)$$

where

$$\begin{aligned} m_a &= \frac{1}{2}\sqrt{2\,(x+y)^2+2\,(x+z)^2-(y+z)^2},\\ m_b &= \frac{1}{2}\sqrt{2\,(y+z)^2+2\,(x+y)^2-(x+z)^2},\\ m_c &= \frac{1}{2}\sqrt{2\,(x+z)^2+2\,(y+z)^2-(x+y)^2},\\ s &= x+y+z \end{aligned}$$

with $x>0,\ y>0,\ z>0$, compute the left, right and border polynomials.

Let

$$\begin{aligned} f_1 &= 4\,m_a^2+(y+z)^2-2\,(x+y)^2-2\,(x+z)^2,\\ f_2 &= 4\,m_b^2+(x+z)^2-2\,(y+z)^2-2\,(x+y)^2,\\ f_3 &= 4\,m_c^2+(x+y)^2-2\,(x+z)^2-2\,(y+z)^2 \end{aligned}$$

and do successive resultant computation

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

we obtain a left polynomial of (4):

$$\begin{aligned} &T^8-6\,(x^2+y^2+z^2+x\,y+y\,z+z\,x)\,T^6+9(x^4+2\,x\,y\,z^2+y^4+2\,x\,z^3\\ &\quad+2\,x^3\,y+z^4+3\,y^2\,z^2+2\,y^2\,z\,x+2\,y^3\,z+2\,y\,z^3+3\,x^2\,z^2+2\,x^3\,z+2\,x^2\,y\,z\\ &\quad+2\,x\,y^3+3\,x^2\,y^2)T^4-(72\,x^4\,y\,z+78\,x^3\,y\,z^2+4\,x^6+4\,y^6+4\,z^6+12\,x\,y^5\\ &\quad-3\,x^4\,y^2-3\,x^2\,z^4-3\,x^2\,y^4-3\,y^4\,z^2-3\,y^2\,z^4-3\,x^4\,z^2-26\,x^3\,y^3-26\,x^3\,z^3\\ &\quad-26\,y^3\,z^3+12\,x\,z^5+12\,y^5\,z+12\,y\,z^5+12\,x^5\,z+12\,x^5\,y+84\,x^2\,y^2\,z^2\\ &\quad+72\,x\,y\,z^4+72\,x\,y^4\,z+78\,x\,y^3\,z^2+78\,x\,y^2\,z^3+78\,x^2\,y\,z^3+78\,x^3\,y^2\,z\\ &\quad+78\,x^2\,y^3\,z)T^2+81\,x^2\,y^2\,z^2\,(x+y+z)^2. \end{aligned} \tag{5}$$

It is trivial to find a right polynomial for this inequality because the right-hand side contains no radicals. We simply take

$$T-2\,(x+y+z). \tag{6}$$

Computing the resultant of (5) and (6) with respect to $T$, we have

$$\begin{aligned} &(144\,x^5\,y+144\,x^5\,z+780\,x^4\,y^2+1056\,x^4\,y\,z+780\,x^4\,z^2+1288\,x^3\,y^3\\ &\quad+3048\,x^3\,y^2\,z+3048\,x^3\,y\,z^2+1288\,x^3\,z^3+780\,x^2\,y^4+3048\,x^2\,y^3\,z\\ &\quad+5073\,x^2\,y^2\,z^2+3048\,x^2\,y\,z^3+780\,x^2\,z^4+144\,x\,y^5+1056\,x\,y^4\,z\\ &\quad+3048\,x\,y^3\,z^2+3048\,x\,y^2\,z^3+1056\,x\,y\,z^4+144\,x\,z^5+144\,y^5\,z+780\,y^4\,z^2\\ &\quad+1288\,y^3\,z^3+780\,y^2\,z^4+144\,y\,z^5)(x+y+z)^2. \end{aligned}$$

Removing the non-vanishing factor $(x+y+z)^2$, we obtain the border surface

$$\begin{aligned}
&144\,x^5\,y + 144\,x^5\,z + 780\,x^4\,y^2 + 1056\,x^4\,y\,z + 780\,x^4\,z^2 + 1288\,x^3\,y^3\\
&\quad + 3048\,x^3\,y^2\,z + 3048\,x^3\,y\,z^2 + 1288\,x^3\,z^3 + 780\,x^2\,y^4 + 3048\,x^2\,y^3\,z\\
&\quad + 5073\,x^2\,y^2\,z^2 + 3048\,x^2\,y\,z^3 + 780\,x^2\,z^4 + 144\,x\,y^5 + 1056\,x\,y^4\,z\\
&\quad + 3048\,x\,y^3\,z^2 + 3048\,x\,y^2\,z^3 + 1056\,x\,y\,z^4 + 144\,x\,z^5 + 144\,y^5\,z + 780\,y^4\,z^2\\
&\quad + 1288\,y^3\,z^3 + 780\,y^2\,z^4 + 144\,y\,z^5 = 0. \qquad (7)
\end{aligned}$$

## 3 A Sketch to Dimension-Decreasing Algorithm

In the present paper, we deal with a class of propositions which take the following form (though the algorithm is applicable to a more extensive class):

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

where $\Phi_0$, $\Phi_1$, …, $\Phi_s$ are algebraic inequalities in $x, y, z, \ldots$ etc., the hypothesis $\Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_s$ defines either an open set[2] or an open set with the whole/partial boundary.

Example 1 may be written as $(x>0) \wedge (y>0) \wedge (z>0) \Rightarrow (2)$, where the hypothesis $(x>0) \wedge (y>0) \wedge (z>0)$ defines an open set in the parametric space $\mathbf{R}^3$, so it belongs to the class we described, so does Example 2. This class covers most of inequalities in Bottema et al.'s book [1] and Mitrinovic et al.'s [9] "*Recent Advances in Geometric Inequalities*". In fact, Example 2 is a geometric inequality encoded in $x$, $y$, $z$, see [1].

We take the following procedures when the conclusion $\Phi_0$ is of type $\leq$. (As for $\Phi_0$ of type $<$, what we need do in additional is to verify if the equation $l_0(x, y, \ldots) - r_0(x, y, \ldots) = 0$ has no real solutions under the hypothesis, where $l_0(x, y, \ldots)$ and $r_0(x, y, \ldots)$ denote the left- and right-hand sides of $\Phi_0$, respectively.)

- Find the border surfaces of the inequalities $\Phi_0$, $\Phi_1$, …, $\Phi_s$.
- These border surfaces decompose the parametric space into a finite number of cells. Among them we just take all the connected open sets, $D_1, D_2, \ldots, D_k$, and discard the lower dimensional cells. Choose at least one test point in every connected open set, say, $(x_\nu, y_\nu, \ldots) \in D_\nu,\ \nu = 0, 1, \ldots, k$. This step can be done by an incomplete cylindrical algebraic decomposition which is much easier than the complete one since all the lower dimensional cells were discarded. Furthermore, we can make every test point a rational point because it is chosen in an open set.
- We need only check the proposition for such a finite number of test points, $(x_1, y_1, \ldots)$, …, $(x_k, y_k, \ldots)$. The statement is true if and only if it holds over these test values.

[2] may be disconnected.

The proof of the correctness of the method is sketched as follows.

By $l_\mu(x,y,\ldots)$, $r_\mu(x,y,\ldots)$ and $P_\mu(x,y,\ldots)=0$ denote the left-, right-hand sides and border surface of $\varPhi_\mu$, respectively, and

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

for $\mu = 0,\ldots,s$.

The set of real zeros of all the $\delta_\mu(x,y,\ldots)$ is a closed set, so its complementary set, say $\varDelta$, is an open set. On other hand, the set

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

is exactly the complementary set of real zeros of all the $P_\mu(x,y,\ldots)$.

We have $D \subset \varDelta$ since any zero of $\delta_\mu(x,y,\ldots)$ must be a zero of $P_\mu(x,y,\ldots)$. By $\varDelta_1,\ldots,\varDelta_t$ denote all the connected components of $\varDelta$, so each one is a connected open set. Every $\varDelta_\lambda$ must contain a point of $D$ for an open set cannot be filled with the real zeros of all the $P_\mu(x,y,\ldots)$. Assume that $\varDelta_\lambda$ contains a point of $D_i$, some connected component of $D$. Then, $D_i \subset \varDelta_\lambda$ because it is impossible that two different components of $\varDelta$ both intersect $D_i$. By step 2, $D_i$ contains a test point $(x_i,y_i,\ldots)$. So, every $\varDelta_\lambda$ contains at least one test point obtained from step 2.

Thus, $\delta_\mu(x,y,\ldots)$ keeps the same sign over $\varDelta_\lambda$ as that of $\delta_\mu(x_{i_\lambda},y_{i_\lambda},\ldots)$ where $(x_{i_\lambda},y_{i_\lambda}\ldots)$ is a test point in $\varDelta_\lambda$, for $\lambda=1,\ldots,t;\ \mu=0,\ldots,s$. Otherwise, if there is some point $(x',y',\ldots)\in\varDelta_\lambda$ that $\delta_\mu(x',y',\ldots)$ has the opposite sign to $\delta_\mu(x_{i_\lambda},y_{i_\lambda},\ldots)$, connecting two points $(x',y',\ldots)$ and $(x_{i_\lambda},y_{i_\lambda},\ldots)$ with a path $\varGamma$ such that $\varGamma\subset\varDelta_\lambda$, then there is a point $(\bar{x},\bar{y},\ldots)\in\varGamma$ such that $\delta_\mu(\bar{x},\bar{y},\ldots)=0$, a contradiction!

By $A\cup B$ denote the set defined by the hypothesis, where $A$ is an open set defined by

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

that consists of a number of connected components of $\varDelta$ and some real zeros of $\delta_0(x,y,\ldots)$, namely $A=Q\cup S$ where $Q=\varDelta_1\cup\cdots\cup\varDelta_j$ and $S$ is a set of some real zeros of $\delta_0(x,y,\ldots)$. And $B$ is the whole or partial boundary of $A$, that consists of some real zeros of $\delta_\mu(x,y,\ldots)$ for $\mu=1,\ldots,s$.

Now, let us verify whether $\delta_0<0$ holds for all the test points in $A$, one by one. If there is a test point whereat $\delta_0>0$, then the proposition is false. Otherwise, $\delta_0<0$ holds over $Q$ because every connected component of $Q$ contains a test point and $\delta_0$ keeps the same sign over each component $\varDelta_\lambda$, hence $\delta_0\le 0$ holds over $A$ by continuity, so it also holds over $A\cup B$, i.e., the proposition is true.

The above procedures sometimes may be simplified. When the conclusion $\varPhi_0$ belongs to an inequality class called “class CGR”, what we need do in step 3 is to compare the greatest roots of left and right polynomials of $\varPhi_0$ over the test values.

**Definition 4.** An algebraic inequality is said to belong to class CGR if its left-hand side is the greatest (real) root of the left polynomial $L(T)$, and the right-hand side is that of the right polynomial $R(T)$.

It is obvious in Example 1 that the left- and right-hand sides of the inequality (2) are the greatest roots of $L(T)$ and $R(T)$, respectively, because all the radicals have got positive signs. Thus, the inequality belongs to class CGR. What we need do is to verify whether the greatest root of $L(T)$ is less than or equal to that of $R(T)$, that is much easier than determine which is greater between two complicated radicals, in the sense of accurate computation.

If an inequality involves only mono-layer radicals, then it always can be transformed into an equivalent one which belongs to class CGR by transposition of terms. Actually, most of inequalities in [1] and [9], including the examples in the present paper, belong to class CGR. For some more material, see [13].

## 4 Inequalities on Triangles

An absolute majority of the hundreds inequalities discussed in [1] are on triangles, so are the thousands appeared in various publications since then.

For geometric inequalities on a single triangle, usually the geometric invariants are used as global variables instead of Cartesian coordinates. By $a, b, c$ denote the side-lengths, $s$ the half perimeter, i.e. $\frac{1}{2}(a+b+c)$, and $x, y, z$ denote $s-a, s-b, s-c$, respectively, as people used to do. In additional, by $A, B, C$ the interior angles, $S$ the area, $R$ the circumradius, $r$ the inradius, $r_a, r_b, r_c$ the radii of escribed circles, $h_a, h_b, h_c$ the altitudes, $m_a, m_b, m_c$ the lengths of medians, $w_a, w_b, w_c$ the lengths of interior angular bisectors, and so on.

People used to choose $x, y, z$ as independent variables and others dependent. Sometimes, another choice is better for decreasing the degrees of polynomials occurred in the process.

An algebraic inequality $\Phi(x, y, z)$ can be regarded as a geometric inequality on a triangle if

- $x > 0,\ y > 0,\ z > 0$;
- the left- and right-hand sides of $\Phi$, namely $l(x, y, z)$ and $r(x, y, z)$, both are homogeneous;
- $l(x, y, z)$ and $r(x, y, z)$ have the same degree.

The item 1 means that the sum of two edges of a triangle is greater than the third edge. The items 2 and 3 means that a similar transformation does not change the truth of the proposition. For example, (4) is such an inequality for its left- and right-hand sides, $m_a + m_b + m_c$ and $2\,s$, both are homogeneous functions (of $x$, $y$, $z$) with degree 1.

In addition, assume that the left- and right-hand sides of $\Phi(x, y, z)$, namely, $l(x,y,z)$ and $r(x,y,z)$, both are symmetric functions of $x$, $y$, $z$. It does not change the truth of the proposition to replace $x$, $y$, $z$ in $l(x,y,z)$ and $r(x,y,z)$ with $x', y', z'$ where $x' = \rho\, x,\ y' = \rho\, y,\ z' = \rho\, z$ and $\rho > 0$.

Clearly, the left and right polynomials of $\Phi(x', y', z')$, namely, $L(T, x', y', z')$ and $R(T, x', y', z')$, both are symmetric with respect to $x', y', z'$, so they can be re-coded in the elementary symmetric functions of $x', y', z'$, say,

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

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

Setting $\rho = \sqrt{\frac{x+y+z}{x\,y\,z}}$, we have $x'y'z' = x' + y' + z'$, i.e., $\sigma_3 = \sigma_1$. Further, letting

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

we can transform $L(T, x', y', z')$ and $R(T, x', y', z')$ into polynomials in $T$, $p$, $s$, say, $F(T, p, s)$ and $G(T, p, s)$. Especially if $F$ and $G$ both have only even-degree terms in $s$, then they can be transformed into polynomials in $T$, $p$ and $q$ where $q = s^2 - 4\,p - 27$. Usually the degrees and the numbers of terms of the latter are much less than those of $L(T, x, y, z)$ and $R(T, x, y, z)$. We thus construct the border surface which is encoded in $p$, $s$ or $p$, $q$, and do the decomposition described in last section on $(p,\ s)$-plane or $(p,\ q)$-plane instead of $\mathbf{R}^3$. This may reduce the computational complexity considerably for a large class of geometric inequalities. The following example is also taken from [1].

**Example 3.** By $w_a$, $w_b$, $w_c$ and $s$ denote the interior angular bisectors and half the perimeter of a triangle, respectively. Prove

$$w_b w_c + w_c w_a + w_a w_b \le s^2.$$

It is well-known that

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

and $s = x + y + z$. By successive resultant computation as above, we get a left polynomial which is of degree 20 and has 557 terms, while the right polynomial $T - (x+y+z)^2$ is very simple, and the border polynomial $P(x,\ y,\ z)$ is of degree 15 and has 136 terms. By this routine, the whole proving process take us 2697 sec on a Pentium III/550 computer.

However, if we encode the left and right polynomials in $p$, $q$, we get

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

and $T - 4\,p - q - 27$, respectively, hence the border polynomial

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

which is of degree 5 and has 20 terms only. The whole proving process in this way spend about 0.10s on the same machine.

## 5 Commands and Syntax

As a prover, the whole program is written in Maple V.5.1 including the cell decomposition, without external packages employed.

On verifying an inequality with BOTTEMA, we need only type in a proving command, then the machine will do everything else. If the statement is true, the computer screen will show "*The inequality holds*"; otherwise, it will show "*The inequality does not hold*" with a counter-example. There are three kinds of proving commands: *prove, xprove* and *yprove*.

**prove** – prove a geometric inequality on a triangle, or an equivalent algebraic inequality.

**Calling Sequence:**

prove(ineq);
prove(ineq, ineqs);

**Parameters:**

ineq – an inequality to be proven, which is encoded in the geometric invariants listed later.

ineqs – a list of inequalities as the hypothesis, which is encoded as well in the geometric invariants listed later.

**Description:**

- The command 'prove' is valid to a geometric inequality on a triangle that must be of type '$\le$' or '$\ge$', with a set of inequalities 'ineqs' as the hypothesis which defines either an open set or an open set with the whole/partial boundary; and 'ineq' and 'ineqs', must be represented by the rational functions and radicals in the geometric invariants listed below.
- The command 'prove' also valid to a statement whose hypothesis and thesis, 'ineqs' and 'ineq', all are homogeneous algebraic inequalities represented by the rational functions and radicals in $x, y, z$ provided $x > 0, y > 0, z > 0$, and of the required types as above. This is equivalent to a geometric one, as shown in last section.

– The following list of geometric invariants is extendable.

**Geometric Invariants on a Triangle:** (extendable)

| | |
|---|---|
| a, b, c, | lengths of sides of a triangle ABC |
| s, | s:=(a+b+c)/2, half the perimeter |
| x, y, z, | x:=s-a, y:=s-b, z:=s-c |
| S, | Area of the triangle |
| R, | circumradius |
| r, | inradius |
| ra, rb, rc, | radii of escribed circles |
| ha, hb, hc, | altitudes |
| ma, mb, mc, | medians |
| wa, wb, wc, | interior-angle-bisectors |
| p, | p:=4*r*(R-2*r) |
| q, | q:=s^2-16*R*r+5*r^2 |
| HA, HB, HC, | distances from orthocenter to vertices |
| IA, IB, IC, | distances from incenter to vertices |
| sin(A), sin(B), sin(C), | sines of the interior angles |
| cos(A), cos(B), cos(C), | cosines of the interior angles |
| tan(A), tan(B), tan(C), | tangents of the interior angles |
| cot(A), cot(B), cot(C), | cotangents of the interior angles |
| sec(A), sec(B), sec(C), | secants of the interior angles |
| csc(A), csc(B), csc(C), | cosecants of the interior angles |
| 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), | |

**Examples:**

```
> read bottema;
> prove(a^2+b^2+c^2>=4*sqrt(3)*S+(b-c)^2+(c-a)^2+(a-b)^2);

                         The theorem holds

> prove(cos(A)>=cos(B),[a<=b]);

                         The theorem holds
```

**xprove** – prove an algebraic inequality with positive variables.

**Calling Sequence:**

xprove(ineq);
xprove(ineq, ineqs);

**Parameters:**

ineq – an algebraic inequality to be proven, with positive variables.
ineqs – a list of algebraic inequalities as the hypothesis, with positive variables.

**Description:**

- The command ‘xprove’ is valid to an algebraic inequality ‘ineq’ which must be of type ‘$\leq$’ or ‘$\geq$’, with a set of inequalities ‘ineqs’ as the hypothesis which defines either an open set or an open set with the whole/partial boundary.
- All the hypothesis and thesis must be represented by the rational functions and radicals.
- All the variables appear in ‘ineq’ are supposed always positive that conditions need not be explicitly included.

**Examples:**

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

*The theorem holds*

```
> f:=(x+1)^(1/3)+sqrt(y-1)+x*y+1/x+1/y^2:
> xprove(f>=42496/10000,[y>1]);
```

*The theorem holds*

```
> xprove(f>=42497/10000,[y>1]);
```

*with a counter example*

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

*The theorem does not hold*

**yprove** – prove an algebraic inequality in general.

**Calling Sequence:**

yprove(ineq);
yprove(ineq, ineqs);

**Parameters:**

ineq – an algebraic inequality to be proven.
ineqs – a list of algebraic inequalities as the hypothesis.

**Description:**

- The command ‘yprove’ is valid to an algebraic inequality ‘ineq’ which must be of type ‘$\leq$’ or ‘$\geq$’, with a set of inequalities ‘ineqs’ as the hypothesis which defines either an open set or an open set with the whole/partial boundary.

- All the hypothesis and thesis must be represented by the rational functions and radicals.

**Examples:**

```
> 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 theorem holds*

## 6 More Examples

The well-known Janous' inequality [5] which was proposed as an open problem in 1986 and solved in 1988.

**Example 4.** By $m_a$, $m_b$, $m_c$ and $2\,s$ denote the three medians and perimeter of a triangle, show that

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

The left-hand side of the some difficult inequality implicitly contains three radicals. BOTTEMA automatically interprets the geometric proposition to algebraic one before proves it. The total CPU time spent for this example on a Pentium III/550 is 29.22s.

The next example was proposed as an open problem, E. 3146*, in the *Amer. Math. Monthly* **93**:(1986), 299.

**Example 5.** By $a, b, c$ and $s$ denote the side-lengths and half perimeter of a triangle, respectively. Prove or disprove

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

The proof took us 48.60s on the same machine.

The following open problem appeared as Problem 169 in *Mathematical Communications* (in Chinese).

**Example 6.** By $r_a, r_b, r_c$ and $w_a, w_b, w_c$ denote the radii of the escribed circles and the interior angle bisectors of a triangle, respectively. Prove or disprove

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

In other words, *the geometric average of* $r_a, r_b, r_c$ *is less than or equal to the arithmetic average of* $w_a, w_b, w_c$.

The right-hand side of the inequality implicitly contains 3 radicals. BOTTEMA proved this conjecture on a Pentium III/550 with CPU time 96.60s. One more conjecture proposed by J. Liu [10] was proven on the same machine with CPU time 375.14s. That is:

**Example 7.** By $a$, $b$, $c$, $m_a$, $m_b$, $m_c$ and $w_a$, $w_b$, $w_c$ denote the side lengths, medians and interior-angle-bisectors of a triangle, respectively. Prove or disprove

$$a\,m_a + b\,m_b + c\,m_c \le \frac{2}{\sqrt{3}}\,(w_a^2 + w_b^2 + w_c^2).$$

The following conjecture was first proposed by J. Garfunkel at *Crux Math.* in 1985, then re-proposed twice again in [9] and [6].

**Example 8.** By $A$, $B$, $C$ denote the three angles of a triangle. Prove or disprove

$$\cos\frac{B-C}{2} + \cos\frac{C-A}{2} + \cos\frac{A-B}{2} \le \\ \frac{1}{\sqrt{3}}\,(\cos\frac{A}{2} + \cos\frac{B}{2} + \cos\frac{C}{2} + \sin A + \sin B + \sin C\,).$$

It was proven on a Pentium III/550 with CPU time 163.54s.

A. Oppenheim studied the following inequality [9] in order to answer a problem proposed by P. Erdös.

**Example 9.** Let $a$, $b$, $c$ and $m_a$, $m_b$, $m_c$ denote the side lengths and medians of a triangle, respectively. If $c = \min\{a,\, b,\, c\}$, then

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

The hypothesis includes one more condition, $c = \min\{a,\, b,\, c\}$, so we type in

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

This took us 547.65s on the same machine. If we type in

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

without the additional condition, the screen will show "*The inequality does not hold*" with a counter-example, $[a = 203,\ b = 706,\ c = 505]$.

A problem of positive semi-definite decision is originated from one of the conjectures proposed by B. Q. Liu [7]:

**Example 10.** Assume that $x > 0$, $y > 0$, $z > 0$. Prove

$$2187(y^4z^4(y+z)^4(2\,x+y+z)^8 + x^4z^4(x+z)^4(x+2\,y+z)^8 + \\ x^4y^4(x+y)^4(x+y+2\,z)^8) - 256(x+y+z)^8(x+y)^4(x+z)^4(y+z)^4 \ge 0.$$

The polynomial after being expanded is of 201 terms with the largest coefficient (absolute value) 181394432. Usually it is non-trivial to decide a polynomial to be positive semi-definite or not, but this one took us CPU time 2.23s only, because of the homogeneity and symmetry which can help decrease the dimension and degree concerned.

There are two well-known geometric inequalities. One is the so-called "Euler's Inequality", $R \geq 2\,r$, another is $m_a \geq w_a$. They are often cited in illustration of various algorithms [2,11,12] for inequality proving. The following example makes a comparison between the two differences, $R - 2\,r$ and $m_a - w_a$.

**Example 11.** By $R$, $r$ denote the circumradius and inradius of a triangle, and $m_a$, $w_a$ the median and the interior angle bisector on a certain side; prove

$$m_a - w_a \leq R - 2\,r.$$

It took us 15.73s.

The geometric inequalities which can be verified by the program, of course, are not limited to those on triangles. To prove the so-called "Ptolemy Inequality", we will use Cartesian coordinates instead of geometric invariants.

**Example 12.** Given four points $A, B, C, D$ on a plane, by $AB, AC, AD, BC, BD, CD$ denote the distances between the points, respectively. Prove

$$AB \cdot CD + BC \cdot AD \geq AC \cdot BD. \tag{8}$$

Put $A = (-\frac{1}{2}, 0)$, $B = (x, y)$, $C = (\frac{1}{2}, 0)$, $D = (u, v)$, and convert (8) to

$$\sqrt{(-\frac{1}{2} - x)^2 + y^2}\,\sqrt{(\frac{1}{2} - u)^2 + v^2} + \sqrt{(x - \frac{1}{2})^2 + y^2}\,\sqrt{(-\frac{1}{2} - u)^2 + v^2}$$
$$\geq \sqrt{(x - u)^2 + (y - v)^2}. \tag{9}$$

We need just type in "`yprove(%)`" where `%` stands for inequality (9). The screen shows "`The inequality holds`" after running 2.70s.

According to our record, the CPU time spent (on a Pentium III/550) and the numbers of the test points for above examples are listed as follows.

| | | |
|---|---:|---:|
| Example 1 | $702.15s$ | 23 test points |
| Example 2 | $0.18s$ | 1 test point |
| Example 3 | $0.10s$ | 1 test point |
| Example 4 | $29.22s$ | 12 test points |
| Example 5 | $48.60s$ | 135 test points |
| Example 6 | $96.60s$ | 4 test points |
| Example 7 | $375.14s$ | 3 test points |
| Example 8 | $163.54s$ | 121 test points |
| Example 9 | $547.65s$ | 287 test points |
| Example 10 | $2.23s$ | 2 test points |
| Example 11 | $15.73s$ | 22 test points |
| Example 12 | $2.70s$ | 48 test points |

The time listed above includes that spent for everything, finding the left, right and border polynomial, cell decomposition, one-by-one sample point test, etc.

## 7 Conclusion

- This program is applicable to any inequality-type theorem whose hypothesis and thesis all are inequalities in rational functions or radicals, but the thesis is of type "$\leq$" or "$\geq$", and the hypothesis defines either an open set or an open set with the whole/partial boundary.
- It is beyond the capacity of this prover to deal with the algebraic functions other than the rational ones and radicals.
- It runs in a completely automatic mode, without human intervention.
- It is especially efficient for geometric inequalities on triangles. The input, in this case, is encoded in geometric invariants.

We will make the package available publicly as soon as possible. The interested reader may contact us by e-mail luyang@guangztc.edu.cn or cdluyang@mail.sc.cninfo.net for further information.

### Acknowledgements

The authors would like to thank the anonymous referees to this paper for their valuable suggestions.

## References

1. Bottema, O., Dordevic, R. Z., Janic, R. R., Mitrinovic, D. S. & Vasic, P. M., *Geometric Inequalities*, Wolters-Noordhoff Publishing, Groningen, The Netherlands, 1969.
2. Chou, S. C., Gao, X. S. & Arnon, D. S., On the mechanical proof of geometry theorems involving inequalities, *Advances in Computing Research*, **6**, JAI Press Inc., pp. 139–181, 1992.
3. Dolzmann, A., Sturm, T. & Weispfenning, V., A new approach for automatic theorem proving in real geometry, *Journal of Automated Reasoning*, **21**(3), 357–380, 1998.
4. Dolzmann, A., Sturm, T. & Weispfenning, V., Real quantifier elimination in practice, *Algorithmic Algebra and Number Theory*, B. H. Matzat, G.-M. Greuel & G. Hiss (eds.), Springer-Verlag, Berlin Heidelberg, pp. 221–247, 1998.
5. Janous, W., Problem 1137, *Crux Math.,* **12**, 79, 177, 1986.
6. Kuang, J. C. *Applied Inequalities* (in Chinese), 2nd ed., Hunan Educational Publishing House, China, 1993.
7. Liu, B. Q., A collection of geometric inequalities discovered with BOTTEMA (in Chinese), *Research Communications on Inequalities*, **31**, 2001 (to appear).
8. McPhee, N. F., Chou, S. C. & Gao, X. S.: Mechanically proving geometry theorems using a combination of Wu's method and Collins' method. *Proc. CADE-12*, LNCS **814**, Springer-Verlag, Berlin Heidelberg, pp. 401–415, 1994.

9. Mitrinovic, D. S., Pecaric, J. E. & Volenec, V., *Recent Advances in Geometric Inequalities*, Kluwer Academic Publishers, Boston Dordrecht, 1989.
10. Shan, Z. (ed.), *Geometric Inequality in China* (in Chinese), Jiangsu Educational Publishing House, China, 1996.
11. Wu W.-t., On a finiteness theorem about problem involving inequalities, *Sys. Sci. & Math. Scis.*, **7**, 193–200, 1994.
12. Wu W.-t., On global-optimization problems, *Proc. ASCM '98*, Lanzhou University Press, Lanzhou, pp. 135–138, 1998.
13. Yang, L., Recent advances in automated theorem proving on inequalities, *J. Comput. Sci. & Technol.*, **14**(5), 434–446, 1999.
14. Yang, L., Hou, X. R. & Xia, B. C., Automated discovering and proving for geometric inequalities, *Automated Deduction in Geometry*, X. S. Gao, D. Wang & L. Yang (eds.), LNAI **1669**, Springer-Verlag, Berlin Heidelberg, pp. 30–46, 1999.
15. 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** **44**(1), 33–49, 2001.

# Randomized Zero Testing of Radical Expressions and Elementary Geometry Theorem Proving

Daniela Tulone[1,⋆], Chee Yap[2,⋆⋆], and Chen Li[2]

[1] Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974
daniela@research.bell-labs.com

[2] Department of Computer Science, Courant Institute, New York University,
251 Mercer Street, New York, NY 10012
{yap,chenli}@cs.nyu.edu

**Abstract.** We develop a probabilistic test for the vanishing of *radical expressions*, that is, expressions involving the four rational operations $(+, -, \times, \div)$ and square root extraction. This extends the well-known Schwartz's probabilistic test for the vanishing of polynomials. The probabilistic test forms the basis of a new theorem prover for conjectures about ruler & compass constructions. Our implementation uses the `Core Library` which can perform exact comparison for radical expressions. Some experimental results are presented.

## 1 Introduction

Several approaches to proving theorems in Elementary Geometry using constructive methods in Computer Algebra were proposed in the 1980s [7]. These were much more successful than earlier approaches based on purely logical or axiomatic approaches. Thus, Kutzler, Stifter [14] and Kapur [12] proposed methods based on Gröbner Bases. Carrà and Gallo [1,8] devised a method using the dimension underlying the algebraic variety. Hong [11] introduced semi-numerical methods ("proof by example" techniques) based on gap theorems. An acclaimed approach in this area is due to Wu [21,23,22] who applied the concept of characteristic sets to geometric theorem proving. Extensive experimentation with Wu's method were reported by Chou [3,5].

All these algebraic approaches begin by translating the geometric statements into algebraic ones. A proposed geometry theorem (also called a *conjecture*) is translated algebraically into two parts: a system $H$ of multivariate polynomials called the *hypothesis*, and a single polynomial $T$ called the *thesis*. The conjecture is true if the vanishing of the hypothesis system implies the vanishing of the thesis polynomial. From the viewpoint of algebraic geometry, proving the conjecture amounts to showing that $Var(H) \subseteq Var(T)$ where $Var(S)$ is the algebraic variety defined by a set $S$ of polynomials. This basic formulation must be refined in order to handle degeneracy conditions.

⋆ This work was performed while Daniela Tulone was at NYU.

⋆⋆ This work is supported in part by NSF Grant #CCR 9402464.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 58–82, 2001.

Wu's "basic method" computes the pseudo-remainder of the polynomial thesis with respect to the Wu-Ritt extended characteristic set of the hypotheses system. If the pseudo-remainder vanishes, then the conjecture is true provided the initials of the extended characteristic set do not vanish. Wu's basic method has been successfully used to prove many classical and some new theorems in plane analytic geometry. The basic method fails if the variety $Var(H)$ is reducible. To handle this, Wu's "complete method" begins by decomposing $Var(H)$ into irreducible components and applying the basic method to each component. A drawback in Wu's method is that it works with an algebraically closed field. In particular, it is not a complete method for the real algebraic varieties. The present paper addresses a special case of real algebraic varieties.

Gröbner bases methods can be doubly exponential in the worst case [17,24]. The complexity for Wu's method is somewhat better but remains an issue. To circumvent the high complexity, we investigate probabilistic methods [20] combined with "proof by example" techniques [11]. In probabilistic theorem proving, we do not prove the validity of a conjecture in the classical sense. Instead, we either prove the invalidity of a conjecture (by showing a counter example) or else classify the conjecture as "true with the high probability $1 - \varepsilon$". This latter classification must be properly understood since, classically, it is nonsense to say that a theorem is true with some probability. What is meant is that, relative to a set of experiments we conduct, the probability that the conjecture is false *and* we failed to discover this, is less than $\varepsilon$.

An interesting approach along these lines was given by Carrà, Gallo and Gennaro [2]. They applied the Schwartz-Zippel [20,27] probabilistic test for the vanishing of pseudo-remainders in Wu's method. They considered conjectures in the classical setting of *ruler & compass constructions*. Such conjectures are examined by testing the vanishing of Wu's pseudo-remainder for randomly chosen examples. Each example is specified by a random choice of values for its parameters. The random choices come from some suitable *test set* whose cardinality depends on the degree of the pseudo-remainder. The extended characteristic set as well as the pseudo-remainder are computed. If the pseudo-remainder is zero, then the example is successful; otherwise, as in Wu's method, further investigation is called for. While implementing their method, one of us (D.T.) discovered a serious efficiency issue. The degree of the pseudo-remainder is very high: if the conjecture involves $C$ ruler & compass construction steps, then, the degree of the pseudo-remainder in [2] (following [9,10]) has the following bound:

$$D = 2^{O(C^3)} C^{O(C^2)} \ .$$

The cardinality of the test set is $2D$, which is too large in practice. This bound applies to the test for "generic truth". For "universal truth", $D$ can be improved to $2P \cdot 3^{C+1}$ where $P$ is the number of points in the construction. Unfortunately, practically no classical theorems are universal truths.

**Summary of New Results.** (1) We develop an extension of the Schwartz-Zippel probabilistic zero test. While the Schwartz-Zippel test is applicable to polynomials, we treat radical expressions by admitting the additional operations of division and square-roots. This adds considerable complexity to the proofs. Furthermore, for efficiency considerations, we use straight line programs to represent radical expressions. The asymptotic time complexity of our probabilistic test is a low-order polynomial. Since radical expressions are common in many applications, we expect this new test to be generally useful.

(2) We address the problem of computer proofs of geometric conjectures about ruler & compass constructions. The zero test of radical expressions is tailor fitted for this problem. Moreover, we combine randomness with the numerical approach of Hong to give additional efficiency. Thus, our approach appears to be intrinsically more efficient than previous general approaches (e.g., Wu's or Gröbner bases).

(3) Our prover is implemented using the `Core library` [15,13,19]. This is an unexpected application of our library, which was designed as a general `C++`-package to support the Exact Geometric Computation [26,25] approach to robust algorithms. Preliminary experimental results are quite promising. We expect further improvements by fine-tuning our library for this specific application. Our prover is currently distributed with version 1.3 of the `Core library` (Aug. 15, 2000) and available from `http://cs.nyu.edu/exact/core/`.

**Overview.** The paper is organized as follows: Section 2 gives an overview of geometric conjectures about ruler & compass constructions. Section 3 gives our extension of Schwartz's probabilistic test to radical expressions. Section 4 addresses the application of our new probabilistic test to theorem proving. We conclude in Section 5.

## 2 Theorem Proving for Ruler & Compass Constructions

We follow the algebraic approach which has been well-summarized by Chou [5]. Ruler & compass operations may be seen as constructing lengths, points, lines and circles, collectively called *geometric objects*. A collection of such geometric objects will be called a *geometric scene*. We consider geometric scenes that are constructed incrementally using ruler & compass operations. The algebraic analogue of constructing a geometric object $O$ amounts to introducing a pair of variables $(x, y)$ and corresponding polynomial equations $h_i(x, y, z, \ldots)$ $(i = 1, 2, \ldots)$ that must be satisfied if $(x, y)$ lies on $O$. Here, $h_i$ may involve other variables $z, \ldots$, from previously constructed objects. We shall classify the *variables* introduced by our constructions into two sorts: *independent* and *dependent* variables. For short, the independent variables will be called *parameters*. It is instructive to give a concrete example (Figure 1 from [5]).

**Example 1** (Pascal's Theorem). Let $A$, $B$, $C$, $D$, $F$ and $E$ be six points on a circle centered at $O$. Let $P = AB \bigcap DF$, $Q = BC \bigcap FE$ and $S = CD \bigcap EA$. Show that $P$, $Q$ and $S$ are collinear.

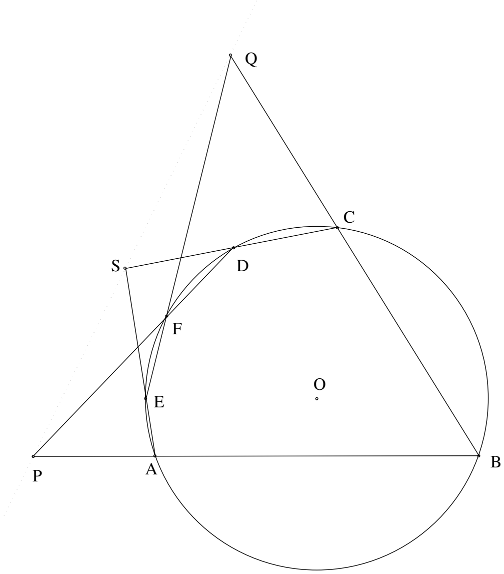


**Fig. 1.** Pascal's Theorem.

Let $A = (0,0)$, $O = (u_1,0)$, $B = (x_1,u_2)$, $C = (x_2,u_3)$, $D = (x_3,u_4)$, $F = (x_4,u_5)$, $E = (x_5,u_6)$, $P = (x_7,x_6)$, $Q = (x_9,x_8)$, and $S = (x_{11},x_{10})$. This gives the following equations for the hypotheses.

| *Equation* | *Geometry* | *Remark* |
|---|---|---|
| $h_1 : x_1^2 - 2u_1x_1 + u_2^2 = 0$ | $[OA \equiv OB]$ | Introduces $x_1, u_2$ |
| $h_2 : x_2^2 - 2u_1x_2 + u_3^2 = 0$ | $[OA \equiv OC]$ | Introduces $x_2, u_3$ |
| $h_3 : x_3^2 - 2u_1x_3 + u_4^2 = 0$ | $[OA \equiv OD]$ | Introduces $x_3, u_4$ |
| $h_4 : x_4^2 - 2u_1x_4 + u_5^2 = 0$ | $[OA \equiv OF]$ | Introduces $x_4, u_5$ |
| $h_5 : x_5^2 - 2u_1x_5 + u_6^2 = 0$ | $[OA \equiv OE]$ | Introduces $x_5, u_6$ |
| $h_6 : \begin{array}{l}(u_5 - u_4)x_7 + (-x_4 + x_3)x_6 + \\ u_4x_4 - u_5x_3 = 0\end{array}$ | $[P \in DF]$ | Introduces $x_6, x_7$ |
| $h_7 : u_2x_7 - x_1x_6 = 0$ | $[P \in AB]$ | Constrains $x_6, x_7$ |
| $h_8 : \begin{array}{l}(u_6 - u_5)x_9 + (-x_5 + x_4)x_8 + \\ u_5x_5 - u_6x_4 = 0\end{array}$ | $[Q \in FE]$ | Introduces $x_8, x_9$ |
| $h_9 : \begin{array}{l}(u_3 - u_2)x_9 + (-x_2 + x_1)x_8 + \\ u_2x_2 - u_3x_1 = 0\end{array}$ | $[Q \in BC]$ | Constrains $x_8, x_9$ |
| $h_{10} : u_6x_{11} - x_5x_{10} = 0$ | $[S \in AE]$ | Introduces $x_{10}, x_{11}$ |
| $h_{11} : \begin{array}{l}(u_4 - u_3)x_{11} + (-x_3 + x_2)x_{10} + \\ u_3x_3 - u_4x_2 = 0\end{array}$ | $[S \in CD]$ | Constrains $x_{10}, x_{11}$ |

The conclusion that $P, Q, S$ are collinear can be translated into the following polynomial:

$$g = (x_8 - x_6)x_{11} + (-x_9 + x_7)x_{10} + x_6x_9 - x_7x_8 = 0. \qquad \blacksquare$$

In general, we get a system of polynomial equations, $h_1 = h_2 = \cdots = h_\ell = 0$ where $h_i \in \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_n]$ ($\mathbb{R}$ is the field of real numbers), the $u_1, \ldots, u_m$ are parameters, and the $x_1, \ldots, x_n$ are dependent variables. The conjecture has the form:

$$(\forall \mathbf{u}, \mathbf{x})[h_1 = h_2 = \cdots = h_\ell = 0 \;\; \Rightarrow \;\; g = 0] \tag{1}$$

where $\mathbf{u} = (u_1, \ldots, u_m)$, $\mathbf{x} = (x_1, \ldots, x_n)$ and $g = g(\mathbf{u}, \mathbf{x}) \in \mathbb{R}[\mathbf{u}, \mathbf{x}]$.

**Degeneracy and Generic Truth.** A theorem of the form (1) is called a *universal truth.* It turns out that the classical notion of theoremhood is more subtle, and this led Wu to formulate the notion of *generic truth.* We formalize it as follows: let $\Delta_1, \ldots, \Delta_k$ be predicates on the variables $\mathbf{u}, \mathbf{x}$. We call each $\Delta_i$ a *non-degeneracy condition.* The conjecture (1) is *generically true* relative to $\{\Delta_1, \ldots, \Delta_k\}$ if

$$(\forall \mathbf{u}, \mathbf{x})[\Delta_1, \Delta_2, \ldots, \Delta_k, h_1 = h_2 = \cdots = h_\ell = 0 \;\; \Rightarrow \;\; g = 0]. \tag{2}$$

Classical ruler-and-compass theorems are indeterminate in that they do not explicitly specify the degenerate conditions. Hence part of “proving a classical theorem” involves discovering a suitable set of non-degeneracy conditions. Hopefully the set is minimal is some sense (but not necessarily unique). The simplest kind of non-degeneracy condition has the form

$$\Delta : d \not\models 0$$

where $d$ is a polynomial. Call this the *first kind* of non-degeneracy condition. The *degree* of the $\Delta$ is equal to the total degree of $d$. If each $\Delta_i$ has degree $d_i$, then the *degree* of $\{\Delta_1, \ldots, \Delta_k\}$ is $\sum_{i=1}^{k} d_i$. Typical examples of the first kind of non-degeneracy may require two points to be distinct or two lines to be non-parallel. It is easy to see that both have degree 2.

**Example 1** (continued). The non-degeneracy conditions require the intersection points $P, S$ and $Q$ be not at infinity. Equivalently, the following pairs of lines are not parallel: $\{AB, DF\}$, $\{BC, FE\}$, $\{CD, EA\}$. So the degree of these non-degeneracy conditions is 6.

**Second Kind of Degeneracy.** The *second kind* of non-degeneracy condition arises for theorems in the real field. For example, when we define a point by the intersection of two circles, we require that these two circles intersect. Or, when we define three collinear points $A, B$ and $C$, we may require $B$ to lie between the other two points. Such non-degeneracy conditions have the form

$$\Delta : d \geq 0$$

where $d$ is a polynomial. We can modify this condition using a well-known trick:

$$\Delta' : \;\; \exists z, \; d - z^2 = 0$$

where $z$ is a new variable. The existential quantifier on $z$ can be pulled out as a prenex universal quantifier. Thus, we can formulate the conjecture as

$$(\forall \mathbf{u}, \mathbf{x}, \mathbf{z}) \ \ (\Delta', H \ \ \Rightarrow \ \ T).$$

In practice, there may be other ways to handle this: in the Pascal example, such non-degeneracies demand that the parameters $u_j$ (for $j = 2, 3, 4, 5, 6$) satisfy $|u_j| \leq |u_1|$. Our prover can handle non-degeneracy conditions of the second kind when put in this form. Indeed, in all the examples we looked at in [5], such a formulation is possible.

**Reduction to Radical Expressions.** In a ruler & compass construction, each dependent variable is a radical function of the previously introduced variables. As exemplified by Pascal's Theorem, all the dependent variables are introduced either (i) singly by a single equation (e.g., $x_1$ is introduced by $h_1 = 0$) or (ii) in pairs by two equations (e.g., $x_6, x_7$ are introduced by $h_6 = h_7 = 0$). As all equations are at most quadratic, the $x_i$'s can be replaced by radical expressions involving the $u_j$'s. Let $G = G(\mathbf{u})$ be the radical expression after such a substitution into $g(\mathbf{u}, \mathbf{x})$. The universal truth conjecture (1) now says

$$(\forall \mathbf{u})[G = 0],$$

with an analogous statement for generic truth. Another issue arises: each radical is determined only up to a ± sign. Hence, if there are $r$ radicals in $G$, we must replace $G = 0$ by the system of $2^r$ radical expressions, $G_1 = G_2 = \cdots = G_{2^r} = 0$, in which each of the $2^r$ possible sign combinations are used. If a single function $G^*(\mathbf{u})$ is desired, we can use $G^* = \sum_{i=1}^{2^r} G_i^2$. The appearance of "$2^r$" in this expression may be disturbing from a complexity viewpoint. Several observations suggest that this is not serious in practice. First, $r$ is typically small ($r = 5$ in Pascal's theorem). Next, we can reduce the number of summands in $G^*$ from the worst case of $2^r$ terms. There are two ways this can happen: (A) Symmetries in the problem may arise so that many of the $G_i$'s can be omitted. (B) Certain sign combinations may be excluded by the nature of the construction and/or theorem so that $G^*$ may represent a sum of less than $2^r$ radical expressions. In particular, using (A) and (B), we can always omit half of the summands in standard geometric theorems. Thus, $2^{r-1}$ terms suffice in $G^*$.

**Example 2** (Butterfly Theorem). We illustrate the reduction in the number of terms in $G^*$ using the Butterfly Theorem in [5, Example 2.4, p. 9]. The theorem concerns 4 co-circular points $A, B, C$ and $D$. Let $O$ be the center of this circle and $E$ be the intersection of $AC$ and $BD$. The points $A, B, C, D, E$ form a "butterfly". If the line perpendicular to $OE$ and passing through $E$ intersects the lines $AD$ and $BC$ at $G$ and $F$ (respectively), then the theorem says that segments $EF$ and $EG$ have the same length. There are 3 quadratic equations in formulating this theorem (so $r = 3$). In the construction described by Chou, the point $E$ is placed at the origin $(0, 0)$ and $O$ is placed at $(u_1, 0)$. $A$ is freely placed at $(u_2, u_3)$. The point $C$ is now completely determined, and has two possible solutions. In one solution, $C$ and $A$ coincide, and the nature of the theorem excludes this case. Next, the points $B$ is freely chosen on the circle (and this introduces one parameter). Again there are two possible solutions. But it is clear by symmetry that we can arbitrarily choose one of them without loss of generality. Therefore, $G^*$ only needs two terms (corresponding to choosing the 2 solutions for $D$). ■

The fact that our prover can address theorems about real geometry is illustrated by the following simple example.

**Example 3** (Triangle Bisectors). Let $A$, $B$, $C$ be three non-linear points, and $D$ be the intersection point of the angle bisectors of $\angle A$ and $\angle B$ in the triangle $\triangle ABC$. We want to prove that $D$ must be on the bisector of $\angle C$ in $\triangle ABC$.

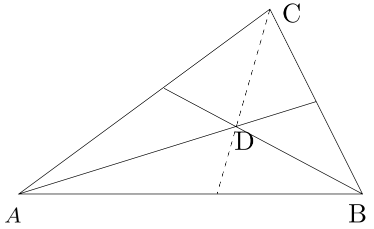


**Fig. 2.** Coincidence of three angle bisectors.

Let $A = (0, 0)$, $B = (u_1, 0)$, $C = (u_2, u_3)$, $D = (x_4, x_5)$. This gives the following equations for the hypotheses.

| *Equation* | *Geometry* | *Remark* |
|---|---|---|
| $h_1 : x_1^2 - u_1^2 = 0, x_1 \geq 0$ | $[x_1 \equiv \|AB\|]$ | Introduces $x_1$ |
| $h_2 : x_2^2 - u_2^2 - u_3^2 = 0, x_2 \geq 0$ | $[x_2 \equiv \|AC\|]$ | Introduces $x_2$ |
| $h_3 : x_3^2 - (u_1 - u_2)^2 - u_3^2 = 0, x_3 \geq 0$ | $[x_3 \equiv \|BC\|]$ | Introduces $x_3$ |
| $h_4 : (x_1u_2 - x_2u_1)x_4 + x_1u_3x_5 = 0$ | $[D \in bisector(\angle A)]$ | Constrains $x_4, x_5$ |
| $h_5 : [(u_2 - u_1)x_1 + u_1x_3](x_4 - u_1) + x_1u_3x_5 = 0$ | $[D \in bisector(\angle B)]$ | Constrains $x_4, x_5$ |

The conclusion that $D$ is on the bisector of angle $\angle C$ can be formulated as the following thesis:

$$g = (x_4 - x_2)(u_1x_2 - u_2x_2 + u_2x_3) - (x_5 - x_3)(x_3 - x_2)u_3 = 0$$

The formulation explicitly introduces inequalities for $x_1, x_2, x_3$ to pick the internal angle bisectors. When regarded as a complex theorem, no such inequalities are allowed. In this case, each "bisector" can refer to either the internal or external bisector of an angle, so there are a total of $8 = 2^3$ choices for these bisectors. The "thesis" is true for exactly four of these choices, which also means that the theorem is false in complex geometry. Let $G(\mathbf{u})$ be the radical expression after eliminating the dependent variables from $g$. The 8 choices of bisectors correspond to different assignment of signs to the three radicals in $G(\mathbf{u})$. Our prover can be used to test the validity of each choice. ■

## 3 Randomized Zero Testing for Radical Expressions

### 3.1 Straight Line Programs

We need to generalize expressions to *straight line programs* (SLP). A SLP $\pi$ is a sequence of *steps* where each step is an assignment to a new *programming variable*. The $i$th step of a SLP has one of the forms

$$z_i \leftarrow x_i \circ y_i, \qquad (\circ \in \{+, -, \times, \div\}) \tag{3}$$

$$z_i \leftarrow \sqrt{x_i} \tag{4}$$

where $z_i$ is a newly introduced programming variable, $x_i$ and $y_i$ are either real constants, *input variables* or programming variables introduced in some earlier steps. Alternatively, we call an input variable an *independent variable* (or, *parameter*) and a programming variable a *dependent variable*. These $x_i$ and $y_i$ are said to be *used* in the $i$th step. The last introduced variable is called the *main variable* and it is never used. In general, a SLP can have branching steps. But this possibility is not considered in this paper.

An *expression* is a SLP where, with the exception of the main variable, each programming variable is used exactly once. Underlying each SLP is a labeled and ordered *dag* (directed acyclic graph) defined in the obvious way: each node corresponds to a constant or variable in the SLP. We often use the terms "nodes" and "variables" interchangeably. For the steps in (4) (resp., (3)), we introduce edges that are directed from $x_i$ (resp., $x_i$ and $y_i$) to $z_i$. We use standard graph-theoretic terminology to talk about this dag: sinks, sources, predecessor/successor nodes, etc. If $(u, v)$ is an edge of the dag, we call $u$ the *predecessor* of $v$, and call $v$ the *successor* of $u$. The nodes labeled by input variables or constants are *source nodes* while the non-source are labeled by programming variables. The sources may be called *leaves* in case the dag is a tree. The non-source nodes are associated with an operation ($\pm, \times, \div, \sqrt{\cdot}$) – so we may speak of "radical nodes", "multiplication nodes", etc. Variables that are not used correspond to *sink nodes* in the dag. The main variable corresponds to a sink node which we call *root*. The *radical depth* of a node $u$ is the maximum number of radical nodes in a path from $u$ to any root node, inclusive of the end points. Thus, if $u$ is a radical node, then the radical depth of $u$ is at least 1. For each node $u$, its *induced dag* is the subdag comprising all the nodes that can reach $u$ along a path. A SLP is said to be *rooted* if the root is the unique sink. The dags corresponding to expressions are ordered trees (hence rooted). Our SLP's are assumed rooted unless otherwise noted.

*Values.* Let $\mathbf{u} = (u_1, \ldots, u_m)$ be the input variables. For each variable $u$ in a SLP $\pi$, we inductively define its *value* to be an appropriate element $val_\pi(u)$

in an algebraic extension of $\mathbb{Q}(\mathbf{u})$. The extension is obtained by adjunction of square roots. The *value* of $\pi$ is the value of its main variable. More precisely, let $Q_0 = \mathbb{Q}(\mathbf{u})$ and define the tower of extensions defined by $\pi$ to be

$$Q_0 \subseteq Q_1 \subseteq Q_2 \subseteq \cdots \subseteq Q_r \tag{5}$$

where $Q_i := Q_{i-1}(\sqrt{\alpha_i})$ and the $i$th square-root in $\pi$ has operand $\alpha_i \in Q_{i-1}$. A SLP $\pi$ is also said to *compute* a collection $V \subseteq Q_r$ of values provided each $v \in V$ is the value of some variable in $\pi$.

**Rational Degrees.** Let $x$ be a node in a SLP $\pi$. We define the *rational degree* $\mathrm{rdeg}_\pi(x)$ of $x$ (the subscript $\pi$ is usually dropped). We need some auxiliary notions. For any node or variable $x$, let $\mathrm{RAD}(x)$ denote the set of radical nodes in the subdag of $\pi$ rooted at $x$. Write $\mathrm{RAD}(x, y)$ for $\mathrm{RAD}(x) \setminus \mathrm{RAD}(y)$ (set difference). Also let $\rho(x) := |\,\mathrm{RAD}(x)|$ and $\rho(x, y) := |\,\mathrm{RAD}(x, y)|$. We will inductively define $\mathrm{rdeg}(x)$ to be a pair of natural numbers $(a, b) \in \mathbb{N}^2$, but usually write it as "$a : b$". These two numbers are the "upper" and "lower" degrees of $x$ and denoted $\mathrm{udeg}(x)$ and $\mathrm{ldeg}(x)$. Thus,

$$\mathrm{rdeg}(x) = \mathrm{udeg}(x) : \mathrm{ldeg}(x).$$

Assuming $\mathrm{rdeg}(x) = a_x : b_x$ and $\mathrm{rdeg}(y) = a_y : b_y$, we inductively define $\mathrm{rdeg}(z)$ using the table:

| $z$ | $\mathrm{udeg}(z)$ | $\mathrm{ldeg}(z)$ |
|---|---|---|
| constant | 0 | 0 |
| parameter | 1 | 0 |
| $x \times y$ | $a_x 2^{\rho(y,x)} + a_y 2^{\rho(x,y)}$ | $b_x 2^{\rho(y,x)} + b_y 2^{\rho(x,y)}$ |
| $x \div y$ | $a_x 2^{\rho(y,x)} + b_y 2^{\rho(x,y)}$ | $b_x 2^{\rho(y,x)} + a_y 2^{\rho(x,y)}$ |
| $x \pm y$ | $\max(a_x 2^{\rho(y,x)} + b_y 2^{\rho(x,y)}, b_x 2^{\rho(y,x)} + a_y 2^{\rho(x,y)})$ | $b_x 2^{\rho(y,x)} + b_y 2^{\rho(x,y)}$ |
| $\sqrt{x}$ | $a_x$ | $b_x$ |

The *rational degree* of the SLP $\pi$ is defined to be $a : b$ where $a = \max_x \mathrm{udeg}(x)$, $b = \max_x \mathrm{ldeg}(x)$, and $x$ ranges over the nodes in $\pi$. Note that if $\pi$ is division-free, then $\mathrm{ldeg}(x) = 0$ for all $x$.

**Alternative Approach.** It is useful to have an alternative approach to rdeg which does not involve $\rho(x, y)$ or $\rho(y, x)$. In particular, we define $\mathrm{rdeg}_2(z) = \mathrm{udeg}_2(z) : \mathrm{ldeg}_2(z)$ inductively using the following table: as before, we assume $\mathrm{rdeg}_2(x) = a_x : b_x$ and $\mathrm{rdeg}_2(y) = a_y : b_y$.

| $z$ | $\mathrm{udeg}_2(z)$ | $\mathrm{ldeg}_2(z)$ |
|---|---|---|
| constant | 0 | 0 |
| parameter | 1 | 0 |
| $x \times y$ | $a_x + a_y$ | $b_x + b_y$ |
| $x \div y$ | $a_x + b_y$ | $b_x + a_y$ |
| $x \pm y$ | $\max\{a_x + b_y, b_x + a_y\}$ | $b_x + b_y$ |
| $\sqrt{x}$ | $\frac{a_x}{2}$ | $\frac{b_x}{2}$ |

Notice that these degrees are no longer natural numbers but binary fractions. The following lemma gives the connection between the two definitions of rdeg.

**Lemma 1.** *For any variable $z$ in a SLP, we have*

$$\mathrm{udeg}(z) = 2^{\rho(z)}\,\mathrm{udeg}_2(z), \qquad \mathrm{ldeg}(z) = 2^{\rho(z)}\,\mathrm{ldeg}_2(z).$$

### 3.2 Equivalent Transformations

Two variables (resp. SLP's) are said to be *equivalent* if they have the same value. Transformations of an SLP that do not change its value are called *equivalent transformations* (but the set of computed values may change). Equivalent transformations may change the rational degree, as when applying the distributive law:

$$z(x+y) \Rightarrow zx + zy. \tag{6}$$

It is easy to verify that the rational degree of the left-hand side is at most that of the right-hand side. We next show that the rational degree is preserved in the absence of division (but allowing radicals):

**Lemma 2.** *If $\pi$ is division-free, then the transformation (6) preserves* rdeg *of $\pi$. In particular,*

$$\mathrm{rdeg}(z(x+y)) = \mathrm{rdeg}(zx+zy).$$

*Proof.* We only need to consider the upper degrees. With $\mathrm{udeg}(x) = a_x$, etc, as before, we have

$$\mathrm{udeg}(z(x+y)) = 2^{\rho(xy,z)}a_z + 2^{\rho(z,xy)}\max\{a_x 2^{\rho(y,x)}, a_y 2^{\rho(x,y)}\}$$

while

$$\begin{aligned}\mathrm{udeg}(zx+zy) &= \max\{a_{zx}2^{\rho(zy,zx)}, a_{zy}2^{\rho(zx,zy)}\}\\ &= \max\{(a_z 2^{\rho(x,z)} + a_x 2^{\rho(z,x)})2^{\rho(zy,zx)}, (a_z 2^{\rho(y,z)} + a_y 2^{\rho(z,y)})2^{\rho(zx,zy)}\}.\end{aligned}$$

The lemma follows if we now verify the following:

$$\begin{aligned}\mathrm{RAD}(xy,z) &= \mathrm{RAD}(x,z) \uplus \mathrm{RAD}(zy,zx),\\ \mathrm{RAD}(xy,z) &= \mathrm{RAD}(y,z) \uplus \mathrm{RAD}(zx,zy),\\ \mathrm{RAD}(z,xy) \uplus \mathrm{RAD}(y,x) &= \mathrm{RAD}(z,x) \uplus \mathrm{RAD}(zy,zx),\\ \mathrm{RAD}(z,xy) \uplus \mathrm{RAD}(x,y) &= \mathrm{RAD}(z,y) \uplus \mathrm{RAD}(zx,zy).\end{aligned}$$

Our notation here, $A \uplus B$, refers to disjoint union of the sets $A$ and $B$. Let us only prove the first equation: the RHS is equivalent to $\mathrm{RAD}(x,z) \uplus \mathrm{RAD}(y,zx)$. We may verify that the union is indeed disjoint, and equal to $\mathrm{RAD}(xy,z)$. The other equations can be proved similarly. We omit the details here. ■

Next, we show that applying the associative laws for multiplication and addition does not affect rational degree. This follows from the following general result:

**Lemma 3.** *Let $x_i$ be variables in $\pi$ and $r_i = |\mathrm{RAD}(x_1, \ldots, x_k) \setminus \mathrm{RAD}(x_i)|$. Then*

$$\mathrm{rdeg}(\prod_{i=1}^{k} x_i) = \sum_{i=1}^{k} \mathrm{rdeg}(x_i) 2^{r_i}$$

$$\mathrm{udeg}(\sum_{i=1}^{k} x_i) = \max_{i=1}^{k} \{\mathrm{udeg}(x_i) 2^{r_i} + \sum_{j=1, j \not= i}^{k} \mathrm{ldeg}(x_j) 2^{r_j}\}$$

$$\mathrm{ldeg}(\sum_{i=1}^{k} x_i) = \sum_{i=1}^{k} \mathrm{ldeg}(x_i) 2^{r_i}$$

The above lemma justifies a generalization of SLP's in which we allow addition nodes and multiplication nodes to take an arbitrary number of arguments. These are called "sum" or $\sum$-nodes, and "product" or $\prod$-nodes, respectively. Such an SLP is called a *generalized SLP.* A path in a generalized SLP dag is said to be *alternating* if along the path, no two consecutive nodes are $\sum$-nodes and no two consecutive nodes are $\prod$-nodes. The SLP is *alternating* if every path is alternating. Clearly, any SLP can be made alternating without changing its rational degree. We can eliminating any non-alternating path in the SLP by aggregating the consecutive additions (or multiplications) using the $\sum$ (or $\prod$) operations. This process will terminate because each elimination reduces the number if nodes in a SLP.

### 3.3 Preparation

A SLP in which the last three steps has the form

$$\begin{aligned} &\ldots \\ &x \leftarrow \sqrt{w_C} \\ &y \leftarrow x \times w_B \\ &z \leftarrow y + w_A \end{aligned}$$

is said to be *prepared* (or in prepared form). Here $w_A, w_B, w_C$ are variables or constants. Thus $z$ is the main variable, and $x$ is the last radical variable to be introduced. Intuitively, the radical $x$ has been brought up as close to the root as possible, in preparation for a transformation (to be introduced) to remove the radical. We also call $x$ the *prepared variable.* If the values of $w_A, w_B, w_C$ are given by the expressions $A, B, C$ (resp.) then the value of $z$ is given by the expression

$$A + B\sqrt{C}.$$

Note special forms of this expression when $A = 0$ or $B = 1$, or both. If the SLP has no square roots, it is considered prepared already. Our goal is to prepare a given SLP, and to bound the resulting rational degree.

Let us now prepare a radical node $A_0$ with radical depth 1. Assume the SLP is division-free. Let $A_n, B_n$ be expressions ($n \geq 0$). The expressions $E_n$, for $n \geq 0$ is defined inductively as follows: $E_0 = A_0 \times B_0$, and for $n \geq 1$,

$$E_n = (E_{n-1} + A_n)B_n = ((E_{n-2} + A_{n-1})B_{n-1} + A_n)B_n = \cdots .$$

To show the dependence of $E_n$ on the $A_n$'s and $B_n$'s, we may also write $E_n = E_n(A_0, B_0, A_1, B_1, \ldots, A_n, B_n)$. Viewed as a tree, $E_n$ is essentially a single alternating path from the root down to $A_0$. This path is left-branching only and the root is a ×-node. Also write: $B_{(n)} := \prod_{j=0}^{n} B_j$.

**Lemma 4.** *For $n \geq 1$, the expression $E_n(A_0, B_0, \ldots, A_n, B_n)$ is equivalent to the expression*

$$E'_n := (A_0 \times B_{(n)}) + E_{n-1}(A_1, B_1, \ldots, A_n, B_n)$$

*Moreover, if $E_n$ is division-free, then* $\mathrm{rdeg}(E_n) = \mathrm{rdeg}(E'_n)$.

*Proof.* Proof by induction. When $n = 1$,

$$\begin{aligned} E_1 &= (A_0 \times B_0) + A_1) \times B_1 \\ &= (A_0 \times B_0 \times B_1) + A_1 \times B_1 . \end{aligned}$$

Assume that this lemma is held for $n \leq k$, then for $n = k + 1$,

$$\begin{aligned} E_{k+1} &= (E_k + A_{k+1}) \times B_{k+1} \\ &= ((A_0 \times B_{(k)}) + E_{k-1}(A_1, B_1, \ldots, A_k, B_k) + A_{k+1}) \times B_{k+1} \\ &= (A_0 \times B_{(k+1)}) + E_k(A_1, B_1, \ldots, A_{k+1}, B_{k+1}). \end{aligned}$$

Thus we know the equivalence of this transformation is held for any $n \in \mathbb{N}$.

In both cases, we only apply the distributive and associative laws, which do not change the rational degree when $E_n$ is division free. ■

This is illustrated in the case $n = 2$ by Figure 3. Note that the variable $A_0$ is prepared in $E'_n$. Actually, $E_n$ in this lemma can be a generalized SLP so that the $A_i, B_i$'s need not be distinct and the nodes can be $\sum$- and $\prod$-nodes. Then there is a corresponding equivalent SLP $E'_n$; this is the version that we will use in the next theorem.

We address the problem of multiple uses of a node. A node $u$ is *used* $k$ times if there are $k$ distinct paths from the root to $u$. If a radical node $u$ of radical depth 1 is used $k$ times, then if we judiciously apply the previous lemma $k$ times, each time eliminating one "use" of $u$, we obtain:

**Theorem 1.** *Suppose $\pi$ is a division-free SLP and $u$ is a radical node in $\pi$ with radical depth of* 1*. Then we can transform $\pi$ into an equivalent SLP $\pi'$ such that* $\mathrm{udeg}(\pi) = \mathrm{udeg}(\pi')$*. Moreover, either no node in $\pi'$ has the value $val_\pi(u)$ or else, there is a node $u'$ in $\pi'$ with the following properties:*

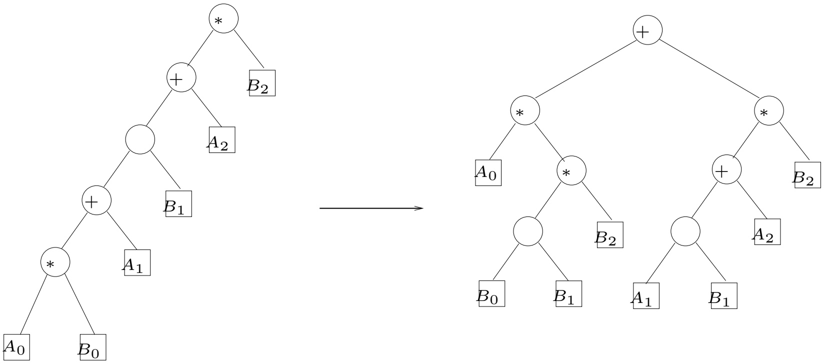


**Fig. 3.** The Transformation $E_2 \mapsto E_2'$.

*1. $u'$ is the prepared variable in $\pi'$*
*2. $u'$ is the unique node in $\pi'$ with value $val_\pi(u)$.*

*Proof.* We may assume that $\pi$ is a generalized, alternating SLP. Fix any path $p$ from $u$ to the root and we may assume that this alternating sum-product path has the same form as the path from $A_0$ to the root of $E_n$ in lemma 4. We then apply the previous lemma in which $u$ now plays the role of the node $A_0$ in $E_n$. This collapses the path $p$ to length 2, as in the lemma and the resulting SLP is in a prepared form $E' = u \times A + B$. If the variable $u$ is used in $A$ and/or $B$, then we can repeat this process for another path $p'$ (if any) in $A$ or $B$. We can repeat this process for the subexpressions $A$ and/or $B$, if they contain references to the node $u$ as well. There are two cases:

1. $u$ is used in $A$, then A is transformed to $A' = u \times A_1 + B_1$ and $E' = u \times B_1 + (A_1 u^2 + B)$. Remember that $u$ is a square root and thus the expression $u^2$ effectively eliminates the square root operation here;
2. $u$ is used in $B$, then $B$ is transformed to $B' = u \times A_2 + B_2$ and $E' = u \times (A + A_2) + B_2$.

In both cases, we can see that $E'$ is still in a prepared form. We keep this process until there is no use of $u$ except the one that is in the prepared position and has a unique path to the root with length 2. Since there must be a finite number of uses of $u$, this iterative process will eventually terminate. At that point, the resulting SLP $\pi'$ has the desired form: $\pi'$ is prepared and $u$ is the main prepared variable. It is also clear that if there are other nodes with the same value as $u$, they can also be merged with $u$ by the same process. Hence, $u$ will be the unique node with value $val_\pi(u)$.

Note that we apply the commutative, associative and distributive laws in these transformations. The commutative and associative transformations do not change the rational degree. Since $\pi$ is division free, Lemma 2 tells us that the distributive transformation preserves the rational degree too. Therefore, the preparation transformation does not change the rational degree of $\pi$. ■

We say that $\pi'$ is obtained by the process of "preparing" $u$ in $\pi$.

### 3.4 Main Result

Let $\pi$ be a SLP whose value is $V = V(\mathbf{u}) \in \mathbb{Q}_r$ (see (5)). We define the real function $f_\pi : \mathbb{R}^m \to \mathbb{R}$ where $f_\pi(a_1, \ldots, a_m)$ is the value of the main variable in $\pi$ when we evaluate each dependent variable at $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{R}^m$, following $\pi$ in a step-by-step fashion. The *domain* of $f_\pi$ comprises those $\mathbf{a} \in \mathbb{R}^m$ where $f_\pi(\mathbf{a})$ is defined. Similarly, we define an associated real function $f_V : \mathbb{R}^m \to \mathbb{R}$. Note that the domain of $f_\pi$ is always a subset of $f_V$. The following example shows that it may be a proper subset: let $\pi$ compute the value $V = \sum_{i=0}^{n-1} x^i$ using Horner's rule, and let $\pi'$ compute the same $V$ using the formula $V = \frac{x^n - 1}{x-1}$. Then $\pi$ and $\pi'$ are equivalent, but $\pi(1) = n$ while $\pi'(1)$ is undefined. The domain of $\pi$ (and $V$) is $\mathbb{R}$ but the domain of $\pi'$ is $\mathbb{R} - \{1\}$.

**Theorem 2.** *Suppose $V = V(\mathbf{u})$ is the non-zero value of a rooted division-free SLP $\pi$. Then there exists a non-zero polynomial $P(\mathbf{u})$ such that $\mathtt{Zero}(V) \subseteq \mathtt{Zero}(P)$ with $\deg P(\mathbf{u}) \leq \mathrm{udeg}(\pi)$.*

*Proof.* We show the existence of the polynomial $P(\mathbf{u})$ by induction on the number $r$ of square roots in $\pi$. For $r = 0$, the result holds because $V$ is already a polynomial of degree $\mathrm{udeg}(\pi)$.

Assume $r > 0$ and let $u$ be a radical node of radical depth 1 in $\pi$. We prepare $u$, leading to an equivalent SLP (which we still call $\pi$). The udeg of $\pi$ is unchanged by this transformation. If $C$ is the value of $u$, then the value of $\pi$ can be written as

$$V = A + B\sqrt{C}$$

where $A, B, C$ belongs to $Q_{r-1}$ (recall that values of programming variable introduced before the $r$th root extraction belongs to the field $Q_{r-1}$, by definition of $Q_{r-1}$). If $B = 0$ then $V = A$ and the result is true by the inductive hypothesis applied to $A$ (which has $\leq r-1$ square roots). Otherwise, by applying some further (obvious) transformations, we transform $\pi$ to some $\pi'$ whose value is

$$V' = A^2 - B^2 C. \tag{7}$$

Note that $\pi'$ has $\leq r-1$ square-roots. If $V' = 0$ then $0 = V' = (A + B\sqrt{C})(A - B\sqrt{C})$. Since $Q_r$ is a UFD and $V = A + B\sqrt{C} \not\models 0$ (by assumption), we conclude that $A - B\sqrt{C} = 0$, i.e., $\sqrt{C} = A/B \in Q_{r-1}$. Thus $V = A + B\sqrt{C} = 2A$. Then $V$ can be computed by some SLP with $\leq r-1$ square-roots, and the result follows by inductive hypothesis.

So assume $V' \not\models 0$. By induction,$\mathtt{Zero}(V') = \mathtt{Zero}(A^2 - B^2C) \subseteq \mathtt{Zero}(P)$ for some $P$ with $\deg(P) \leq \mathrm{udeg}(V')$. Since $\mathtt{Zero}(V) \subseteq \mathtt{Zero}(V')$, it remains to show that $\mathrm{udeg}(V') \leq \mathrm{udeg}(V)$. We have

$$\begin{aligned}
&\mathrm{udeg}(V) = \mathrm{udeg}(A + B\sqrt{C})\\
&= \max\{\mathrm{udeg}(A)2^{\rho(B\sqrt{C},A)}, \mathrm{udeg}(B\sqrt{C})2^{\rho(A,B\sqrt{C})}\}\\
&\geq \max\{\mathrm{udeg}(A)2^{1+\rho(B^2C,A)}, \left[\mathrm{udeg}(B)2^{\rho(\sqrt{C},B)} + \mathrm{udeg}(C)2^{\rho(B,\sqrt{C})}\right] 2^{\rho(A,B^2C)}\}
\end{aligned}$$

$$= \max\{2\,\mathrm{udeg}(A)2^{\rho(B^2C,A)}, \left[\frac{\mathrm{udeg}(B^2)}{2}2^{1+\rho(C,B^2)} + \mathrm{udeg}(C)2^{\rho(B^2,C)}\right]2^{\rho(A,B^2C)}\}$$

$$\geq \max\{\mathrm{udeg}(A^2)2^{\rho(B^2C,A)}, \left[\mathrm{udeg}(B^2)2^{\rho(C,B^2)} + \mathrm{udeg}(C)2^{\rho(B^2,C)}\right]2^{\rho(A^2,B^2C)}\}$$

$$= \mathrm{udeg}(A^2 - B^2C) = \mathrm{udeg}(V').$$

■

### 3.5 Presence of Division

What if the SLP is not division-free? Note that the presence of division is very common. For instance, when we intersect two lines in the construction, it gives rise to an expression with division. There is a well-known transformation to move all divisions towards the root, merging them as we go. An instance of this transformation is

$$\frac{A}{B} + \frac{A'}{B'} \Rightarrow \frac{AB' + A'B}{BB'}.$$

Unfortunately, the number of radical nodes may be doubled because if we move a division node past a radical node, we obtain two radical nodes:

$$\sqrt{\frac{A}{B}} \Rightarrow \frac{\sqrt{A}}{\sqrt{B}}. \qquad (8)$$

Hence we give two versions of this transformation in the following lemma: in version (i) we do not move any division node past a radical node, and in version (ii) we remove all but at most one division node.

**Lemma 5 (Elimination of Division).** *Let $\pi$ be a rooted SLP.*
*(i) There is an equivalent SLP $\pi'$ in which each division node is either the root of $\pi$ or the child of a radical node. Moreover,* $\mathrm{rdeg}(\pi') = \mathrm{rdeg}(\pi)$ *and $\pi'$ has the same number of radical nodes as $\pi$.*
*(ii) There is an equivalent SLP $\pi''$ with only one division node which is also the root. In this case* $\mathrm{rdeg}(\pi'') \leq 2^r\,\mathrm{rdeg}(\pi)$.

The proof of (ii) exploits the alternative definition of udeg($u$). Because the justification of the alternative definition is long, we only refer to the details in [15].

The value of the SLP $\pi''$ has the form $A/B$ where $A, B$ are division-free. Intuitively, to check if $A/B = 0$, we check if $A = 0$ subject to $B \not\models 0$. SinceA is division-free, we may apply main theorem (see next Section). This effectively amounts to doubling the number of square roots to prove a theorem involving division.

### 3.6 Improved Square Root Transformation

It turns out that we can exploit another trick motivated by [18] in order to avoid the doubling of the number of square roots. Instead of (8), we use the following transformation to extract division out of square roots:

$$\sqrt{\frac{A}{B}} \Rightarrow \begin{cases} \frac{\sqrt{AB}}{B} & \text{if } \operatorname{udeg}(A) \geq \operatorname{udeg}(B), \\ \\ \frac{A}{\sqrt{AB}} & \text{if } \operatorname{udeg}(A) < \operatorname{udeg}(B). \end{cases} \tag{9}$$

Suppose our transformations for eliminating divisions, using the new rule (9), transform an arbitrary expression $z$ into $U(z)/L(z)$ where $U(z), L(z)$ are division free. Let $u_z$ and $\ell_z$ denote the udeg$(U(z))$ and udeg$(L(z))$. To exploit the advantages of this new rule, we now give an explicit set of inductive rules for computing $u_z$ and $\ell_z$:

| $z$ | $u_z$ | $l_z$ |
|---|---|---|
| constant | 0 | 0 |
| parameter | 1 | 0 |
| $x \times y$ | $u_x + u_y$ | $l_x + l_y$ |
| $x \div y$ | $u_x + l_y$ | $l_x + u_y$ |
| $x \pm y$ | $\max\{u_x + l_y, l_x + u_y\}$ | $l_x + l_y$ |
| $\sqrt{x}$ | $\frac{1}{2}(u_x + l_x)$, $(u_x \geq l_x)$;<br>$u_x$, $(u_x < l_x)$. | $l_x$, $(u_x \geq l_x)$;<br>$\frac{1}{2}(u_x + l_x)$, $(u_x < l_x)$. |

Note that [18] only uses one of two clauses in (9) unconditionally. But the effect of using the two conditional clauses is that the resulting bound $u_z$ is never worse than $2^r$ udeg$(z)$, which is the bound in Lemma 5. The proofs may be found in [15].

## 4 Proving by Random Examples

We show how to use our main result to prove theorems about ruler & compass constructions. According to Section 2, this amounts to verifying if a radical expression $G^*(\mathbf{u})$ is identically zero (subject to non-degeneracy conditions). Let $\pi(\mathbf{u})$ be the natural SLP which computes the values of all the dependent variables in a ruler & compass construction, and whose value is the polynomial thesis $G^*(\mathbf{u})$. We give a simple upper estimate on the rdeg of each node in $\pi$.

Each "stage" of our construction introduces new points, lines or circles. Let us now be more precise: assume that our system maintains three kinds of geometric objects: points, lines and circles. These are constructed as follows:

- Points: There are three cases. **Case 0**: We can introduce an arbitrary point, $P$. Then $P.x$ and $P.y$ are free variables (i.e., parameters). **Case 1**: We can introduce an arbitrary point, $P$ on an existing line $L$ or circle $C$. We may specify either $P.x$ or $P.y$ to be a parameter. The other coordinate is therefore a dependent variable, constrained by an equation. **Case 2**: We can introduce a point $P$ that arises from the intersection of a line/circle with another line/circle. In this case, $P.x$ and $P.y$ are both dependent variables constrained by a pair of simultaneous equations. There is a variation of Case 2, which arises when at least one of the two intersecting objects is a circle. In this case, we allow the user to obtain both the points of intersection[1].

- Lines: Given two existing points, we can construct the line through them.
- Circles: Given three points $P, Q, R$, we can construct the circle centered at $P$ of radius equal to the distance between $Q$ and $R$. As a special case, if $P$ is equal to $Q$ or $R$, we can just use two arguments for this construction.

**Lemma 6.** *If the dependent variable $x$ is introduced at stage $i$ , then* $\mathrm{rdeg}_2(x) \leq 85^i$*, i.e.,* $\mathrm{udeg}_2(x) \leq 85^i$*,* $\mathrm{ldeg}_2(x) \leq 85^i$*.*

*Proof.* Proof by induction. Let $S_k$ be the set of objects (points, lines, etc.) available after $k$ construction stages. This lemma is trivially true when $k = 0$ because $S_0$ is empty.

Let $r_k = 85^k$. By the induction hypothesis, we assume that the coordinate (e.g., for points) or coefficient (e.g., in a line or circle equation) variables for all the objects in $S_k$ have rational degrees at most $r_k$.

Let us first consider the construction of lines and circles. Recall that in our system, a line refers to one that is constructed by linking two points in $S_k$; while a circle means one that is constructed with the center in $S_k$ and the radius being the length of some segment between two points in $S_k$. We represent a line by a linear equation $ax + by + c = 0$. It is easily verified that the rational degrees of $a$, $b$ and $c$ are at most $2r_k, 2r_k$ and $6r_k$, respectively. Similarly, we represent a circle by an equation in the form of $(x - a)^2 + (y - b)^2 = c^2$ where the rational degrees of $a, b$ and $c$ are at most $r_k, r_k$ and $4r_k$, respectively.

Next, we consider the construction of points. As discussed above, we can have one of the three types of construction (Cases 0, 1, 2) in stage $(k + 1)$. Case 0 is trivial because all the parameters have the rational degree $1 : 0$. Case 1 can be viewed as a simplified Case 2. In the following, we focus on the more interesting Case 2 constructions.

There are three possible constructions in a Case 2 stage.

First, we consider the intersection of two lines $L_1 : a_1x + b_1y + c_1 = 0$ and $L_2 : a_2x + b_2y + c_2 = 0$ where $a$'s, $b$'s and $c$'s can be at most $r_k$. We obtain the intersection point $(x, y)$ of these two lines as follows,

$$\left(\frac{c_1b_2 - c_2b_1}{a_1b_2 - a_2b_1}, \frac{c_1a_2 - c_2a_1}{a_2b_1 - a_1b_2}\right).$$

From the definition (see Section 3.1), the rational degrees for $x$ and $y$ are at most $8r_k$.

Next, let us consider the intersection of a line $L : a_1x + b_1y + c_1 = 0$ and a circle $C : (x - a_2)^2 + (y - b_2)^2 = c_2^2$. We eliminate $y$ and get a quadratic equation

[1] It should be possible to allow the user to pick one of the two points using some criteria, but we defer this to a future paper on implementation. This additional power is sometimes needed in ruler-and-compass theorems.

for $x$ as follows:

$$(1+\frac{a_1^2}{b_1^2})x^2+(-2a_2+2\frac{a_1}{b_1}(\frac{c_1}{b_1}+b_2))x+((\frac{c_1}{b_1}+b_2)^2+a_2^2-c_2^2)=0.$$

Let $A, B$ and $C$ be the three coefficients in the above equation. It can be shown that the rational degrees of them can at most be $4r_k$, $6r_k$ and $10r_k$ respectively. From the above equations, we get $x = \frac{-B\pm\sqrt{B^2-4AC}}{2A}$ and $y = -\frac{a_1x+c_1}{b_1}$. Thus, $\mathrm{rdeg}_2(x) \leq 23r_k$ and $\mathrm{rdeg}_2(y) \leq 26r_k$.

Thirdly, we consider the intersection of two circles: $C_1 : (x-a_1)^2+(y-b_1)^2 = c_1^2$ and $C_2 : (x-a_2)^2+(y-b_2)^2 = c_2^2$. We subtract them first to obtain a linear equation first. Then by arguments similar to those used for the intersection of a line and a circle, we can show that the rational degrees for $x$ and $y$ are at most $69r_k$ and $85r_k$, respectively.

Therefore, we know that $\mathrm{rdeg}_2(x) \leq 85^i$ for all the nodes at the stage $i$. ■

REMARK: The constant 85 in the above lemma is clearly very conservative. This bound can be refined, for example, by classifying the stages into the various types of construction.

**Corollary 1.** *Let the thesis polynomial be $g(\mathbf{u},\mathbf{x})$ with $\deg(g) = d$, and $G(\mathbf{u})$ be any of the $2^r$ radical expressions derived from $g(\mathbf{u},\mathbf{x})$ by eliminating dependent variables. Then $\mathrm{rdeg}_\pi(G) \leq td2^r85^k$ where $g(\mathbf{u},\mathbf{x})$ has $t$ terms and $k$ is the number of construction stages.*

*Proof.* For Lemma 6, we know that the rational degrees for all the dependent and independent variables are at most $85^k$. The thesis $G$ has $t$ terms with total degree at most $d$. By the inductive definitions of rational degrees, we know that $\mathrm{rdeg}_\pi(G) \leq td2^r85^k$. ■

Assume an incremental construction with $m$ parameters, $n$ dependent variables, $k$ stages, and $r$ quadratic equations. Note that $t$ is at most $\binom{m+n+d}{d}$. Moreover, $d \leq 2$ in most classical geometric theorems. In our implementation, instead of relying on this crude upper bound, we actually compute the actual bounds on rdeg to achieve better performance. By applying Lemma 5(ii) to $\pi$, we obtain $\pi''$ with one division at the root, and $\mathrm{rdeg}(\pi'') \leq 2^r\,\mathrm{rdeg}(\pi)$. Now the value of $\pi''$ (which is $G^*$) has the form $A/B$ where $A, B$ are division-free. Moreover, $\mathrm{rdeg}_{\pi''}(G^*) \leq td2^{2r}85^k$. Clearly, $\mathtt{Zero}(A/B) \subseteq \mathtt{Zero}(A)$. Without loss of generality, assume $A \not\models 0$. By our main theorem,$\mathtt{Zero}(A) \subseteq \mathtt{Zero}(P)$ for some polynomial $P$ of degree $\leq td2^{2r}85^k$. Then we invoke a simple form of Schwartz's lemma:

**Fact 1.** *Let $P(\mathbf{u})$ be a non-zero polynomial of degree at most $D$. If each $a_i$ $(i = 1,\ldots,m)$ is randomly chosen from a finite set $S \subseteq \mathbb{R}$. Then the probability that $P(a_1,\ldots,a_m) = 0$ is at most $D/|S|$.*

If we randomly pick the values $\mathbf{a} = (a_1,\ldots,a_m) \in S^m$, and $|S| = td2^{c+2r}85^k$ (for any $c \geq 1$) then the “error probability” of our procedure is given by $\mathtt{Pr}\{A(\mathbf{a}) =$

$0\} \leq \texttt{Pr}\{P(\mathbf{a}) = 0\} \leq 2^{-c}$. This constitutes our probabilistic verification of the universal truth of "$G^*(\mathbf{u}) = 0$".

An alternative to testing $G^*(\mathbf{u}) = 0$ is viewing the problem as testing the simultaneous vanishing of a set of polynomial $\mathcal{G} := \{G_1(\mathbf{u}), \ldots, G_{2^r}(\mathbf{u})\}$. This reduces the complexity in two ways:

– The root bound (which determines the precision necessary to numerically determine the sign of radical expressions in the Core Library) is smaller.
– The size of the test set $S$ is smaller.

We also have a further choice when testing $\mathcal{G}$: we can randomly choose some $G_i$ to test for its vanishing, or we can choose to randomly test all the $G_i$'s for their vanishing. However, the random choice of $G_i$ does not seem to be the most efficient way to test a theorem.

**Degeneracies of the First Kind.** We now address the generic truth of "$G^*(\mathbf{u}) = 0$". The notion of "error probability" becomes an interesting issue. First consider only non-degeneracy conditions of the first kind, $\Delta : \delta \not\models 0$. For simplicity, assume the $i$th ruler & compass construction step introduces exactly one such condition, $\delta_i \not\models 0$, of degree$\leq 2$. Since there are $k$ stages of construction, the non-degeneracy condition becomes $\delta^* := \delta_1\delta_2 \cdots \delta_k \not\models 0$. The degree of $\delta^*$ is thus at most $2k$.

There are two natural models of what it means to have an "error probability" $\leq 2^{-c}$: (A) The "strict model" says that our sample space is now restricted to $S^m \setminus \{\mathbf{a} : \delta(\mathbf{a}) = 0\}$. (B) Alternatively, we can say that the sample space is still $S^m$ but the theorem is trivially true at $S^m \cap \{\mathbf{a} : \delta(\mathbf{a}) = 0\}$. Given a finite test set $S$, the possible zeros of $\delta^*$ (i.e., degenerate configurations) in $S^m$ is at most $2^{2r}\,\mathrm{udeg}(\delta^*)|S|^{m-1}$. With a large enough test set $S$, we can make the probability that degenerate cases are chosen in the test (i.e., $2^{2r}\,\mathrm{udeg}(\delta^*)/|S|$) arbitrarily small. We adopt the model A in the next theorem:

**Theorem 3.** *Conjectures about ruler & compass constructions with s non-degenerate conditions of the first kind can be verified with error probability* $\leq 2^{-c}$ *in time polynomial in the parameters* $2^r, 2^s, k, c, \lg(t)$ *and* $\lg(d)$*, where* $r$ *is the number of square roots in the thesis radical expression* $G(\mathbf{u})$*,* $k$ *is the number of construction stages,* $t$ *is the number of monomials in the thesis polynomial* $g(\mathbf{u}, \mathbf{x})$*, and* $d$ *is the total degree of* $g$*.*

*Proof.* Each construction introduces a constant number of new operations into the final radical thesis expression $G^*(\mathbf{a})$. Thus, the cost to construct the thesis expressions $G^*(\mathbf{a})$ is bound by $O(k)$. Next, let us consider the complexity in verifying $G(\mathbf{a})$ for some sample configuration $\mathbf{a} = (a_1, a_2, \ldots a_m)$ randomly chosen from a finite test set $S$ with a cardinality of $2^{2r+c}85^k td$. From the discussion above, we know that the failure probability of this test is at most $2^{-c}$. Without loss of generality, we can assume all the elements in $S$ are integers. So the bit length of each instance value is bounded by $L = \lg(|S|) = O(r + c + \lg(t) + \lg(d) + k)$. In our root bound based approach to determine the exact sign of an algebraic expression [16], the number of bits which need to compute in the verification is bounded by $O(pL2^{2r})$, where $p$ is the total number of operations in $G^*$ which is bounded by $O(k)$. It is known that the time complexity of arithmetic operations among multiple precision numbers are no more than $O(\ell^2)$ where $\ell$ is the bit length of operands. We have a total of $2^r$ radical thesis expressions to verify. So the complexity to verify the vanishing of $G^*$, when exact arithmetic is employed, is polynomial in $2^r, k, c, \lg(t)$ and $\lg(d)$.

In presence of $s$ non-degeneracy conditions of the first kind, let $\Delta(\mathbf{u})$ be the product of all of them. It is a radical expression in $\mathbf{u}$. By our main theorems, the number of zeros of $\Delta$ in $S^m$, $N$, is polynomial in $2^s$ and $2^r$. In the worst case, we may meet at most $N$ degenerate cases before we get the first non-degenerate one. So the worst case complexity for our complete method is polynomial in $2^r, 2^s, k, c, \lg(t)$ and $\lg(d)$. ■

**Degeneracies of the Second Kind.** As noted, degeneracies of the second kind can often be reduced to simple constraints on the domains of the parameters, possibly depending on the values of other parameters. For instance, we noted that in Pascal's Theorem, the parameters $u_i$ $(i = 2, \dots, 6)$ must satisfy $|u_i| \leq |u_1|$. Our prover can handle such degeneracies by exploiting the following more general form of fact 1: define the *generalized degree* of $p(x_1, \dots, x_n)$ to be $(d_1, \dots, d_n)$ where the degree of $p$ is $d_1$ when viewed as a polynomial in $x_1$ and its leading coefficient inductively has generalized degree $(d_2, \dots, d_n)$. Suppose $S_1, \dots, S_n$ are finite sets of real numbers, then it can be shown that if we choose $(u_1, \dots, u_n)$ randomly from $S_1 \times S_2 \times \cdots \times S_n$, the probability that $p$ is non-zero and $p(u_1, \dots, u_n) = 0$ is at most

$$\frac{d_1}{|S_1|} + \cdots + \frac{d_n}{|S_n|}.$$

The main extra complexity caused by this version of our prover is that we need to evaluate the parameters at rational values (instead of just at integer values).

The current implementation does not handle the second kind of degeneracy in the above way, but we plan to rectify this in the future. Instead, it detects when an example $\mathbf{a} \in S^m$ is degenerate, discards it and generates another example, etc. Under probability model (A) above, this means that we do not have an á priori bound on the running time, but the error probability is correct. Of course, under model (B), there is no need to generate another example; but this does not seem like a reasonable model.

**Degenerate Ruler-and-Compass Constructions.** Certain theorems amount to detecting the validity of construction steps. We give a simple example from [6] of a theorem true in real geometry but false in the complex geometry. The construction amounts to picking two points $P_1(0,0)$ and $P_2(u,0)$ where $u$ is a free parameter. Also let $P_3$ be the midpoint of $P_1P_2$, and $P_4$ the midpoint of

$P_1P_3$. Let $L$ be the bisector of the segment $P_1P_2$, and $C$ be the circle centered at $P_1$ with radius $P_1P_4$. Let $P_5$ be the intersection of $L$ and $C$. The thesis is $P_1 = P_2$ or equivalently $u = 0$. This conjecture is true in real geometry, but it is false in the complex plane because $u = \sqrt{-1}$ is a solution. This is an interesting example because the thesis does not depend on the construction at all. It is an indirect way of asserting the validity of the construction steps. In implementing a prover that takes inputs from the user, we need to guard against being asked to prove such theorems. This amounts to an extreme form of the second kind of degeneracy.

**Timing.** The following table lists some theorems from Chou [5]. However, the last row (Tri-Bisector theorem) is the real geometry example from Section 2. The timings are for two values of $c$ (this means the probability of error is at most $2^{-c}$). We also arbitrarily "perturb" the hypothesis of each theorem by randomly changing one coefficient of one of the input polynomials, and report their timings as well. These are all false theorems, naturally. Our tests were performed on a Sun UltraSPARC-IIi (440 MHz, 512 MB). The times are all in seconds, and represent the average of 6 runs each. The prover uses Core Library, Version 1.3. Actually, the library is directly modified so that we compute the exact rational degrees of the expressions (rather than use the estimates of the Lemma 6). For comparison, we include the timings reported by Chou [5] using the approaches of Wu and of Gröbner Bases. The final column in the table gives the page number in Chou's book [5].

| No. | Theorem | $c = 10$ | $c = 20$ | Perturbed | Char Set | Gröbner | Page |
|---|---|---|---|---|---|---|---|
| 1 | Pappus | 0.020 | 0.020 | 0.007 | 1.52 | 33.32 | 100 |
| 2 | Pappus Point | 0.110 | 0.113 | 0.023 | 4.87 | 67.62 | 100 |
| 3 | Pappus-dual | 0.020 | 0.020 | 0.013 | 1.45 | 25.53 | 111 |
| 4 | Nehring | 8.300 | 8.390 | 0.107 | 4.15 | 159.3 | 115 |
| 5 | Chou-46 | 0.070 | 0.073 | 0.020 | 88.13 | 37.65 | 124 |
| 6 | Ceva | 0.030 | 0.033 | 0.017 | 1.12 | 3.47 | 264 |
| 7 | Simson | 193.22 | 262.49 | 0.023 | 1.22 | 5.02 | 240 |
| 8 | Pascal | 1715.8 | 2991.6 | 0.037 | 29.6 | >14400 | 103 |
| 9 | Tri-Bisector | 20.027 | 38.350 | 0.010 | – | – | – |

Let $r$ be the number of square roots in the radical expression representing a theorem. If $r = 0$, we say the theorem is linear. A large part[2] of the 512 theorems in Chou's book are linear. Only the last two theorems (Simson and Pascal) in the above list are non-linear, with $r = 1$ and $r = 5$, respectively. Evidently non-linear theorems represent a challenge for our current system. Recall that there are $2^r$ (or $2^{r-1}$ by symmetry) possible sign assignments to the radicals in $G(\mathbf{u})$.

[2] The theorems in Chou's book include an original list of 366 theorems from [4], of which 219 are reported to be linear [5, p. 12].

Our prover has three verification modes: (1) random mode, (2) exhaustive mode, and (3) specified mode. These correspond, respectively, to testing (1) a random sign assignment, (2) all sign assignments and (3) a user-specified assignment. For linear theorems, these modes are irrelevant. In the above table, we test Simson's theorem in the exhaustive mode, Pascal's theorem in the random mode and Tri-bisector in the specified mode. So our timing for Pascal's theorem should really be multiplied by $2^4 = 16$.

It is interesting to note that we have never observed a single wrong conclusion from our probabilistic tests – all true theorems are reported as true, and all perturbed theorems are reported as false. In some sense, that is not surprising because the probabilistic bounds based on Schwartz's lemma seem overly conservative in all real situations.

The running times for linear theorems are pretty consistent across different runs. However, for the non-linear theorems, the timing can show much more variation. This is not unexpected since the running time depends on the bit size of the random example. A more prominent behavior comes from the clustering of times around certain values. For instance, for Simson ($c = 20$), the times cluster around 10 seconds and around 70 seconds. This "multimodal" behavior of the timings are again seen in Pascal. This can be attributed to the random choice of signs for the radicals in non-linear theorems. This may also account for the curious relative times for Simson $c = 10$ and $c = 20$.

The performance of our library is critically dependent of good root bounds (an area of research that we are actively working on [16]). It should be possible to exploit prover-specific techniques to improve the speed, but this has not been done. There are several issues to bear in mind when comparing our method with Wu's method:

- Chou's timings would look considerably better using hardware available today.
- The actual theorems proved by Wu's method are not strictly comparable to ours in two important aspects: Wu's method proves theorems about complex geometry while ours is about real geometry. On the other hand, Chou's algorithm is deterministic while ours is probabilistic.
- Our method is extremely effective for discarding wrong or perturbed conjectures. It is unclear if Wu's method will be much faster for perturbed theorems, since the algorithm would still have to execute the same basic steps. The ability to quickly reject false theorems is extremely useful in applications where the user has many conjectures to check but most of the conjectures are likely to be false.
- One of the strengths of Wu's methods (as compared to Gröbner bases, say) is its ability to discover non-degeneracy conditions. A similar capability is embedded in our approach – this simply amounts to detecting when a construction step is ill-defined.

## 5 Final Remarks

In this paper, we have developed a generalization of the Schwartz-Zippel randomized zero test for the class of radical expressions. Such a test is expected to have many applications as radical expressions are quite common. Here, we focus on their use in proving theorems about ruler & compass constructions. Some features of our prover are:

- It proves theorems about real (rather than complex) geometry, under the limitation that there is no inequalities appearing in the thesis.
- It is probabilistic, so that speed can be traded-off against error probability.
- It detects wrong conjectures very quickly.
- It is extremely effective for linear theorems (the majority of the theorems in [5]).
- It exploits the special nature of ruler & compass constructions.

Because of the last feature, our approach may ultimately prove to be more efficient for *this* class of problems than other more general techniques. However, our results so far have not been conclusive in the case of non-linear theorems. The following are some open problems:

- Improve our zero test for straight line programs that involve division.
- Develop techniques to make our approach faster for non-linear theorems.
- Extend our randomized techniques to theorems that have inequalities in the theses. This seems to call for radically new ideas.

### Acknowledgments

Daniela Tulone would like to acknowledge Alfredo Ferro and Pina Carrà for their advice and discussions in the initial stages of this project.

## References

1. G. Carrà Ferro and G. Gallo. A procedure to prove geometrical statements. In L. Huguet and A. Poli, editors, *Proc. 5th Int. Conf. on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 356 of *LNCS*, pages 141–150. Springer, Berlin, 1989.
2. G. Carrà Ferro, G. Gallo, and R. Gennaro. Probabilistic verification of elementary geometry statements. In D. Wang, editor, *Proc. Int. Workshop on Automated Deduction in Geometry* (ADG-96), volume 1360 of *LNAI*, pages 87–101. Springer, Berlin, 1997.
3. S.-C. Chou. Proving elementary geometry theorems using Wu's algorithm. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years*, volume 29 of *Contemporary Mathematics*, pages 243–286. American Mathematical Society, Providence, Rhode Island, 1984.
4. S.-C. Chou. Proving geometry theorems using Wu's method: A collection of geometry theorems proved mechanically. Technical Report 50, Institute for Computing Science, University of Texas, Austin, July 1986.

5. S.-C. Chou. *Mechanical Geomtry Theorem Proving.* D. Reidel Publishing Company, 1988.
6. P. Conti and C. Traverso. Proving real geometry theorems and the computation of the real radical. In J. Richter-Gebert and D. Wang, editors, *Proc. 3rd Int. Workshop on Automated Deduction in Geometry* (ADG 2000), pages 109–120. Zurich, Switzerland, Sept. 2000.
7. A. Ferro and G. Gallo. Automatic theorem proving in elementary geometry. *Le Matematiche*, XLIII(fasc. I):195–224, 1988.
8. G. Gallo. *La Dimostrazione Automatica in Geometria e Questioni di Complessitá Correlate.* Tesi di dottorato, University of Catania, Italy, 1989.
9. G. Gallo and B. Mishra. Efficient algorithms and bounds for Wu-Ritt characteristic sets. In F. Mora and C. Traverso, editors, *Effective Methods in Algebraic Geometry* (Proc. MEGA'90), volume 94 of *Progress in Mathematics*, pages 119–142. Birkhäuser, Boston, 1991.
10. G. Gallo and B. Mishra. Wu-Ritt characteristic sets and their complexity. In *Computational Geometry: Papers from the DIMACS Special Year*, volume 6, pages 111–136. AMS and ACM, New York, 1991.
11. J.-W. Hong. Proving by example and gap theorem. In *Proc. 27th Annual Symposium on Foundations of Computer Science*, pages 107–116. IEEE, 1986.
12. D. Kapur. Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2:399–412, 1986.
13. V. Karamcheti, C. Li, I. Pechtchanski, and C. K. Yap. A core library for robust numeric and geometric computation. In *Proc. 15th ACM Symp. on Computational Geometry*, pages 351–359. ACM Press, New York, 1999.
14. B. Kutzler and S. Stifter. Automated geometry theorem proving using Buchberger's algorithm. In *Proc. Symp. on Symbolic and Algebraic Computation*, pages 209–214. ACM Press, New York, 1986.
15. C. Li. *Exact Geometric Computation: Theory and Applications.* PhD thesis, Courant Institute of Mathematical Sciences, New York University, Jan. 2001. URL: http://www.cs.nyu.edu/csweb/Research/theses.html.
16. C. Li and C. K. Yap. A new constructive root bound for algebraic expressions. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms* (SODA 2001), pages 496–505. ACM and SIAM, 2001.
17. E. W. Mayr and A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46:305–329, 1982.
18. K. Mehlhorn and S. Schirra. A generalized and improved constructive separation bound for real algebraic expressions. Technical Report MPI-I-2000-004, Max-Planck-Institut für Informatik, Nov. 2000.
19. K. Ouchi. Real/Expr: Implementation of an exact computation package. Master thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, Jan. 1997.
20. J. T. Schwartz. Probabilistic verification of polynomial identities. *J. ACM*, 27(4): 701–717, 1980.
21. W.-T. Wu. On decision problem and the mechanization of theorem proving in elementary geometry. *Scientia Sinica*, 21:157–179, 1978.
22. W.-T. Wu. Some recent advances in mechanical theorem proving of geometries. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years*, volume 29 of *Contemporary Mathematics*, pages 235–242. American Mathematical Society, Providence, Rhode Island, 1984.

23. W.-T. Wu. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning*, 2(4):221–252, 1986.
24. C. K. Yap. A new lower bound construction for commutative Thue systems, with applications. *Journal of Symbolic Computation*, 12:1–28, 1991.
25. C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 35, pages 653–668. CRC Press LLC, 1997.
26. C. K. Yap. Towards exact geometric computation. *Computational Geometry: Theory and Applications*, 7:3–23, 1997. Invited talk, Proc. 5th Canadian Conference on Computational Geometry, Waterloo, Aug. 5–9, 1993.
27. R. Zippel. *Effective Polynomial Computation.* Kluwer Academic Publishers, 1993.

# Algebraic and Semialgebraic Proofs: Methods and Paradoxes*

Pasqualina Conti and Carlo Traverso

Dipartimento di Matematica
Via Buonarroti 2
I-56127 Pisa, Italy
{conti,traverso}@dm.unipi.it

**Abstract.** The aim of the present paper is the following:
– Examine critically some features of the usual algebraic proof protocols, in particular the "test phase" that checks if a theorem is "true" or not, depending on the existence of a non-degenerate component on which it is true; this form of "truth" leads to paradoxes, that are analyzed both for real and complex theorems.
– Generalize these proof tools to theorems on the real field; the generalization relies on the construction of the real radical, and allows to consider inequalities in the statements.
– Describe a tool that can be used to transform an algebraic proof valid for the complex field into a proof valid for the real field.
– Describe a protocol, valid for both complex and real theorems, in which a statement is supplemented by an example; this protocol allows us to avoid most of the paradoxes.

## 1 Introduction

The methods of algebraic proof in Euclidean geometry, notably the Wu-Ritt characteristic set method [30] and Kapur's Gröbner basis method [19,20], have been a striking success, as for example testified by [8], in which hundreds of theorems in Euclidean geometry are proved. See also [9,21,22].

These methods start from an informally and incompletely specified theorem (i.e. a theorem in which some implicit assumptions — e.g. the three vertices of a triangle are not aligned — are made) and, through a series of steps, obtain a formal statement of a theorem and a proof (or a disproof) of it.

Some of the steps of an algebraic proof method can be fully automated through completely defined sub-algorithms, and some instead require human expertise, without excluding that an automatic equivalent may be obtained through expert systems.

To specify a method of algebraic proof one has to specify the different steps, the algorithmic steps and the "expertise" steps, the languages in which the different steps are expressed, etc; we call such a specification a *protocol.*

* Work partly performed within the Project "Tecniche per Immagini", cluster C15, progetto n. 7 "Sviluppo, Analisi ed Implementazione di Metodi Matematici Avanzati per il Trattamento di Immagini", with the contribution of MURST.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 83–103, 2001.

A common feature of many such protocols is the following: the theorem hypothesis and thesis are translated into systems of equations between the coordinates of the geometric entities involved (usually points, since all entities may be reduced to a set of defining points); these equations define two affine algebraic varieties, the *hypothesis variety* $\mathcal{H}$ and the *thesis variety* $\mathcal{T}$; some of the components of the hypothesis variety are considered degenerate; a theorem is considered true if a non-degenerate component (or every non-degenerate component, depending on the protocol) of the hypothesis variety is contained in the thesis variety. The theorem is hence completed by adding to the hypothesis some non-degeneracy condition (usually in the form of an inequation) that allows to discard the components on which the thesis is not true.

One of the most used proof protocols requires us to identify a construction; this is both a serious limitation, since many interesting theorems cannot be formalized in this way, and human expertise is necessary to identify the construction steps: this is especially well explained in [8].

The algebraic proof protocols in use have often two serious drawbacks:

- it is possible to prove "intuitively false" theorems, i.e. theorems that are true only on a component whose points are all intuitively degenerate, and false on components whose points are intuitively non-degenerate;
- the proofs are valid on an algebraically closed field (and not on the real field).

For some classes of theorems the proof of a theorem on the real ground field, when the theorem is true on the complex field, is straightforward, but the problem of the identification of such classes, to our knowledge, has not been fully addressed. Moreover there are two classes of theorems on the reals for which the standard methods of algebraic proof are useless:

- theorems that are false on the complex field;
- theorems for which the statement is impossible over the complex field, since the statement uses the ordering in the real field.

For these classes of theorems other protocols have been developed, for example real quantifier elimination [11,18,14]; these methods, however, are often much slower than the purely algebraic methods that use Wu-Ritt characteristic sets or Gröbner bases.

The aim of the present paper is the following:

- Examine critically some features of the usual algebraic proof protocols, in particular the "test phase" that checks if a theorem is "true" or not, depending on the existence of a non-degenerate component on which it is true; this form of "truth" leads to paradoxes, that are analyzed both for real and complex theorems.
- Generalize these proof tools to theorems on the real field; the generalization relies on the construction of the real radical, both of ideals and of *semi-ideals*,

the generalization to systems of equations and inequalities to the concept of ideal of a polynomial ring for a system of equations.
- Describe a tool that can be used to transform an algebraic proof valid for the complex field into a proof valid for the real field. This tool may fail, without giving a proof that the theorem is false on the real field.
- Describe a protocol, valid for both complex and real theorems, in which a statement is supplemented by an example, and allows us to avoid most of the paradoxes. This protocol can prove a true theorem, but may fail to disprove a false theorem.
- Give some hints of a more difficult problem concerning real theorems that are algebraically false but for which a semialgebraic completion and a corresponding proof might be automatically determined.

We do not discuss the choice of an algorithm for the steps of the protocols described; hence, for example, we do not take a position between different characteristic set or Gröbner basis methods, as far as they are usable to perform some commutative algebra operation such as discovering the existence of a component of an affine algebraic variety with some property needed by a protocol. The only exception is a sketch of an algorithm for the computation of the real radical, since practically efficient algorithms for this computation are not present in the literature.

Many of the items discussed in the paper are not new; many issues are already discussed in [8,30] as well as in more recent literature; our contribution proposes a different accent that might help expanding the automatic deduction methods into some of the less explored regions of algebraic and semialgebraic theorems, notably the theorems that do not correspond to a construction. In particular, we are as much interested in disproving false "theorems" (and avoiding to "prove" false "theorems") as in proving true theorems; this is especially important if one wants to establish a protocol that can mechanically produce conjectures and prove or disprove them without further human intervention.

## 2 Geometric Theorems and Algebraic Proofs

### 2.1 Proof Protocols

An algebraic proof protocol consists in a series of steps that, starting from an informally stated theorem, produce a formal theorem with a proof or a disproof, completing correctly the hypothesis with additional conditions that were implicit or overlooked in the initial statement. Some protocols moreover start from the theorem hypothesis only, and try to discover some possible thesis that, together with some additional conditions as above, produces a true (and non-trivial) theorem with a proof.

The aim of an automatic proof protocol is to have as much as possible of these steps to be carried out without human intervention. This might be especially difficult for the translation from the initial informal statement expressed in natural language into a formalized statement; for this an expert system might be conceived, but seems presently beyond reach. Recognizing and re-translating into geometric form the additional conditions needed is also difficult, especially if an automatic translation is required, but can sometimes be done,

A proof protocol may consist, for example, in the following phases:

1. **Formalization**: give a formalized expression of the hypothesis and the thesis as conjunctions of relations between geometric entities (points, lines, conics, etc.).
2. **Formalization through points**: translate into a form involving only points, (with the possible addition of auxiliary points).
3. **Algebraic translation**: translate into a form involving polynomial equations between the coordinates of the points.
4. **Test phase**: prove that the theorem can be completed with algebraic side conditions in the form of inequations yielding a true theorem.
5. **Algebraic completion**: find inequations yielding a true theorem.
6. **Geometric completion**: find geometric conditions (*side conditions*) to add to the hypothesis to ensure the truth of the theorem.
7. **Formal proof**: this can be either an algebraic proof (algebraic manipulation of the coordinates) or an human-readable geometric proof, or something in between.

As an example of an application of such a protocol, consider the theorem:

**Theorem 1.** *The bisectors of the angles of a triangle meet in the center of the in-circle.*

This is first translated into a statement like:

**Theorem 2.** *Let $A, B, C$ be three points, let $AB, BC, CA$ be lines connecting them, let a (resp. $b, c$) be a line through $A$ (resp. $B, C$) and a point equidistant from $AB$ and $CA$ (resp. $AB$ and $BC$, $BC$ and $CA$); let $D$ be a point on a and b; then $D$ lies on c and the circle with center $D$ and tangent to $AB$ is tangent to $BC$ and $CA$.*

Remark here that we have avoided to quote *the* line, *the* point, etc. to avoid implicit uniqueness assumptions: the language in which we have to specify these translations does not allow this form of statements.

The theorem as expressed now is false for several reasons: it does not rule out degeneracies (like the three points $A, B, C$ being aligned) but is also false if two bisectors are internal and one is external: this is due to a first translation that is not clever (we cannot expect an automatic translation to be clever; a clever translation requires a clear view of the geometric situation, and might be available only to somebody — man or machine — that knows the proof).

The theorem is now translated, eliminating any reference to lines and circles, into a theorem on points only of the type:

**Theorem 3.** *Let $A, B, C$ be points, $P_1$ (resp. $P_2, P_3$) be a point equidistant from the lines through $A, B$ and $A, C$ (resp. $A, B$ and $B, C$, $A, C$ and $B, C$); then $P_1, P_2, P_3$ coincide, call it $D$; let moreover $Q_1$ (resp. $Q_2, Q_3$) be a point on a line* *passing through $D$ and perpendicular to a line passing through $A, B$ (resp. $A, C$, $B, C$). Then the distances $\overline{DQ_1}$, $\overline{DQ_2}$, $\overline{DQ_3}$ coincide.*

(This theorem too, being a translation of the previous theorem, is of course logically false.)

The theorem hypothesis and thesis are now translated into sets of equations between the coordinates of the points $A, B, C, P_1, P_2, P_3, D, Q_1, Q_2, Q_3$, and one has to prove that a component of the hypothesis variety defined by the ideal of the set of hypothesis equations is contained in the thesis variety. This is the test phase.

We have then to find algebraic side conditions, and to discover that the initial translation was incomplete: one has to avoid that $A, B, C$ are aligned and that $c$ is perpendicular to the line through $C$ and $P_1$; hence the correct statement is

**Theorem 4.** *Let $A, B, C$ be three points, not aligned, let $AB, BC, CA$ be lines connecting them, let $a$ (resp. $b$) be lines through $A$ (resp. $B$) and a point equidistant from $AB$ and $CA$ (resp. $AB, BC$); let $a$ and $b$ meet in a point $D$; then $D$ is equidistant from $AC, BC$, and the circle with center $D$ and tangent to $AB$ is tangent to $BC$ and $CA$.*

This is, however, not yet an accurate translation of the initial statement, since a conditions implicit in it (that the bisectors are internal to the triangle) has not been taken into account; this hypothesis however cannot be formalized algebraically, since it needs an ordering on the ground field, and the four tritangent circles are in the same algebraic component anyway. We have proved a slightly less general theorem than that we intended. In other cases, however, we may end with a completely different theorem than the one that was initially stated; sometimes this might end up with a proof of the truth of a theorem that in its initial statement is intuitively false (we'll give an example later).

The test phase is the one that is usually considered the phase providing the proof of the truth of the theorem. Several paradoxes arise however from considering true a theorem that has passed the test phase. The test phase usually consists in proving that the affine algebraic variety defined by the polynomials in which the hypothesis has been translated (the *hypothesis variety*) has a *non-degenerate* (algebraic) component on which the theorem is true (an alternative being to require that the theorem is true on every non-degenerate variety); the definition of non-degeneracy is a part of the protocol.

We will not formalize further the notion of proof protocol; and we will concentrate mainly on the test phase.

## 2.2 Configuration Equational Theorems

We consider a restricted class of theorems in Euclidean geometry that we call *configuration equational theorems.* These theorems have the form:

$$\mathbf{h}_1 \wedge \mathbf{h}_2 \wedge \ldots \wedge \mathbf{h}_n \Rightarrow \mathbf{k}_1 \wedge \mathbf{k}_2 \wedge \ldots \wedge \mathbf{k}_m$$

where $\mathbf{h}_i, \mathbf{k}_j$ are algebraic relations, i.e. relations between geometric entities (points, lines, circles, ...) that can be translated into the vanishing of one or more polynomials in the corresponding coordinates.

It is usual to represent all the relations as relations between points. The hypothesis and the thesis of a theorem of this kind are hence the conjunction of equations between the coordinates of a set of indeterminate points. In our examples we will use only relations of this type, but it is possible to handle other geometric entities through their coordinates; for example, handle lines in space through their Grassman coordinates. In this case of course the Grassman equations have to be considered.

Some of the relations considered between points (the ones that we use in our examples) are the following (we use the notation :≡ to denote that the right hand side is a definition of the left hand side, and the coordinates of a point denoted by an uppercase letter $X$ are denoted by the corresponding lowercase letters $(x_1, x_2)$):

- $X, Y, Z$ are aligned:

$$col(X,Y,Z) :\equiv \det\begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ 1 & 1 & 1 \end{pmatrix} = 0$$

- $Z$ is the midpoint of $(X, Y)$:

$$Z = mid(X,Y) :\equiv (x_1 + y_1 = 2z_1, x_2 + y_2 = 2z_2)$$

- $(X, Y)$ and $(Z, T)$ are equidistant:

$$\overline{XY} = \overline{ZT} :\equiv (x_1 - y_1)^2 + (x_2 - y_2)^2 = (z_1 - t_1)^2 + (z_2 - t_2)^2$$

- The lines through $(X, Y)$ and $(Z, T)$ are parallel:

$$XY//ZT :\equiv \det\begin{pmatrix} x_1 - y_1 & z_1 - t_1 \\ x_2 - y_2 & z_2 - t_2 \end{pmatrix} = 0$$

As a first definition of degeneracy, we consider a component of the hypothesis variety non-degenerate if it is contained in the thesis variety; hence the test phase has to test that a component of the hypothesis variety exists that is contained in the thesis variety. An algebraic proof performs computations in the ideals defined by the hypothesis and thesis conditions to prove this fact. A stricter definition of non-degeneracy is considered in the *construction protocol*, that will be defined later.

In the paper [5], in these proceedings, five different “levels of truth” are considered; this classification can be seen as a combination of two criteria: considering a theorem true if it is verified on every (on at least one) non-degenerate component; and defining every component non-degenerate, or use the definition of non-degeneracy of the construction protocol. His fifth level of truth (*rarely true*, when the set of points where the theorem is true is not empty) does not match our definition of truth, and every theorem that can be translated into homogeneous algebraic equations in the coordinates is at least rarely true (being verified at least when all the coordinates coincide).

After the test phase, one has to identify (algebraic) additional conditions (in the form of inequations, $\phi(X) \neq 0$, where$\phi$ is a polynomial in the coordinates) that ensure that the thesis is verified; these are called *side conditions*, and usually one aims at a geometric description of these conditions, e.g. in the form of a statement of the form "$P_i, P_j, P_k$ are not aligned".

To obtain a degeneracy condition, it is sufficient to find a polynomial vanishing on those components that are not contained in the thesis locus. This polynomial is of course not unique, and expressing it in an optimal way is the key of the success of the further steps of the protocol. Hence the logically true theorem that is obtained at the end is not unique.

## 2.3 Construction Protocols

A *construction* in dimension $d$ is given by

- a sequence $\mathcal{P}$ of indeterminate points $(P_1, \ldots, P_n)$, each one being a $d$-tuple of indeterminate coordinates;
- for every $i$, $1 \leq i \leq n$, a set $\Lambda_{i,1}, \Lambda_{i,m_i}$ of geometrical conditions on the points $P_1, \ldots, P_i$ with the restriction that for every $i$, the corresponding polynomials in the coordinates of the $P_i$ are algebraically independent on the field generated by the coordinates of $P_1, \ldots, P_{i-1}$.

The case $d = 2$ is usual, but everything can be carried over unchanged to any $d$.

The construction protocol requires us to express the hypothesis of a theorem in the form of a construction, and gives a corresponding definition of non-degeneracy.

If $\mathcal{P} = (P_1, \ldots, P_n)$ is a construction, let $\mathcal{P}_i = (P_1, \ldots, P_i)$ be a sub-construction; let $\mathcal{H}$ be the hypothesis variety, and $\mathcal{H}_i$ the corresponding hypothesis varieties for the sub-constructions $\mathcal{P}_i$; a component $\bar{\mathcal{H}}$ of $\mathcal{H}$ is non-degenerate if for every $i$ the natural projection $\bar{\mathcal{H}} \to \mathcal{H}_i$ induced by the natural projection $(P_1, \ldots, P_n) \to (P_1, \ldots, P_i)$ is generically surjective on a component of $\mathcal{H}_i$.

The construction protocol is often specified asking us to identify a set of *free variables*; our definition is more intrinsic and invariant under change of coordinates; moreover one can include a rule to choose the free variables once the coordinates are fixed — for example choose the first such coordinates that are algebraically independent. This reduces the part on which human expertise is needed to a more geometric part, leaving the algebraic specification to the automatic phase. When free variables are chosen, one can define non-degeneracy by a projection on the free variables (testing that the variables are parameters in a component); this is clearly equivalent to our formulation.

A construction theorem that is true on a non-degenerate component is often called *generically true*, on the ground that the construction is possible on the generic point of the parameter space; in our opinion the term is misleading, since it cannot be applied outside of construction theorems, when there is no a-priori definition of the parameters (not even of the number of parameters, since the intuitively non-degenerate components may have different dimensions). And the non-degeneracy of a component may depend on the choice of the construction (we will show examples of both later).

Transforming an informally stated theorem into a construction theorem is often hard, and sometimes impossible; in any case, human expertise is needed; see, for example, Wu's ingenious specification for the Morley trisector theorem ([8], p. 39; we remark, moreover, that the construction given there does not match the definition of construction given in the same book); we quote moreover from [8], p. 21 (immediately after having shown how to prove a false theorem):

> "The experienced reader might immediately find that the second construction of the configuration is absurd.
> Our prover requires the user's responsibility not to introduce "absurd" constructions similar to above. Though for the above absurd construction and more hidden ones, some heuristics are built into our prover to detect them and give the user warnings, we do not have a general definition of all "absurd" constructions. We require the user to take full responsibility.
> To specify construction sequence of a geometric statement of constructive type is actually to specify the exact meaning of the statement. Thus the above requirement of the user's responsibility by our prover is reasonable, *because the user should understand the meaning of the geometry statement before starting to prove it*" (our emphasis).

We will show later that transforming an hypothesis into a construction might sometimes hide useful theorems.

## 3 Paradoxes

> Paradox: a statement or proposition seemingly self-contradictory or absurd, and yet explicable as expressing a truth. [29]

For example, Zeno's paradox shows that if space is continuous, time should be continuous too; and Russel's paradox explains that extensional definitions of sets should be bounded.

The definition of "truth" of a geometric theorem (truth on a component) may have paradoxical consequences: it may happen that

1. $p \Rightarrow q$ is true, $p \Rightarrow r$ is true but $p \Rightarrow q \wedge r$ is false;
2. $p \Rightarrow q$ is true, $q \Rightarrow r$ is true but $p \Rightarrow r$ is false;
3. $p \Rightarrow q$ is true but $p \wedge r \Rightarrow q$ is false.

The possibility of such paradoxes is obvious; for example in the first paradox it may happen that $p$ is true on a component on which $q$ is true and $r$ is false and conversely, but $q$ and $r$ may be generically incompatible (compatible on a proper subvariety of the hypothesis variety). Nevertheless, this possibility is often overlooked: for example, some protocols consider only theorems whose thesis can be expressed by a unique equation; but this is not always the case: for example, to express a thesis that a point is the midpoint of two other points, more than one condition is required, and testing the conditions separately is insufficient in view of the first paradox.

The paradoxes as usual vanish with a proper formalization. When we state that "$p \Rightarrow q$ is true" our notation is ambiguous; if "is true" means "can be derived by a system of axioms" then we can use the *modus ponens* inference rule; if "is true" means "passes an algebraic test (on a component)" we cannot. The paradoxes are ruled out if one defines truth as truth on every non-degenerate component, but this throws away the possibility of proving several interesting theorems, like the above quoted Morley trisector theorem.

A simple example of the third paradox is the following: given 4 points in the plane, $P_1, P_2, P_3, P_4$, that constitute a parallelogram, (i.e. $(P_1P_2//P_3P_4) \wedge (P_1P_3//P_2P_4)$) then the two diagonals meet in a point that is the midpoint of both diagonals. If we add the condition that $P_1, P_2, P_3$ are aligned, the theorem becomes false, since the diagonals coincide, and a point where they meet can be any point on them. This example however does not correspond to a construction.

For a construction theorem we require that the thesis is verified for a non-degenerate component. Hence a theorem may be true, but not true when considered as a construction theorem; and the truth of a construction theorem may depend on the ordering of the set of points.

Consider the construction $(P_1, P_2, P_3, P_4)$, with the conditions

- $col(P_1, P_2, P_3)$, $\overline{P_1P_2} = \overline{P_1P_3}$;
- $P_4 = mid(P_2, P_3)$.

Both $P_4 = P_1$ and $P_4 = P_2$ are true, but $(P_4 = P_1) \wedge (P_4 = P_2)$ is false. This happens whenever there is more than one component that is non-degenerate with respect to a construction.

Consider now $(P_1, P_3, P_2, P_4)$ with the same conditions; this is still a construction, but now the component $P_1 = P_3$ is degenerate; hence the theorem $P_4 = P_1$ is false. Hence the truth of a construction theorem may depend on the ordering of the points.

Consider a continuation of the construction with $P_5, P_6, P_7$, and the conditions:

- $\overline{P_1P_5} = \overline{P_4P_5}$
- $\overline{P_1P_6} = \overline{P_4P_6}$
- $\overline{P_1P_7} = \overline{P_4P_7}$

We have now two components, one with $P_1 = P_4$ and the triangle $P_5, P_6, P_7$ non-degenerate, and the other with $P_2 = P_4$ and the triangle $P_5, P_6, P_7$ degenerate; both components are hence degenerate or non-degenerate depending on the ordering. Moreover the component with $P_1 = P_4$ (degenerate if the constructions starts with $P_1, P_3, P_2$) supports the following useful theorem:

**Theorem 5.** *Let $P_5, P_6, P_7$ be points in the plane, and let $P_1$ be another point. If $P_4$ is a point in the plane having from $P_5, P_6, P_7$ the same distances as $P_1$ then $P_4 = P_1$ (unless $P_5, P_6, P_7$ are aligned).*

A further simple reasoning gives for this construction $(P_1, \dots, P_7)$ a paradox of the second type: let $q$ be the hypothesis given by the construction, let $p$ be $q \wedge (P_1 = P_4)$, and let $r$ be $col(P_5, P_6, P_7)$; then $p \Rightarrow q$ and $q \Rightarrow r$ are true, but $p \Rightarrow r$ is false.

### 3.1 Real Theorems vs. Complex Theorems

There are several differences between theorems in complex Euclidean geometry and theorems in real Euclidean geometry. Some are evident:

- theorems in real Euclidean geometry that use inequalities do not make sense on the complex field;
- existence theorems true on the complex field might be false on the reals: e.g., "given three segments, a triangle exists having sides of the same lengths as the three segments".

It is less evident that there are purely equational theorems such that the truth on the real field does not imply the truth on the complex field, and conversely. Hence a purely algebraic proof can neither prove nor disprove a real theorem, unless we have additional results.

In particular, the existence of purely equational theorems true over the complex field and false over the real field is a paradox: the hypothesis does not have quantifiers, all the variables are free, and adding the reality condition is hence simply adding an hypothesis — a paradox of the type $p \Rightarrow q$ is true but $p \wedge r \Rightarrow q$ is false, where $r$ is the hypothesis that the free variables take real values.

### 3.2 A Simple Example

We give simple examples of theorems true on the real field and false on the complex field and conversely.

Consider the following construction of 5 points:

1. $P_3$ is the midpoint of $P_1$ and $P_2$;
2. $P_4$ is the midpoint of $P_1$ and $P_3$;
3. $P_5$ is equidistant from $P_1$ and $P_2$, and has from them distance equal to the distance of $P_1$ from $P_4$.

The thesis that $P_1 = P_2$ is true on the real field (and false on the complex field, it is true only on a degenerate component) since if $P_1$ and $P_5$ are different, then taking $P_1, P_5$ as unit distance then $P_1, P_2, P_5$ constitute a triangle with sides of lengths 4,1,1, and this is impossible on the real field (but possible on the complex field).

Continue the construction above with one more points as follows:

1. $P_6$ is equidistant from $P_1$ and $P_2$.

The thesis that $P_3, P_5, P_6$ are aligned is true on the complex field: $P_3, P_5, P_6$ lie on the axis of $P_1, P_2$; but it is false on the real field since the only possibility is that $P_1$ and $P_2$ coincide, hence the two last conditions are empty.

The hypothesis variety on the reals has only one component, that corresponds to a degenerate component of the complex variety.

The complex construction, while on the reals, should not be considered a construction, since conditions involving $P_5$ imply a condition on $P_1, P_2$; this implication is however not equationally algebraic, depending on an inequality $(1 + 1 < 4)$; but a possible definition of real construction built on this consideration is quite difficult to verify.

### 3.3 A Weird Example

We now consider a configuration that is not a construction, and that shows that the decision on degenerate configurations is much harder: depending on the thesis one would choose a different component as non-degenerate. The configuration also shows an essential difference between the real and complex cases.

This theorem only requires collinearity and parallelism, hence it is an affine configuration; it is also possible to build a more complex configuration with the same properties that only uses collinearity, i.e. with a projective configuration.

Consider 8 points $P_1, \ldots, P_8$ with the following condition (called $8_3$):

$$P_i, P_{i+1}, P_{i+3} \text{ are collinear (where } P_{i+8} = P_i).$$

The condition $8_3$ is not a construction, and no construction can contain an $8_3$ condition, since every point is involved in three alignment conditions.

A theorem of Maclane states that if the eight points are real and distinct, then they are collinear.

On the complex field, the set of configurations has 10 components:

- A component $H_{10}$ of dimension 10, composed of 8 aligned points.
- A component $H_9$ of dimension 8, in which the points $P_1, P_3, P_5, P_7$ are arbitrary, and for every generic choice of them one has two possible configurations for $P_2, P_4, P_6, P_8$; if $P_1, P_3, P_5, P_7$ are real, the two choices of $P_2, P_4, P_6, P_8$ are complex conjugate; the real points of $H_9$ constitute 8 components $H_{9,i}$ of codimension 7, composed of configurations in which 5 points coincide.
- Eight components $H_i$, $1 \leq i \leq 8$ of dimension 8, $H_i$ is composed of configurations in which $P_{i+1} = P_{i+3}$, $P_{i+2} = P_{i+7}$, $P_{i+5} = P_{i+6}$, $P_{i+1}, P_{i+2}, P_{i+4}$ and $P_{i+5}$ are aligned, $P_i$ is arbitrary.

See e.g. [12,10]. In any case for a real configuration the points $P_1, P_3, P_5, P_7$ never constitute a proper parallelogram.

The theorem stating that $P_1, \ldots, P_8$ are collinear is true on the reals (the exceptional locus consists of the components $H_i, H_{9,i}$, $1 \leq i \leq 8$ in which the 8 points are not distinct).

The theorem is intuitively false on the complex field, since the component $H_9$ cannot be considered "degenerate". But if we adhere to our definition of (algebraic) truth, the theorem should be considered true: an example of an obviously false theorem that is true.

Consider now points $P_1, \dots P_9$ such that $P_1, \dots P_8$ satisfy the $8_3$ condition, $P_1, P_3, P_5, P_7$ build a parallelogram, and $P_9$ is collinear with $P_1, P_5$ and $P_3, P_7$. The only possible real configuration has 9 aligned points (this can be checked considering all the possible configurations of the real theorem, see [12]).

The thesis that $P_9$ is the midpoint of $P_1$ and $P_5$ is true on the complex field and false on the real field: the complex component that corresponds to a proper parallelogram has no real point (more precisely, its real points are contained in another component that is not contained in the thesis locus).

# 4 Partial Conclusions

Here are some of the partial conclusions that we want to draw at this point:

- Testing a theorem is only a part of the way to the statement of a true theorem; the theorem that will be eventually proved may be unexpectedly different from the initial informal statement: we might end proving something that is so different from the intuitive meaning of the starting formulation that what we prove, although formally true, could be described as a "false theorem".
- Constructions are adequate sometimes, but not always; they are no guarantee that a real theorem can be derived from a complex theorem.
- The result of the testing of a theorem depends from the protocol; not every protocol is suitable for every theorem.

# 5 Proving Real Theorems: The Real Radical

We have seen that testing a theorem is insufficient to determine a logically true theorem; in the rest of the paper however we will concentrate on the analysis of some methods of algebraic proof, and on the extension of these methods to the real ground field (or more generically to ordered fields).

## 5.1 Algebraic Proof Tools

We have seen that in an algebraic proof we have to show that a component of the hypothesis variety is contained in the thesis variety.

Basic tools to identify components are transporter and saturation (see e.g. [16,6]).

Let $I$, $J$ be ideals of a noetherian ring $R$; the *transporter* $I : J$ and the *saturation* $I :^* J$ are defined as follows: $I : J = \{a \in R \mid aJ \subseteq I\}$; $I :^* J$ denotes the (constant) ideal $I : J^n$ for $n$ sufficiently large. See [7] for an account on algorithms for the computation of $I : J$ and $I :^* J$.

Algebraically, $I : J$ enlarges the primary components corresponding to associated primes containing $J$, and $I :^* J$ deletes such components; if the ground field is algebraically closed, and $R$ is a polynomial ring, the associated components of the radical correspond to the irreducible components of the corresponding variety, hence computing $I : J$ if $I$ is radical, or computing $I :^* J$ in general corresponds to killing the components contained in the locus of $J$.

To prove that a theorem is true, let $H$ be the hypothesis ideal, and $T$ the thesis ideal; one of the two following computations is sufficient.

- Let $H'$ be the radical of $H$. Prove that $H' : T \neq H'$.
- Prove that $H :^* (H :^* T) \neq (1)$.

Indeed, let $\mathcal{H}$ be the locus of $H$ and $\mathcal{T}$ be the locus of $T$; the locus of $H' : T$ is equal to the closure of $\mathcal{H} \setminus \mathcal{T}$, and the locus of $H :^* (H :^* T)$ is the closure of $\mathcal{H} \setminus (\mathcal{H} \setminus \mathcal{T})$, hence coincides with the union of the components of $\mathcal{H}$ contained in $\mathcal{T}$.

The first computation is much harder, but can be generalized more easily.

When we are over the real field, we no longer have the correspondence between associated prime ideals of the radical and irreducible components, hence the methods are not sufficient.

The first method however can be straightforwardly generalized to the real case taking the real radical instead of the radical: the problem is that the computation of the real radical is much harder and less known. Eventually, in Section 5.5 we will generalize the second method too.

Let $K$ be an ordered field, and $R$ its real closure. The real radical of an ideal $I \subseteq K[X]$ is the largest ideal $\sqrt{I}$ that has the same zero set as $I$ in $R$.

With this definition, the first criterion above can be generalized to the real field:

- Let $H'$ be the real radical of $H$. Prove that $H' : T \neq H'$.

## 5.2 Semi-ideals, Theorems with Inequalities

The notion of real radical can be generalized to semi-ideals, that generalize the concept of ideal taking into account inequalities.

A *sign condition* is a predicate $p\#0$ where $p$ is a polynomial and # is one of $>, \geq, <, \leq, =, \neq$. A sign condition is*strict* if # is $>, <$ or $\neq$.

A semi-ideal is $(I, \sigma_1 \dots, \sigma_n)$, where $I$ is an ideal and the $\sigma_i$ are sign conditions. A semi-ideal defines a locus (a semialgebraic set) and one defines its radical as the largest ideal $J$ such that $(J, \sigma_1 \dots, \sigma_n)$ has the same locus.

The considerations of semi-ideals allows us to introduce inequalities in the hypothesis of theorems; we do not investigate here the straightforward generalization (with some extra difficulties one can introduce inequalities in the thesis too).

Semi-ideals are also needed in the algorithms for the real radical, even in the case that that we only need to compute the real radical of an ideal.

### 5.3 Computation of the Real Radical: An Outline

The real radical can be defined non-constructively using the theorem [4]:

**Theorem 6 (Real Nullstellensatz).** *Let $I$ be an ideal of $\mathbf{R}[X]$, and $f \in \mathbf{R}[X]$ vanishing identically on the real locus of $I$; then $\exists m, n \in \mathbf{N}$, $f_1, \ldots, f_m \in \mathbf{R}[X]$ such that $f^{2n} + \sum_i f_i^2 \in I$.*

The converse implication is of course true. This theorem, however, is non-constructive, as opposed to the Hilbert Nullstellensatz, which implicitly gives an algorithm to test radical membership; the algorithms for the computation of the radical cannot be mimicked to construct the real radical or to test real radical membership. One of the difficulties is the following: while the radical commutes with separable extensions (if $J = \sqrt{I} \subseteq K[X]$ and $K'$ is a separable extension of $K$ then $J \otimes K' = \sqrt{I \otimes K'}$) this is not true for the real radical: the real radical of $(x^3 - 2)$ is $(x^3 - 2)$ or $x - \sqrt[3]{2}$ depending on the existence of $\sqrt[3]{2} \in K$.

Algorithms for the computation of the real radical have been discovered only recently; most of the algorithms in the literature are theoretical, and unfeasible even for very simple examples, see [3], and [15,26]; the main concern of these paper is just to prove complexity bounds. See however also [2], that describes a feasible algorithm for a special class of ideals.

An algorithm that is more attentive to practical complexity than to theoretical complexity is described in [13]; see also [27,1,28] for an important sub-algorithm with a slightly different approach. We outline the algorithm of [13], which, following the classical algorithms of Morse theory [23], tries to minimize the use of operations that, while innocent-looking from the theoretical complexity point of view, may make unfeasible a computation that would be otherwise feasible.

The computation of the real radical is much harder than the computation of the radical; as opposed to the computation of the radical, which requires an equidimensional decomposition, it requires an irreducible decomposition; the key ingredient is the following theorem:

**Theorem 7.** *Let $V$ be a complex algebraic variety, defined and irreducible over the real field. If $V$ has a real point $P$ that is non-singular as a complex point then the set of real points is Zariski dense in $V$, hence the defining ideal of $V$ coincides with its real radical.*

The proof of the theorem is simple and classical: for the implicit function theorem the fixed locus of the complex involution at the point $P \in V$ has real dimension equal to the complex dimension of $V$; hence the set of real points is Zariski-dense.

A common feature of most algorithms for the real radical is the following: decompose a semialgebraic set into subsets for which the defining ideal is easy to compute; then take the intersection of these ideals.

The basic idea of our algorithm is the following: decompose $V$ into components $V_i$ irreducible over the ground field, (more explicitly, find a prime decomposition of the defining ideal) and let $S_i$ be the corresponding singular locus (algebraically defined: the locus where the rank of the Jacobin ideal of a set of generators is not maximal); if $V_i \setminus S_i$ is empty, replace $V_i$ with an irreducible decomposition of $S_i$. At the end we get a decomposition of the real locus of $V$ into irreducible algebraic varieties whose defining ideal coincides with its real radical. The real radical of the original ideal is the intersection of these radicals.

To prove emptiness of $V_i \setminus S_i$ it is sufficient to prove the emptiness of $V_i \setminus S'_i$, where $S'_i$ is any subvariety of $V$ of codimension 1 containing $S_i$, since if the real locus is contained in a 1-codimensional subset then by Theorem 7 it is contained in the singular locus.

$V_i \setminus S'_i$ is defined by a semi-ideal. The proof thus reduces to a test of emptiness for semi-ideals; this can be split in two parts:

1. Prove that the locus of the semi-ideal is compact;
2. Prove that a function has no critical point (no point where the differential vanishes).

Since the locus of the critical points is recursively of smaller dimension (unless the variety has a very special position, to be handled separately) the emptiness can be handled recursively, reducing to dimension 0, where one can use methods for real root counting, see e.g. [24,17].

The function can be any polynomial; we take a coordinate function, that has the advantage of being linear, hence does not increase the complexity of the calculations; if the semi-ideal is an ideal (no inequalities), a more classical alternative (see [27,1] and also [23]) takes a non-negative function (a sum of squares) thus avoiding the need of proving compactness. With classical methods (Rabinowitz trick) one can transform inequalities into equalities, but at the cost of adding a variable and a non-linear equation, hence threatening feasibility.

To prove the compactness of a semialgebraic set $S$ defined by a semi-ideal $\mathcal{I} = (I, \sigma_1 \ldots, \sigma_n)$, one can prove that it is closed in the projective space, and for this it is sufficient to prove that it is closed in every affine chart.

Given a semi-ideal $\mathcal{I} = (I, \sigma_1 \ldots, \sigma_n)$, define $\mathcal{I}_j = (I, \sigma_1 \ldots, \sigma_n, \tilde{\sigma}_j)$, where $\tilde{\sigma}_j$ is defined as follows: let $\epsilon$ be a positive infinitesimal, i.e. a new indeterminate added to the ground ordered field $K$, positive and smaller than any positive element of $K$; then:

- if $\sigma_j$ is $g_j > 0$ then $\tilde{\sigma}_j$ is $g_j = \epsilon$;
- if $\sigma_j$ is $g_j < 0$ then $\tilde{\sigma}_j$ is $g_j = -\epsilon$;
- if $\sigma_j$ is $g_j \neq 0$ then$\tilde{\sigma_j}$ is $g_j = \pm\epsilon$;
- in every other case $\tilde{\sigma}_j$ is $1 = 0$, i.e. $\mathcal{I}_j$ has empty locus.

One proves that if every $\mathcal{I}_j$ defines an empty semialgebraic set, then $S$ is closed, and if $S$ is compact then every $\mathcal{I}_j$ has empty locus. Hence closedness is recursively reduced to emptiness. This proof is the key step of the theorem; since it mixes topological arguments with non-Archimedean arguments (use of the infinitesimals), the Tarski-Seidenberg quantifier elimination theorem is largely used; but this is only for the proof, no quantifier elimination is needed for the algorithm, hence the efficiency is not impaired.

Two remarks on combinatorial complexity:

- Every $g_j \neq 0$ gives rise to a $g_j = \pm\epsilon$, that can be reduced either to $g_j^2 = \epsilon^2$ or to two disjoint cases $g_j = \epsilon$ and $g_j = -\epsilon$. The best way is the second, although it appears of exponential complexity: the two cases $g_j = \epsilon$ and $g_j = -\epsilon$ are algebraically identical, the difference being only in the real root counting phase, that can anyway be handled simultaneously for all the signs of the added $\epsilon$. Adding a quadratic equation instead doubles the degree, increasing the complexity of all the subsequent steps.
- There is an element of exponential growth in the recursive handling of the affine charts; however the computations are anyway exponential in the number of variables in average case complexity, and even doubly exponential in worst case complexity, and the recursion reduces the number of variables cutting with a hyperplane, hence the most time-consuming parts of the algorithm are the top-dimensional ones.

Moreover, in general, for evaluating the practical performance of an algorithm requiring computations with polynomial ideals, one has to remember that the limiting factor is usually space, not time, and computing a very large number of small problems may succeed, while a single slightly larger problem may fail for space problems.

A full description of the algorithm is in [13].

### 5.4 A Simpler Algorithm for the Proof of Real Theorems

To (algebraically) prove a theorem it is not necessary to compute the real radical: we do not need to identify all the components of the real locus, we only need to prove that a component is contained in the locus of the thesis. As soon as this is achieved, we can exit the algorithm.

It is hence possible to avoid some of the more difficult points of the algorithm, in particular to avoid finding an irreducible decomposition (an equidimensional decomposition is sufficient, and is anyway needed for the computation of the radical). And we do not need to investigate smaller dimensional components when we have identified one component with smooth real points.

The algorithm strictly follows the real radical algorithm, using unchanged some of its sub-algorithms, and simplifying it at some points. We leave the details to a further paper.

### 5.5 A Criterion to Pass from a Complex Proof to a Real Proof

From the algorithms for the computation of the real radical, in particular Theorem 7, it is clear that the following criterion holds:

**Theorem 8.** *Let $\mathcal{H}$ be the hypothesis variety, and $\mathcal{T}$ the thesis variety of an algebraically true complex theorem. Let $\hat{\mathcal{H}}$ be the Zariski closure of $\mathcal{H} \setminus \mathcal{T}$. Assume that $P$ is a smooth real point of the Zariski closure of $\mathcal{H} \setminus \hat{\mathcal{H}}$. Then the theorem is algebraically true on the real field.*

Indeed $P$ is contained in a component of $\mathcal{H}$ that is contained in $\mathcal{T}$; this component, having a smooth real point, is a component of the real radical. Hence the real theorem is algebraically true.

We remark that it is not sufficient to prove that $P$ is smooth in $\mathcal{H}$ and contained in $\mathcal{T}$, since it may happen that $P$ is a smooth point of a component on which the thesis is not generically true; nor it is sufficient to find a point of a component of the complex variety where the theorem is true, since it may happen that this complex component has a real locus fully contained in the intersection with another component having real points on which the theorem is false.

A simple example for both remarks is the following: let $H = (xz, y^2z + z^3) = (x, y^2 + z^2) \cap (z)$, $T = (x)$. We have $\sqrt[R]{(x, y^2 + z^2)} = (x, y, z)$, $\sqrt[R]{(z)} = (z)$ hence $\sqrt[R]{H} = (z)$, hence the complex theorem is true and the real theorem is false; but $(0, 1, 0)$ is a smooth point where the thesis is true, and $(0, 0, 0)$ is a real point of $(x, y^2 + z^2)$ that is the component of the hypothesis variety where the theorem is true.

To check the hypothesis of Theorem 8 one has to find a defining ideal of $\mathcal{H} \setminus \hat{\mathcal{H}}$; one such ideal in the algebraically closed case is $H :^* (H :^* T)$. This proof criterion can hence be seen as a generalization of the second proof criterion in Section 5.1.

Usually, it is fairly easy to provide a real point $P \in \mathcal{H}$ if the theorem is true over the real field: it is just an example of the theorem. Checking the smoothness is just a rank computation of the Jacobian matrix at the chosen point. We have moreover to check that $P$ is a point of a complex component contained in $\mathcal{T}$; this corresponds to checking that the maximal ideal corresponding to $P$ contains the ideal $H :^* (H :^* T)$.

If in every component on which the thesis holds we do not have a smooth real point, the truth of the theorem on the real field remains undecided: it may happen that a real component of $I$ contained in the singular locus is contained in the thesis locus. The real theorem will be true if and only if this component is not contained in a larger real component of $H$. Here are two examples:

- for the configuration of 3.2, all the real configurations are contained in the singular locus (since otherwise they would be dense, but we have seen that for them we have $P_1 = P_2$, that is not verified for the generic complex point). The thesis that $P_1, P_2, P_3$ are aligned is true without exceptions;
- the configuration of 3.3 is an example of the other possibility, since the real part of the component of the complex variety satisfying the thesis is contained in another component, composed of 9 aligned points, for which the thesis is not verified.

To decide the issue when a smooth point is not available (either it does not exist or we have been unable to find it) we can proceed with the following (probabilistic) algorithm:

- compute $I = H :^* (H :^* T)$ (that is an ideal defining $\mathcal{H} \setminus \hat{\mathcal{H}}$);

- find $g \in H :^* T$, $g \notin H$, i.e. a polynomial vanishing on $\hat{\mathcal{H}}$ but not identically vanishing on $\mathcal{H}$;
- consider the semi-ideal $\mathcal{I} = (H, g \;\; /\!\!= 0)$ and prove that the locus of$\mathcal{I}$ is non-empty.

This proves the existence of a real point $P$ not contained in any component where the theorem is generically false; hence in every component containing $P$ the theorem is true; but this neither provides an example nor proves that an algebraically smooth example exists; it is however sufficient to test the theorem.

The algorithm can be made deterministic by repeating it for $g$ in a set $\{g_i\}$ of generators of $H :^* T$. In that case if every $(H, g_i \;\; /\!\!= 0)$ is empty we have a proof of the falseness of the theorem.

To summarize, the test method outlined in this section can be viewed as a generalization of the second criterion in Section 5.1: after proving that the ideal $I = H :^* (H :^* T)$ is not trivial, to test the theorem on the real field one can either prove that a smooth real point exists, or prove that a real point that is not a point of $H :^* T$ exists. In the test of a complex theorem, the second part is unnecessary because of Hilbert's Nullstellensatz.

## 6 A Protocol Proposal: Theorems with Examples

The real proof criterion of Theorem 8 suggests a slightly different specification for configurational theorems; consider an ideal $H$ generated by a set of configurational geometric equations, and let $\mathcal{P}$ be a configuration satisfying the conditions (i.e. in the locus $\mathcal{H}$ of $H$), an *example*. If a point $\mathcal{P}$ is smooth, (i.e. the localization of $\mathbf{C}[X]/H$ at $\mathcal{P}$ is a regular local ring) then a unique component of the hypothesis variety contains $\mathcal{P}$, and this component is declared non-degenerate. More generally, we may give a set $\mathcal{P}_i$ of examples, and consider non-degenerate any component containing one of the smooth examples. The test points may also be given approximately, provided that they are far from the singular locus and near to an unique component: this can be tested numerically.

We remark that giving a smooth point allows us to find a set of parameters: a choice of coordinates exists such that the projection at the smooth point is a surjective diffeomorphism, and we can take these coordinates as independent variables, the other being dependent (on the component under consideration). This generalizes the usual construction protocol, but only on an a posteriori basis: the existence of the independent variables is a conclusion, not an hypothesis; and if several test points are given, the set of independent variables may vary for different components (they may even be of different cardinality — the $8_3$ configuration is an example).

Theorems in which a unique smooth example of the hypothesis is given are unambiguous, and all the paradoxes discussed above are solved. The praxis of exemplifying theorems with a drawing is moreover current in Euclidean geometry; a drawing is a smooth example, since (and if…) it is approximate: the points can be slightly moved provided that the geometric relations indicated by the lines and circles included in the drawing are maintained; this movement realizes the representation of a Euclidean neighborhood of the example, showing the smoothness. We remark that this protocol is already implicitly implemented in the system Cinderella [25]: when you draw a construction this corresponds to an example; the deductions made by the system correspond to finding a thesis that is valid on the component of the construction defined by the example, since the movements used to find the random examples on which the hypothesis is tested avoid the singular locus, hence remain on the same algebraic component.

### 6.1 The Real Case: Further Extensions

In a theorem with example, if the complex theorem is true, and the example point is real, the real theorem is true too.

In the real case it may be possible, following these ideas, to develop proof protocols for theorem with examples that require inequalities in the side conditions (hence are algebraically false).

Consider for example the following theorem:

**Theorem 9.** *In a triangle, the radius of the in-circle is not greater than half the radius of the circumcircle.*

On the complex field the theorem cannot be stated algebraically since it involves inequalities; if one allows complex conjugation the theorem can be stated, but is false since one cannot tell the difference between the four circles that are tangent to the three sides of a triangle (it is impossible to distinguish between the inside and the outside of the triangle, hence from in-circle and ex-circle).

The theorem is however true on the reals. In that case, the interior of the triangle can be identified (it is the set of points where the functions defining the lines of each side have the same sign as the opposite vertex). Hence the inscribed circle is identified as the circle tangent to the three sides that has the center inside the triangle.

In the set of (non-degenerate) configurations of the variety of tritangent circles, the inscribed circles are not an algebraic component; but indeed they are a connected component: the degenerate triangles constitute a subvariety of the non-degenerate component, and are non-smooth points (belonging to two different components, the non-degenerate and the degenerate). The set of configurations corresponding to inscribed circles is a connected component of the set of smooth points (to pass continuously from an in-circle to an ex-circle one has to pass through a degenerate triangle; and conversely one can pass continuously from one non-degenerate triangle to another). An algebraic proof (indeed, a semialgebraic proof. . . ) is possible: if one considers the semialgebraic set of smooth points of the configuration, the in-circle and the ex-circles lie in different connected components, since the degenerate triangles constitute a different algebraic component, (one for which infinitely many circles tangent to the three coinciding sides exist), and one can pass continuously from an in-circle to an ex-circle only through a degenerate triangle or a complex triangle. The example indicates the correct connected component, that is defined from the properties that a linear function vanishing on any two vertices has the same sign on the third vertex and on the center of the circle (i.e. the center of the circle is internal to the triangle).

Consider the following “theorem”:

**Theorem 10.** *In a triangle, the radius of one of the ex-circles is smaller than the radius of the circumcircle.*

An automatic semialgebraic proof correctly and optimally completing the hypothesis with a set of inequalities from a true example remains a challenge (the theorem is true e.g. if one angle is obtuse, for the ex-circle opposed to the smaller angle, and is false in a neighborhood of a regular triangle).

### Acknowledgments

We want to thank Fabrizio Broglia for useful discussions and references on the semialgebraic geometry issues discussed in this paper.

We thank the anonymous referees for many useful remarks.

## References

1. P. Aubry, F. Rouillier, M. Safey El Din, *Real solving for positive dimensional systems*, Report LIP6, `http://www.lip6.fr/reports/lip6.2000.009.html` (2000).
2. E. Becker, R. Grobe, M. Niermann, *Real zeros and real radicals of binomial ideals*, J. Pure Appl. Algebra 117 & 118, 41–75 (1997).
3. E. Becker, R. Neuhaus, *Computation of real radicals of polynomial ideals*, Computational Algebraic Geometry, Progress in Math. 109, 1–20, Birkhäuser, Boston (1993).
4. J. Bochnak, M. Coste, M.-F. Roy, *Géométrie Algébrique Réelle*, Erg. der Mathematik 12, Springer-Verlag, Berlin Heidelberg (1987).
5. M. Bulmer, D. Fearnley-Sander, T. Stokes, *The kinds of truth of geometry theorems*, Automated Deduction in Geometry (J. Richter-Gebert, D. Wang, eds.), LNAI 2061, 129–142, Springer-Verlag, Berlin Heidelberg (2001).
6. M. Caboara, P. Conti, C. Traverso, *Yet another ideal decomposition algorithm*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes — AAECC-12 (T. Mora, H. F. Mattson, eds.), LNCS 1255, 39–54, Springer-Verlag, Berlin Heidelberg (1997).
7. M. Caboara, C. Traverso, *Efficient algorithms for ideal operations*, Proc. ISSAC 98, 147–152, ACM Press, New York (1998).
8. S.-C. Chou, *Mechanical Geometry Theorem Proving*, D. Reidel Pub. C., Dordrecht (1988).
9. S.-C. Chou, X.-S. Gao, *Ritt-Wu’s decomposition algorithm and geometry theorem proving*, 10th International Conference on Automated Deduction (M. E. Stickel, ed.), LNCS 449, 207–220, Springer-Verlag, Berlin Heidelberg (1990).
10. S.-C. Chou, X.-S. Gao, N. McPhee, *A combination of Ritt-Wu’s method and Collins’ method*, TR-89-28, CS Department, The Univ. of Texas at Austin, USA (1989).

11. G. Collins, *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, Autom. Theor. Form. Lang., 2nd GI Conf., LNCS 33, 134–183, Springer-Verlag, Berlin Heidelberg (1975).
12. P. Conti, C. Traverso, *A case study of semiautomatic proving: The Maclane* $8_3$ *theorem*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes — AAECC-11, (G. Cohen, M. Giusti, T. Mora, eds.), LNCS 948, 183–193, Springer-Verlag, Berlin Heidelberg (1995).
13. P. Conti, C. Traverso, *Algorithms for the real radical*, Technical Report, `http://www.dm.unipi.it/~traverso/Papers/RealRadical.ps` (1998).
14. A. Dolzmann, T. Sturm, V. Weispfenning, *A new approach for automatic theorem proving in real geometry*, J. Automat. Reason. 21 (3), 357–380 (1998).
15. A. Galligo, N. Vorobjov, *Complexity of finding irreducible components of a semialgebraic set*, J. Complexity 11, 174–193 (1995).
16. P. Gianni, B. Trager, G. Zacharias, *Gröbner bases and primary decomposition of polynomial ideals*, J. Symb. Comput. 6 (2–3), 149–167 (1988).
17. L. Gonzalez-Vega, F. Rouillier, M.-F. Roy, *Symbolic recipes for real polynomial system solving*, Some Tapas of Computer Algebra (A. M. Cohen, et al., eds.), Algorithms Comput. Math. 4, 121–167, Springer-Verlag, Berlin Heidelberg (1999).
18. A. Guergueb, J. Mainguené, M. F. Roy, *Examples of automatic theorem proving in real geometry*, Proc. ISSAC 94, 20–23, ACM Press, New York (1994).
19. D. Kapur, *Using Gröbner bases to reason about geometry problems*, J. Symb. Comput. 2, 399–408 (1986).
20. D. Kapur, *A refutational approach to geometry theorem proving*, Artif. Intell. 37, 61–93 (1988).
21. B. Kutzler, *Algebraic approaches to automated geometry theorem proving*, Ph.D thesis, RISC-Linz, Johannes Kepler Univ., Austria (1988).
22. B. Kutzler, S. Stifter, *Collection of computerized proofs of geometry theorems*, Tech. Rep. 86-12, RISC-Linz, Johannes Kepler Univ., Austria (1986).
23. J. Milnor, *Morse Theory*, Annals of Mathematics Studies 51, Princeton University Press, Princeton (1963).
24. P. Pedersen, M.-F. Roy, A. Szpirglas, *Counting real zeros in the multivariate case*, Computational Algebraic Geometry (F. Eyssette, A. Galligo, eds.), 203–223, Birkhäuser, Boston (1993).
25. J. Richter-Gebert, U. Kortenkamp, *The Interactive Geometry Software Cinderella*, Springer-Verlag, Berlin Heidelberg (1999).
26. M.-F. Roy, N. Vorobjov, *Computing the complexification of semialgebraic sets*, Proc. ISSAC 96, 26–34, ACM Press, New York (1996).
27. F. Rouillier, M.-F. Roy, M. Safey El Din, *Testing emptiness of real hypersurfaces, real algebraic sets and semi-algebraic sets*, FRISCO Technical Report (1998).
28. F. Rouillier, M. Safey El Din, E. Schost, *Solving the Birkhoff interpolation problem via the critical point method: An experimental study*, Automated Deduction in Geometry (J. Richter-Gebert, D. Wang, eds.), LNAI 2061, 26–40, Springer-Verlag, Berlin Heidelberg (2001).
29. *New Webster's Dictionary of the English Language*, The English Language Institute of America (1971).
30. W.-t. Wu, *Mechanical Theorem Proving in Geometries: Basic Principles* (translated from the Chinese by X. Jin and D. Wang), Springer-Verlag, Wien New York (1994).

# Remarks on Geometric Theorem Proving

Laura Bazzotti, Giorgio Dalzotto, and Lorenzo Robbiano

Dipartimento di Matematica, Via Dodecaneso 35, 16146 Genova, Italy
{bazzotti,dalzotto}@poly.dima.unige.it, robbiano@dima.unige.it
http://cocoa.dima.unige.it

> *The* intelligent *reader clearly understands*
> *that the work is* artificial *in its essence.*
> (Anonymous)

**Abstract.** The mathematical literature related to automatic methods for proving theorems in Euclidean geometry is immense. However, it is the opinion of the authors that the theory behind this topic would profit from more algebraic tools and more methods from commutative algebra. The scope of this paper is to begin to fill such a gap. In particular we bring to the forefront important notions such as *computing field*, *optimal hypothesis ideal*, and *good set of conditions*.

## 1 Introduction

As is well explained in the survey paper [6], the *dream* of deducing all the theorems of a specific sector of mathematics from a small set of axioms has fascinated many mathematicians not least of which David Hilbert. The dream did not come true due to the fundamental work of Kurt Gödel, from which it became clear that it is not possible to construct a machine which proves *all possible theorems*. However, the door was left open to the possibility of using a machine to prove *some theorems*.

The idea of constructing such a machine was revived by the appearance of electronic computers, and the goal of proving theorems automatically soon became a central part of that academic discipline, with the awe-inspiring name of Artificial Intelligence.

Already in the early sixties, Gelernter, Hansen and Loveland in [5] introduced a "Geometry Machine" with the purpose of proving theorems in Euclidean geometry. A major breakthrough in the subject was the work of Wen-tsün Wu (see for instance [15–17]) and his coworkers of the Chinese school. An impressive amount of material on the subject was collected by Shang-Ching Chou in the book [3].

Another source of inspiration was the advent of Gröbner basis theory, due to Bruno Buchberger (see for instance [1, 2]), and which inspired many fundamental contributions, such as those of Deepak Kapur (see [7] and the important paper [8]), Bernhard Kutzler and Sabine Stifter [10], and more recently Tomas Recio with his coworkers [12, 11] and Dongming Wang (see for instance [14]).

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 104–128, 2001.

In this brief overview of the history we have clearly skipped over many important paths and omitted many authors. Fortunately, a huge list of papers on the subject is contained in the web site managed by Dongming Wang at

`http://calfor.lip6.fr/~wang/GRBib/Welcome.html`

This reference is of fundamental importance for the adventurous reader who wants to navigate inside the huge quantity of work done on the topic of proving theorems automatically.

The purpose of this paper is to contribute to the subject in many ways. First of all, we want to look at the fundamentals of the theory on a purely algebraic basis, where ideal theory provides the necessary devices. Along the way, we introduce important notions such as *computing field*, *optimal hypothesis ideal*, and *good set of conditions*.

We also provide an implementation of our method in the system CoCoA (see [4]), which uses Gröbner bases. But, since we mainly concentrate on the representation of knowledge, the tools for proving (Gröbner bases, Wu-Ritt bases or other devices) are considered to be of lesser importance. Good sources of inspiration for us were the papers [8, 11], where an algebro-geometric approach was taken via the celebrated Nullstellensatz of Hilbert.

Before giving a more detailed description of the content of the paper at the end of the introduction, we want to supply some background to help appreciate the novelties of our approach.

We recall that the main step in proving a theorem automatically is to construct an *explicit representation* of its hypothesis and thesis. This should be done in such a way that the thesis follows from the hypothesis with the help of a set of rules, and both the representation and the rules can be manipulated by a computer.

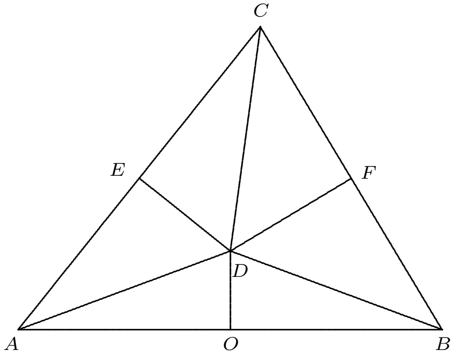
Given a theorem in Euclidean geometry, the fundamental steps usually considered are:

a) introduce Cartesian coordinates in the Euclidean plane or space;
b) translate the hypotheses and thesis into algebraic relations among the fundamental geometric data such as coordinates of points and lengths of segments;
c) assuming that these algebraic relations are expressed as the vanishing of suitable polynomials, prove the theorem by showing that the "thesis polynomial" is a consequence of the "hypothesis polynomials".

All of these steps require some clarification in order to avoid critical obstacles. But before raising the many questions related to the concept of automatically proving theorems, let us say that even the traditional methods are not totally reliable, as the following beautiful example shows. We learned of and borrowed it from the book [3].

*Example 1 (A Fake Theorem).* Every triangle is isosceles.

*Proof.* Let $ABC$ be a triangle as shown in the picture. We want to prove that $CA = CB$. Let $D$ be the intersection of the perpendicular bisector of $AB$ and the internal bisector of angle $ACB$. Let $DE \perp AC$ and $DF \perp CB$. It is easy to see that $\Delta CDE \cong \Delta CDF$ and $\Delta ADE \cong \Delta BDF$. Hence $CE + EA = CF + FB$, i.e. $CA = CB$.

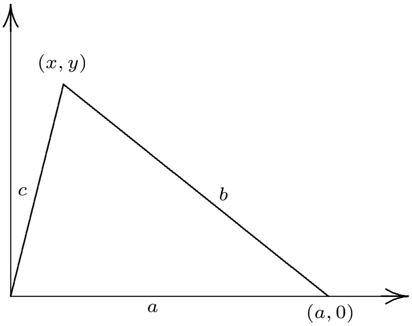


What is wrong with this proof? If you feel a little nervous about this example, or you are simply surprised by this apparent paradox, please relax and you will discover the trick. Mathematics is full of difficulties and also the validity of many theorems relies more on general consensus than on the logical correctness of the steps of a formal proof. It is not our intention to delve into these philosophical oddities, we merely want to show that the idea of automating proofs leads to a fair number of new and subtle problems.

To begin with, we say that it is practically almost impossible to automatize the entire process. For instance the choice of the coordinates and some clever simplifications of the input data are generally part of a preprocessing, which comes before the automatic part is left to the computer.

Now, to highlight some of the problems we are going to encounter, we discuss a number of examples.

*Example 2 (Heron's Formula).* In any given triangle the area $s$ is given by $s^2 = p(p-a)(p-b)(p-c)$, where $a, b, c$ are the lengths of the sides, and $p = (a+b+c)/2$.



Can we deduce it automatically? The answer is yes and the idea is the following.

We introduce an orthogonal system of coordinates, we use the possibility of freely choosing the system, and position the triangle as shown in the picture. Then we use Pythagoras' Theorem to show that $b^2 = (a-x)^2 + y^2$, $c^2 = x^2 + y^2$ and we observe that $2s = ay$.

We construct the ideal $I = (b^2 - (a-x)^2 - y^2,\ c^2 - x^2 - y^2,\ 2s - ay)$ in the polynomial ring $P = \mathbb{R}[x,y,a,b,c,s]$, we look for a polynomial relation among the indeterminates $a,b,c,s$ and we try to deduce it from the given set of relations.

One approach is to compute $I \cap \mathbb{R}[a,b,c,s]$. It works and produces a principal ideal, whose generator can be rewritten as Heron's Formula after the substitution $p = (a+b+c)/2$.

*Example 3 (Equal Cubes).* Cubes of equal volume have equal sides.

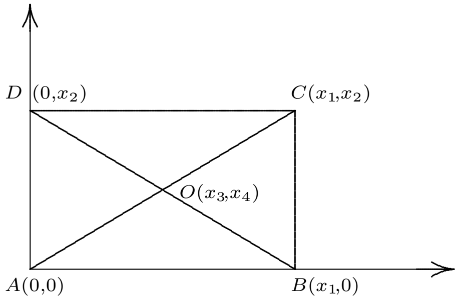
This absolutely trivial theorem is not provable by using a method independent of the base field, simply because it is false over the complex numbers. Unlike the case of Heron's Formula, where we want to deduce an algebraic relation from others, we could content ourselves to check that the *thesis polynomials* vanish on the common zeros of the *hypothesis polynomials.* A first big difficulty arises. The Nullstellensatz translates this vanishing into an algebraic verification only in the case of *algebraically closed fields.*

The thesis polynomial $x-y$ does not *follow* from the hypothesis polynomial $x^3 - y^3$, in the sense that $x - y \notin \sqrt{(x^3 - y^3)}$. Then we observe that $x^3 - y^3 = (x-y)(x^2+xy+y^2) = (x-y)(x-\alpha_1 y)(x-\alpha_2 y)$, where $\alpha_1, \alpha_2$ are the two conjugate complex roots of the univariate polynomial $x^2+x+1$.

This means that the *hypothesis ideal* has three complex components, two real components, but the *hypothesis variety* has only one real component, which is the *thesis variety.* Therefore the statement is true, and hence a theorem, over the reals but it should be considered false over the complex numbers, since, for instance

$$2^3 = (-1 + \sqrt{3}\, i)^3$$

*Example 4 (Bisecting Diagonals).* The diagonals of a rectangle cross each other at their midpoint.



In the picture the two diagonals of the rectangle $ABCD$ are $AC$ and $DB$, while $O$ is their point of intersection. We have to show that $AO = OC$ and $DO = OB$. We proceed as follows. We introduce an orthogonal system of coordinates as shown in the picture and we name the coordinates of the relevant points by putting

$$A = (0,0),\ B = (x_1,0),\ C = (x_1,x_2),\ D = (0,x_2),\ O = (x_3,x_4)$$

In this way we have already expressed that $ABCD$ is a rectangle. The only hypotheses still to be translated are that $O$ belongs to the line $AC$ and to the line $BD$ too. From elementary considerations we get

$$h_1 = x_4x_1 - x_3x_2 \quad \text{and} \qquad h_2 = x_1x_2 - x_2x_3 - x_1x_4$$

So we produce two *hypothesis polynomials*, $h_1, h_2$, whose vanishing represents the two described conditions. Analogously we produce the *thesis polynomials*, $\boldsymbol{t}_1, \boldsymbol{t}_2$, whose vanishing represents the property that $AO = OC$ and $BO = OD$. We get

$$\boldsymbol{t}_1 = x_1^2 + x_2^2 - 2x_1x_3 - 2x_2x_4 \qquad \boldsymbol{t}_2 = x_1^2 - x_2^2 - 2x_1x_3 + 2x_2x_4$$

We have already seen that the Nullstellensatz translates the vanishing into an algebraic verification only in the case of *algebraically closed fields*. However a second important difficulty shows up. No matter over which field we work, the statement is false.

Indeed, it is well known that many theorems from Euclidean geometry turn out to be false in the algebraic setting, simply because the algebraic translation of some condition can encode more cases. This certainly happens in the case of the bisecting diagonals, but we can exhibit an even simpler example.

Suppose we want to prove the following apparently trivial theorem.

Let $r$ be a straight line in the Euclidean plane, $A, B \in r$, and $C$ a point in the plane. Then $C$ is aligned with $A$, $B$, if and only if $C \in r$.

We may try to prove the theorem in the following way. We let $r$ be the $x$-axis and $A(0,0)$ the origin. Then we let $B(x,0)$ be a point on the $x$-axis and $C(a,y)$ be a point on the plane. What is the condition which expresses the alignment of $A$, $B$, $C$? A first look at the problem suggests that the answer is $y = 0$. But of course there is a more complete answer, namely $xy = 0$, which takes in consideration also the case $A = B$. Indeed in such a situation every point is obviously aligned with $A$ and $B$. If we really want to consider only the case $A \neq B$, then we need to disregard the case $x = 0$. This can be done by saturating the hypothesis ideal $I = (xy)$ with respect to $x$. The new hypothesis ideal is now $(y)$, and since the thesis polynomial is clearly $y$, the wanted conclusion follows trivially.

Looking from the point of view of algebra, the hypothesis ideal $I = (xy)$ and the hypothesis ideal $J = (y)$ are different ideals. And the consequence is dramatic. Namely the statement is true with the hypothesis ideal $J$, but false with $I$.

Many authors have already pointed out that it is not always easy to make a full a priori analysis of the degenerate cases of a geometric construction (see Example 8). But even more important is the consideration that the radical of the hypothesis ideal need not be prime. It is perfectly legitimate to consider a statement of the type: "Let $T$ be a triangle which is isosceles or right angled. Prove that ..." In this case, even if we disregard degenerate components, the hypothesis ideal has at least two prime components. Following this approach, we are naturally lead to consider statements which are algebraically true on some prime components of the radical of the hypothesis ideal (see Definition 5).

Now we have seen some of the problems, so let us have a closer look at the content of the sections.

In Section 2, following for instance the approach of [8], we restrict ourselves to proving or disproving *algebraically true statements*. In this way we can stay within the realm of commutative algebra, since the Nullstellensatz allows us to formulate the definition of algebraically true statements in terms of the radical of the hypothesis ideal (see Definition 1).

However, the ring on which we consider our data is a polynomial ring over the reals, and it is well-known that $\mathbb{R}$ is not a computable field. With the aid of the notion of field of definition we define a computing field for a given statement (see Definition 3). It will play a fundamental role in the theoretical aspect of our approach. A first result is that if $K$ is a computing field for a statement $\mathcal{T}$, then the validity of $\mathcal{T}$ can be expressed completely in a polynomial ring over $K$ (see Thereom 2).

A careful study of the hypothesis ideal in the case of the intriguing Example 4, shows that some statements are true on more than one component. In fact, the final part of Section 2 is devoted to giving a graphical representation of the *space of components* of our hypothesis. This space is closely related to what geometers call *realization spaces* (see for instance [13]).

More algebra is needed to lay down the theoretical basis, and Section 3, which is the heart of the paper, takes care of that. After recalling or proving some easy facts form ideal theory, we define the condition ideal (see Definition 6) and we prove Theorem 3 and Corollary 2, where all the facts about a statement in Euclidean geometry are expressed in purely algebraic terms. In particular we concentrate on the intrinsic notion of *optimal hypothesis ideal* (see Definition 8) and on that of *almost good* and *good set of conditions* (see Definition 11). These are central notions, and Proposition 2 describes a way of computing almost good and good sets of conditions.

Then the final section treats the algorithmic part of the paper. It states and proves the main algorithm (see Theorem 4) and describes the CoCoA code which can be freely used. With the help of CoCoA we illustrate some examples which, in different respects, illustrate some of the subtleties encountered in the paper.

## 2 Algebraically True Statements

As a first step we follow a classical approach, and define the notion of an algebraically true statement.

**Definition 1.** *Let $\mathcal{T}$ be a statement in Euclidean geometry, whose hypotheses and thesis are expressed by the vanishing of polynomials $h_1, \ldots, h_r, \boldsymbol{t}$ in $\mathbb{R}[x_1, \ldots, x_n]$. We assume that the ideal $(h_1, \ldots, h_r)$ is proper. Then it is called the* hypothesis ideal *of $\mathcal{T}$, and denoted by $\mathcal{I}(\mathcal{T})$. The polynomial $\boldsymbol{t}$ is called the* thesis polynomial. *We say that the statement is* algebraically true *if $\boldsymbol{t} \in \sqrt{\mathcal{I}(\mathcal{T})}$.*

*Likewise, we say that the statement is algebraically false if $\boldsymbol{t} \notin \sqrt{\mathcal{I}(\mathcal{T})}$. If there is no ambiguity, we simply say that a statement is true or false.*

The first simplification is that we are only going to discuss how to give automatic proofs of algebraically true statements.

The next fundamental step, which is most of the times overlooked in the literature, is to be able to do all the field computations inside a field which is *computable*. What does it mean that a field is computable? Roughly speaking, it means that it is possible to store an element of the field in finitely many memory cells of the computer, that one can check in finitely many steps whether two such representations correspond to the same field element, and there are algorithms for performing the four basic operations $+, -, \times, \div$.

The problem is that $\mathbb{R}$ is not computable. However, we may use the notion of *field of definition*, which we recall briefly (see [9] Section 2.4 for more details).

**Definition 2.** *Let $K$ be a field, $P = K[x_1, \ldots, x_n]$ a polynomial ring, and $I$ an ideal in $P$.*

*a) Let $k \subseteq K$ be a subfield. We say that $I$ is* defined over *$k$ if there exist elements in $k[x_1, \ldots, x_n]$ which generate $I$ as a $P$-module.*

*b) A subfield $k \subseteq K$ is called a* field of definition *of $I$ if $I$ is defined over $k$ and there exists no proper subfield $k' \subset k$ such that $I$ is defined over $k'$.*

**Theorem 1.** *Let $I$ be a non-zero ideal in $K[x_1, \ldots, x_n]$.*

*a) There exists a unique field of definition of $I$.*

*b) Given any term ordering $\sigma$, let $G$ be the corresponding reduced $\sigma$-Gröbner basis of $I$. Then the field of definition of $I$ is the field generated over the prime field of $K$ by the coefficients of the terms in the support of the vectors in $G$.*

*Proof.* See [9], Section 2.4.

**Corollary 1.** *Let $f_1, \ldots, f_r$ be polynomials in the ring $\mathbb{R}[x_1, \ldots, x_n]$ and let $I$ be the ideal $(f_1, \ldots, f_r)$. Then the field of definition of $I$ is a finite extension of $\mathbb{Q}$, hence it is computable.*

*Proof.* The claim follows immediately from the theorem, since the prime field of $\mathbb{R}$ is $\mathbb{Q}$, and a reduced Gröbner basis contains finitely many polynomials.

**Definition 3.** *Let be given a statement $\mathcal{T}$ in Euclidean geometry, whose hypothesis and thesis polynomials $h_1, \ldots, h_r, \boldsymbol{t}$ lie in $\mathbb{R}[x_1, \ldots, x_n]$. A field $K$ is called a* computing field *for $\mathcal{T}$, if it is finitely generated over $\mathbb{Q}$ and contains the field of definition of the ideal $(\mathcal{I}(\mathcal{T}), \boldsymbol{t}) = (h_1, \ldots, h_r, \boldsymbol{t})$. It is a computable field. The ideal generated by $(h_1, \ldots, h_r)$ in $K[x_1, \ldots, x_n]$ will be denoted by $I_K(\mathcal{T})$ or simply $I(\mathcal{T})$ or even $I$, if no confusion arises.*

**Definition 4.** *Let $R$ be a commutative ring, let $I$ and $J$ be ideals in $R$. Then the set*

$$I : J^\infty = \bigcup_{i \in \mathbb{N}} I : J^i = \{r \in R \mid J^i \cdot r \subseteq I \text{ for some } i \in \mathbb{N}\}$$

*is an ideal in $R$. It is called the* saturation *of $I$ by $J$.*

Now we need an extra piece of information. In the following, if $K \subseteq L$ is a field extension, and $I$ is an ideal in $K[x_1, \dots, x_n]$, we denote by $I^e$ the ideal generated by $I$ in $L[x_1, \dots, x_n]$.

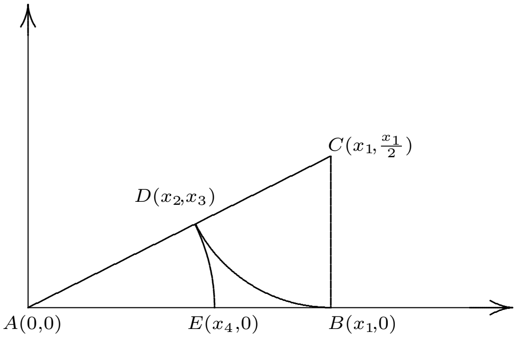
**Proposition 1.** *Let $K \subseteq L$ be a field extension, $I$, $J$ ideals in $K[x_1, \dots, x_n]$. Then:*

*a) $I^e \cap K[x_1, \dots, x_n] = I$. In particular, we have $1 \in I^e$ if and only if $1 \in I$.*
*b) For $I = \cap_{i=1}^r I_i$, we get $I^e = \cap_{i=1}^r I_i^e$.*
*c) We have $I^e : (J^e)^\infty = (I : J^\infty)^e$.*
*d) If moreover $K$ is perfect, then $\sqrt{I^e} = (\sqrt{I})^e$.*

*Proof.* Claims $a), b), c)$ are standard in commutative algebra and depend on the faithful flatness of the homomorphism $K[x_1, \dots, x_n] \hookrightarrow L[x_1, \dots, x_n]$. The proof of claim $d)$ is based on its 0-dimensional version, namely on the fact that if $I$ is a 0-dimensional radical ideal and $K$ is perfect, then $I^e$ is also a radical ideal. The proof of this fact can be found in [9], Proposition 3.7.18.

With the help the following example we illustrate the importance of the notion of the computing field.

*Example 5 (Golden Ratio, Golden Section, Golden Mean).*



Let $AB$ be a segment as shown in the picture. Let $CB$ be perpendicular to $AB$ and such that the equality $AB = 2CB$ holds true. Let $D$ be the intersection of $AC$ with the circle centered in $C$ and passing through $B$. Let $E$ be a point on $AB$ such that $AE = AD$.

We have to show that $AB$, $AE$ satisfy the *golden ratio*, i.e. $\frac{AB}{AE} = \frac{1+\sqrt{5}}{2}$.

We introduce an orthogonal system of coordinates as shown in the picture. We get

$$A = (0,0),\ B = (x_1, 0),\ C = (x_1, \frac{x_1}{2}),\ D = (x_2, x_3),\ E = (x_4, 0)$$

In this way we have already expressed that $CB$ is perpendicular to $AB$ and that $AB = 2CB$. It remains to express the facts that $D$ belongs to the line $AC$, that $CD = CB$, and that $AE = AD$. We get the following hypothesis polynomials:

$$h_1 = -\frac{1}{2}x_1x_2 + x_1x_3 \qquad h_2 = -x_1^2 + 2x_1x_2 - x_2^2 + x_1x_3 - x_3^2 \qquad h_3 = -x_2^2 - x_3^2 + x_4^2$$

The thesis polynomial is

$$\boldsymbol{t} = \sqrt{5}x_4 - 2x_1 + x_4$$

Therefore a computing field is $\mathbb{Q}(\sqrt{5})$ and we may perform all the computations in $\mathbb{Q}(\sqrt{5})[x_1, x_2, x_3, x_4]$.

An alternative way of posing the problem is to require that $AE$ is the *golden section* of $AB$, i.e. that $\frac{AB}{AE} = \frac{AE}{EB}$. The thesis polynomial is

$$\boldsymbol{t} = x_1^4 - 2x_1^3x_4 + x_1^2x_4^2 - x_4^4$$

Therefore a computing field is $\mathbb{Q}$ and we may work in $\mathbb{Q}[x_1, x_2, x_3, x_4]$. We will see later (see Section 4) that the two approaches lead to different computations.

In the following, we are only going to use the operations on ideals described above. This implies that all the theory and the computation can be carried on in the ring $P = K[x_1, \ldots, x_n]$, where $K$ is a computing field for $\mathcal{T}$. Since $K$ is computable, the procedures which we are going to develop in the paper are *true algorithms.*

For the moment, we content ourselves to get the first step toward the automatization of the proof of theorems in Euclidean geometry.

**Theorem 2.** *With the above assumptions, let $K$ be a computing field for $\mathcal{T}$. Then the following conditions are equivalent.*

*a) Statement $\mathcal{T}$ is algebraically true.*
*b) The thesis polynomial $\boldsymbol{t}$ is in $\sqrt{I_K(\mathcal{T})}$.*

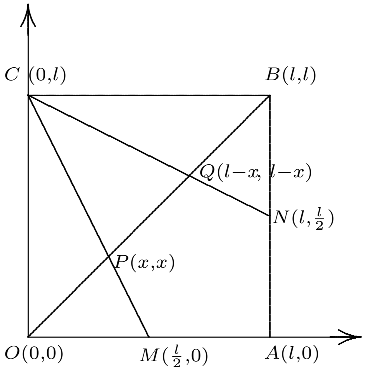
*Proof.* Let us prove the only non trivial implication, i.e. a) $\Rightarrow$ b). Let $\mathcal{T}$ be algebraically true. Then there exists $d \in \mathbb{N}$ such that $\boldsymbol{t}^d \in (h_1, \ldots, h_r)\mathbb{R}[x_1, \ldots, x_n]$. But $\boldsymbol{t}^d \in K[x_1, \ldots, x_n]$, hence $\boldsymbol{t}^d \in I_K(\mathcal{T})\mathbb{R}[x_1, \ldots, x_n] \cap K[x_1, \ldots, x_n]$, and the latter is $I_K(\mathcal{T})$ by Proposition 1.a. We have checked that $\boldsymbol{t} \in \sqrt{I_K(\mathcal{T})}$, which concludes the proof.

*After the above discussion, we assume that for every given statement $\mathcal{T}$ we work over a fixed computing field $K$ for $\mathcal{T}$, and we call $I_K(\mathcal{T})$ the hypothesis ideal of $\mathcal{T}$.*

Unfortunately also this simplification is not enough to avoid further difficulties. Let us go back to the Bisecting Diagonals statement. We saw that the hypothesis ideal is $I = (x_4x_1 - x_3x_2,\ x_1x_2 - x_2x_3 - x_1x_4)$. We take the thesis polynomials $\boldsymbol{t}_1, \boldsymbol{t}_2$ and check if they belong to $\sqrt{I}$. *The answer is no.* Should we conclude that the Bisecting Diagonals statement is not a theorem? Are we facing the same difficulty as in the example of the equals cubes?

Let us have a look at another example.

*Example 6 (Trisection of Line Segment).* The two straight lines passing through a vertex of a square and the medium points of the opposite sides cut the opposite diagonal into three equal parts.



We introduce Cartesian coordinates, and place the square as in the picture. Then we have

$$A = (l, 0),\ B = (l, l),\ C = (0, l),\ M = (l/2, 0),\ N = (l, l/2)$$

Furthermore we introduce a new indeterminate $x$, we work in the polynomial ring $\mathbb{Q}[l, x]$, and we have

$$P = (x, x),\ Q = (l - x, l - x)$$

In this way we have already expressed that $P$ and $Q$ belong to the line $OB$, and that $OP = BQ$. The only hypotheses still to be expressed are that $C$, $M$,$P$ are aligned and $Q$ belongs to the line $CN$. Both lead to the polynomial $h = l^2 - 3lx$, hence the hypothesis ideal is the principal ideal $I = (h)$. The thesis comes from expressing that $OP = PQ$, hence $\boldsymbol{t} = l^2 - 4lx + 3x^2$. A computing field is $\mathbb{Q}$. We check if $\boldsymbol{t} \in \sqrt{I}$, the answer is no, and again we must conclude that the statement is *algebraically false*.

And now we have to face the problem of discovering why this statement turns out to be algebraically false. In this case the answer is fairly easy. The ideal $I$ is a principal ideal generated by $l^2 - 3lx = l(l - 3x)$. In other words it describes two different situations. One is given by $l = 0$, a component over which the statement is algebraically false, the other one is given by $l - 3x = 0$ and the thesis polynomial $\boldsymbol{t} = l^2 - 4lx + 3x^2$ is such that $\boldsymbol{t} \in (l - 3x)$.

We conclude that the statement is algebraically true (hence a theorem) if the square is not a point. It is also interesting to observe that this case is *not a limit case of our construction, but a truly different component.* This suggests the following

**Definition 5.** *Let $\mathcal{T}$ be a statement, $K$ a computing field for $\mathcal{T}$. Let $I$ be the hypothesis ideal of $\mathcal{T}$ in $K[x_1, \ldots, x_n]$, $\boldsymbol{t}$ its thesis polynomial, and let $\mathfrak{p}_1, \ldots, \mathfrak{p}_c$ be the minimal primes of $\sqrt{I}$, so that $\mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_c$ is the primary decomposition of $\sqrt{I}$. We say that $\mathcal{T}$ is* algebraically true on $\mathfrak{p}_i$ *if $\boldsymbol{t} \in \mathfrak{p}_i$.*

Therefore a statement can be algebraically false, but true on a component. This is what happens in the case of the trisection of a line segment.

Let us consider again the case of the Bisecting Diagonals (see Example 4). In this case it is easy to see that $I = \sqrt{I} = \mathfrak{p}_1 \cap \mathfrak{p}_2 \cap \mathfrak{p}_3 \cap \mathfrak{p}_4$, where

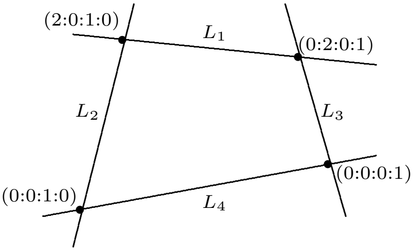
$$\mathfrak{p}_1 = (x_1 - 2x_3,\ x_2 - 2x_4) \qquad \mathfrak{p}_2 = (x_2, x_4) \qquad \mathfrak{p}_3 = (x_1, x_3) \qquad \mathfrak{p}_4 = (x_1, x_2)$$

and $\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathfrak{p}_1 \cap \mathfrak{p}_4$, while $\boldsymbol{t}_1, \boldsymbol{t}_2$ do not belong to $\mathfrak{p}_2$ and to $\mathfrak{p}_3$. So the statement is algebraically true only on $\mathfrak{p}_1$ and $\mathfrak{p}_4$. But what is the geometric interpretation of the hypotheses expressed by the generators of $\mathfrak{p}_1$ and $\mathfrak{p}_4$? For instance, it is not so clear how to read $\mathfrak{p}_1$ as a set of hypotheses. It is instead clear how to read $\mathfrak{p}_2$, $\mathfrak{p}_3$, $\mathfrak{p}_4$. The vanishing of $\mathfrak{p}_2$ represents the degeneracy of the rectangle to the line segment $AB$. The vanishing of $\mathfrak{p}_3$ represents the degeneracy of the rectangle to the line segment $AD$. The vanishing of $\mathfrak{p}_4$ represents the degeneracy of the rectangle to the point $O$.

So now it is clear that the vanishing of $\mathfrak{p}_1$ represents the same hypotheses as before (i.e. that $O$ belongs to the line $AC$ and to the line $BD$) *applied to non degenerate rectangles.* This is fine. It looks like we can find *good hypotheses* and *good conditions*, which corresponds to the obvious remark that a finite subset of a finite set can be described as the complementary set of its complementary set.

But something even more intriguing is emerging from this example. Namely the algebraic analysis leads to the conclusion that the statement is true (hence a theorem) for rectangles which do not degenerate to a line segment *plus rectangles which degenerate to a point.* Even more than in the case of the trisection of a line segment, something is really defeating the intuition, namely the fact that this strongly degenerate case is not in the limit of the other degenerate cases.

Let us explain why, with the help of the next picture, which is of a different nature with respect to the other pictures in the paper. It represents the components of the hypothesis ideal above. It is a homogeneous ideal, hence the four components $\mathfrak{p}_1 = (x_1 - 2x_3,\ x_2 - 2x_4)$, $\mathfrak{p}_2 = (x_2, x_4)$, $\mathfrak{p}_3 = (x_1, x_3)$, $\mathfrak{p}_4 = (x_1, x_2)$ represent lines in $\mathbb{P}^3$, which we call $L_1$, $L_2$, $L_3$, $L_4$ respectively.



Clearly $L_1$, $L_4$ are skew, as well as $L_2$, $L_3$. The interpretation goes as follows.

The line $L_1$ represents the component of non-degenerate rectangles. The lines $L_2$ and $L_3$ represent the components of rectangles which degenerate to the side $AB$ and $AD$ respectively. The line $L_4$ represents the component of rectangles which degenerate to the point $A$. The projective point $(2 : 0 : 1 : 0)$ represents the rectangles which degenerate to the side $AB$, while $O$ is a point of the plane (even outside $AB$), such that the ratio of the $x$-coordinates of $B$ and $O$ is 2. Analogously, the projective point $(0 : 2 : 0 : 1)$ represents the rectangles which degenerate to the side $AD$, while $O$ is a point of the plane, such that the ratio of the $x$-coordinates of $D$ and $O$ is 2. The two points $(2:0:1:0)$ and $(0:2:0:1)$ are the *true limits* of the degeneration of the rectangle to one of its sides.

On $L_2$, $L_3$ the statement is false, but then the statement is again true on $L_4$. It represents the rectangles which degenerate to the point $A$. Here $O$ can be taken arbitrarily. We said that on $L_3$, $L_4$ the statement is false, but there are exceptions represented by the projective points $(0:0:1:0)$, $(0:0:0:1)$. Let us explain the first one. It represents rectangles degenerated to the point $A$, with $O$ on the $x$-axis. Therefore it is a limit case of the rectangles represented by $L_2$. Although in general the statement is false on $L_2$, it is true on the limit, because we keep $O$ on the $x$-axis. The same explanation holds for $(0:0:0:1)$.

This example shows that to investigate the situation we need more tools. They are developed in the next section, which is the heart of the paper.

## 3 Optimal Hypothesis Ideals and Good Conditions

We have seen that in general the radical of the hypothesis ideal is not prime, so it can be expressed as the intersection of several prime ideals. A way to proceed could be the following: compute $\sqrt{I}$, then compute its prime decomposition, and finally check on which prime components $\mathcal{T}$ is algebraically true.

This approach is *impractical* for two different reasons. The first reason is the complexity of computing the prime components of the radical of $I$. But even more important is the difficulty of understanding the geometric meaning of some components. After a heavy computation we might conclude that *we have proved a theorem, but we do not know which.*

Let us go back for a moment to the Bisecting Diagonals, and suppose we decide to exclude *a priori* the degenerate cases $x_1 = 0$ and $x_2 = 0$. Then our hypothesis ideal becomes

$$I : (x_1 x_2)^\infty = (x_1 - 2x_3,\ x_2 - 2x_4)$$

on which it is immediate to verify that the statement is true. We can safely say that the statement is verified under the condition that $x_1 \not= 0$ and$x_2 \not= 0$. It means that it is verified providing the rectangle is truly a rectangle. With this condition it is a theorem.

Another possibility which may happen is that $\boldsymbol{t}$ does not belong to any of the prime components of the hypothesis ideal. Then the statement should be considered to be absolutely false.

To clarify these situations we need more technical devices from Commutative Algebra.

**Lemma 1.** *Let $R$ be a noetherian ring, $I$ an ideal in $R$, $f, f_1, \ldots, f_r \in R \setminus \{0\}$, and $J = (f_1, \ldots, f_r)$. Then*

*a)* $I : f^\infty = IR_f \cap R$
*b)* $I : J^\infty = \bigcap\limits_{i=1}^{r} I : f_i^\infty$

*Proof.* The proof of these elementary facts can be found for instance in [9], Section 3.5.

**Lemma 2.** *Let $R$ be a noetherian ring, $I$ an ideal in $R$, and $f, g \in R \setminus \{0\}$. Then the following conditions are equivalent:*

*a)* $g \in \sqrt{I : f^\infty}$
*b)* $f \in \sqrt{I : g^\infty}$
*c)* $1 \in I : (fg)^\infty$

*Proof.* Let us prove a) $\Rightarrow$ c). Using $g \in \sqrt{I : f^\infty}$ we deduce that there exists a natural number $d$ such that $g^d \in I : f^\infty$, hence there exists a natural number $e$ such that $(gf)^e \in I$, which implies that $1 \in I : (fg)^e$. Conversely, if $1 \in I : (fg)^e$, then $g^e \in I : f^\infty$, hence $g \in \sqrt{I : f^\infty}$. Condition c) is symmetric on $f$ and $g$, therefore the proof is complete.

**Lemma 3.** *Let $R$ be a noetherian ring, let $I, J$ be ideals in $R$, and let $g \in R$. Then*

*a)* $\sqrt{I : J^\infty} = \sqrt{I} : J = \sqrt{I} : \sqrt{J}$
*b)* $I : J^\infty = (1)$ *if and only if* $J \subseteq \sqrt{I}$
*c)* $I : g^\infty \subseteq \sqrt{I}$ *if and only if* $g \nmid 0 \bmod \sqrt{I}$

*Proof.* Let us prove the first equality of a). To show that $\sqrt{I : J^\infty} \subseteq \sqrt{I} : J$, we pick an element $a \in \sqrt{I : J^\infty}$. There exists a natural number $d$ such that $a^d \in I : J^\infty$, hence there exists a natural number $e$ such that $(aJ)^e \in \sqrt{I}$. Therefore $aJ \subseteq \sqrt{I}$, and we may conclude that $a \in \sqrt{I} : J$. Conversely, let $a \in \sqrt{I} : J$. Then there exists a natural number $e$ such that $(aJ)^e \in I$. Therefore $a \in \sqrt{I : J^e} \subseteq \sqrt{I : J^\infty}$.

To prove the second equality, it suffices to show that $\sqrt{I} : J \subseteq \sqrt{I} : \sqrt{J}$, since the other inclusion is obvious. For $aJ \subseteq \sqrt{I}$ there exists a natural number $d$ such that $(aJ)^d \subseteq I$. We have to show that for $b \in \sqrt{J}$, then $ab \in \sqrt{I}$. Now, for $b \in \sqrt{J}$ there exists a natural number $e$ such that $b^e \in J$, hence $(ab^e)^d \in I$, hence $ab \in \sqrt{I}$.

It is clear that b) follows from a), since $I : J^\infty = (1)$ is equivalent to $\sqrt{I : J^\infty} = (1)$.

Finally we prove c). From $I : g^\infty \subseteq \sqrt{I}$ we deduce that $\sqrt{I : g^\infty} \subseteq \sqrt{I}$, therefore $\sqrt{I} : g = \sqrt{I}$, by a). Conversely, if $g \nmid 0 \bmod \sqrt{I}$, then $\sqrt{I} : g = \sqrt{I}$, which implies $\sqrt{I : g^\infty} = \sqrt{I}$, by a). The conclusion is that $I : g^\infty \subseteq \sqrt{I}$.

**Lemma 4.** *Let $R$ be a noetherian ring, let $I$ be an ideal in $R$, and let $g \in R$. Assume that $\sqrt{I} = \mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_s \cap \mathfrak{q}_1 \cap \ldots \cap \mathfrak{q}_t$ is a minimal prime decomposition of $\sqrt{I}$, where $g \in \mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_s$, and $g \notin \mathfrak{q}_j$ for $j = 1, \ldots, t$. Then*

*a)* $\sqrt{I : g^\infty} = \mathfrak{q}_1 \cap \ldots \cap \mathfrak{q}_t$
*b)* $\sqrt{I : (I : g^\infty)^\infty} = \mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_s$

*Proof.* To show a), we recall from Lemma 3.a that $\sqrt{I : g^\infty} = \sqrt{I} : g$, and the claim follows easily. Now let us prove b). We use again Lemma 3.a to see that $\sqrt{I : (I : g^\infty)^\infty} = \sqrt{I} : \sqrt{(I : g^\infty)}$. The assumption and a) imply that we have to show the equality $(\mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_s \cap \mathfrak{q}_1 \cap \ldots \cap \mathfrak{q}_t) : (\mathfrak{q}_1 \cap \ldots \cap \mathfrak{q}_t) = \mathfrak{p}_1 \cap \ldots \cap \mathfrak{p}_s$, whose proof is immediate.

In the following we let $h_1, \ldots, h_r, \boldsymbol{t}$ be polynomials in $\mathbb{R}[x_1, \ldots, x_n]$. Then we let $\mathcal{T}$ be a statement in Euclidean geometry, whose hypothesis ideal is the ideal in $\mathbb{R}[x_1, \ldots, x_n]$ generated by $(h_1, \ldots, h_r)$, and whose thesis polynomial is $\boldsymbol{t}$. Let $K$ be a computing field for $\mathcal{T}$ and let $I(\mathcal{T})$ be the ideal generated by $h_1, \ldots, h_r$ in $P = K[x_1, \ldots, x_n]$.

Due to Proposition 1, all the operations described in the above lemmas can be considered in a polynomial ring over $K$.

To say that $\boldsymbol{t} \in \sqrt{I(\mathcal{T}) : f^\infty}$ means that $\mathcal{T}$ is algebraically true under the condition $f \neq 0$, as we can see from Lemma 1. Therefore Lemma 2 shows that $\sqrt{I(\mathcal{T}) : \boldsymbol{t}^\infty}$ is the container of all the conditions. We see the ideal $I(\mathcal{T}) : \boldsymbol{t}^\infty$ becoming an important actor. It deserves a name.

**Definition 6.** *The ideal $I(\mathcal{T}) : \boldsymbol{t}^\infty$ is called the* condition ideal *of $\mathcal{T}$. It is denoted by $I_c(\mathcal{T})$ or simply by $I_c$ if no confusion arises.*

**Definition 7.** *Let $J = (f_1, \ldots, f_s)$ be an ideal in $P$. A statement $\mathcal{T}$ is called* algebraically true under the conditions expressed by $J$ *(or* algebraically true if at least one of the conditions $f_1 \neq 0, \ldots, f_s \neq 0$ holds true*), if the revised statement, whose hypothesis ideal is $I(\mathcal{T}) : J^\infty$ and whose thesis polynomial is $\boldsymbol{t}$, is algebraically true. A statement $\mathcal{T}$ is called* absolutely false *if the thesis polynomial does not belong to any minimal prime of the hypothesis ideal.*

**Theorem 3.** *Let $\mathcal{T}$ be a statement in Euclidean geometry, whose hypothesis and thesis are expressed by the vanishing of polynomials $h_1, \ldots, h_r, \boldsymbol{t}$. Let $K$ be a computing field for $\mathcal{T}$ and let $I(\mathcal{T})$ be the proper ideal generated by $h_1, \ldots, h_r$ in $K[x_1, \ldots, x_n]$. Then the following facts hold true.*

*a) We have $\boldsymbol{t} \in \sqrt{I(\mathcal{T}) : I_c(\mathcal{T})^\infty}$.*

*b) The following are equivalent.*

  *1) The statement $\mathcal{T}$ is algebraically true.*
  *2) $I_c(\mathcal{T}) = (1)$*
  *3) $\sqrt{I(\mathcal{T}) : I_c(\mathcal{T})^\infty} = \sqrt{I(\mathcal{T})}$*

*c) The following are equivalent.*

  *1) The statement $\mathcal{T}$ is absolutely false.*
  *2) $I_c(\mathcal{T}) \subseteq \sqrt{I(\mathcal{T})}$*
  *3) $I(\mathcal{T}) : I_c(\mathcal{T})^\infty = (1)$*

*d) If the statement $\mathcal{T}$ is neither algebraically true nor absolutely false, then we have the chain of strict inclusions $\sqrt{I(\mathcal{T})} \subset \sqrt{I(\mathcal{T}) : I_c(\mathcal{T})^\infty} \subset (1)$.*

*Proof.* Claim a) is immediate from the definition. We prove Claim b). From Theorem 2 and Lemma 3.b we deduce the equivalence between 1) and 2). Since 2) $\Rightarrow$ 3) is clear, it remains to show that 3) $\Rightarrow$ 1). This implication follows directly from a).

Now we prove c). From Definition 7 and Lemma 3.c. we see the equivalence between 1) and 2). The equivalence between 2) and 3) is clear.

Finally we prove d). From 3) of b) we get the first strict inclusion, from 3) of c) the second one.

The above theorem indicates that most of the information about the validity of $\mathcal{T}$ is stored in the ideal $I(\mathcal{T}) : I_c(\mathcal{T})^\infty$. In particular we see that if $\mathcal{T}$ is neither algebraically true nor absolutely false, then what is algebraically true is the revised statement, whose hypothesis ideal is $I(\mathcal{T}) : I_c(\mathcal{T})^\infty$, and thesis polynomial is $\boldsymbol{t}$. But we can say more with the help of Lemma 4.

**Definition 8.** *An ideal which contains the hypothesis ideal $I$, and whose prime components are exactly those on which $\mathcal{T}$ is algebraically true is called an* optimal hypothesis ideal *for $\mathcal{T}$.*

**Corollary 2.** *With the same assumptions as in Theorem 3, an optimal hypothesis ideal for $\mathcal{T}$ is the ideal $I(\mathcal{T}) : I_c(\mathcal{T})^\infty$.*

*Proof.* The proof follows immediately from Lemma 4.b.

This fact motivates the following definition

**Definition 9.** *The ideal $I(\mathcal{T}) : I_c(\mathcal{T})^\infty$ will be denoted by $I_{op}(\mathcal{T})$.*

What happens if there are more than one thesis polynomials? Suppose that there are $r > 1$ thesis polynomials $\boldsymbol{t}_i$. In that case it is easy to see that the following equality $I_c(\mathcal{T}) = I(\mathcal{T}) : (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r)^\infty$ holds true.

*Remark 1.* With this terminology some parts of Theorem 3 can be rephrased in the following way.

The statement $\mathcal{T}$ is algebraically true if and only if $\sqrt{I_{op}(\mathcal{T})} = \sqrt{I(\mathcal{T})}$.
The statement $\mathcal{T}$ is absolutely false if and only if $I_{op}(\mathcal{T}) = (1)$.

This could be the final point of our investigation. However, if we do the computation in the case of the Bisecting Diagonals Theorem, we get that the optimal hypothesis ideal $(I : (I : (\boldsymbol{t}_1, \boldsymbol{t}_2)^\infty)^\infty)$ is equal to

$$(x_1^2 - 2x_1x_3,\ x_1x_2 - 2x_1x_4,\ x_2^2 - 2x_2x_4,\ x_2x_3 - x_1x_4)$$

We are unable to read these hypotheses. As we had warned before, we have proved a theorem, but we do not know exactly which. Even if we are able to decompose this ideal, we get the two components $\mathfrak{p}_1 = (x_1 - 2x_3,\ x_2 - 2x_4)$, $\mathfrak{p}_4 = (x_1, x_2)$. As we observed, we understand $\mathfrak{p}_4$, but it is not clear how to interpret $\mathfrak{p}_1$.

This fact suggests the idea that sometimes it could be better to describe the validity of a statement not by describing the components on which it is true, but by listing the components on which it is algebraically false. The advantage is that we can do that by *throwing away hypersurfaces*, i.e. by imposing conditions. The disadvantage can be, as it happens in the case of the Bisecting Diagonals Theorem when we put $x_1x_2 \not\models 0$, that we throwaway even a *good component.* This is life.

In any event, it is better to further investigate conditions.

*Remark 2.* We recall that in the paper [11] a polynomial $f$ is called an *irrelevant condition* for $\mathcal{T}$, if $(I, f) = (1)$. Now, either $\mathcal{T}$ is algebraically true or not. In the latter case $I_c \not= (1)$ by Theorem 3.a. If $f \in I_c$ is a condition, then we get $(I, f) \subseteq (I_c, f) \subseteq I_c \not= (1)$, hence $f$ is not an irrelevant condition. This observation implies that there is *no need to consider irrelevant conditions.* This observation was already stated in [11] (see Exercise 4.7).

Inside the ideal $I_c(\mathcal{T})$ we may encounter polynomials $f_i \in \sqrt{I(\mathcal{T})}$ and polynomials $f_j \notin \sqrt{I(\mathcal{T})}$. A polynomial $f_i$ of the first type should be considered useless, since $I(\mathcal{T}) : f_i^\infty = (1)$, so that if we consider the statement under the condition $f_i \not= 0$, *there is no statement anymore.* These conditions are called *trivial* in [11]. Let us formalize this notion.

**Definition 10.** *A condition $f \in I_c(\mathcal{T})$ is called* trivial *if $f \in \sqrt{I(\mathcal{T})}$.*

We observe that if $f$ is a non-trivial condition (hence $\mathcal{T}$ is not algebraically true), then $\sqrt{I(\mathcal{T}) : f^\infty} = \sqrt{I(\mathcal{T})} : f$ is a proper ideal which contains $\sqrt{I(\mathcal{T})}$ strictly. Therefore the condition deletes some but not all of the prime components of $\sqrt{I(\mathcal{T})}$, possibly some good ones. A first simplification can be done in the following way. We decide to sacrifice the possibility of checking the validity of $\mathcal{T}$ on all the components, by imposing all the non-degeneracy conditions, which can be spotted immediately from the assumptions. For instance in Example 6 it is clear that we may assume $l \not= 0$. The observation suggests that we can substitute the hypothesis ideal $I = (l^2 - 3lx)$ with the new hypothesis ideal $I : l^\infty$, and this choice leads immediately to the conclusion that the statement is algebraically true.

Now the next question is: can we find a *good* set $S$ of conditions, i.e. a set $S$ with good properties, and such that $I_{op}(\mathcal{T}) = I(\mathcal{T}) : I_c(\mathcal{T})^\infty = I(\mathcal{T}) : (S)^\infty$? We propose the following definition.

**Definition 11.** *Given a set of generators $\Sigma$ of $I_c(\mathcal{T})$, a subset $S \subseteq \Sigma$ is called an* almost good set of conditions *for $\mathcal{T}$ if*

*a) $I_{op}(\mathcal{T}) = I(\mathcal{T}) : (S)^\infty$*

*b) For every pair $(f, g)$ of elements in $S$, such that $I(\mathcal{T}) : f^\infty \not= I(\mathcal{T}) : g^\infty$, there is no inclusion relation between $I(\mathcal{T}) : f^\infty$ and $I(\mathcal{T}) : g^\infty$.*

*It is called a* good set of conditions *for $\mathcal{T}$ if*

*a) It is almost good.*

*b) For every pair $(f, g)$ of elements in $S$, $I(\mathcal{T}) : f^\infty \not= I(\mathcal{T}) : g^\infty$ holds true.*

**Proposition 2.** *With the same assumptions as in Theorem 3, let $G$ be a set of generators of the ideal $I_c$. We define a partial ordering on $G$ by putting $a \prec b$ if $I : a^\infty \subset I : b^\infty$. We let $S'$ be the subset of $G$ of the minimal elements. Then we consider the equivalence relation on $S'$ which is defined by putting $a \equiv b$ if $I(\mathcal{T}) : a^\infty = I(\mathcal{T}) : b^\infty$. Let $S$ be a set of representatives of the equivalence classes.*

*a) The set $S'$ is an almost good set of conditions for $\mathcal{T}$.*

*b) The set $S$ is a good set of conditions for $\mathcal{T}$.*
*c) If $\mathcal{T}$ is not absolutely false, then $S$ does not contain trivial conditions.*

*Proof.* The first two claims follow from the definitions, so let us prove the third. If $\mathcal{T}$ is not absolutely false, then $I_c \not\subseteq \sqrt{I}$ by Theorem 3.c.2, hence there exists at least one non-trivial condition $b$ among the elements in $G$. Let $a$ be a trivial condition in $G$. Then $I : a^\infty = (1)$, while $I : b^\infty \subset (1)$, so that $a$ is discarded.

*Remark 3.* It is possible to get better set of conditions, by imposing the following constraints

a) $I_{op}(\mathcal{T}) = I(\mathcal{T}) : (S)^\infty$
b) $\bigcap_{j \neq i} (I(\mathcal{T}) : f_j^\infty) \not\subseteq I(\mathcal{T}) : f_i^\infty$, for every $i = 1, \ldots, s$.

Such a set can be called an optimal set of conditions. However, the experience shows that to find optimal set of conditions is too demanding from the computational point of view.

*Remark 4.* While optimal hypothesis ideals depend only on the data of the problem, good and optimal sets of conditions depend on the chosen set of generators of $I_c(\mathcal{T})$, so that they are not intrinsic to the problem. Nevertheless good sets of conditions play an important role in the algorithms which we are going to describe in the next section.

## 4 The Algorithms, the CoCoA Code and Some Examples

At this point we have the possibility of describing an algorithm, which largely automatizes the process of proving that a statement in Euclidean geometry is a theorem. The beginning of the process is a step which aims at optimizing the hypotheses. However, it may delete some degeneracy components on which the statement is true.

**Theorem 4.** *Let $\mathcal{T}$ be a statement in Euclidean geometry, whose hypothesis and thesis are expressed by the vanishing of polynomials $h_1, \ldots, h_r, \boldsymbol{t}$. Let $K$ be a computing field for $\mathcal{T}$ and let $I$ be the ideal generated by $h_1, \ldots, h_r$ in $P = K[x_1, \ldots, x_n]$. Consider the following sequence of instructions.*

*0) (Optional preprocessing) Let $g$ be the product of all the degeneracy conditions to be excluded a priori. Compute $I : g^\infty$, and call this ideal $I$ again.*
*1) Compute $I_c$.*
*2) If $1 \in I_c$, then return* ‘‘`The statement is algebraically true`" *and stop.*
*3) Let $G$ be a set of non-zero generators of $I_c$.*
*4) Compute $I : f^\infty$ for every $f \in G$. If $1 \in I : f^\infty$ for all $f \in G$, then return* ‘‘`The statement is absolutely false`" *and stop.*
*5) Consider the partial order on $G$ given by $a \prec b$ if $I : a^\infty \subseteq I : b^\infty$, and let $S'$ be the set of the minimal elements of $G$ with respect to $\prec$. Then consider the equivalence relation on $S'$ given by $a \equiv b$ if $I : a^\infty = I : b^\infty$, and let $S \subseteq S'$ be a subset of representatives of the equivalence classes.*

*6) Compute* $J = I : I_c^\infty$
*7) Return* $J$, $S$ *and stop.*

*This is an algorithm which decides whether* $\mathcal{T}$ *is algebraically true or absolutely false. In the case that* $\mathcal{T}$ *is neither algebraically true nor absolutely false, the algorithm finds an optimal hypothesis ideal and a good set of conditions for* $\mathcal{T}$*.*

*Proof.* The first instruction requires the computation of $I_c$. If $I_c = (1)$ then $\mathcal{T}$ is algebraically true by Theorem 3.a. If $I_c$ is a proper ideal, then there exist non-constant polynomials $f_1, \ldots, f_s$ such that $I_c = (f_1, \ldots, f_s)$. Let us consider the ideals $I : f_i^\infty$. We know that $f_i \in I_c$ hence $f_i \in \sqrt{I_c} = \sqrt{I : \boldsymbol{t}^\infty}$. Then Lemma 3 implies that $\boldsymbol{t} \in \sqrt{I : f_i^\infty}$. Since we know that $\boldsymbol{t} \notin \sqrt{I}$, we may conclude that $\sqrt{I : f_i^\infty}$, which is equal to $\sqrt{I} : f_i$ by Lemma 3.a, is strictly bigger than $\sqrt{I}$.

If all the $f_i$ belong to $\sqrt{I}$, then $I_c \subseteq \sqrt{I}$ and $\mathcal{T}$ is absolutely false by Theorem 3.b.

It remains to show that if $\mathcal{T}$ is neither algebraically true nor absolutely false, then $S$ is a good set of conditions for $\mathcal{T}$. This fact follows from Proposition 2.

*Remark 5.* It is clear how to modify Step 5 to get optimal sets of conditions instead of good sets of conditions. It is also clear that Steps 5, 6 can be used interchangeably. If one deletes step 5, then Step 7 reads: *Return* $J$ *and stop.* If one deletes step 6, then Step 7 reads: *Return* $S$ *and stop.*

Now we illustrate the CoCoA code, which transforms the algorithm described above into an executable code. We insist on the notion of good set of conditions, which is the most important from a computational point of view.

### Algorithm: Almost Good Set of Conditions

```
Define AlmostGoodSetOfConditions(L)
  AlmostGood := [];
  Foreach ElementOfL In L Do
    Inserted := FALSE;
    I := 1;
    While I <= Len(AlmostGood) And Not Inserted Do
      If AlmostGood[I].Ideal = ElementOfL.Ideal Then
        Append(AlmostGood[I].Polys, ElementOfL.Poly);
        Inserted := TRUE;
      ElsIf AlmostGood[I].Ideal > ElementOfL.Ideal Then
        AlmostGood[I] := Record(Ideal=ElementOfL.Ideal,
                                Polys=[ElementOfL.Poly]);
        Inserted := TRUE;
      ElsIf AlmostGood[I].Ideal < ElementOfL.Ideal Then
        Inserted := TRUE;
      EndIf;
      I := I+1;
    EndWhile;
    If Not Inserted Then
      Append(AlmostGood, Record(Ideal=ElementOfL.Ideal,
                                Polys=[ElementOfL.Poly]));
    EndIf;
  EndForeach;
  Return [X.Polys | X In AlmostGood]
EndDefine; -- AlmostGoodSetOfConditions
```

## Algorithm: Proving

```
Define Proving(Hypothesis,Thesis)
  ConditionsIdeal := Saturation(Hypothesis,Ideal(Thesis));
  If ConditionsIdeal=Ideal(1) Then
    Return Record(Statement=TRUE);
  EndIf;
  Conditions := Gens(ConditionsIdeal);
  ConditionsIdealList :=
      [ Record(Ideal=Saturation(Hypothesis,Ideal(C)), Poly=C) |
        C In Conditions];
  If [ X In ConditionsIdealList | X.Ideal<>Ideal(1) ] = [] Then
    Return  Record(Statement=FALSE, Conditions=[]);
  EndIf;
  Return Record(Statement=FALSE, Conditions =
               AlmostGoodSetOfConditions(ConditionsIdealList));
EndDefine; -- Proving
```

or

```
  Return Record(Statement=FALSE, OptimalHypothesisIdeal =
       IntersectionList([I.Ideal | I In ConditionsIdealList]));
```

*Remark 6.* It should be noticed that Algorithm Proving does not return the good set of conditions $S$, as described in Theorem 4, Step 5. It returns an almost good set of conditions $S'$. The transition from $S'$ to $S$ is done by *human judgment*, so it belongs to a phase which could be called *postprocessing*, as we shall see with the help of some examples.

The first example that we are going to investigate is Example 5.

*Example 5 (continued).* In the preprocessing phase we decide to exclude the possibility that the segment $AB$ degenerates to a point. We do that by replacing the hypothesis ideal $(h_1, h_2, h_3)$ with the new hypothesis ideal $I = (h_1, h_2, h_3) : (x_1)^\infty$. Then we compute a set of conditions by using Algorithm Proving, and get

$$[x_3 x_4 - 1/10\sqrt{5}x_1^2 - 1/10\sqrt{5}x_1 x_4 + 1/10\sqrt{5}x_4^2 - 1/10x_1^2 - 1/2x_1 x_4 - 1/10x_4^2,\\ \sqrt{5}x_1^2 x_4 + \sqrt{5}x_1 x_4^2 - \sqrt{5}x_4^3 + 2x_1^3 + x_1^2 x_4 - 3x_1 x_4^2 + x_4^3,\\ \sqrt{5}x_1^3 - 2\sqrt{5}x_1 x_4^2 + \sqrt{5}x_4^3 - x_1^3 + 2x_1^2 x_4 + 4x_1 x_4^2 - 3x_4^3].$$

All the conditions are equivalent. We conclude that a good set of conditions is

$$\{\sqrt{5}x_1^2 x_4 + \sqrt{5}x_1 x_4^2 - \sqrt{5}x_4^3 + 2x_1^3 + x_1^2 x_4 - 3x_1 x_4^2 + x_4^3\}$$

We can factor the polynomial and get

$$(\sqrt{5}x_4 + 2x_1 - x_4)(x_1 + 1/2x_4 - \sqrt{5}/2x_4)(x_1 + 1/2x_4 + \sqrt{5}/2x_4)$$

We conclude that the statement is algebraically true under the conditions that $\frac{x_1}{x_4} \not= \frac{-1-\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}, \frac{\sqrt{5}-1}{2}$. The last condition excludes the situation where $B$ is external to the segment $AC$.

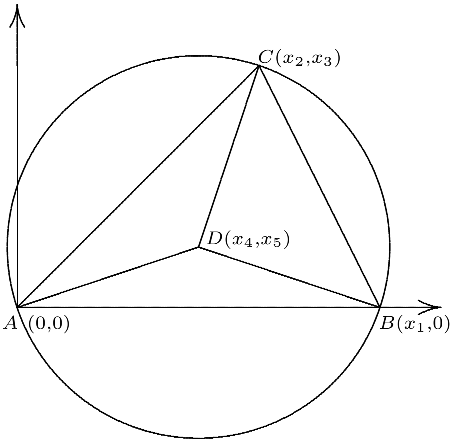
You remember that a proposed alternative way was to impose that $AE$ is the golden section of $AB$. In this way we were able to use $\mathbb{Q}$ as the computing field.

Then we compute a set of conditions by using Algorithm Proving, and get

$[x_1^2 + x_1 x_4 - x_4^2,$
$x_3 - 2/5x_1 - 1/5x_4,$
$x_2 - 4/5x_1 - 2/5x_4]$

All the conditions are equivalent. We conclude that the statement is algebraically true under one of these conditions. The last two are not clearly interpretable. However, if we work in $\mathbb{Q}[\sqrt{5}][x_1, x_2, x_3, x_4]$, we factor $x_1^2 + x_1 x_4 - x_4^2 = (x_1 + 1/2x_4 + \sqrt{5}x_4)(x_1 + 1/2x_4 - \sqrt{5}x_4)$. We can conclude that the statement is algebraically true under the condition $\frac{x_1}{x_4} \not= \frac{-1-\sqrt{5}}{2}, \frac{\sqrt{5}-1}{2}$. As before, the second condition excludes the situation where $B$ is external to the segment $AC$.

*Example 7 (Isosceles or Right-angled Triangles).* Let be given a triangle which is isosceles or right-angled. Show that the center of the circumscribed circle belongs to one of the sides of the triangle.



We introduce Cartesian coordinates, and place the triangle as in the picture. Then we have

$$A = (0,0),\ B = (x_1,0),\ C = (x_2,x_3),\ D = (x_4,x_5)$$

$DA = DB$ yields the polynomial $h_1 = 2x_1x_4 - x_1^2$.
$DA = DC$ yields the polynomial $h_2 = x_2^2 + x_3^2 - 2x_2x_4 - 2x_3x_5$.

We assume that the triangle is isosceles or right-angled in $C$.
$AC = CB$ or $AC \perp CB$ yields the polynomial $h_3 = x_1^3x_2 - 3x_1^2x_2^2 + 2x_1x_2^3 - x_1^2x_3^2 + 2x_1x_2x_3^2$.

The thesis $D \in AB$ yields the polynomial $\boldsymbol{t} = x_1x_5$. Therefore a computing field is $\mathbb{Q}$ and we may work in the polynomial ring $\mathbb{Q}[x_1, x_2, x_3, x_4, x_5]$.

We run Algorithm Proving without any preprocessing, and get the following almost good set of conditions
$[x_3x_4 - x_3x_2,\ x_3^3 - x_3x_4^2 - 2x_3^2x_5]$.

The statement is not a theorem. Using the equivalence relation introduced in Theorem 4, Step 5, we check that the two conditions found are equivalent. Therefore a good set of conditions is a single polynomial, and we can choose between $x_3x_4 - x_3x_2$ and $x_3^3 - x_3x_4^2 - 2x_3^2x_5$.

We enter the phase of the postprocessing. We observe that it is easier to interpret the first condition. It means that the statement is algebraically true under the conditions that $x_3 \not\models 0$ and$x_4 \not\models x_2$. Clearly, $x_3 \not\models 0$ excludes the degenerate case that the triangle degenerates to the segment $AB$, while $x_4 \not\models x_2$ excludes the case that the triangle is isosceles.

We run the second variant of the Algorithm Proving and get the Optimal Hypothesis Ideal

$$(x_2^2 + x_3^2 - 2x_2x_4 - 2x_3x_5,\ x_1^2 - 2x_1x_4,\ x_1x_5)$$

As it happens frequently, the optimal hypothesis ideal is difficult to read. In this case we can understand why. Namely, in this case we can compute

$$\sqrt{I} = \mathfrak{p}_1 \cap \mathfrak{p}_2 \cap \mathfrak{p}_3 \cap \mathfrak{p}_4 \cap \mathfrak{p}_5$$

where

$$\mathfrak{p}_1 = (x_1, x_2^2 + x_3^2 - 2x_2x_4 - 2x_3x_5) \qquad \mathfrak{p}_2 = (x_3, x_1 - x_2, 2x_4 - x_1)$$
$$\mathfrak{p}_3 = (x_3, x_2, 2x_4 - x_1) \qquad \mathfrak{p}_4 = (x_4 - x_2, x_3^2 - x_4^2 - 2x_3x_5, 2x_4 - x_1)$$
$$\mathfrak{p}_5 = (x_5, x_2^2 + x_3^2 - 2x_2x_4, 2x_4 - x_1)$$

It is easy to check that $\boldsymbol{t} \in \mathfrak{p}_5$ e $\boldsymbol{t} \in \mathfrak{p}_1$, but $\boldsymbol{t} \notin \mathfrak{p}_2$, $\boldsymbol{t} \notin \mathfrak{p}_3$, $\boldsymbol{t} \notin \mathfrak{p}_4$.

In conclusion the statement is true on the component which represents the right-angled triangles, and on the component which represents the triangles which degenerate to the segment $AC$. This is why the optimal hypothesis ideal is complicated.

We could follow a different path by doing some preprocessing. Suppose we want to avoid the cases where the triangle degenerates to a segment. This means that the ideal $I$ should be replaced by the ideal $I : (x_1x_3)^\infty$ and $\boldsymbol{t} = x_5$. Let us do that. We compute and get

$$I : (x_1x_3)^\infty = (x_1 - 2x_4,\ x_2x_5 - x_4x_5,\ x_2^2 + x_3^2 - 2x_2x_4 - 2x_3x_5,\ x_3^2x_5 - x_4^4x_5 - 2x_3x_5^2)$$

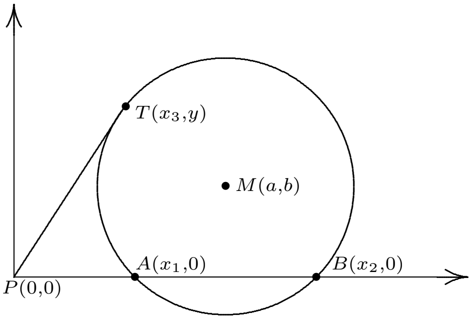
We run Algorithm Proving on this new ideal, and get the following set of almost good conditions $[x_2 - x_4,\ x_3^2 - x_4^2 - 2x_3x_5]$.

Again we check that the two conditions are equivalent, we use the first and conclude that the statement is true unless the triangles are isosceles, hence it is true for right-angled triangles. The optimal hypothesis ideal turns out to be

$$(x_5,\ x_1 - 2x_4,\ x_2^2 + x_3^2 - 2x_2x_4)$$

Now it should be clear that it expresses that the triangle is right-angled in $C$.

*Example 8 (Tangent and Secant).* Let be given a circle $C$ and an external point $P$. Through $P$ consider a tangent line $l$ and a secant line $s$ to the circle. Then the square of the distance of $P$ to the point of contact of $l$ with $C$ is equal to the product of its distances to the intersection points of $s$ with $C$.



We introduce Cartesian coordinates, and place the axes as in the picture. Then we have

$$P=(0,0),\ A=(x_1,0),\ B=(x_2,0),\ M=(a,b),\ T=(x_3,y)$$

$TM = AM$ yields the polynomial $h_1=(x_3-a)^2+(y-b)^2-(x_1-a)^2-b^2$.
$AM = BM$ yields the polynomial $h_2=(x_1-a)^2-(x_2-a)^2$.
$PT \perp MT$ yields the polynomial $h_3=x_3(x_3-a)+y(y-b)$.

The thesis $\overline{PT}^2=\overline{PA}\cdot\overline{PB}$ yields the polynomial $x_3^2+y^2-x_1x_2$.

Therefore a computing field is $\mathbb{Q}$ and we may work in the polynomial ring $\mathbb{Q}[x_1,x_2,x_3,y,a,b]$.

We run Algorithm Proving without any preprocessing, and get the following good set of conditions $[x_1-x_2]$.

This comes somehow as a little surprise. It means that the statement is false on the component determined by the choice $A=B$, i.e. when we do not specify that $A$, $B$ are the *two points* of intersections of the secant line with the circle.

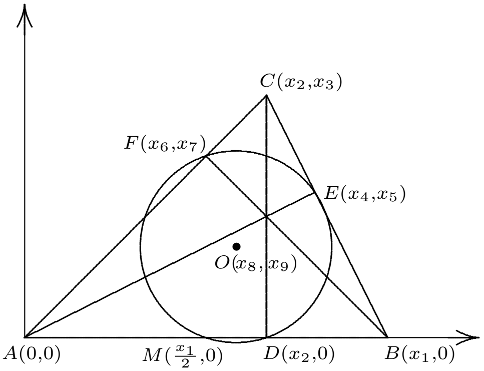
It is a typical situation where it is very easy to overlook a degeneracy condition.

*Example 9 (Feet and Midpoints).* In every triangle the circle passing through the feet of the three altitudes intersects the sides of the triangle in their midpoints.

We introduce Cartesian coordinates, and place the geometric objects as in the picture. Then we have that $A=(0,0)$, $B=(x_1,0)$, $C=(x_2,x_3)$ are the vertices of the triangle, $D=(x_2,0)$, $E=(x_4,x_5)$, $F=(x_6,x_7)$ are the feet of the altitudes. Finally let $O=(x_8,x_9)$ be the center of the circle, $M=(x_1/2,0)$ the medium point of $AB$.

$F\in AC$ yields the polynomial $h_1=-x_3x_6+x_2x_7$,
$AC\perp FB$ yields the polynomial $h_2=x_1x_2-x_2x_6-x_3x_7$,
$E\in CB$ yields the polynomial $h_3=-x_1x_3+x_3x_4+x_1x_5-x_2x_5$,

$AE \perp BC$ yields the polynomial $h_4 = -x_1x_4 + x_2x_4 + x_3x_5$,
$OE = OD$ yields the polynomial $h_5 = -x_2^2 + x_4^2 + x_5^2 + 2x_2x_8 - 2x_4x_8 - 2x_5x_9$,
$OD = OF$ yields the polynomial $h_6 = x_2^2 - x_6^2 - x_7^2 - 2x_2x_8 + 2x_6x_8 + 2x_7x_9$.



The thesis $OM = OD$ yields the polynomial $\boldsymbol{t} = 1/4x_1^2 - x_2^2 - x_1x_8 + 2x_2x_8$. Therefore a computing field is $\mathbb{Q}$ and we may work in the polynomial ring $\mathbb{Q}[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9]$. In the preprocessing phase we decide to exclude some degenerate cases. Namely we exclude the possibility that the triangle degenerates to a segment, by replacing the hypothesis ideal $(h_1, h_2, h_3, h_4, h_5, h_6)$ with the new hypothesis ideal $I = (h_1, h_2, h_3, h_4, h_5, h_6) : (x_1x_3)^\infty$.

Then we compute an almost good set of conditions by using Algorithm Proving, and get

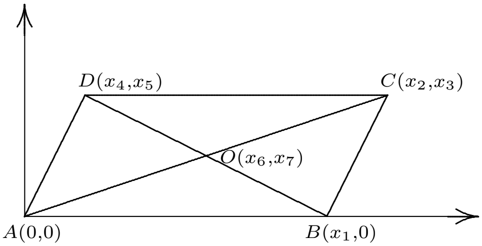
$[x_5x_7x_8 - 2x_5x_6x_9 + x_4x_7x_9 - 2x_7x_8x_9 + 2x_6x_9^2,$
$x_6^2x_7^2 + x_7^4 - 2x_6x_7^2x_8 - 2x_7^3x_9 + 4x_6x_7x_8x_9 - 4x_6^2x_9^2,$
$x_1x_5 + x_5x_6 - x_4x_7 - 2x_5x_8 + 2x_7x_8 - 2x_1x_9 + 2x_4x_9 - 2x_6x_9,$
$x_3x_5 + 2x_2x_8 - 2x_4x_8 - 2x_5x_9,$
$x_3x_6 - x_4x_7,$
$x_4x_6 - x_6^2 + x_5x_7 - x_7^2,$
$x_1x_3 - x_4x_7 - 2x_3x_8 + 2x_7x_8 - 2x_1x_9 + 4x_2x_9 - 2x_6x_9,$
$x_2x_4 - x_6^2 - x_7^2 - 2x_2x_8 + 2x_6x_8 + 2x_7x_9,$
$x_3x_4 - 2x_5x_6 - 2x_3x_8 + 2x_5x_8 + 4x_2x_9 - 2x_4x_9,$
$x_1x_2 - x_6^2 - x_5x_7 - x_7^2 - 2x_2x_8 + 2x_6x_8 + 2x_7x_9,$
$x_2x_5 - x_5x_6,$
$x_2x_6 - x_6^2 + x_5x_7 - x_7^2,$
$x_3x_7 - 2x_5x_7 - 2x_2x_8 + 2x_6x_8 + 2x_7x_9,$
$x_2x_7 - x_4x_7,$
$x_5x_6x_7 - 2x_5x_6x_9,$
$x_5^2x_7 - 2x_5x_7x_9,$
$x_4x_5x_7 - 2x_5x_6x_9,$
$x_5x_7^2 - 2x_5x_7x_9,$
$x_4x_7^2 - 2x_7^2x_8 - 2x_4x_7x_9 + 2x_6x_7x_9 + 4x_7x_8x_9 - 4x_6x_9^2,$
$x_5x_6^2 - 1/2x_6^2x_7 - 1/2x_7^3 - x_5x_6x_8 + x_6x_7x_8 - x_6^2x_9 + x_5x_7x_9,$

$x_5^2x_6 - 2x_5x_6x_9,$
$x_1x_7^2 - 2x_7^2x_8 - 2x_1x_7x_9 + 2x_6x_7x_9 + 4x_7x_8x_9 - 4x_6x_9^2]$

All the conditions are equivalent. We conclude that a good set of conditions is $\{x_2x_5 - x_5x_6\}$, and that the statement is algebraically true under the condition that $x_2x_5 - x_5x_6 \not\models 0$, which means that the triangle should not be right-angled.

In this case the Optimal Hypothesis Ideal is difficult to read.

*Example 10 (Parallelogram).* Given a parallelogram, the intersection point of the diagonals lies on a side.



We introduce Cartesian coordinates, and place the geometric objects as in the picture. Then we have that $A = (0,0)$, $B = (x_1, 0)$, $C = (x_2, x_3)$, $D = (x_4, x_5)$ are the vertices of the parallelogram, and $O = (x_6, x_7)$ is the intersection point of the diagonals.

$AB \parallel DC$ yields the polynomial $h_1 = -x_1x_3 + x_1x_5$,
$AD \parallel BC$ yields the polynomial $h_2 = -x_3x_4 - x_1x_5 + x_2x_5$,
$O \in AC$ yields the polynomial $h_3 = -x_3x_6 + x_2x_7$,
$O \in DB$ yields the polynomial $h_4 = -x_1x_5 + x_5x_6 + x_1x_7 - x_4x_7$.

The thesis $O \in AB$ yields the polynomial $\boldsymbol{t} = x_1x_7$. Therefore a computing field is $\mathbb{Q}$ and we may work in $\mathbb{Q}[x_1, x_2, x_3, x_4, x_5, x_6, x_7]$.

Then we compute an almost good set of conditions by using Algorithm Proving, and get

$[x_1^2 - x_4^2 - 2x_1x_6 + 2x_4x_6,$
$x_1x_2 - x_1x_6 - x_2x_6 + x_4x_6,$
$x_2^2 - 2x_2x_6,$
$x_2x_4 + x_1x_6 - x_2x_6 - x_4x_6]$

This is a case where, no matter which condition we take, its understanding is not trivial. However, we may compute the Optimal Hypothesis Ideal and get that the Optimal Hypothesis Ideal is the ideal

$$(x_1x_5,\ x_1x_3,\ x_5x_6 - x_4x_7,\ x_3x_6 - x_2x_7,\ x_3x_4 - x_2x_5,\ x_1x_7)$$

whose meaning is clear. We conclude that the statement is algebraically true for the parallelograms which degenerate into a segment.

In this case the possibility of working with two options was useful for the correct interpretation of the output.

## References

1. B. Buchberger, On Finding a Vector Space Basis of the Residue Class Ring Modulo a Zero-dimensional Polynomial Ideal (in German), PhD Thesis, Universität Innsbruck, Innsbruck (1965).
2. B. Buchberger, Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, Chapter 6 in N. K. Bose ed., Multidimensional Systems Theory, pp. 184–232, D. Reidel Publ. Comp., Dordrecht (1985).
3. S.-C. Chou, Mechanical Geometry Theorem Proving (Mathematics and Its Applications 41), D. Reidel Publ. Comp., Dordrecht (1988).
4. A. Capani, G. Niesi, L. Robbiano, CoCoA, a System for Doing Computations in Commutative Algebra. Version 4.0 is available at `http://cocoa.dima.unige.it`.
5. H. Gelernter, J. R. Hansen, D. W. Loveland, Empirical Exploration of the Geometry Theorem Proving Machine, in A. Feigenbaum, A., J. Feldman eds., Computer and Thought, pp. 153–163, McGraw–Hill, New York (1963).
6. F. De Giovanni, T. Landolfi, Le Dimostrazioni di Teoremi fondate sull'uso del Calcolatori, Bollettino U.M.I., La matematica nella Società e nella Cultura (8) 2-A: 69–81 (1999).
7. D. Kapur, Using Gröbner Bases to Reason about Geometry Problems, J. Symb. Comp. 2: 399–408 (1986).
8. D. Kapur, A Refutational Approach to Geometry Theorem Proving, Artificial Intelligence 37: 61–93 (1988).
9. M. Kreuzer, L. Robbiano, Computational Commutative Algebra 1, Springer-Verlag, Berlin Heidelberg (2000).
10. B. Kutzler, S. Stifter, On the Application of Buchberger's Algorithm to Automated Geometry Theorem Proving, J. Symb. Comput. 2: 389–397 (1986).
11. T. Recio, H. Sterk, M. Pilar Vélez, Automatic Geometry Theorem Proving, in A. Cohen, H. Cuypers, H. Sterk eds., Some Tapas of Computer Algebra (Algorithms and Computation in Mathematics 4), pp. 276–296, Springer-Verlag, Berlin Heidelberg (1999).
12. T. Recio, M. Pilar Vélez, Automatic Discovery of Theorems in Elementary Geometry, J. Automat. Reason. 23: 63–82 (1999).
13. J. Richter-Gebert, Realization Spaces of Polytopes (Lecture Notes in Mathematics 1643), Springer-Verlag, Berlin Heidelberg (1996).
14. D. Wang, Gröbner Bases Applied to Geometric Theorem Proving and Discovering, in B. Buchberger and F. Winkler eds., Gröbner Bases and Applications (London Mathematical Society Lecture Notes Series 251), pp. 281–301, Cambridge University Press, Cambridge (1998).
15. W.-T. Wu, On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry, Scientia Sinica 21: 150–172 (1978).
16. W.-T. Wu, Toward Mechanization of Geometry — Some Comments on Hilbert's "Grundlagen der Geometrie", Acta Math. Scientia 2: 125–138 (1982).
17. W.-T. Wu, Basic Principles of Mechanical Theorem Proving in Elementary Geometries, J. Syst. Sci. Math. Sci. 4: 207–235 (1984).

# The Kinds of Truth of Geometry Theorems

Michael Bulmer[1], Desmond Fearnley-Sander[2], and Tim Stokes[3]

[1] University of Queensland, Queensland, Australia
mrb@maths.uq.edu.au
[2] University of Tasmania, Tasmania, Australia
Desmond.FearnleySander@utas.edu.au
[3] Murdoch University, Western Australia, Australia
stokes@prodigal.murdoch.edu.au

**Abstract.** Proof by refutation of a geometry theorem that is not universally true produces a Gröbner basis whose elements, called *side polynomials*, may be used to give inequations that can be added to the hypotheses to give a valid theorem. We show that (in a certain sense) all possible subsidiary conditions are implied by those obtained from the basis; that what we call the *kind of truth* of the theorem may be derived from the basis; and that the side polynomials may be classified in a useful way. We analyse the relationship between side polynomials and kinds of truth, and we give a unified algorithmic treatment of side polynomials, with examples generated by an implementation.

## 1 Algebraic Preliminaries

Throughout, let $n$ be a positive integer and $L$ a fixed field containing the field $\mathbb{Q}$ of rational numbers. Let $\mathbb{Q}[X_n]$ be the ring of polynomials with $n$-variable set $X_n = \{x_1, x_2, \ldots, x_n\}$ over $\mathbb{Q}$.

Let $F \subseteq \mathbb{Q}[X_n]$ and $f \in \mathbb{Q}[X_n]$. We call $(F, f)$ a *possible theorem.* $(F)_{X_n}$ denotes the ideal generated by $F$ in $\mathbb{Q}[X_n]$. For $a \in L^n$ we denote by $f(a)$ the result of substituting $a$ for their corresponding variables in $f$ and evaluating.

For $F \subseteq \mathbb{Q}[X_n]$, let

$$C_{X_n}(F) = \{f \in \mathbb{Q}[X_n] \mid \text{ for all } a \in L^n, h(a) = 0 \text{ for all } h \in F \text{ implies } f(a) = 0\},$$

an ideal of $\mathbb{Q}[X_n]$ as is easily checked. If the choice of polynomial ring is clear we often write just $C(F)$. The ideal $C_{X_n}(F)$ depends on the field $L$. For example if $n = 1$ and $F = \{x^3+x\}$, then if $L = R$ is the field of real numbers, $C_{X_1}(F) = (x)$, because $x^3 + x$ has only one root $x = 0$ in $R$. If $L = C$, the field of complex numbers, then $C_{X_1}(F) = (x^3 + x)$, because $x^3 + x$ has three roots $x = 0, i, -i$ in $C$. If $L$ is algebraically closed, then by Hilbert's Nullstellensatz, $C_{X_n}(F) = \{f \in \mathbb{Q}[X_n] \mid f^k \in (F)_{X_n}\}$, the *radical ideal generated by* $F$ in $\mathbb{Q}[X_n]$.

Dually, for $F \subseteq \mathbb{Q}[X_n]$, let

$$\mathcal{V}_{X_n}(F) = \{a \mid a \in L^n, f(a) = 0 \text{ for all } f \in F\},$$

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 129–142, 2001.

the *variety* associated with $F \subseteq \mathbb{Q}[X_n]$. Again, we often write just $\mathcal{V}(F)$ if the context is clear, and if $F = \{f\}$, a singleton set, we shall often write $\mathcal{V}_{X_n}(f)$ (or just $\mathcal{V}(f)$) rather than $\mathcal{V}_{X_n}(\{f\})$. Varieties are closed under arbitrary intersections and finite unions.

There is a familiar dual isomorphism between the lattices of varieties of the form $\mathcal{V}_{X_n}(G)$ and ideals of the form $C_{X_n}(G)$, $G \subseteq \mathbb{Q}[X_n]$. Because the lattice of varieties is distributive and satisfies the descending chain condition, every variety $\mathcal{V}$ has a unique decomposition

$$\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \cdots \cup \mathcal{V}_n$$

into distinct *irreducible components* (varieties which cannot themselves be expressed non-trivially as unions of two or more proper subvarieties).

## 2 The Kind of Truth of a Possible Theorem

In geometrical theorem proving, the higher level statement of a valid geometry theorem naturally translates into an equational implication involving polynomials, with geometrical predicates such as "points $A, B$ and $C$ are collinear" becoming polynomial equations via coordinatisation. Further, one can require that certain variables be treated as independent, in the sense that no algebraic relations are assumed to hold amongst them. Chou has argued in [2] that the specification of the independent variables in a geometry theorem is an integral part of the algebraic formulation: such variables are chosen according to a notional "construction" that takes place when the geometry theorem hypotheses are read in order. It is this approach we adopt.

Formally, suppose the variables in $U \subseteq X_n$ are specified as being independent; view $\mathbb{Q}[U]$ as a subring of $\mathbb{Q}[X_n]$. We say that a variety $\mathcal{V}$ is *$U$-generic* (or simply *generic* if the choice of $U$ is clear) if no polynomial in $\mathbb{Q}[U]$ is zero on all of $\mathcal{V}$. If $L$ is algebraically closed, then $\mathcal{V}_{X_n}(F)$ is $U$-generic if and only if $(F)_{X_n} \cap \mathbb{Q}[U] = \{0\}$; this is called *$U$-independence* in [5]. Let $\mathcal{G}_{X_n}(F)$ denote the union of all $U$-generic irreducible components of $\mathcal{V}_{X_n}(F)$.

We can now define the four basic *kinds of truth* of a possible theorem: for $F \cup \{f\} \subseteq \mathbb{Q}[X_n]$, we say the possible theorem $(F, f)$ is

1. *universally true* if $f$ is zero on all of $\mathcal{V}_{X_n}(F)$, that is, $f(a) = 0$ for all $a \in \mathcal{V}_{X_n}(F)$;
2. *generically true* if $f$ is zero on $\mathcal{G}_{X_n}(F)$;
3. *generically conditionally true* if there exists an irreducible component of $\mathcal{G}_{X_n}(F)$ on which $f$ is zero but $f$ is not zero on all of $\mathcal{G}_{X_n}(F)$;
4. *generically false* there is no irreducible component of $\mathcal{G}_{X_n}(F)$ on which $f$ is zero.

We are further able to separate the final category into two subcategories. A generically false possible theorem is:

- *degenerately true* if there is no irreducible component of $\mathcal{G}_{X_n}(F)$ on which $f$ is zero, yet there is at least one irreducible component of $\mathcal{V}_{X_n}(F)$ on which $f$ is zero;
- *rarely true* if there is no irreducible component of $\mathcal{V}_{X_n}(F)$ on which $f$ is zero.

We call these kinds of truth 1 to 4 respectively, with kind 4 split into 4(a) and 4(b) as above. Note that a possible theorem is of at least one of these five types, and exactly one of types 2 to 4 (b). We shall show how to determine the kind of truth of a possible theorem using Gröbner bases for the case where $L$ is algebraically closed.

Generic truth reflects the idea that the conclusion holding on the generic irreducible components of the hypotheses is what is "really intended" by the author of a theorem. Generically conditional truth occurs when there is some ambiguity in the hypotheses of the theorem and the conclusion will not hold on all generic irreducible components; this is in practice a rare situation but can occur as illustrated in [2] and elsewhere. Generic falsity occurs when the conclusion is valid on no generic irreducible components. In contrast to rare truth, degenerate truth is at least a form of conditional truth, and the fact that it can be algorithmically distinguished from rare truth has led us to define it separately.

Most of these kinds of truth have in essence been considered elsewhere along with algorithmic ways of determining the kind of truth of a possible theorem (see [2,1] and more recently [7,6] for instance). However, so far as we know, no unified algorithmic treatment of the kind presented here, applying to all the kinds of truth discussed here and providing a complete set of side polynomials for each, has appeared previously.

## 3 Side Polynomials and Kinds of Truth

For the remainder of the article, $F$ and $f$ will be an arbitrary subset and an element of $\mathbb{Q}[X_n]$ respectively.

In [4], a method described by Kapur as refutational theorem proving was considered for proving geometry theorems translated into polynomial equations in this way. Kapur's approach was based on the idea of considering the conjunction of the hypotheses of the theorem with the negation of the conclusion (in a certain sense), forming a Gröbner basis and determining if it was $(1)_{X_n}$; if so, then the theorem was validated and if not the polynomials in the basis could be used to give side conditions in the form of inequations which could be added to the hypotheses in order to give a valid theorem.

Such side conditions prove necessary because most standard Euclidean geometry "theorems", as normally stated, are not universally true, owing to the absence from the hypotheses of certain additional non-degeneracy conditions that may be represented algebraically as inequations. These may correspond to hypotheses of the form "points $A, B$ and $C$ are *not* collinear", and so forth. Often, such extra hypotheses are not easy to guess, as is discussed at length in

[2], although synthetic proofs of geometry theorems make at least tacit use of them. Note that any finite number of inequations may be expressed as a single inequation.

We show that all possible side conditions (in a certain natural sense) are implied by those obtained from the Gröbner basis used in Kapur's method, and that the kind of truth of the theorem may be derived from this basis; moreover, the side polynomials may be classified in a useful way.

Formally, we say that $g \in \mathbb{Q}[X_n]$ is a *side polynomial* for $(F, f)$ if $f(a) = 0$ for all $a \in \mathcal{V}(F)$ for which $g(a) \neq 0$. Let the set of all side polynomials for $(F, f)$ be denoted $side(F, f)$.

**Theorem 1.** *For the possible theorem* $(F, f)$, $C(side(F, f)) = (side(F, f))_{X_n} = side(F, f)$.

*Proof.* Now of course $side(F, f) \subseteq (side(F, f))_{X_n} \subseteq C(side(F, f))$. Suppose $h \in C(side(F, f))$ and that $h(a) \neq 0$ for some $a \in \mathcal{V}(F)$. Then because $h \in C(side(F, f))$, by definition there exists $g \in side(F, f)$ such that $g(a) \neq 0$, so $f(a) = 0$. Hence $h \in side(F, f)$ by definition, and so $C(side(F, f)) \subseteq side(F, f)$. Hence the two sets are equal. □

For example, letting $F = \{x_1x_2\}$ and $f = x_1$, one element of $side(F, f)$ is $g = x_2$, since for $c, d \in L$, if $cd = 0$ and $d \neq 0$, then necessarily $c = 0$. In other words, for all $a \in L^2$, if $a \in \mathcal{V}(F)$ and $g(a) \neq 0$, then $f(a) = 0$. In fact in this case, $side(F, f) = (x_2)$, the ideal generated by $x_2$ in $\mathbb{Q}[X_2]$.

There are various kinds of side polynomial, corresponding to the various kinds of truth as we shall show. Thus a side polynomial $g$ for $(F, f)$ is

1. *generic* if $g \in \mathbb{Q}[U]$;
2. *generically resolving* if $g \notin \mathbb{Q}[U]$ and $(F, g)$ is not generically true;
3. *degenerate* if $g \notin \mathbb{Q}[U]$ and $(F, g)$ is generically true;
4. *extraneous* if $g \notin \mathbb{Q}[U]$ and $(F, g)$ is universally true.

**Theorem 2.** *The possible theorem* $(F, f)$ *is*

1. *universally true if and only if every polynomial is a side polynomial;*
2. *generically true if and only if there exists a generic side polynomial;*
3. *generically conditionally true if and only if there exist no generic side polynomials, but there does exist a generically resolving side polynomial;*
4. *degenerately true if and only if there exist no generic or generically resolving side polynomials, but there does exist a degenerate side polynomial;*
5. *rarely true if and only if all side polynomials are extraneous.*

*Proof.* Condition 1 is immediate, and condition 2 is proved in [2].

Now suppose $(F, f)$ has no generic or generically resolving side polynomials. If $g \in side(F, f)$ then $(F, g)$ is generically true, so if $h(a) = 0$ for all $h \in F$ and $g(a) \neq 0$ then $a \notin \mathcal{G}(F)$. Suppose $h \notin \mathbb{Q}[U]$ and $h$ is a non-degenerate side polynomial for $(F, f)$; hence $fh$ is zero on $\mathcal{V}(F)$, so certainly $fh$ is zero on $\mathcal{G}_{X_n}(F)$, so $\mathcal{G}_{X_n}(F) \subseteq \mathcal{V}(fh) = \mathcal{V}(f) \cup \mathcal{V}(h)$. Hence $\mathcal{G}_{X_n}(F) = [\mathcal{G}_{X_n}(F) \cap \mathcal{V}(f)] \cup$

$[\mathcal{G}_{X_n}(F) \cap \mathcal{V}(h)]$. But $h$ is non-degenerate, so $h$ is not zero on all of $\mathcal{G}_{X_n}(F)$, so $\mathcal{G}_{X_n}(F) \ /\subseteq \mathcal{V}h)$, so $\mathcal{G}_{X_n}(F) \cap \mathcal{V}(h) \subset \mathcal{G}_{X_n}(F)$, and so $\mathcal{G}_{X_n}(F) \cap \mathcal{V}(f)$ is a finite non-empty union of irreducible components of $\mathcal{G}_{X_n}(F)$: let $\mathcal{V}(G')$ be one of these irreducible components. Then $\mathcal{V}(G')$ is a generic irreducible component of $\mathcal{V}(F)$ also, and because $\mathcal{V}(G') \subseteq \mathcal{G}_{X_n}(F) \cap \mathcal{V}(f) \subseteq \mathcal{V}(f)$, it follows that $f$ is zero on $\mathcal{V}(G')$. Hence $(F, f)$ is not generically false.

Conversely, suppose $(F, f)$ is not generically false. Now $\mathcal{V}(F) \ /=\emptyset$, so $C(F) \ /= \mathbb{Q}[X_n]$. Suppose $f$ is zero on the generic irreducible component $\mathcal{V}(F')$ of $\mathcal{V}(F)$; then $\mathcal{V}(F') \subseteq \mathcal{V}(f)$. If $\mathcal{V}(F)$ has only one irreducible component (namely itself), then $\mathcal{V}(F) = \mathcal{V}(F')$ and then $f$ is zero on $\mathcal{V}(F)$ and so any polynomial $h \ /\in C(F)$ is a non-degenerate side polynomial for $(F, f)$; these exist because $C(F) \ /= \mathbb{Q}[X_n]$. On the other hand, if $\mathcal{V}(F)$ has more than one irreducible component, let $\mathcal{V}(H)$ be the union of the irreducible components of $\mathcal{V}(F)$ other than $\mathcal{V}(F')$. Let $h \in C(H)\backslash C(F')$; such a non-zero $h$ exists since $\mathcal{V}(F') \ /\subseteq \mathcal{V}H)$, so $C(H)$ is not a subset of $C(F')$, and $C(H)$ is non-empty. Further, if $F(a) = 0$ yet $h(a) \ /= 0$ then $a \in \mathcal{V}(F)\backslash\mathcal{V}(H) \subseteq \mathcal{V}(F') \subseteq \mathcal{V}(f)$, so $h \in \mathit{side}(F, f)$. But $h \ /\in C(F')$ and $\mathcal{V}(F') \subseteq \mathcal{G}_{X_n}(F)$, so $h$ does not vanish on $\mathcal{G}_{X_n}(F)$ and so the possible theorem $(F, h)$ is not generically true. Hence $h$ is a non-degenerate side polynomial for $(F, f)$. Thus every side polynomial for $(F, f)$ is degenerate if and only if $(F, f)$ is generically false.

Let $U' = \emptyset$. Then all irreducible components of $\mathcal{V}(F)$ are $U'$-generic, $(F, g)$ is $U'$-generically true if and only if $(F, g)$ is universally true, and a side polynomial is $U'$-degenerate if and only if it is extraneous. Then $(F, f)$ is rarely true if and only if $(F, f)$ is $U'$-degenerately or rarely true if and only if every side polynomial for $(F, f)$ is $U'$-degenerate, that is to say, extraneous.

Finally, there is at least one generically resolving side polynomial for $(F, f)$ yet no generic side polynomials for $(F, f)$ if and only if $(F, f)$ is not generically true and (by the above) neither degenerately nor rarely true, that is, if and only if $(F, f)$ is generically conditionally true. □

## 4 Proof by Refutation and the Kind of Truth

Because of Hilbert's Nullstellensatz, algebraically closed fields are algorithmically convenient to work with, and we frequently assume algebraic closure of $L$ in what follows; moreover all fields have characteristic zero since we work with polynomials over the rational numbers. When $L$ is the field of real numbers, a geometry theorem being true in any of the senses just defined means that it is *true in the theory of Euclidean geometry.* Of course, if a possible theorem is universally true over an algebraically closed field of characteristic zero, then it is true over the complex numbers and hence over the reals also. Although the converse fails, it seems to do so rarely, a fact which apparently generalises to the other kinds of truth as we see in examples to follow. Certainly any side polynomial over algebraically closed $L$ is a side polynomial over the reals also. Thus the assumption that $L$ is algebraically closed is not totally artificial and corresponds to a certain well-defined level of geometrical reasoning which in practice seems only a little weaker than full Euclidean geometry, namely *metric geometry*. We recommend the book [2] to the reader interested in a more detailed account of some of these matters, which have been discussed by many authors.

There are methods which allow one to test whether a given guess is a side polynomial for a possible theorem (and these are considered in detail in [4]), but guessing side polynomials is generally difficult. Furthermore, the existence of (say) a resolving side polynomial for $(F, f)$ does not preclude the existence of generic side polynomials, so the kind of truth is not necessarily established by a correctly guessed side polynomial. Moreover, one can never be sure of having a complete set of side polynomials (so that the disjunction of the associated inequations covers all possibilities for side conditions) using such an approach. A method which is able both to produce a complete set of side polynomials and then to read off the kind of truth of the possible theorem is desirable. It turns out that the set obtained using Kapur's method in [4], based on constructing the Gröbner basis of $F \cup \{fz - 1\}$, does this job.

Recall from Theorem 1 that $C(side(F, f)) = side(F, f)$. We shall call any set of polynomials $G \subseteq \mathbb{Q}[X_n]$ for which $C(G) = side(F, f)$ a *complete set* of side polynomials for $(F, f)$. Then certainly, for any $a \in L^n$, $h(a) = 0$ for all $h \in F$ and $g_1(a) \neq 0 \vee g_2(a) \neq 0 \vee \cdots \vee g_k(a) \neq 0$ imply $f(a) = 0$, and moreover any side polynomial $p$ is such that $p(a) \neq 0$ implies $g_1(a) \neq 0 \vee g_2(a) \neq 0 \vee \cdots \vee g_k(a) \neq 0$. So the disjunction of the side conditions of the form $g_i \neq 0$ is the weakest possible such disjunction. We similarly define a complete set of generic side polynomials for $(F, f)$ to be any finite $G \subseteq \mathbb{Q}[U]$ for which $C(G) \cap \mathbb{Q}[U] = side(F, f) \cap \mathbb{Q}[U]$.

For $F \cup \{f\} \subseteq \mathbb{Q}[X_n]$ and $U$ a non-empty subset of $X_n$, let $(F : f^\infty)_U$ be the ideal $(F \cup \{fz - 1\})_{X_n \cup \{z\}} \cap \mathbb{Q}[U]$. (If $U = X_n$, this is the *saturation* of $f$ with respect to the ideal $(F)_{X_n}$.)

Results similar to the following have already appeared in the literature.

**Theorem 3.** *For the possible theorem $(F, f)$, $side(F, f) = C_{X_n \cup \{z\}}(F \cup \{fz - 1\}) \cap \mathbb{Q}[X_n]$.*

*Proof.* Let $B(F, f) = C_{X_n \cup \{z\}}(F \cup \{fz - 1\}) \cap \mathbb{Q}[X_n]$. The following are equivalent.

- $g \in B(F, f)$;
- $g \in \mathbb{Q}[X_n]$, and if $a \in \mathcal{V}_{X_n}(F)$ and $f(a)b = 1$ for some $b \in L$, then $g(a) = 0$;
- $g \in \mathbb{Q}[X_n]$ and if $a \in \mathcal{V}_{X_n}(F)$ and $f(a) \neq 0$, then $g(a) = 0$;
- $g \in \mathbb{Q}[X_n]$ and if $a \in \mathcal{V}_{X_n}(F)$ and $g(a) \neq 0$, then $f(a) = 0$;
- $g \in side(F, f)$.

Hence $side(F, f) = B(F, f)$. □

**Theorem 4.** *Suppose that $L$ is algebraically closed. Then $side(F, f) = C((F : f^\infty)_{X_n})$, and if a lexicographic order is used in which $z$ is the biggest variable and the variables in $U$ are all ordered below those in $X_n \backslash U$, then $GB(F \cup \{fz - 1\}) \cap \mathbb{Q}[U]$ is a complete set of generic side polynomials for $(F, f)$.*

*Proof.*

$$\begin{aligned}
C_U((F:f)_U) &= C_U((F\cup\{fz-1\})_{X_n\cup\{z\}}\cap\mathbb{Q}[U]) \\
&\subseteq C_{X_n\cup\{z\}}((F\cup\{fz-1\})_{X_n\cup\{z\}}\cap\mathbb{Q}[U])\cap\mathbb{Q}[U] \\
&\subseteq C_{X_n\cup\{z\}}((F\cup\{fz-1\})_{X_n\cup\{z\}})\cap\mathbb{Q}[U] \\
&= C_{X_n\cup\{z\}}(F\cup\{fz-1\})\cap\mathbb{Q}[U] \\
&= \mathit{side}(F,f)
\end{aligned}$$

from Theorem 3. (Note that this part of the argument works even if $L$ is not algebraically closed.)

Conversely, if $L$ is algebraically closed and $g\in \mathit{side}(F,f)\cap\mathbb{Q}[U]$, then by Theorem 3, $g\in C_{X_n\cup\{z\}}(F\cup\{fz-1\})\cap\mathbb{Q}[U]$, so by Hilbert's Nulltellensatz, there exists $n>0$ for which $g^n\in(F\cup\{fz-1\})_{X_n\cup\{z\}}$ and hence $g^n\in C_U((F:f^\infty)_U)$, so by Hilbert's Nullstellensatz, $g\in C_U((F:f)_U)$. Hence $\mathit{side}(F,f)\cap\mathbb{Q}[U]\subseteq C_U((F:f)_U)$ and so $\mathit{side}(F,f)=C_U((F:f)_U)$.

From the theory of Gröbner bases, if a lexicographic order is used in which the variables in $U$ are all ordered below those in $X_n\cup\{z\}\backslash U$, then $GB((F\cup\{fz-1\})\cap\mathbb{Q}[U])=GB(F\cup\{fz-1\})\cap\mathbb{Q}[U]$, which is therefore a generating set for the ideal $(F:f^\infty)_U$ and hence is a complete set of side polynomials for $(F,f)$, since if an ideal $I$ is a complete set of side polynomials for $(F,f)$ then so is any generating set for $I$. □

The following is immediate if one lets $U=X_n$.

**Corollary 1.** *Suppose $L$ is algebraically closed. If a lexicographic order is used in which $z$ is the biggest variable, then $GB(F\cup\{fz-1\})\cap\mathbb{Q}[X_n]$ is a complete set of side polynomials for $(F,f)$.*

This generalises a result in [4], where it was shown that if a side polynomial $h$ consistent with the hypotheses (in other words a non-extraneous side polynomial) exists, then there will be one in $GB(F\cup\{fz-1\})\cap\mathbb{Q}[X_n]$. Our result shows that $GB(F\cup\{fz-1\})\cap\mathbb{Q}[X_n]$ spans all possible side conditions in the natural sense described earlier, and moreover this extends to generic side polynomials.

Suppose $L$ is algebraically closed. Let $G(F,f)=GB(F\cup\{fz-1\})\cap\mathbb{Q}[X_n]$, computed with respect to a fixed lexicographic order in which $z$ is the biggest variable and the variables in $U$ are all ordered below those in $X_n\backslash U$.

Let

$$\begin{aligned}
&G_1=G(F,f)\cap\mathbb{Q}[U],\\
&G_2=\{g\mid g\in G(F,f), g\notin\mathbb{Q}[U], G(F,g)\cap\mathbb{Q}[U]=\emptyset\},\\
&G_3=\{g\mid g\in G(F,f), g\notin\mathbb{Q}[U], G(F,g)\cap\mathbb{Q}[U]\neq\emptyset, G(F,g)\neq\{1\}\},\\
&G_4=\{g\mid g\in G(F,f), g\notin\mathbb{Q}[U], G(F,g)=\{1\}\}.
\end{aligned}$$

We need the condition $g\notin\mathbb{Q}[U]$, in the definitions of $G_2,\ldots,G_4$ in order that the sets $G_i$ are disjoint. $G_1$ consists of a complete set of generic side polynomials as we have just shown, $G_2$ consists of generically resolving side polynomials,

$G_3$ consists of non-extraneous degenerate side polynomials and $G_4$ consists of extraneous side polynomials. Thus the $G_i$ partition $G(F,f)$: $G = \cup_{i=1}^{4} G_i$ and $G_i \cap G_j = \emptyset$ for $i /= j$.

**Theorem 5.** *Suppose $L$ is an algebraically closed field and $U \subseteq X_n$ is a set of independent variables. The possible theorem $(F,f)$ is*

1. *universally true if and only if $G_1 = \{1\}$;*
2. *generically true if and only if $G_1 /= \emptyset$;*
3. *generically conditionally true if and only if $G_1 = \emptyset$, $G_2$ emptyset;*
4. *degenerately true if and only if $G_1 = G_2 = \emptyset$, $G_3 /= \emptyset$;*
5. *rarely true if and only if $G_1 = G_2 = G_3 = \emptyset$, $G_4 /= \emptyset$.*

*Proof.* Since $L$ is algebraically closed, by Corollary 1, $G(F,f) = GB(F \cup \{fz - 1\}) \cap \mathbb{Q}[X_n]$ is a complete set of side polynomials for $(F,f)$. Consequently, for any side polynomial $h$ for $(F,f)$, there exists $k \in G(F,f)$ such that $h(a) /= 0$ implies $k(a) /= 0a \in L^n$. Hence $k$ is zero on no more irreducible components of $\mathcal{V}_{X_n}(F)$ than is $h$.

If $(F,f)$ is generically true, then by Theorem 4, $G_1 /= \emptyset$; the converse is obvious.

Suppose $(F,f)$ is generically conditionally true. Then $G_1 = \emptyset$ as otherwise there would be a generic side polynomial for $(F,f)$. Furthermore, there exists a side polynomial $h$ for $(F,f)$ which is generically resolving. Hence by the above, there exists $k \in G(F,f)$ which vanishes on no more irreducible components of $\mathcal{V}(F)$ than does $h$. Hence $k$ is either generically resolving or generic. It cannot be generic since $G_1 = \emptyset$, so $k \in G_2$ and so $G_2 /= \emptyset$.

Conversely, suppose $G_1 = \emptyset$, $G_2 /= \emptyset$. Then $(F,f)$ is not generically true since $G_1$ is a complete set of generic side polynomials, but there is a generically resolving side polynomial, so $(F,f)$ is generically conditionally true.

Suppose $(F,f)$ is degenerately true. Then as above, $G_1 = G_2 = \emptyset$. Also as above, because there is a degenerate side polynomial for $(F,f)$, there must be one in $G(F,f)$, and so $G_3 /= \emptyset$. Conversely, if $G_1 = G_2 = \emptyset$ and $G_3 /= \emptyset$, then $(F,f)$ is certainly not generically true since $G_1 = \emptyset$, but neither is it generically conditionally true, as if it were, $G_2 /= \emptyset$ from earlier in the proof. Thus $(F,f)$ is degenerately true since there are no generic or generically resolving side polynomials, but there is at least one degenerate side polynomial.

Suppose $(F,f)$ is rarely true. Then, again, $G_1 = G_2 = G_3 = \emptyset$ yet $G_4 /= \emptyset$. Conversely, if $G_1 = G_2 = G_3 = \emptyset, G_4 /= \emptyset$, then, as above, $(F,f)$ is not generically true, nor generically conditionally true, nor degenerately true, and hence is rarely true. □

Thus $G(F,f)$ provides a complete set of side polynomials for $(F,f)$ from which the kind of truth may be determined along with the relevant side polynomials and, with luck, a geometrical interpretation of each such side polynomial, leading to a geometrical side condition for each. Such computations are illustrated in the following section.

## 5 Implementation and Examples

The classification procedure given by Theorem 5 can be attached to a standard implementation of a refutational prover for algebraic geometry theorems. This has been coded in Mathematica using the following algorithm:

1. Translate geometric predicates in the hypotheses and conclusion to algebraic polynomials, giving $F \subseteq \mathbb{Q}[X]$ and $f \in \mathbb{Q}[X]$, respectively. For example, `Collinear[`$a$`,`$b$`,`$c$`]` (meaning that the points $a$, $b$, and $c$ are collinear) translates to the coordinate polynomial

   $$(x[a] - x[b])(y[b] - y[c]) - (y[a] - y[b])(x[b] - x[c]).$$

   Additionally, the construction sequence is used to determine which variables are independent, the elements of $U \subseteq X$, as described in [2].
2. Compute the Gröbner basis $G(F, f) = GB(F \cup \{fz - 1\}) \cap \mathbb{Q}[X]$, removing the polynomials involving $z$.
3. Split $G(F, f)$ into the four sets $G_1, G_2, G_3, G_4$, as defined in the preamble to Theorem 5. This involves additional Gröbner basis computations.
4. If $G_1 = \{1\}$ then return `True`. Otherwise, attempt to translate the polynomials back into geometric predicates using pattern matching. (If no predicate can be determined then the polynomial is returned for inspection by the user.)

The Mathematica code for our implementation can be downloaded from

`http://www.maths.utas.edu.au/People/dfs/dfs.html`

We now give some simple examples to show the kinds of truth of geometry theorems and the results of this algorithm.

### Parallel Pappus

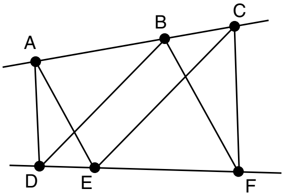


**Fig. 1.** The Parallel Pappus Theorem.

The following is a famous theorem of Pappus:

```
Hyps[Pappus] = { Collinear[A,B,C], Collinear[D,E,F],
                 Parallel[A,E,B,F], Parallel[B,D,C,E] };
Conc[Pappus] = Parallel[A,D,C,F];
```

The function `Prove[`$F$`,`$f$`]` returns the kind of truth of the possible theorem $(F, f)$ as a 4-tuple $\{generic, conditional, degenerate, extraneous\}$ of sets of equations. In the case where the first of these equals $\{1\}$ with all others empty, the output is rendered as `True`.

```
Prove[Hyps[Pappus],Conc[Pappus]]

True
```

Thus this possible theorem is universally true. Any instance of the hypotheses is an instance of the conclusion, without restriction. Universal truth is an uncommonly strong property of possible theorems. It means that the entailment holds for arbitrary choices of the points. For example, the Parallel Pappus theorem holds even when all the points are collinear.

### Collinearity Theorem

Consider the following statement carefully:

```
Hyps[CollinearityThm] = { Collinear[A,B,C], Collinear[A,B,D] };
Conc[CollinearityThm] = Collinear[B,C,D];
```

Here the prover chooses $U = \{x_A, y_A, x_B, y_B, x_C, x_D\}$ from the construction and computes

$$G(F, f) = \{-y_A + y_B, x_A - x_B\}.$$

Each polynomial is in $\mathbb{Q}[U]$ and so $G_1 = G(F, f)$, $G_2 = G_3 = G_4 = \emptyset$. Invoking `Prove` gives the partition of $G(F, f)$ and interprets the polynomials:

```
Prove[Hyps[CollinearityThm],Conc[CollinearityThm]]

{{!Identical[A, B]}, {}, {}, {}}
```

Thus we have a generic side polynomial for the possible theorem. The theorem is generically true, and the associated side condition asserts that points `A` and `B` are not identical. The prover has established that `!Identical[A, B]` is a weakest possible (generic) side condition: any other side condition is at least as strong. (Readers should draw a diagram for the case where `A` and `B` coincide to see the problem.)

### Parallelogram Theorem

The following possible theorem says that the diagonals of a parallelogram bisect each other.

```
Hyps[ParallelogramThm] = { Parallel[A,B,D,C],
  Parallel[D,A,C,B], Collinear[O,B,D], Collinear[O,A,C] };
Conc[ParallelogramThm] = EqualLength[A,O,O,C];
```

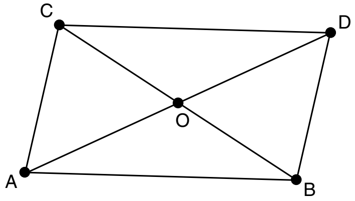


**Fig. 2.** The Parallelogram Theorem.

Here the predicate `EqualLength[`$a$`,`$b$`,`$c$`,`$d$`]` says that the line segments $ab$ and $cd$ are of equal length.

The prover chooses $U = \{x_A, y_A, x_B, y_B, x_C, y_C\}$ from the construction and computes

$$G(F, f) = \left\{ \begin{array}{l} -x_B y_A + x_C y_A + x_A y_B - x_C y_B - x_A y_C + x_B y_C, \\ x_C y_B - x_O y_B - x_B y_C + x_O y_C + x_B y_O - x_C y_O, \\ -x_C y_A + x_O y_A + x_A y_C - x_O y_C - x_A y_O + x_C y_O, \\ x_C y_B - x_D y_B - x_B y_C + x_D y_C + x_B y_D - x_C y_D, \\ -x_C y_A + x_D y_A + x_A y_C - x_D y_C - x_A y_D + x_C y_D, \\ -x_D y_C + x_O y_C + x_C y_D - x_O y_D - x_C y_O + x_D y_O \end{array} \right\}.$$

Here

$$\begin{aligned} G_1 &= \{-x_B y_A + x_C y_A + x_A y_B - x_C y_B - x_A y_C + x_B y_C\}, \\ G_2 &= \left\{ \begin{array}{l} x_C y_B - x_O y_B - x_B y_C + x_O y_C + x_B y_O - x_C y_O, \\ x_C y_B - x_D y_B - x_B y_C + x_D y_C + x_B y_D - x_C y_D, \\ -x_C y_A + x_D y_A + x_A y_C - x_D y_C - x_A y_D + x_C y_D, \\ -x_D y_C + x_O y_C + x_C y_D - x_O y_D - x_C y_O + x_D y_O \end{array} \right\}, \\ G_3 &= \emptyset, \\ G_4 &= \{-x_C y_A + x_O y_A + x_A y_C - x_O y_C - x_A y_O + x_C y_O\}. \end{aligned}$$

Again `Prove` gives this partition and interprets the side conditions:

```
Prove[Hyps[ParallelogramThm],Conc[ParallelogramThm]]

{{!Collinear[A,B,C]},
 {!Collinear[A,C,D],!Collinear[B,C,D],!Collinear[B,C,O],
 !Collinear[C,D,O]},
 {},
 {!Collinear[A,C,O]}}
```

This theorem is generically true: the associated side condition states that the points `A`, `B`, and `C` are not collinear, and again, any other generic side condition is at least as strong as this one. We also have five non-generic side conditions. One of these, the side condition `Collinear[A, C, O]` is a consequence of the hypotheses and hence is *extraneous*. The remaining four side conditions are generically resolving.

### Isosceles Theorem

In this example, `EqualAngle[`$a, b, c, d, e, f$`]` says that the angle $\angle abc$ is equal to the angle $\angle def$.

```
Hyps[IsoscelesThm] = {EqualAngle[A,B,C,C,A,B]};
Conc[IsoscelesThm] = EqualLength[A,C,B,C];
```

We obtain

```
Prove[Hyps[IsoscelesThm],Conc[IsoscelesThm]]

{{}, {!Collinear[A, B, C]}, {}, {}}
```

So the possible theorem is generically conditionally true. The conditional predicate `!Collinear[A, B, C]` identifies which of the two generic components gives a theorem.

### A Rarely True Theorem

Rarely true theorems are not of great interest, but here is an example. Letting `Midpoint[`$a$`,`$b$`,`$c$`]` be the predicate that the midpoint of the line segment between $a$ and $b$ is the point $c$, we obtain the following:

```
Hyps[NonThm] = {Midpoint[A,B,C]};
Conc[NonThm] = Midpoint[A,C,B];

Prove[{Midpoint[A,B,C]},Midpoint[A,C,B]]

{{}, {}, {}, {!Midpoint[A,B,C]}}
```

Thus any side conditions for the possible theorem are at least as strong as the negation of the hypothesis! That is, there is no component of the hypothesis on which the conclusion holds. The theorem fails to hold in a most comprehensive way.

## 6 Conclusion

Universal truth was considered in [2,3], as was the Gröbner basis characterisation given above. Universal and conditional truth were also considered by Wang [8]. The definitions of generic truth and non-degeneracy conditions originate with Wu [9,10], and have been considered also by Chou in [2], where a variant on the Gröbner basis method using fields of rational functions is featured. Conditional truth in general (meaning neither universal truth nor rare truth) was considered in [3] along with the Gröbner basis method of proof. Generically conditional truth was considered in [2] though no Gröbner basis method was given. In [1], two strengths of generic truth were defined in terms of the highest dimension irreducible components of the hypothesis variety, an approach often giving a different notion of generic truth to the one used here and one which Chou argues in [2] is not always the one intended by the user. The notion of a complete set of side polynomials, though hinted at in [3], seems not to have been explicitly considered elsewhere.

More recently, the article [7] takes a similar approach to ours, in that a possible theorem $(F, f)$ is classifiable as universally true (called "geometrically true" in [7]), generically true, neither generically true nor generically false (generically conditionally true in our terms), and generically false. However, this is done by computing with both (elimination ideals generated by) $F \cup \{fz - 1\}$ and $F \cup \{f\}$, whereas our approach considers only Gröbner bases of the former kind of set (that is, side polynomial calculations). The approach in [7] does not seem to be able to provide information in the generically conditionally true case (other than to flag the need for a decomposition), whereas our approach is able to provide side polynomials which eliminate the generic irreducible components on which the conclusion fails to hold. An advantage of the approach in [7] is the possibility of generating additional hypotheses of equational type (rather than just inequations) in the generically false case, although the approach is not guaranteed to do this. Nonetheless, it would be possible to use a combination of the approach in [7] and our approach in such cases: first, that a theorem is generically false could be established using our approach, and then $F \cup \{f\}$ could be considered in an attempt to obtain sufficient additional equational hypotheses.

The main contribution of the current work is to bring together facts which show that a single Gröbner basis calculation for $F \cup \{fz - 1\}$ yields a complete set of side polynomials $\{g_1, g_2, \ldots, g_k\}$ for the possible theorem $(F, f)$, and moreover that this (plus perhaps similar calculations of Gröbner bases for some of the $F \cup \{g_i z - 1\}$) is all that is needed to classify the kind of truth of the theorem and to provide the appropriate complete set of side conditions.

**Acknowledgements**

We thank Professor Giuseppa Carrà Ferro for advice on the paper. The comments of anonymous referees were also very helpful. The work reported here was partially supported by Australian Research Council Large Grants A49132001 and A49331346.

## References

1. Carrà Ferro, G. and Gallo, G.: A Procedure to Prove Geometrical Statements, *Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, AAECC-5* (L. Huguet and A. Poli, eds.), Lecture Notes in Computer Science 356, Springer-Verlag (1989), 141–150.
2. Chou, S.-C.: *Mechanical Geometry Theorem Proving*, D. Reidel (1988).
3. Kapur, D.: Geometry Theorem Proving Using Hilbert's Nullstellensatz, *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation* (Waterloo, July 21–23, 1986), ACM Press, 202–208.

4. Kapur, D.: A Refutational Approach to Theorem Proving in Geometry, *Artificial Intelligence* 37 (1988), 61–93.
5. Kutzler, B. and Stifter, S.: Automated Geometry Theorem Proving Using Buchberger's Algorithm, *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation* (Waterloo, July 21–23, 1986), ACM Press, 209–214.
6. Recio, T., Sterk, H. and Velez, M.: Automatic Geometry Theorem Proving, *Some Tapas of Computer Algebra* (A. M. Cohen, H. Cuipers and H. Sterk, eds.), Algorithms and Computation in Mathematics 4, Springer-Verlag (1999), 276–296.
7. Recio, T. and Velez, M.: Automatic Discovery of Theorems in Elementary Geometry, *Journal of Automated Reasoning* 23 (1999), 63–82.
8. Wang, D.: Elimination Procedures for Mechanical Theorem Proving in Geometry, *Annals of Mathematics and Artificial Intelligence* 13 (1995), 1–24.
9. Wu, W.-t.: On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry, *Scientia Sinica* 21 (1978), 157–179.
10. Wu, W.-t.: *Mechanical Theorem Proving in Geometries: Basic Principles*, Springer-Verlag (1994).

# A Complex Change of Variables for Geometrical Reasoning

Tim Stokes[1] and Michael Bulmer[2]

[1] Murdoch University, Western Australia, Australia
stokes@prodigal.murdoch.edu.au
[2] University of Queensland, Queensland, Australia
mrb@maths.uq.edu.au

**Abstract.** We use complex vectors in geometrical reasoning, specifically automated theorem proving. The calculations are embedded in Clifford algebras, but commutative polynomial techniques can be used. Using the Gröbner basis package in the computer algebra package *Maple*, this approach is shown to have efficiency benefits in situations where distance and angle relations amongst geometrical entities predominate.

## 1 Introduction

The connection between the complex numbers with conjugation, $\mathbf{C}$, and the orthogonal transformations of $\mathbf{R}^2$ is well-known, and leads to an algebraic representation of geometrical quantities involving plane vectors. The key to this is that the inner product of two plane vectors $u, v \in \mathbf{C}$ is given by $(u\bar{v} + \bar{u}v)/2$ where $\bar{u}$ is the usual complex conjugate of $u$. Thus commonly used geometrical relations may be expressed in terms of this complex number formalism, and the algebraic forms are often more natural and succinct than the corresponding forms involving coordinates. We shall exploit this natural advantage for the purpose of automated reasoning about geometrical configurations.

We give below a comparison of three formalisms in expressing a number of basic geometrical relations and quantities. A point $A$ will have *coordinate representation* $(a_1, a_2) \in \mathbf{R}^2$ and *vector representation* $a = a_1 + a_2 i \in \mathbf{C}$. One expression can be obtained from the other using the relations

$$a \leftrightarrow a_1 + a_2 i, \quad \bar{a} \leftrightarrow a_1 - a_2 i,$$

$$a_1 \leftrightarrow \frac{1}{2}(a + \bar{a}), \quad a_2 \leftrightarrow \frac{1}{2i}(a - \bar{a}),$$

and dividing through by $i$ if necessary (although in practice, this last step is scarcely ever needed). This is not simply a linear change of variables since the imaginary unit lies outside the scalar domain.

We also consider a *Clifford algebra representation* $A = a_1 X + a_2 Y$, where $X, Y$ are orthogonal unit vectors ($X^2 = Y^2 = 1$ and $XY = -YX$) in the rank two real Clifford algebra.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 143–153, 2001.

The complex vector representation can also be thought of as derived from the Clifford algebra representation; indeed the complex numbers are naturally embedded in the rank two real Clifford algebra, by virtue of the fact that the subalgebra of bivectors (which contains all reals as squares of vectors) is isomorphic to the complex numbers (see [7] for the details of the gemoetrical significance of this in terms of angles). Thus an arbitrary vector $X$ of length 1 ($X^2 = 1$ in the algebra) is chosen, and for any other vector $A$, the product $XA$ is represented by $a \in \mathbf{C}$ while $AX$ is represented by $\bar{a} \in \mathbf{C}$. Since $X^2 = 1$, all expressions can be rewritten to allow this transformation: for example, $AB$ is rewritten as $AXXB$ which becomes $\bar{a}b$ in the complex representation.

## 2 Geometrical Statements in Various Formalisms

The following list gives examples of geometrical quantities and relations expressed in the three formalisms. The standard coordinate form is given first in each case, followed by the Clifford form and then the complex vector form. Of interest is the number of terms in each polynomial.

1. The length of segment $AB$.
   (a) $(a_1 - b_1)^2 + (a_2 - b_2)^2$, yielding an expression consisting of six terms on expansion.
   (b) $(A - B)^2$, yielding four terms on expansion.
   (c) $(\bar{a} - \bar{b})(a - b)$, yielding four terms on expansion.
2. The tangents of the angles $ABC$ and $DEF$ are equal.
   (a) $[(a_2-b_2)(c_1-b_1)-(c_2-b_2)(a_1-b_1)][(d_1-e_1)(f_1-e_1)+(d_2-e_2)(f_2-e_2)] = [(d_2-e_2)(f_1-e_1)-(f_2-e_2)(d_1-e_1)][(a_2-b_2)(c_2-b_2)+(a_1-b_1)(c_1-b_1)]$, which, after expansion and simplifications, yields an equation consisting of 96 terms.
   (b) $(B - C)(A - B)(D - E)(E - F) - (A - B)(B - C)(E - F)(D - E)$, an expression in 30 terms.
   (c) $(\bar{a} - \bar{b})(b - c)(d - e)(\bar{e} - \bar{f}) - (a - b)(\bar{b} - \bar{c})(\bar{d} - \bar{e})(e - f) = 0$, which ultimately yields an expression in 30 terms.
3. The product of two squares of segment lengths, $AB^2 \cdot CD^2$.
   (a) $[(a_1 - b_1)^2 + (a_2 - b_2)^2][(c_1 - d_1)^2 + (c_2 - d_2)^2]$, which expands to 36 terms.
   (b) $(A - B)^2(C - D)^2$, an expression in 16 terms.
   (c) $(\bar{a} - \bar{b})(a - b)(\bar{c} - \bar{d})(c - d)$, yielding 16 terms.
4. Line $AC$ is parallel to line $BD$.
   (a) $(c_2 - a_2)(d_1 - b_1) - (d_2 - b_2)(c_1 - a_1) = 0$, which has 8 terms upon expansion.
   (b) $(C - A)(D - B) - (D - B)(C - A)$, 8 terms.
   (c) $(\bar{a} - \bar{c})(b - d) - (a - c)(\bar{b} - \bar{d}) = 0$, 8 terms.
5. Point $A$ lies on line $BC$.
   Use the previous case with $AB$ parallel to $BC$, giving expressions with 6 terms for each formulation.

6. Line $AC$ is perpendicular to line $BD$.
   (a) $(c_2 - a_2)(d_2 - a_2) + (c_1 - a_1)(d_1 - a_1) = 0$, yielding an expression with 8 terms.
   (b) $(C - A)(D - B) + (D - B)(C - A)$, 8 terms.
   (c) $(\bar{a} - \bar{c})(b - d) + (a - c)(\bar{b} - \bar{d}) = 0$, again an 8 term expression.
7. $A$ is the midpoint of $B$ and $C$.
   (a) $a_1 - (b_1 - c_1)/2 = 0$, $a_2 - (b_2 - c_2)/2 = 0$.
   (b) A = (B+C)/2.
   (c) $a - (b + c)/2 = 0$, hence also $\bar{a} - (\bar{b} + \bar{c})/2 = 0$.

It is evident that the advantages of the Clifford algebra vector formalism over the usual coordinate approach in terms of succinctness are also possessed by the complex vector formalism, even though this is a commutative formalism. However, the use of conjugation means that essentially twice as many variables are needed when using complex vectors as compared to Clifford vectors, the same as for the coordinatised versions.

In the following sections we compare the standard coordinate formalism with the complex vector formalism. In particular, in Sections 5 and 6 we describe implementations and give timing results of proofs using the two representations. This is appropriate since, having expressed a theorem in the respective formalism, both approaches use the same algorithm to carry out the proof. This is not the case for the Clifford representation; we leave discussion of this approach to later work.

Throughout, we shall be interested in the solution of various problems using the Gröbner basis approach [1]. For geometrical theorem proving, an alternative method is Wu's [16,15]. Wu's method was the first approach to geometrical theorem proving using polynomial algorithms, and is essentially equivalent to the Gröbner basis approach for this class of problems as is demonstrated in [2]. We anticipate the same sorts of efficiency improvements using Wu's prover as occur using the Gröbner basis approach. However, the Gröbner basis approach is more readily generalised to non-commutative algebras, and packages based on Buchberger's Gröbner basis algorithm are in wider use than are implementations of Wu's algorithm. We use *Maple*'s built-in *Gröbner* package in the computed examples.

## 3 Viewing the Complex Vector Formalism as a Change of Coordinates

Geometry theorems may be expressed in terms of polynomial algebra, using coordinatisation. Once a field of scalars $\mathbf{K}$ is fixed, the hypotheses and conclusion of the theorem correspond to certain polynomials in the coordinates of points being zero. In standard methods, the field $\mathbf{K}$ is assumed to be algebraically closed in order to ensure completeness, since then Hilbert's Nullstellensatz can be used. Typically one takes $\mathbf{K}$ to be $\mathbf{C}$, the complex number field, which has the effect of replacing the real Euclidean plane $\mathbf{R}^2$ by $\mathbf{C}^2$.

The ramifications of working with $\mathbf{C}^2$ rather than $\mathbf{R}^2$ are explored fully in [2], but essentially the difference is that one proves theorems of *metric geometry*, a generalisation of Euclidean geometry with the property that a theorem is true in metric geometry if and only if the corresponding coordinatised algebraic theorem is true over any algebraically closed field of characteristic zero. As discussed in [2], in practice little is lost in working with $\mathbf{C}$ rather than $\mathbf{R}$ .

In the present context, working with $\mathbf{C}$ rather than $\mathbf{R}$ has special significance. Letting $i$ be the imaginary unit in $\mathbf{C}$, the invertible linear transformation $\theta : \mathbf{C}^2 \to \mathbf{C}^2$ given by $(a_1, a_2) \mapsto (a, \bar{a})$, where $a = a_1 + a_2 i$ and $\bar{a} = a_1 - a_2 i$, induces an invertible linear change of variables $\psi$ from the polynomial ring $\mathbf{C}[x_1, y_1, x_2, y_2, \ldots, x_n, y_n]$ to $\mathbf{C}[v_1, \bar{v}_1, v_2, \bar{v}_2, \ldots, v_n, \bar{v}_n]$ in which $v_j = x_j + iy_j$ and $\bar{v}_j = x_j - iy_j$ for each $j$. (Note that for any $(a_1, a_2) \in \mathbf{C}^2$, the complex conjugate of $a_1 + ia_2$ is *not* $a_1 - ia_2$ unless $a_1, a_2 \in \mathbf{R}$. However, we retain the over-bar notation at the risk of some confusion.)

Thus, writing $f'_j = \psi(f_j)$ and $g' = \psi(g)$, if a theorem's hypotheses are captured by the equations $f_1 = 0, f_2 = 0, \ldots, f_k = 0$ and the conclusion by $g = 0$, then in the transformed coordinates, these become $f'_1 = 0, f'_2 = 0, \ldots, f'_k = 0$ and $g' = 0$. The second set of equations are all expressed in terms of the complex vector formalism, so the more efficient algebraic forms given in the first section will feature. Hence the conversion between formalisms can be performed *internally*, with no need to introduce an imaginary unit $i$: it is already part of the field. The significance of this is that the complex vector formulation can be viewed simply as an alternative coordinate-based formulation. In a theorem proving context, this will mean that all the usual cooordinate-based methods apply in full.

## 4 Generic Truth and Gröbner Bases

The standard notion of generic truth, discussed at length by Chou in [2] and in many places since, turns out to be the appropriate one for most geometry theorems encountered in practice. (Most "theorems" turn out to be false due to degenerate cases not easily excluded by the user.) This approach requires that certain of the variables in a coordinatised possible theorem be specified as *parameters*. The idea is that these coordinates can be chosen in a mechanical way on the basis of the ordering of the hypotheses. This ordering contains implicit information about a notional "construction" associated with a theorem statement: the first-mentioned points are chosen "arbitrarily", or in *general position*, and then subsequent points are either partially or wholly constrained by the theorem hypotheses. In this formulation, theorems are proved to be "generically true" in a sense made rigorous in [15,2]. The idea is that one is only concerned with cases of the hypotheses in which the parameters are algebraically independent, and Chou gives strong arguments as to why this is a logically and geometrically sound approach. Algorithmically, using rational expression fields is shown in [2] to be a very efficient way of checking generic truth,and also allows one to obtain *non-degeneracy conditions*, sufficient for the truth of the theorem and expressible as algebraic inequations.

Issues of genericity are readily dealt with in our scheme: if a point is in general position, both it and its "conjugate" are viewed as parameters and added to the coefficient field for purposes of computations. If the point is viewed as totally dependent, both it and its "conjugate" are added to the variable set. If a point represented by $x$ in the complex vector formalism has a single notional degree of freedom (for instance, if it is constrained to lie on a particular line or circle), then $x$ is made a parameter but not $\bar{x}$. (Note that this would make no sense unless we viewed both $x$ and $\bar{x}$ as separate and independent complex scalars rather than as a vector in the complex plane and its conjugate.) All other variables are left as polynomial variables. Once this is done, the methods applying to the standard formulation also apply here.

The results which are the basis of provers of generic truth are well known and we state them here. Let

$$F = \{f_1, f_2, \ldots, f_k\} \subseteq \mathbf{K}(m)[n] = \mathbf{K}(u_1, u_2, \ldots, u_m)[x_1, x_2, \ldots, x_n],$$

and let $GB(F)$ be the reduced Gröbner basis of $F$ in $\mathbf{K}(m)[n]$ with respect to a fixed admissible term ordering on the monomials in the $x_i$. With the definition of generic truth given in [2], we have the following extension of an approach first introduced by Kapur (see [9,10] where this *refutational* approach is discussed in detail), and extended to the generic context via the use of fields of rational expressions in [2].

**Theorem 1.** *1. If $g \in GB(F)$, then the possible theorem represented by the algebraic implication $f_1 = f_2 = \cdots = f_k = 0 \Rightarrow g = 0$ is generically true.*

*2. The possible theorem $f_1 = f_2 = \cdots = f_k = 0 \Rightarrow g = 0$ is generically true if and only if $GB(F \cup \{gx_{n+1} - 1\}) = \{1\}$.*

In our computed examples, only the first part of Theorem 1 is needed in order to verify the generic truth of theorems, reflecting the experience of Chou in [2].

Thus our complex vector approach is really a coordinate approach in disguise. This contrasts with genuinely vector approaches, such as those of Chou, Gao and Zhang [3,4,5], in which theorems of two and three dimensional geometry are proved using a radically different formalism based on areas and volumes, allowing higher level interpretation of the resultant proofs. Similarly, the point-based algebraic approach to automated reasoning in projective geometry featured in the work of White [14], Sturmfels and Whitely [12] uses a fundamentally different algebraic algorithm, Cayley factorization in Cayley algebras, to do reasoning in projective geometry. In [8], Fearnley-Sander and Stokes use a variant of Buchberger's algorithm which applies to Grassmann algebras to do automated reasoning in affine geometry. Wang [13] demonstrated that Clifford algebras with genuine algebraic term rewriting could be used for a range of geometrical reasoning. A summary of recent advances in the use of Clifford representations is given in [11]. The complex translation related to Clifford algebra, mentioned in the Introduction, also contrasts with the rule-based methods [17] that have been used for reasoning with the Clifford formalism.

## 5 Implementation

A feature of Chou's book [2] is the very large number of theorems proved mechanically using the methods described in the first half of the book. We used the implementation of the Buchberger algorithm built into the computer algebra package *Maple V*, running on a SPARC Station 1 (Sun 4/60), to test the advantages of our approach for a number of the theorems considered in [2]. The choice was not random: a theorem was chosen if it featured many distance or angle relations, a situation we might expect to be better handled by the complex vector formulation. However, we have included all results obtained, and we believe that the process of identifying theorems that are likely to be more efficiently proved by our technique could be fully automated.

We made no attempt to obtain non-degeneracy conditions, only to prove generic truth. The *Maple* package does not permit the recording of polynomials which arise as denominators during a computation. However, a package such as the one referred to in [2], which does this and is able to interpret non-degeneracy conditions geometrically, could easily be modified to do these things for polynomials in the complex vector formalism. The sole object here has been to show that our change of variables leads to faster processing for a wide class of easily identified theorems.

We remark by the way that some other types of theorems from [2] were tested, in particular some purely affine theorems. Here, the results if anything favoured the usual coordinate-based approach, but not markedly. No advantage was anticipated from the use of the transformed system for such problems.

Given a finite set of geometrical hypotheses $H_1, H_2, \ldots, H_k$ and a single conclusion $C$ in the points $P_1, P_2, \ldots, P_n$, the procedure was as follows:

1. Convert each $H_i$ to a polynomial $f_i(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ using the conversion rules of the first section; similarly convert $C$ to $g$.
2. Select independent points $A_1, A_2, \ldots, A_r$, semidependent points $A_{r+1}, A_{r+2}, \ldots, A_s$ and dependent points $A_{s+1}, A_{s+2}, \ldots, A_n$.
3. Compute $G = GB(f_1, f_2, \ldots, f_k)$ using the total degree order with $y_{r+1} < y_{r+2} < \cdots < y_s < y_{s+1} < x_{s+1} < y_{s+2} < x_{s+2} < \cdots < y_n < x_n$.
4. Reduce $g$ using $G$.

Only the third and fourth stages take a significant amount of processing time, and the times given in Table 1 are for the total of those parts only.

The procedure for the complex vector formulation was the same, the only difference being the rules for converting geometrical predicates into polynomials.

For the standard coordinate approach, the choice of which variables were parameters was specified in [2], having been obtained automatically by Chou's prover, and we used the same choice. Determining parameters for the corresponding complex vector formulation of the theorem was done according to the choice used for the standard formulation. For testing purposes, we used a total degree order for both formalisms, whereas [2] used a lexicographic order, so the choice of order was less than optimal for the coordinatised system, and likewise for the corresponding complex vector formulation. Also, the variable ordering choice in [2] was predicated on subsequent use by Wu's method, and a slightly different variable ordering may have been better for maximising the efficiency of the Gröbner basis method.

There were some refinements on the above algorithm that should be mentioned. Some efficiency gains were achieved using the standard variables by choosing axes conveniently, as was done in [2] as a matter of course. For instance, the first point could be chosen as the origin and the second as on the $X$-axis, of the form $(x, 0)$; thus three of the parameters were set to zero, without affecting the generic truth of each possible theorem. Mostly these gains could be carried over to the complex vector representation: for instance, a vector $x$ on the $X$-axis satisfies $x = \bar{x}$. However, using standard coordinates also permits the assertion of collinearity and perpendicularity conditions by a careful choice of axes, a technique not possible in the complex vector formalism. For example, the line from $(x_1, x_2)$ to $(x_1, x_3)$ is parallel to the line $(0, 0)$ to $(0, u_1)$, a fact which must be expressed equationally using the complex vector variables.

Because of the reducibility of the coordinate-based algebraic formulation of a theorem, ambiguities in the geometrical interpretation occasionally arise. These are dealt with in [2] by introducing certain subsidiary equations which rule out the unwanted possibilities. Where this arose in our examples, we included a complex vector version of each such subsidiary polynomial. The resultant complex vector polynomials tend to be slightly less simple than the original ones, but the advantage of the method for such examples was still clear.

Were it not for these various refinements, it would be possible to convert directly between the standard and complex vector formulations using the substitution scheme

$$x_j \mapsto (u_j + v_j)/2, \quad y_j \mapsto (u_j - v_j)/2i$$

for converting from standard to complex vector variables, and

$$u_j \mapsto x_j + iy_j, \quad v_j \mapsto x_j - iy_j$$

for the inverse process.

## 6 Brianchon's Theorem for Circles

We will illustrate the potential computational benefit from the complex vector formulation using a collection of examples from [2]. We will show the details of the application to Brianchon's theorem for circles (Example 19 in [2]), with the remaining timing results summarised in Section 7.

Brianchon's theorem is illustrated in Figure 1. The theorem states that if $A$, $B$, $C$, $D$, $E$, $F$ are six points on a circle then the three lines through the opposite verticies of the hexagon formed by the tangent lines to the circle at these points are concurrent.

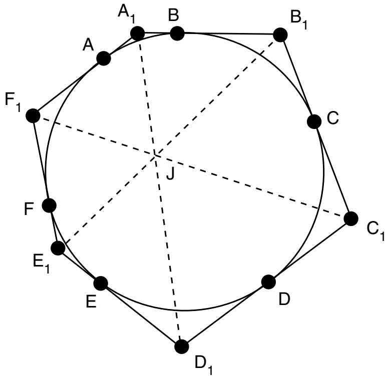


**Fig. 1.** Brianchon's Theorem.

We will start by showing the standard conversion into polynomials using the coordinate approach. We construct the hypothesis polynomials by starting with point $O$, the centre of the circle, at the origin and point $A$ on the horizontal axis. The remaining points, including the vertices of the hexagon, $A_1$, $B_1$, $C_1$, $D_1$, $E_1$, and $F_1$, are constructed (in order) as follows.

Firstly, we fix the points on the circle by setting $BO \equiv OA$, $CO \equiv OA$, $DO \equiv OA$, $EO \equiv OA$, and $FO \equiv OA$. This large number of equal-length relationships will be naturally suited to the complex vector method since each once can be captured in fewer terms, as seen in Section 2. Secondly, we construct the hexagon's vertices using the tangent relationships $A_1B \perp BO$, $A_1A \perp AO$, $B_1C \perp CO$, $B_1B \perp BO$, $C_1D \perp DO$, $C_1C \perp CO$, $D_1E \perp EO$, $D_1D \perp DO$, $E_1F \perp FO$, $E_1E \perp EO$, and $F_1F \perp FO$, and (for simplicity) saying $F_1$ is on line $A_1A$. Finally we describe the point of intersection, $J$, by saying $J$ is on line $B_1E_1$ and on line $A_1D_1$. The conclusion is then that points $C_1, F_1$ and $J$ are collinear.

This construction gives the following coordinate representation of the points: $O = (0,0)$, $A = (u_1, 0)$, $B = (x_1, u_2)$, $C = (x_2, u_3)$, $D = (x_3, u_4)$, $E = (x_4, u_5)$, $F = (x_5, u_6)$, $A_1 = (u_1, x_6)$, $B_1 = (x_8, x_7)$, $C_1 = (x_{10}, x_9)$, $D_1 = (x_{12}, x_{11})$, $E_1 = (x_{14}, x_{13})$, $F_1 = (u_1, x_{15})$, $I = (x_{17}, x_{16})$.

The list of hypothesis polynomials (each notionally set equal to zero) is then

$$F = \left\{ \begin{array}{l} x_1^2 + u_2^2 - u_1^2, x_2^2 + u_3^2 - u_1^2, x_3^2 + u_4^2 - u_1^2, x_4^2 + u_5^2 - u_1^2, x_5^2 + u_6^2 - u_1^2, \\ x_1u_1 - x_1^2 + x_6u_2 - u_2^2, x_8x_2 - x_2^2 + x_7u_3 - u_3^2, \\ x_8x_1 - x_1^2 + x_7u_2 - u_2^2, x_{10}x_3 - x_3^2 + x_9u_4 - u_4^2, \\ x_{10}x_2 - x_2^2 + x_9u_3 - u_3^2, x_{12}x_4 - x_4^2 + x_{11}u_5 - u_5^2, \\ x_{14}x_5 - x_5^2 + x_{13}u_6 - u_6^2, x_{14}x_4 - x_4^2 + x_{13}u_5 - u_5^2, \\ x_5u_1 - x_5^2 + x_{15}u_6 - u_6^2, x_{12}x_3 - x_3^2 + x_{11}u_4 - u_4^2, \\ -x_{16}x_{14} - x_7x_{17} + x_7x_{14} + x_{16}x_8 + x_{13}x_{17} - x_{13}x_8, \\ -x_{16}x_{12} - x_6x_{17} + x_6x_{12} + x_{16}u_1 + x_{11}x_{17} - x_{11}u_1 \end{array} \right\},$$

with conclusion polynomial

$$g = x_9u_1 - x_9x_{17} + x_{15}x_{17} - x_{15}x_{10} + x_{16}x_{10} - x_{16}u_1.$$

We find the Gröbner basis using the total degree term order with $x_1 < x_2 < \cdots < x_{17}$. Generating the Gröbner basis of the hypotheses and reducing the conclusion to zero took in excess of 5000 seconds.

The complex vector formulation proceeds similarly but the refinements made in the coordinate case will not necessarily work in the new setting. For example, we can still choose $O = (0, 0)$ but to specify that $A$ is on the horizontal axis we use $A = (u_1, u_1)$, saying that the point and its conjugate are the same. Similarly, in the coordinate case we could set the point $A_1$ and $F_1$ by noting that both would have the same $X$ coordinate as $A$. This is not possible in the complex case and so we must in troduce new dependent variables $x'_5$ and $x'_14$, setting $A_1 = (x'_5, x_6)$ and $F_1 = (x'_{14}, x_{15})$, and also the explicit perpendicularity and collinearity constraints that define where they are.

Thus there are now 19 hypothesis polynomials

$$F_{\mathbf{C}} = \left\{ \begin{array}{l} x_1u_2 - u_1^2, x_2u_3 - u_1^2, x_3u_4 - u_1^2, x_4u_5 - u_1^2, x_5u_6 - u_1^2, \\ \frac{1}{2}x'_5u_2 + \frac{1}{2}x_6x_1 - x_1u_2, \frac{1}{2}u_1x'_5 + \frac{1}{2}u_1x_6 - u_1^2, \frac{1}{2}x_8u_3 + \frac{1}{2}x_7x_2 - x_2u_3, \\ \frac{1}{2}x_8u_2 + \frac{1}{2}x_7x_1 - x_1u_2, \frac{1}{2}x_{10}u_4 + \frac{1}{2}x_9x_3 - x_3u_4, \\ \frac{1}{2}x_{10}u_3 + \frac{1}{2}x_9x_2 - x_2u_3, \frac{1}{2}x_{12}u_5 + \frac{1}{2}x_{11}x_4 - x_4u_5, \\ \frac{1}{2}x_{14}u_6 + \frac{1}{2}x_{13}x_5 - x_5u_6, \frac{1}{2}x_{14}u_5 + \frac{1}{2}x_{13}x_4 - x_4u_5, \\ \frac{1}{2}x'_{14}u_6 + \frac{1}{2}x_{15}x_5 - x_5u_6, \frac{1}{2}x_{12}u_4 + \frac{1}{2}x_{11}x_3 - x_3u_4, \\ x_8x_{13} + x_{16}x_{14} - x_{17}x_{13} + x_7x_{17} - x_7x_{14} - x_8x_{16}, \\ x_{16}x_{12} - x_{17}x_{11} + x_6x_{17} - x_6x_{12} - x'_5x_{16} + x'_5x_{11}, \\ -x_{15}u_1 + x'_{14}u_1 - x'_{14}x_6 + u_1x_6 + x_{15}x'_5 - u_1x'_5\} \end{array} \right\},$$

compared to 17 polynomials in $F$ above. The conclusion polynomial is

$$g = -x_9x_{17} + x_9x'_{14} - x_{10}x_{15} + x_{10}x_{16} + x_{15}x_{17} - x'_{14}x_{16}.$$

We use the same total degree term ordering as before but with $x'_5$ inserted between $x_5$ and $x_6$, and $x'_{14}$ inserted between $x_{14}$ and $x_{15}$, to maintain the order of construction. The new proof took only 34 seconds.

## 7 Further Timing Comparisons

To conclude we give a timing comparison for a selection of other theorems chosen from [2], identified according to their numbering there. Table 1 gives comparisons between proofs using the standard formulation and the complex vector formulation. The superiority of the vector formulation is fairly consistent, and often quite considerable.

**Table 1.** Comparison between proofs using standard representation and complex vector representation (time in seconds).

| Example | Chou | Chou | Chou | Chou | Chou | Chou | Chou | Chou |
|---|---|---|---|---|---|---|---|---|
| from [2] | 19 | 21 | 39 | 40 | 43 | 44 | 45 | 48 |
| Standard | > 5000 | 394 | 81 | 247 | 20 | 22 | 32 | > 5000 |
| Complex | 34 | 82 | 47 | 149 | 18 | 21 | 34 | 33 |

| Example | Chou | Chou | Chou | Chou | Chou | Chou | Chou | Chou |
|---|---|---|---|---|---|---|---|---|
| from [2] | 63 | 73 | 80 | 94 | 96 | 106 | 109 | 115 |
| Standard | 47 | 97 | > 5000 | 139 | 3791 | 2994 | > 5000 | 432 |
| Complex | 170 | 25 | 897 | 28 | 8 | 28 | 33 | 29 |

| Example | Chou | Chou | Chou | Chou | Chou | Chou | Chou | Chou |
|---|---|---|---|---|---|---|---|---|
| from [2] | 128 | 144 | 162 | 240 | 243 | 271 | 277 | 302 |
| Standard | 1600 | 3297 | 56 | 85 | 171 | 33 | 1755 | 490 |
| Complex | 620 | 336 | 120 | 23 | 90 | 23 | 43 | 31 |

| Example | Chou | Chou | Chou | Chou | Chou | Chou | Chou | Chou |
|---|---|---|---|---|---|---|---|---|
| from [2] | 303 | 309 | 310 | 311 | 315 | 317 | 347 | 379 |
| Standard | 468 | 510 | 2302 | 36 | 340 | 25 | > 5000 | 88 |
| Complex | 25 | 390 | 87 | 31 | 78 | 2723 | 2190 | 15 |

| Example | Chou | Chou | Chou | Chou | Chou | Chou | Chou | Chou |
|---|---|---|---|---|---|---|---|---|
| from [2] | 390 | 393 | 395 | 401 | 409 | 440 | 456 | 487 |
| Standard | > 5000 | 1115 | 71 | > 5000 | 423 | 1545 | 10 | 41 |
| Complex | 10 | 336 | 19 | 37 | 23 | 53 | 10 | 39 |

### Acknowledgements

We thank the anonymous referees of the first draft of this paper for their helpful suggestions. We would also like to thank Desmond Fearnley-Sander for introducing us to the concept of automated theorem proving in geometry and many discussions over the years on that topic.

## References

1. Buchberger, B., Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, *Multidimensional Systems Theory* (ed. N. K. Bose), D. Reidel, 184–232 (1985).
2. Chou, S.-C., *Mechanical Geometry Theorem Proving*, D. Reidel (1988).
3. Chou, S.-C., Gao, X.-S. and Zhang, J.-Z., Automated Geometry Theorem Proving Using Vector Calculation, *Proc. ISSAC '93*, Kiev, 284–291 (1993).
4. Chou, S.-C., Gao, X.-S. and Zhang, J.-Z., Automated Production of Traditional Proofs for Constructive Geometry Theorems, *Proc. 8th IEEE Symbolic Logic in Computer Science*, Montreal, 48–56 (1993).
5. Chou, S.-C., Gao, X.-S. and Zhang, J.-Z., Automated Production of Traditional Proofs in Solid Geometry, *J. Automated Reasoning* 14, 257–291 (1995).

6. Cox, D., Little, J. and O'Shea, D., *Ideals, Varieties and Algorithms — An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Undergraduate Texts in Mathematics, Springer-Verlag (1992).
7. Fearnley-Sander, D., Plane Euclidean Reasoning, *Automated Deduction in Geometry* (eds. X.-S. Gao, D. Wang, L. Yang), Lecture Notes in Artificial Intelligence 1669, Springer-Verlag, 86–110 (1999).
8. Fearnley-Sander, D. and Stokes, T., Area in Grassmann Geometry, *Automated Deduction in Geometry* (ed. D. Wang), Lecture Notes in Artificial Intelligence, Springer-Verlag, 141–170 (1997).
9. Kapur, D., Geometry Theorem Proving Using Hilbert's Nullstellensatz, *Proc. SYMSAC '86*, Waterloo, 202–208 (1986).
10. Kapur, D., A Refutational Approach to Theorem Proving in Geometry, *Artificial Intelligence* 37, 61–93 (1988).
11. Li, H., Some Applications of Clifford Algebras to Geometries, *Automated Deduction in Geometry* (eds. X.-S. Gao, D. Wang, L. Yang), Lecture Notes in Artificial Intelligence 1669, Springer-Verlag, 156–179 (1999).
12. Sturmfels, B. and Whiteley, W., On the Synthetic Factorization of Projectively Invariant Polynomials, *J. Symbolic Computation* 11, 439–453 (1991).
13. Wang, D., Clifford Algebraic Calculus for Geometric Reasoning with Application to Computer Vision, *Automated Deduction in Geometry* (ed. D. Wang), Lecture Notes in Artificial Intelligence 1360, Springer-Verlag, 115–140 (1997).
14. White, N. L., Multilinear Cayley Factorization, *J. Symbolic Computation* 11, 421–438 (1991).
15. Wu, W.-t., On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry, *Scientia Sinica* 21, 157–179 (1978).
16. Wu, W.-t., *Mechanical Theorem Proving in Geometries: Basic Principles*, Springer-Verlag (1994).
17. Yang H., Zhang, S. and Feng, G., A Clifford Algebraic Method for Geometric Reasoning, *Automated Deduction in Geometry* (eds. X.-S. Gao, D. Wang, L. Yang), Lecture Notes in Artificial Intelligence 1669, Springer-Verlag, 111–129 (1999).

# Reasoning about Surfaces Using Differential Zero and Ideal Decomposition

Philippe Aubry and Dongming Wang

Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie – CNRS
4, place Jussieu, F-75252 Paris Cedex 05, France

**Abstract.** This paper presents methods for zero and ideal decomposition of partial differential polynomial systems and the application of these methods and their implementations to deal with problems from the local theory of surfaces. We show how to prove known geometric theorems and to derive unknown relations automatically. In particular, an algebraic relation between the first and the second fundamental coefficients in a very compact form has been derived, which is more general and has smaller degree than a relation discovered previously by Z. Li. Moreover, we provide symmetric expressions for Li's relation and clarify his statement. Some examples of theorem proving and computational difficulties encountered in our experiments are also discussed.

## 1 Introduction

Automated reasoning in differential geometry was initiated by W.-t. Wu [26] in the later 1970s as part of his work on mechanical theorem proving and discovering. Unlike the case of elementary geometry in which Wu's method has proven extremely efficient for proving many theorems, the situation in the differential case is quite different. Wu [27,29] and his followers (for example, Chou and Gao [8] and the second author [22]) have applied the method to prove theorems and derive relations mainly in the local theory of curves and mechanics, where the involved differential algebraic equations are ordinary (i.e., with only one derivation variable). The computational complexity becomes much higher in the partial differential case, where integrability conditions have to be taken into account and term orderings among derivatives need be introduced. In the case of two derivation variables, one may apply the method to deal with surfaces which has been investigated by Li [17] and recently by the second author in [25], a preliminary version of this paper. Along different directions, Carrà Ferro and Gallo [7] tried to generalize the Gröbner basis method [6] for automated theorem proving in differential geometry. Li and Cheng [15,16] have proposed to combine Clifford algebra formalism with Wu's method in order to produce short and readable proofs.

The algebraic bases of Wu's method are general algorithms for triangularizing systems of differential polynomials and for decomposing such systems into finitely many characteristic systems. These algorithms are developed from the classical work of M. Janet, J. M. Thomas, E. R. Kolchin, and Ritt [19], and are

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 154–174, 2001.

applicable to many other problems beyond geometry. The design and implementation of efficient algorithms for decomposing differential polynomial systems have become the topics of several other researchers. For example, we may mention the algorithm Rosenfeld–Groebner developed by Boulier and others [3,4], the subresultant-based algorithm suggested by Li and Wang [18] for computing differential simple systems, the factorization-free algorithm proposed recently by Hubert [10], and other relevant work by these authors. Some of these algorithms will be reviewed in Sect. 3 of this paper. In Sect. 4, we shall present an algorithm for decomposing any regular differential triangular system into simple sets such that the intersection of their saturated differential ideals is the same as the saturated differential ideal of the triangular system. This algorithm makes use of some ideas from the work of Kalkbrener [12] with improvement by the first author [1] and is similar to the algorithms given by Hubert [10], Boulier and Lemaire [5], but they are different in the fact that our algorithm avoids computing explicitly the inverse of some polynomials in an extension field and works in the general frame of positive dimension (there is no need to reduce the problem to one of dimension 0).

The main objective of this paper is to apply Wu's and other related methods based on zero and ideal decomposition and their implementations to deal with partial differential polynomial systems formulated from geometric problems, with a particular intention to study the local properties of surfaces. In fact, we have proved several well-known geometric theorems and derived some unknown relations about surfaces automatically (see the examples in Sects. 5.1 and 6). An algebraic relation between the first and the second fundamental coefficients in a very compact form has been derived (Sect. 5.2), which is more general and has smaller degree than a relation discovered previously by Li [17]. Moreover, we provide symmetric expressions for Li's relation and clarify his statement (Sect. 5.3). Some basics from the local theory of surfaces are collected in the following section and the paper concludes with a mention of a few computational difficulties encountered in our experiments.

The purpose of our work about surfaces is twofold: on the one hand, we take differential polynomial systems formulated from geometric problems as practical, not artificially constructed, examples to test the applicability and efficiency of decomposition algorithms, and on the other hand, we illustrate how differential geometric problems can be solved in an automatic way by using such algorithms. The results reported in this paper are still primitive and may serve to motivate further research on this subject. To achieve a full and effective automation of theorem proving and discovering in the theory of surfaces, considerable research and effort on both decomposition algorithms and algebraic formulation of geometric problems are needed.

## 2 The Local Theory of Surfaces

We recall some basic concepts from the local theory of surfaces. Let

$$\boldsymbol{r} = \boldsymbol{r}(u, v) = (x(u, v), y(u, v), z(u, v))$$

be a parametric surface $\mathfrak{S}$ in three-dimensional Euclidean space. Then

$$\boldsymbol{r}_u = \frac{\partial \boldsymbol{r}}{\partial u} \quad \text{and} \quad \boldsymbol{r}_v = \frac{\partial \boldsymbol{r}}{\partial v}$$

are the tangent vectors along the lines of $u$ and $v$ in the tangent plane at point $P\,(x, y, z)$ of $\mathfrak{S}$. The *first fundamental form* of $\mathfrak{S}$ is

$$|d\boldsymbol{r}|^2 = E du^2 + 2F dudv + G dv^2,$$

where

$$\begin{aligned} E &= \boldsymbol{r}_u^2 = \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2, \\ F &= \boldsymbol{r}_u \cdot \boldsymbol{r}_v = \frac{\partial x}{\partial u}\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u}\frac{\partial y}{\partial v} + \frac{\partial z}{\partial u}\frac{\partial z}{\partial v}, \\ G &= \boldsymbol{r}_v^2 = \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2 \end{aligned}$$

are called the *coefficients* of the first fundamental form (*first fundamental coefficients* for short) of $\mathfrak{S}$. Except for singular points on $\mathfrak{S}$, we have

$$E > 0, \; G > 0, \quad \delta = EG - F^2 > 0.$$

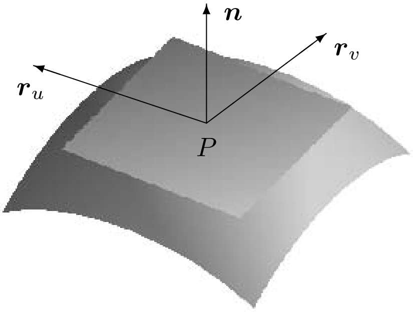


Fig. 1

Assume that the tangent vectors $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ are not parallel at a regular point $P$. Then

$$\boldsymbol{n} = \frac{\boldsymbol{r}_u \times \boldsymbol{r}_v}{|\boldsymbol{r}_u \times \boldsymbol{r}_v|}$$

is a unit normal vector perpendicular to the tangent plane of $\mathfrak{S}$ at $P$. The quadratic form

$$-d\boldsymbol{n} \cdot d\boldsymbol{r} = L du^2 + 2M dudv + N dv^2$$

is called the *second fundamental form* of $\mathfrak{S}$, where

$$L = -\boldsymbol{r}_u \cdot \boldsymbol{n}_u, \quad M = -\boldsymbol{r}_u \cdot \boldsymbol{n}_v, \quad N = -\boldsymbol{r}_v \cdot \boldsymbol{n}_v$$

are the *second fundamental coefficients* of $\mathfrak{S}$. The function

$$K = (LN - M^2)/\delta$$

is defined to be the *Gaussian curvature* of $\mathfrak{S}$. We recall the following famous theorem of Gauss.

**Theorem 1 (Theorema egregium).** *The Gaussian curvature $K$ of any surface depends only on the first fundamental coefficients $E, F, G$ and their first and second derivatives. More precisely,*

$$K = \frac{1}{\delta^2}\left(\begin{vmatrix} E & F & F_v - \frac{1}{2}G_u \\ F & G & \frac{1}{2}G_v \\ \frac{1}{2}E_u & F_u - \frac{1}{2}E_v & \alpha \end{vmatrix} - \begin{vmatrix} E & F & \frac{1}{2}E_v \\ F & G & \frac{1}{2}G_u \\ \frac{1}{2}E_v & \frac{1}{2}G_u & 0 \end{vmatrix}\right),$$

*where* $\alpha = -\frac{1}{2}E_{vv} + F_{uv} - \frac{1}{2}G_{uu}$.

When expanded, the right-hand side of the above equality is a rational expression, whose numerator consists of 17 terms, in $E, F, G$ and their derivatives. We shall demonstrate in Sect. 5 how this expression can be derived automatically.

The three vectors $[\boldsymbol{r}_u, \boldsymbol{r}_v, \boldsymbol{n}]$ form a moving frame of $\mathfrak{S}$ at $P$. Taking their partial derivatives with respect to $u$ and $v$, we have

$$\begin{cases} \boldsymbol{r}_{uu} = \Gamma_{11}^1 \boldsymbol{r}_u + \Gamma_{11}^2 \boldsymbol{r}_v + L\boldsymbol{n}, \\ \boldsymbol{r}_{uv} = \Gamma_{12}^1 \boldsymbol{r}_u + \Gamma_{12}^2 \boldsymbol{r}_v + M\boldsymbol{n}, \\ \boldsymbol{r}_{vv} = \Gamma_{22}^1 \boldsymbol{r}_u + \Gamma_{22}^2 \boldsymbol{r}_v + N\boldsymbol{n}; \end{cases} \tag{1}$$

$$\begin{cases} \boldsymbol{n}_u = [(MF - LG)\boldsymbol{r}_u + (LF - ME)\boldsymbol{r}_v]/\delta, \\ \boldsymbol{n}_v = [(NF - MG)\boldsymbol{r}_u + (MF - NE)\boldsymbol{r}_v]/\delta, \end{cases} \tag{2}$$

where

$$\begin{array}{ll} \Gamma_{11}^1 = (GE_u - 2FF_v + FE_v)/(2\delta), & \Gamma_{11}^2 = (2EF_u - EE_v - FE_u)/(2\delta), \\ \Gamma_{12}^1 = (GE_v - FG_u)/(2\delta), & \Gamma_{12}^2 = (EG_u - FE_v)/(2\delta), \\ \Gamma_{22}^1 = (2GF_v - GG_u + FG_v)/(2\delta), & \Gamma_{22}^2 = (EG_v - 2FF_v - FG_u)/(2\delta) \end{array}$$

are the *Christoffel symbols of the second kind*. The equations (1) and (2) are called *Gauss formulas* and *Weingarten formulas*, respectively. Their integrability conditions are given respectively by the following equations:

$$\begin{cases} KF = (\Gamma_{12}^1)_u - (\Gamma_{11}^1)_v + \Gamma_{12}^2\Gamma_{12}^1 - \Gamma_{11}^2\Gamma_{22}^1, \\ KE = (\Gamma_{11}^2)_v - (\Gamma_{12}^2)_u + \Gamma_{11}^1\Gamma_{12}^2 + \Gamma_{11}^2\Gamma_{22}^2 - \Gamma_{12}^1\Gamma_{11}^2 - (\Gamma_{12}^2)^2, \\ KG = (\Gamma_{22}^1)_u - (\Gamma_{12}^1)_v + \Gamma_{22}^2\Gamma_{12}^1 + \Gamma_{22}^1\Gamma_{11}^1 - \Gamma_{12}^2\Gamma_{22}^1 - (\Gamma_{12}^1)^2, \\ KF = (\Gamma_{12}^2)_v - (\Gamma_{22}^2)_u + \Gamma_{12}^1\Gamma_{12}^2 - \Gamma_{22}^1\Gamma_{11}^2; \end{cases} \tag{3}$$

$$\begin{cases} L_v - M_u = L\,\Gamma_{12}^1 + M(\Gamma_{12}^2 - \Gamma_{11}^1) - N\,\Gamma_{11}^2, \\ M_v - N_u = L\,\Gamma_{22}^1 + M(\Gamma_{22}^2 - \Gamma_{12}^1) - N\,\Gamma_{12}^2. \end{cases} \tag{4}$$

As usual, (3) is called the *Gauss equations* and (4) the *Codazzi–Mainardi equations*.

The concepts and results about the local theory of surfaces reviewed above are classical and can be found in standard textbooks of differential geometry, for example [13,14].

# 3 Decomposition of Differential Polynomial Systems

In order to reason about problems of surfaces, we shall formulate geometric conditions as partial differential polynomial (*d-pol* for short) equations and inequations with two derivation variables $u$ and $v$. Proving a known theorem may be reduced to determining whether the conclusion-d-pols vanish on the set of differential zeros of the hypothesis-d-pol system, and if not, on which part of the zero set they do. This is closely related to the problem of radical differential ideal membership. Discovering a new theorem amounts to deriving an unknown relation from the given geometric hypotheses expressed as d-pol equations. The determination and derivation may be made easier when the system of hypothesis-d-pols is decomposed into special subsystems that have certain triangular form and for which the radical differential ideal membership may be tested by simple reductions. In this section, we give a short review of major techniques for the decomposition of d-pol systems, which may be used for differential geometric reasoning.

## 3.1 Differential Polynomials and Triangular Systems

Let $\mathcal{K}$ be a differential field of characteristic 0 with $m$ *derivation operators* $\delta_1, \ldots, \delta_m$. We call $\theta = \delta_1^{i_1} \ldots \delta_m^{i_m}$ a *derivative operator* and $i_1 + \cdots + i_m$ the *order* of $\theta$. Let $x_1, \ldots, x_n$ be $n$ differential indeterminates over $\mathcal{K}$. For any $k$, the symbol $\theta x_k$ denotes the derivative of $x_k$ with respect to (*wrt*) $\theta$; $\theta x_k$ is said to be *proper* if the order of $\theta$ is positive. We denote by $\mathcal{R} = \mathcal{K}\{x_1, \ldots, x_n\}$ the ring of polynomials in the derivatives of $x_1, \ldots, x_n$ with coefficients in $\mathcal{K}$, by $[P]$ or $[\mathbb{P}]$ the differential ideal generated by the d-pol $P$ or the d-pols in $\mathbb{P}$, and by $(P)$ or $(\mathbb{P})$ the algebraic ideal generated by $P$ or the elements of $\mathbb{P}$ considered as ordinary polynomials in the derivatives.

There are different *admissible* orderings by which derivatives can be ordered. We shall consider d-pols with a fixed admissible ordering $\prec$ for their derivatives. Let $P$ be any d-pol in $\mathcal{R} \setminus \mathcal{K}$. The highest derivative appearing in $P$ is called the *lead* of $P$ and denoted by $\mathrm{ld}(P)$. When $\mathrm{ld}(P) = \theta x_k$, $k$ is called the *class* of $P$. We call the leading coefficient of $P$ wrt $\mathrm{ld}(P)$ the *initial* of $P$, and the formal partial derivative of $P$ wrt $\mathrm{ld}(P)$ the *separant* of $P$; they are denoted by $\mathrm{ini}(P)$ and $\mathrm{sep}(P)$, respectively.

A d-pol $Q$ is said to be *partially reduced* wrt $P$ if no proper derivative of $\mathrm{ld}(P)$ appears in $Q$, and *reduced* wrt $P$ if $Q$ is partially reduced wrt $P$ and the degree of $Q$ is lower than that of $P$ in $\mathrm{ld}(P)$. In any case, one can compute a *partial d-pseudo-remainder* $R'$ and a *d-pseudo-remainder* $R$ of $Q$ wrt $P$; then there exist integers $\alpha$ and $\beta$ such that

$$\mathrm{sep}(P)^{\alpha}Q - R' \in [P], \quad \mathrm{sep}(P)^{\alpha}\mathrm{ini}(P)^{\beta}Q - R \in [P],$$

$R'$ is partially reduced wrt $P$, and $R$ is reduced wrt $P$. The d-pseudo-remainder of $Q$ wrt $P$ is denoted by d-prem$(Q, P)$.

Let $\mathbb{P}$ be any finite set of d-pols in $\mathcal{R}$. A *differential zero* (*d-zero*) of $\mathbb{P}$ is an $n$-tuple $(z_1, \ldots, z_n)$ in a universal differential extension field of $\mathcal{K}$ such that every $P \in \mathbb{P}$ becomes zero after $\theta x_k$ is replaced by $\theta z_k$, for all $\theta x_k$ occurring in the d-pols of $\mathbb{P}$. We denote by d-Zero$(\mathbb{P})$ the set of all d-zeros of $\mathbb{P}$. The set of ordinary zeros of $\mathbb{P}$, regarded as a set of polynomials with the occurring derivatives as indeterminates, is denoted by Zero$(\mathbb{P})$. For any $\mathbb{Q} \subset \mathcal{R}$, we define

$$\text{d-Zero}(\mathbb{P}/\mathbb{Q}) := \text{d-Zero}(\mathbb{P}) \setminus \text{d-Zero}\left(\left\{\prod\nolimits_{Q\in\mathbb{Q}} Q\right\}\right),$$

and similarly for Zero$(\mathbb{P}/\mathbb{Q})$.

A nonempty ordered set $\mathbb{T} = \langle T_1, \ldots, T_r\rangle$ of d-pols in $\mathcal{R} \setminus \mathcal{K}$ is called a *differential triangular* (*d-tri*) set if $\mathrm{ld}(T_1) \prec \cdots \prec \mathrm{ld}(T_r)$ and every $T_j$ is partially reduced wrt $T_i$ for $j > i$. A d-tri set is *autoreduced* if every $T_j$ is reduced wrt $T_i$ for $j > i$.

Let $Q$ be any d-pol and $\mathbb{U}$ a finite set of nonzero d-pols in $\mathcal{R}$. We define

$$\begin{aligned}&\text{d-prem}(Q, \mathbb{T}) := \text{d-prem}(\ldots \text{d-prem}(Q, T_r), \ldots, T_1),\\ &\mathrm{si}(\mathbb{T}) := \{\mathrm{sep}(T_i), \mathrm{ini}(T_i) \mid 1 \le i \le r\}.\end{aligned}$$

The pair $\langle\mathbb{T}, \mathbb{U}\rangle$ is called a *d-tri* system if every d-pol in si$(\mathbb{T})$ does not vanish on d-Zero$(\mathbb{T}/\mathbb{U})$, and an *elementary triangular* (*e-tri*) system if every polynomial in si$(\mathbb{T})$ does not vanish on Zero$(\mathbb{T}/\mathbb{U})$.

### 3.2 Decomposition into Regular Systems

Let $F$ and $G$ be two d-pols in $\mathcal{R} \setminus \mathcal{K}$ of same class. A *$\Delta$-polynomial* (*$\Delta$-pol*) of $F$ and $G$ is defined as

$$\Delta(F, G) := \mathrm{sep}(G)\theta F - \mathrm{sep}(F)\phi G, \tag{5}$$

where $\theta$ and $\phi$ are proper derivative operators with lowest order such that $\mathrm{ld}(\theta F) = \mathrm{ld}(\phi G)$.

A d-tri set $\mathbb{T}$ is said to be *coherent* if, for any $F, G \in \mathbb{T}$ with same class and any $\Delta(F, G)$ of the form (5), we have

$$J\Delta(F, G) = Q_1\theta_1 T_1 + \cdots + Q_r\theta_r T_r,$$

where $J$ is a product of d-pols in si$(\mathbb{T})$, $Q_i \in \mathcal{R}$, and $T_i \in \mathbb{T}$ with $\theta_i T_i < \mathrm{ld}(\theta F)$ for $1 \le i \le r$.

A d-tri system $\langle\mathbb{T}, \mathbb{U}\rangle$ is said to be *coherent* if $\mathbb{T}$ is coherent and each d-pol in $\mathbb{U}$ is partially reduced wrt $\mathbb{T}$. A d-tri system $\langle\mathbb{T}, \mathbb{U}\rangle$ is said to be *regular* if it is a coherent and e-tri system.

Given a pair $\langle \mathbb{P}, \mathbb{Q} \rangle$ of d-pol sets (called a *d-pol system*) in $\mathcal{R}$, one can decompose $\langle \mathbb{P}, \mathbb{Q} \rangle$ into finitely many regular d-tri systems $\langle \mathbb{T}_1, \mathbb{U}_1 \rangle, \ldots, \langle \mathbb{T}_e, \mathbb{U}_e \rangle$ such that

$$\text{d-Zero}(\mathbb{P}/\mathbb{Q}) = \bigcup_{i=1}^{e} \text{d-Zero}(\mathbb{T}_i/\mathbb{U}_i). \tag{6}$$

This can be done by using the characteristic set method of Ritt [19] and Wu [27,28], or the elimination techniques of Seidenberg [21] further developed by Boulier and others [3,4,18,23]. Besides their different elimination strategies, the two approaches are also distinguished by their ways of dealing with integrability conditions: the former uses *passive ascending sets* and integrability d-pols are computed by means of *completion* and *M/I-reduction* [19,28], while for the latter integrability conditions are handled with coherence and computed by means of $\Delta$-pols [20,3,4]. It is clear that the two methods and their ways of dealing with integrability conditions may be combined or interchanged. Some relations between *passive* and coherent d-tri sets have been established in [18].

The significance of regularity of a d-tri system lies partly in the following remarkable result [20]: if a d-tri system $\langle \mathbb{T}, \mathbb{U} \rangle$ is regular, then

$$\text{d-Zero}(\mathbb{T}/\mathbb{U}) = \varnothing \iff \text{Zero}(\mathbb{T}/\mathbb{U}) = \varnothing.$$

Moreover, by means of computing regular d-tri systems the problem of deciding whether a d-pol $P$ vanishes on d-Zero($\mathbb{T}/\mathbb{U}$) may be reduced to a similar problem for *ordinary* polynomials. To make this precise, we recall the *saturation* of (differential) ideals. Let $\mathfrak{I}$ be an ideal and $\mathbb{F}$ a nonempty subset of $\mathcal{R}$. The *saturated ideal* of $\mathfrak{I}$ wrt $\mathbb{F}$ is

$$\mathfrak{I} : \mathbb{F}^{\infty} := \{P \in \mathcal{R} \mid FP \in \mathfrak{I} \text{ for some product } F \text{ of d-pols in } \mathbb{F}\}.$$

$\mathfrak{I} : \mathbb{F}^{\infty}$ is a differential ideal if $\mathfrak{I}$ is a differential ideal, and that $P$ vanishes on d-Zero($\mathbb{T}/\mathbb{U}$) if and only if (iff) $P$ belongs to the radical of $[\mathbb{T}] : \mathbb{U}^{\infty}$. Thus the zero decomposition (6) implies that

$$\text{d-Zero}([\mathbb{P}] : \mathbb{Q}^{\infty}) = \bigcup_{i=1}^{e} \text{d-Zero}([\mathbb{T}_i] : \mathbb{U}_i^{\infty}). \tag{7}$$

The following result [3,4,18] shows how to reduce the radical ideal membership problem from the differential case to the algebraic case. Let a d-tri system $\langle \mathbb{T}, \mathbb{U} \rangle$ be regular. Then,

- a d-pol $P$ belongs to $[\mathbb{T}] : \mathbb{U}^{\infty}$ iff a partial d-pseudo-remainder of $P$ wrt $\mathbb{T}$ belongs to $(\mathbb{T}) : \mathbb{U}^{\infty}$; and
- both $[\mathbb{T}] : \mathbb{U}^{\infty}$ and $(\mathbb{T}) : \mathbb{U}^{\infty}$ are radical.

### 3.3 Irreducibility, Gröbner Bases, and Simple Systems

In the decomposition (6) it is possible that d-Zero($\mathbb{T}_i/\mathbb{U}_i$) = $\varnothing$ for some $i$, and radical ideal membership cannot readily be tested without further computation.

However, in view of the results mentioned in the preceding section we now can work with ordinary polynomials over $\mathcal{K}$. There are several techniques that have been developed to deal with this algebraic problem deduced from d-pol systems.

The first is to impose an irreducibility requirement, which was proposed by Ritt [19] and Wu [28]. In this case, one further decomposes each regular d-tri system $\langle \mathbb{T}_i, \mathbb{U}_i \rangle$, considered as an e-tri system, into *irreducible* e-tri systems, where polynomial factorization over successive algebaric extension fields may have to be used. In the process of decomposition, an e-tri system that has no zero is detected automatically, and the obtained irreducible ones all have zeros. For any regular and irreducible d-tri system $\langle \mathbb{T}, \mathbb{U} \rangle$, a d-pol $P$ vanishes on $\text{d-Zero}(\mathbb{T}/\mathbb{U})$ iff $\text{d-prem}(P, \mathbb{T}) = 0$. Therefore, after a d-pol set $\mathbb{P}$ has been decomposed into regular and irreducible d-tri systems, whether a d-pol vanishes on $\text{d-Zero}(\mathbb{P})$ can be tested simply by computing d-pseudo-remainders.

The second technique, proposed by Boulier and others [3,4], works by computing a Gröbner basis $\mathbb{G}$ of the saturated algebraic ideal $(\mathbb{T}) : \mathbb{U}^\infty$ using a standard method [6]. The e-tri system $\langle \mathbb{T}, \mathbb{U} \rangle$ has no zero iff $\mathbb{G}$ contains a nonzero element of $\mathcal{K}$. Otherwise, a d-pol $P$ belongs to $[\mathbb{T}] : \mathbb{U}^\infty$ iff the *normal form* of a partial d-pseudo-remainder of $P$ wrt $\mathbb{T}$ modulo $\mathbb{G}$ is 0. So, in this case the radical ideal membership may be tested by means of computing partial d-pseudo-remainders and normal forms.

The third technique, suggested by Li and Wang [18], proceeds by decomposing each regular d-tri system, considered as an e-tri system, into *simple* systems [24]. A simple system is an e-tri system $\langle \mathbb{T}, \mathbb{U} \rangle$, in which $\mathbb{T}$ is not necessarily irreducible but every polynomial of $\mathbb{T}$ is conditionally *squarefree* in some technical sense. The essential process for this decomposition is to use the polynomials of $\mathbb{T}$ to eliminate polynomials from $\mathbb{U}$ by computing *regular subresultants*. The squarefreeness condition may be easily accomplished because of the regularity of the d-tri system. A simple system $\langle \mathbb{T}, \mathbb{U} \rangle$ must have zeros, and a d-pol $P$ vanishes on $\text{d-Zero}(\mathbb{T}/\mathbb{U})$ iff $\text{d-prem}(P, \mathbb{T}) = 0$.

More recently, Hubert [10], Boulier and Lemaire [5] have devised alternative (and more specialized) algorithms for computing (normalized/simple) e-tri sets representing regular differential ideals. We shall propose another specialized algorithm in the next section.

The above general setting of d-pols will be specialized in our study of surfaces with $m = 2$ and $\delta_1 = \partial/\partial v$, $\delta_2 = \partial/\partial u$. Speaking about partial derivatives in this case, we usually mean *proper* derivatives.

## 4 An Algorithm for Ideal Decomposition of Regular Systems

In this section we present a new algorithm for decomposing regular d-tri systems into *d-simple sets*. A coherent d-simple set $\mathbb{T}$ is a d-tri set that has d-zeros, and for which $P \in [\mathbb{T}] : \text{si}(\mathbb{T})^\infty$ iff $\text{d-prem}(P, \mathbb{T}) = 0$. The underlying idea of our method is similar to the one described in [5], but our definition of d-simple sets is weaker than the definition of characteristic presentations introduced by

Boulier and others, and our algorithm avoids computing explicitly the inverse of some polynomials in an extension of $\mathcal{K}$. Moreover, our method works in the general frame of positive dimension and we do not need to reduce the problem to a zero-dimensional one as in [10,5]. This may simplify the problem to be solved and make it clearer and thus is a point of theoritical interest.

We say that a d-tri set $\mathbb{T}$ is a *d-simple set* if

$$[\mathbb{T}] : \mathrm{si}(\mathbb{T})^{\infty} = \{P \in \mathcal{R} \mid \text{d-prem}(P, \mathbb{T}) = 0\}.$$

Let $\langle \mathbb{T}, \mathbb{U} \rangle$ be a regular d-tri system. We are interested in computing an *irredundant characteristic decomposition* of $[\mathbb{T}] : \mathbb{U}^{\infty}$, that is a finite family of d-simple sets $\mathbb{C}_1, \ldots, \mathbb{C}_m$ such that

$$[\mathbb{T}] : \mathbb{U}^{\infty} = \bigcap_{i=1}^{m} [\mathbb{C}_i] : \mathrm{si}(\mathbb{C}_i)^{\infty}$$

and the sets of associated prime ideals of $[\mathbb{C}_i] : \mathrm{si}(\mathbb{C}_i)^{\infty}$ form a partition of the set of associated prime ideals of $[\mathbb{T}] : \mathbb{U}^{\infty}$.

This problem may actually reduce to a purely algebraic problem. To explain this, we have to introduce similar notions for algebraic ideals. Our main tool is the algorithm triangSplit given below. For any d-tri set $\mathbb{T} = \langle T_1, \ldots, T_r \rangle$, we define $\mathrm{ini}(\mathbb{T}) := \{\mathrm{ini}(T_i) \mid 1 \leq i \leq r\}$ and $\mathrm{sat}(\mathbb{T}) := (\mathbb{T}) : \mathrm{ini}(\mathbb{T})^{\infty}$.

A d-tri set $\mathbb{T}$ considered as an e-tri set is called a *proper triangular set*[1] if $\mathrm{sat}(\mathbb{T}) = \{P \in \mathcal{R} \mid \mathrm{prem}(P, \mathbb{T}) = 0\}$. An *irredundant algebraic characteristic decomposition* of an ideal $\mathfrak{I}$ in $\mathcal{R}$ is a family of proper triangular sets $\mathbb{C}_1, \ldots, \mathbb{C}_m$ such that $\mathfrak{I} = \bigcap_{i=1}^{m} \mathrm{sat}(\mathbb{C}_i)$ and the sets of associated prime ideals of $\mathrm{sat}(\mathbb{C}_i)$ form a partition of the set of minimal associated primes of $\mathfrak{I}$. For simplicity, we sometimes omit the adjective "irredundant."

The following result slightly generalizes Theorem 6.2 of [10] and shows that the problem can be treated in a purely algebraic way.

**Theorem 2.** *Let $\langle \mathbb{T}, \mathbb{U} \rangle$ be a regular d-tri system in $\mathcal{R}$. If $(\mathbb{T}) : \mathbb{U}^{\infty} = \bigcap_i \mathit{sat}(\mathbb{C}_i)$ is an irredundant algebraic characteristic decomposition, then each $\mathbb{C}_i$ is a coherent d-simple set, and $[\mathbb{T}] : \mathbb{U}^{\infty} = \bigcap_i [\mathbb{C}_i] : \mathrm{si}(\mathbb{C}_i)^{\infty}$ is an irredundant characteristic decomposition of $[\mathbb{T}] : \mathbb{U}^{\infty}$.*

Hubert [10], Boulier and Lemaire [5] have proposed algorithms for computing algebraic characteristic decompositions from coherent autoreduced sets. They have to work with zero-dimensional ideals in the main step of their algorithms. Our algorithm deals directly with triangular sets of positive dimension, using the techniques introduced by Kalkbrener [11] and developed by the first author, and thus it avoids projecting the triangular sets to dimension zero and then lifting them up.

[1] A proper triangular set is usually called a *regular set* or *regular chain* (see, e.g., [12,2]). We use the term "proper" because "regular" is used with a different meaning in this paper.

Let $\mathbb{C}$ be a proper triangular set and $P$ a polynomial in $\mathcal{R}$. We say that $\mathbb{C}$ is a *simple set* if the ideal $\mathrm{sat}(\mathbb{C})$ is radical. The algorithm $\mathsf{split}(\mathbb{C}, P)$ presented in [12] provides two families $\mathbb{C}_1, \ldots, \mathbb{C}_m$ and $\mathbb{C}_{m+1}, \ldots, \mathbb{C}_p$ of proper triangular sets such that (see [1])

- $\sqrt{\mathrm{sat}(\mathbb{C})} = \sqrt{\mathrm{sat}(\mathbb{C}_1)} \cap \cdots \cap \sqrt{\mathrm{sat}(\mathbb{C}_p)}$,
- the sets of minimal associated prime ideals of $\mathrm{sat}(\mathbb{C}_i)$ for all $i$ form a partition of the set of minimal associated primes of $\mathrm{sat}(\mathbb{C})$,
- $\forall i,\ 1 \le i \le m$, $P$ is not a zero divisor modulo $\mathrm{sat}(\mathbb{C}_i)$,
- $\forall i,\ m+1 \le i \le p$, $P$ is zero modulo $\mathrm{sat}(\mathbb{C}_i)$.

Moreover, when $\mathbb{C}$ is a simple set, the $\mathbb{C}_i$ are also simple sets and thus form an algebraic characteristic decomposition of $\mathrm{sat}(\mathbb{C})$. We then define $\mathsf{invComp}(\mathbb{C}, P)$ as the set $\{\mathbb{C}_1, \ldots, \mathbb{C}_m\}$ and $\mathsf{nullComp}(\mathbb{C}, P)$ as the set $\{\mathbb{C}_{m+1}, \ldots, \mathbb{C}_p\}$.

Let $\mathbb{U} \subset \mathcal{R}$ and $\mathbb{C}$ be a proper triangular set. It follows clearly from the definition of $\mathsf{invComp}$ above that the algorithm $\mathsf{invCompSet}(\mathbb{C}, \mathbb{U})$ below returns a set $\{\mathbb{C}_1, \ldots, \mathbb{C}_m\}$ of proper triangular sets such that

$$\sqrt{\mathrm{sat}(\mathbb{C})} : \mathbb{U}^\infty = \sqrt{\mathrm{sat}(\mathbb{C}_1)} \cap \cdots \cap \sqrt{\mathrm{sat}(\mathbb{C}_m)}.$$

Moreover, if $\mathrm{sat}(\mathbb{C}) : \mathbb{U}^\infty$ is radical, then $\mathbb{C}_1, \ldots, \mathbb{C}_m$ are simple sets forming an algebraic characteristic decomposition of $\mathrm{sat}(\mathbb{C}) : \mathbb{U}^\infty$.

```
$\mathsf{invCompSet}(\mathbb{C}, \mathbb{U})$
    $\Psi := \{\mathbb{C}\}$
    for $U$ in $\mathbb{U}$ repeat
        $\Psi := \bigcup_{\mathbb{B} \in \Psi} \mathsf{invComp}(\mathbb{B}, U)$
    return $\Psi$
```

Using these algorithms we devise our main algorithm $\mathsf{triangSplit}$:

- **Input:** $\langle \mathbb{T}, \mathbb{U} \rangle$ either a regular d-tri system in $\mathcal{R}$, or a d-pol system in $\mathcal{R}$ with $\mathbb{T} = \varnothing$;
- **Output:** $\{\mathbb{C}_1, \ldots, \mathbb{C}_m\}$ a set of simple sets, which form an irreducible characteristic decomposition of $(\mathbb{T}) : \mathbb{U}^\infty$.

```
$\mathsf{triangSplit}(\mathbb{T}, \mathbb{U})$
    if $\mathbb{T} = \varnothing$ then return $\{\mathbb{T}\}$
    let $\mathbb{T} = \langle T_1, \ldots, T_r \rangle$
    $\mathbb{U} := \mathbb{U} \cup \mathrm{ini}(\mathbb{T})$
    $\mathbb{V} := \{U \in \mathbb{U} \mid \ \mathrm{ld}(U) = \mathrm{ld}(T_r)\}$
    $\mathbb{U}' := \mathbb{U} \setminus \mathbb{V}$
    $\Phi := \varnothing$
    for $\mathbb{C}$ in $\mathsf{triangSplit}(\langle T_1, \ldots, T_{r-1} \rangle, \mathbb{U}')$ repeat
        $\mathbb{B} := \mathbb{C} \cup \{T_r\}$
        for $\mathbb{A}$ in $\mathsf{invCompSet}(\mathbb{B}, \mathbb{V})$ repeat
                $\Phi := \Phi \cup \{\mathbb{A}\}$
    return $\Phi$
```

*Proof.* We show the correctness of the algorithm. If $\mathbb{T}$ is empty, then $(\mathbb{T}) = (0)$ and the result is obvious. Now assume that $\mathbb{T} = \langle T_1, \ldots, T_r\rangle$ with $r > 0$, and let $\mathbb{T}' = \langle T_1, \ldots, T_{r-1}\rangle$ and $P = T_r$.

Let $\langle\mathbb{T}, \mathbb{U}\rangle$ be a regular d-tri system. By definition, $\mathrm{ini}(T)(x) \not= 0$ for any$T \in \mathbb{T}$ and $x \in \mathrm{Zero}(\mathbb{T}/\mathbb{U})$. It follows that $\mathrm{ini}(T)$ does not belong to any associated prime ideal of $(\mathbb{T}) : \mathbb{U}^\infty$ and this justifies line 3 of the algorithm. Moreover, we have

$$\begin{aligned}(\mathbb{T}) : \mathbb{U}^\infty &= (\mathbb{T}) : \mathbb{U}^\infty : \mathrm{ini}(P)^\infty\\ &= (\mathbb{T}' \cup \{P\}) : \mathbb{U}'^\infty : \mathrm{ini}(P)^\infty : \mathbb{V}^\infty.\end{aligned}$$

Let $\Phi_1$ be the output of $\mathsf{triangSplit}(\mathbb{T}', \mathbb{U}')$. One may easily verify that $(\mathbb{T}' \cup \{P\}) : \mathbb{U}'^\infty = (\mathbb{T}' : \mathbb{U}'^\infty \cup \{P\}) : \mathbb{U}'^\infty$, and by induction we obtain

$$\begin{aligned}(\mathbb{T}) : \mathbb{U}^\infty &= \bigcap_{\mathbb{C}\in\Phi_1} (\mathrm{sat}(\mathbb{C}) \cup \{P\}) : \mathrm{ini}(P)^\infty : \mathbb{V}^\infty : \mathbb{U}'^\infty\\ &= \bigcap_{\mathbb{C}\in\Phi_1} \mathrm{sat}(\mathbb{C} \cup \{P\}) : \mathbb{V}^\infty : \mathbb{U}'^\infty\end{aligned}$$

according to Proposition 4.3.2 in [1].

Let $\boldsymbol{X}$ denote the indeterminates, other than $\mathrm{ld}(P)$, occurring in $\mathbb{T}$. Let $\mathcal{P}$ be an associated prime ideal of $\mathrm{sat}(\mathbb{C} \cup \{P\})$. According to [1] (Theorem 4.3.6), $\mathcal{P} \cap \mathcal{K}[\boldsymbol{X}]$ is an associated prime of $\mathrm{sat}(\mathbb{C})$ and consequently an associated prime of $(\mathbb{T}') : \mathbb{U}'^\infty$. It follows that $U' \not\in \mathcal{P}$for any $U' \in \mathbb{U}'$, and thus

$$(\mathbb{T}) : \mathbb{U}^\infty = \bigcap_{\mathbb{C}\in\Phi_1} \mathrm{sat}(\mathbb{C} \cup \{P\}) : \mathbb{V}^\infty.$$

Let $\mathbb{C} \in \Phi_1$ and $\{\mathbb{A}_1, \ldots, \mathbb{A}_m\} = \mathsf{invCompSet}(\mathbb{C} \cup \{P\}, \mathbb{V})$. Since $(\mathbb{T}) : \mathbb{U}^\infty$ is radical, so is the ideal $\mathrm{sat}(\mathbb{C} \cup \{P\}) : \mathbb{V}^\infty$; and from the specification of $\mathsf{invCompSet}$ the family $\mathbb{A}_1, \ldots, \mathbb{A}_m$ is an algebraic characteristic decomposition of $\mathrm{sat}(\mathbb{C} \cup \{P\}) : \mathbb{V}^\infty$. We thus deduce that

$$(\mathbb{T}) : \mathbb{U}^\infty = \bigcap_{\mathbb{A}\in\Phi} \mathrm{sat}(\mathbb{A}).$$

This proves the correctness of $\mathsf{triangSplit}$. The termination of the algorithm is obvious.

Let $\langle\mathbb{T}, \mathbb{U}\rangle$ be a regular d-tri system and $\{\mathbb{C}_1, \ldots, \mathbb{C}_m\}$ be the output of our algorithm for $\langle\mathbb{T}, \mathbb{U}\rangle$. Then, $P$ vanishes on $\text{d-Zero}(\mathbb{T}/\mathbb{U})$ iff $P \in [\mathbb{T}] : \mathbb{U}^\infty$ iff the partial d-pseudo-remainder of $P$ wrt $\mathbb{T}$ belongs to $(\mathbb{T}) : \mathbb{U}^\infty$ iff $\text{d-prem}(P, \mathbb{C}_i) = 0$ for all $i = 1, \ldots, m$. Therefore, with (6), whether a d-pol vanishes on $\text{d-Zero}(\mathbb{P}/\mathbb{Q})$ can be completely decided by decomposing the d-pol system $\langle\mathbb{P}, \mathbb{Q}\rangle$ into regular d-tri systems and then to d-simple systems and by d-pseudo-remainder computations.

# 5 The Fundamental Coefficients of Surfaces Revisited

In this section, we apply Wu's and other related methods based on zero decomposition of d-pol systems to study the fundamental coefficients of surfaces and their relationships. The application of these methods to deal with geometric problems that can be expressed algebraically by means of d-pol equations and inequations is quite straightforward. For differential geometry in three-dimensional Euclidean space, sometimes we also need to handle inequalities which are defined over the field of reals. In this case, some alternative devices have to be adopted. For example, we shall use the weaker condition $\delta \not= 0$ instead of the real condition $\delta > 0$ in the computation.

Let us introduce an ordering $\Omega$ for the derivatives of $x_1 \prec \cdots \prec x_n$ as follows: $\delta_1^{i_1}\delta_2^{i_2}x_k \prec \delta_1^{j_1}\delta_2^{j_2}x_l$ if either $i_1+i_2 < j_1+j_2$, or $i_1+i_2 = j_1+j_2$ and $k<l$, or $i_1+i_2=j_1+j_2$, $k=l$ and $i_2<j_2$. This ordering is denoted by `grlexA` in the *diffalg* package of F. Boulier and E. Hubert [9], which has been used together with the authors' implementation of simple systems and the algorithm triangSplit for our experiments.

## 5.1 Automated Rediscovery of Theorema Egregium

To derive the representation of the Gaussian curvature $K$ in terms of the first fundamental coefficients and their derivatives, we form a set $\mathbb{P}$ of d-pols corresponding to the equations in (1) and (3). Let the derivatives of $E \prec F \prec G \prec L \prec M \prec N$ and those of $\boldsymbol{r}_u \prec \boldsymbol{r}_v \prec \boldsymbol{n}$ be ordered according to $\Omega$, and any derivative of $E, F, G, L, M, N$ be ordered smaller than any derivative of $\boldsymbol{r}_u, \boldsymbol{r}_v, \boldsymbol{n}$. Under this (admissible) ordering, a decomposition of the form (6) for the d-pol system $\langle \mathbb{P}, \{\delta, M\}\rangle$ (e.g., using `Rosenfeld_Groebner` in *diffalg* without computation of Gröbner bases) contains only one element (i.e., $e=1$). One may find that in the d-tri set there is a d-pol of the form

$$g = \delta(LN - M^2) - \beta, \tag{8}$$

where $\beta$ is identical to the difference of the two determinants in Theorem 1. Therefore, the Theorema egregium of Gauss is now rediscovered automatically (under the condition $M \not= 0$). Moreover, the d-pol set$\mathbb{P}$ may be decomposed into 12 regular d-tri systems. It can be easily verified that $g$ has d-pseudo-remainder 0 wrt all the 12 d-tri sets and thus $g$ vanishes on the zeros of all these d-tri systems. Hence, the Theorema egregium holds true as well in the case $M=0$.

## 5.2 Relations between First and Second Fundamental Coefficients

In the process of computing integrability conditions with $F = M = 0$, Li [17] discovered a polynomial of degree 4 in $L$, whose coefficients are d-pols in $E$ and $G$, and thus concluded that the second fundamental form is algebraically determined by $E, G$ and their derivatives. This d-pol consists of 434 terms and thus is quite complex. It is difficult to analyze its geometric meaning. Using a slightly different way of elimination, we show how to derive a more general relation in a very compact form. This compact form consists of only 11 terms and thus provides us with some possibility to investigate its geometric significance. We shall clarify the statement of Li and give symmetric expressions for this special case. Moreover, a quadratic relation of $L$ has also been obtained.

Consider the case $M = 0$. Then the Codazzi–Mainardi equations become

$$\begin{aligned} c_1 &= L_v - L\,\mathit{\Gamma}^1_{12} + N\,\mathit{\Gamma}^2_{11} = 0,\\ c_2 &= N_u + L\,\mathit{\Gamma}^1_{22} - N\,\mathit{\Gamma}^2_{12} = 0. \end{aligned}$$

Let $\gamma = \beta/\delta$; then (8) implies that

$$c = LN - \gamma = 0.$$

Now eliminate the variable $N$ from $c_1$ and $c_2$ using $c$ by means of d-pseudo-division; we have

$$p_1 = \text{d-prem}(c_1, c), \quad p_2 = \text{d-prem}(c_2, c).$$

Next, compute the $\mathit{\Delta}$-pol $p_3$ of $p_1$ and $p_2$ (whose leads are $L_v$ and $L_u$ respectively). Finally, eliminating the derivatives $L_u$ and $L_v$ from $p_3$ using $p_1$ and $p_2$ by d-pseudo-division, we get

$$r = \text{d-prem}(p_3, \langle p_1, p_2\rangle),$$

which is a d-pol of degree 4 in its lead $L$ and does not contain terms of odd degrees in $L$. Therefore, the following relation is derived:

$$r = r_4L^4 + r_2L^2 + r_0 = 0,$$

where

$$\begin{aligned} r_4 &= 2\,\gamma\mathit{\Gamma}^1_{12}\mathit{\Gamma}^1_{22} - \gamma_v\mathit{\Gamma}^1_{22} + \gamma(\mathit{\Gamma}^1_{22})_v,\\ r_2 &= -[\gamma^2(\mathit{\Gamma}^1_{12})_u + \gamma^2(\mathit{\Gamma}^2_{12})_v + 4\,\gamma^2\mathit{\Gamma}^2_{11}\mathit{\Gamma}^1_{22} + \gamma_u\gamma_v - \gamma\gamma_{uv}],\\ r_0 &= \gamma^3(\mathit{\Gamma}^2_{11})_u - \gamma^2\gamma_u\mathit{\Gamma}^2_{11} + 2\,\gamma^3\mathit{\Gamma}^2_{11}\mathit{\Gamma}^2_{12}. \end{aligned}$$

In fact, the relation $r = 0$ can be derived automatically by triangularizing the d-pol set $\mathbb{H} = \{c_1, c_2, c\}$ with $L, N, \gamma$ and $\mathit{\Gamma}^k_{ij}$ as variables. For this purpose, order the derivatives of $\mathit{\Gamma}^1_{22} \prec \mathit{\Gamma}^2_{11} \prec \mathit{\Gamma}^2_{12} \prec \mathit{\Gamma}^1_{12} \prec \gamma$ according to $\mathit{\Omega}$ and to be smaller than any derivative of $L$, which is ordered smaller than any derivative of $N$. Then the d-pol $r$ as well as $f$ given below appears in the process of decomposing $\mathbb{H}$ into regular d-tri systems. However, the complete decomposition of $\mathbb{H}$ could not be obtained due to the occurrence of very large d-pols.

After $\mathit{\Gamma}^k_{ij}$ and $\gamma$ are substituted by their corresponding expressions, the numerator of $r$, when expanded, is a d-pol consisting of 17 666 terms in $E, F, G$ and their derivatives. In the case $F = 0$, $r$ simplifies to 434 terms, yielding the d-pol found by Li [17].

From $r = 0$ and $c = 0$, the following theorem may be established.

**Theorem 3.** *For any surface with $M = 0$, either $r_4 = r_2 = r_0 = 0$, or the second fundamental coefficient $L$ is algebraically determined by the first fundamental coefficients $E, F, G$ and their derivatives. In the latter case, if $L \neq 0$, then $N$ and thus the second fundamental form are also algebraically determined by $E, F, G$ and their derivatives.*

Moreover, pseudo-dividing $p_1$ by $r$, we obtain the following quadratic relation

$$f = \text{d-prem}(p_1, r) = \gamma(f_2 L^2 + f_0) = 0,$$

where $f_2$ and $f_0$ are d-pols consisting of 46 and 43 terms, respectively, in $\gamma, \Gamma^1_{12}, \Gamma^2_{11}, \Gamma^1_{22}, \Gamma^2_{12}$ and their derivatives. Therefore, we have the following new theorem.

**Theorem 4.** *For any surface with $M = 0$, either $\gamma f_2 = \gamma f_0 = 0$, or the square of the second fundamental coefficient $L$ is a rational expression of the first fundamental coefficients $E, F, G$ and their derivatives, viz.,*

$$L^2 = -\frac{f_0}{f_2}.$$

*If $\gamma f_0 \neq 0$, then $N$ and thus the second fundamental form are algebraically determined by $E, F, G$ and their derivatives.*

Note that $f$ was first discovered from the triangularization process of decomposing $\mathbb{H}$. We realize that this d-pol can be easily obtained as the d-pseudo-remainder of $p_1$ wrt $r$ after we have already seen it. After substitution of the expressions of $\gamma, \Gamma^1_{12}$, etc., the numerator of $f_2 L^2 + f_0$, if expanded, is a very large d-pol in $E, F, G$ and their derivatives. When $F = 0$, the numerator of $f_2 L^2 + f_0$ simplifies to 3 696 terms.

### 5.3 The Case $F = M = 0$

Now we come to the special case $F = M = 0$. Then the d-pols $r_4, r_2, r_0$ can be written as

$$\begin{aligned}
r_4 &= -\frac{1}{2E}(EGKG_{uv} - GKE_vG_u - EKG_vG_u - EGG_uK_v),\\
r_2 &= -\frac{1}{2}(2EGK^2E_vG_u - EG^2K^2E_{uv} + G^2K^2E_uE_v - E^2GK^2G_{uv}\\
&\qquad + E^2K^2G_uG_v - 2E^2G^2KK_{uv} + 2E^2G^2K_uK_v),\\
r_0 &= -\frac{E^2GK^2}{2}(EGKE_{uv} - GKE_uE_v - EKE_vG_u - EGK_uE_v),
\end{aligned}$$

where the Gaussian curvature simplifies to

$$K = \frac{\gamma}{EG} = \left(\frac{E_uG_u + E_v^2}{4E} + \frac{E_vG_v + G_u^2}{4G} - \frac{E_{vv} + G_{uu}}{2}\right) \Big/ (EG).$$

**Theorem 5.** *For any surface with $F = M = 0$, one of the following holds:*

*(a)* $G_u K = 0$;

*(b)* $G_u K \not\models 0$ *and* $\dfrac{G_{uv}}{G_u} = \dfrac{E_v}{E} + \dfrac{G_v}{G} + \dfrac{K_v}{K}$;

*(c) the second fundamental coefficient $L$ is algebraically determined by the first fundamental coefficients $E, G$ and their derivatives; if $L \not\models 0$ then $N$ and thus the second fundamental form are also algebraically determined by $E, G$ and their derivatives.*

In the case $E_v G_u K \not\models 0, r_4$, $r_2$, and $r_0$ can be written in the following symmetric form:

$$r_4 = -\frac{GKG_u}{2}\left(\frac{G_{uv}}{G_u} - \frac{E_v}{E} - \frac{G_v}{G} - \frac{K_v}{K}\right),$$
$$r_2 = -\frac{(EGK)^2}{2}\left[2\frac{E_v}{E}\frac{G_u}{G} - \left(\frac{E_u}{E}\right)_v - \left(\frac{G_u}{G}\right)_v - 2\left(\frac{K_u}{K}\right)_v\right],$$
$$r_0 = -\frac{E^3G^2K^3E_v}{2}\left(\frac{E_{uv}}{E_v} - \frac{E_u}{E} - \frac{G_u}{G} - \frac{K_u}{K}\right).$$

Of course, if $r_4 = 0$ and $r_2 \not\models 0$, then$L$ is also algebraically determined by $E, G$ and their derivatives. In the case $r_4 = r_2 = r_0 = 0$, computation shows that in general there does not exist any polynomial relation among $L, E, G$ and the derivatives of $E$ and $G$, that follows formally from the equations (1)–(4).

Now an interesting question is how to geometrically characterize the three classes of surfaces. The first two cases, which were not considered in Li's conclusion, are not trivial. For example, we have looked at the following three families of surfaces. The first is the general surface of revolution

$$\boldsymbol{r}(u, v) = (h(u)\cos v, h(u)\sin v, k(u)),$$

where it is assumed that $h'^2 + k'^2 \not\models 0$ and$h \not\models 0$. The surface includes the sphere with $h(u) = a\cos u$ and $k(u) = a\sin u$, the ellipsoid (Fig. 2) with $h(u) = a\cos u$ and $k(u) = b\sin u$, and the torus (Fig. 3) with $h(u) = a + b\cos u$ and $k(u) = b\sin u$ as special cases. For this surface, we have $E_v = G_v = K_v = G_{uv} = 0$, so it belongs to class $(b)$.

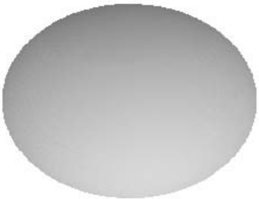


Fig. 2

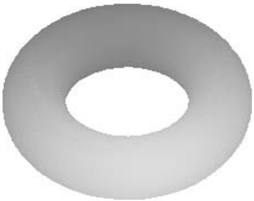


Fig. 3

Another surface is the helicoid (Fig. 4)

$$\boldsymbol{r}(u, v) = (v \cos u, -v \sin u, bu),$$

which is not a surface of revolution. For this surface, $G_u = 0$ and thus case $(a)$ holds.

Obviously, all surfaces of zero Gaussian curvature (i.e., $K = 0$) also belong to class $(a)$. The cylinder (Fig. 5) is one of such surfaces. Moreover, for all the above examples we have $r_4 = r_2 = r_0 = 0$.

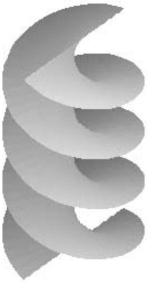


Fig. 4

On the other hand, it may be verified that $r_i = 0$ $(i = 4, 2, 0)$ and $f_j = 0$ $(j = 2, 0)$ are not formal consequences of the equations (1)–(4). However, this fact does not imply that there must exist surfaces such that $r_i \not= 0$ or$f_j \not= 0$. The reason is that the relations between the fundamental coefficients $E, F, G, L, M, N$ and the vectors $\boldsymbol{r}_u, \boldsymbol{r}_v, \boldsymbol{n}$ expressed by means of the inner product are not taken into account in the hypothesis. To include these relations, we need to take the components of the vectors $\boldsymbol{r}_u, \boldsymbol{r}_v$ and $\boldsymbol{n}$ as variables. This leads to a considerable increase in the number of variables and of equations and thus makes the computation much more complex. We have proved that, in the case $F = M = 0$, the d-pols $r_4, r_2, r_0$ do not belong to the radical of the differential ideal generated by the d-pols obtained from (1)–(4) and the relations between $E, G, L, N$ and $\boldsymbol{r}_u, \boldsymbol{r}_v, \boldsymbol{n}$ by taking the components of the involved vectors as variables. We have also observed that the regular d-tri systems whose d-zeros do not make the vanishing of $r_4, r_2, r_0$ represent some nontrivial components of the zero set. So there should indeed exist nontrivial surfaces that belong to class $(c)$. The remaining problem is to find such surfaces and their geometric characterization.

## 6 Proving Theorems about Surfaces

As shown in the preceding section, deriving an unknown geometric relation or discovering a new theorem may be done by triangularizing the geometric hypotheses with specially arranged variable (and term) orderings. The case of proving a geometric theorem is relatively easy because the conclusion-relations are given. Theoretically, the theorem may be proved by verifying that the conclusion-d-pols vanish on the d-zeros of the d-pol set $\mathbb{P}$ expressing the geometric hypotheses. However, this theoretical approach does not work well because geometric theorems are true usually under certain nondegeneracy conditions. It is a good strategy to introduce d-pol inequations to rule out some of the degenerate cases. Thus, in practice we should formulate the hypothesis of the geometric theorem in question as a d-pol system $\langle \mathbb{P}, \mathbb{Q} \rangle$ and decide on which part of d-Zero$(\mathbb{P}/\mathbb{Q})$ the conclusion-d-pols vanish. This can be done by decomposing the hypothesis-d-pol system so that its zero set is split into subsets (corresponding to the geometric situations), and then deciding on which subsets the conclusion-d-pols vanish.

We have applied the zero decomposition techniques reviewed in Sect. 3 to d-pol systems formulated from problems in the local theory of surfaces, experimenting with several theorems selected from standard textbooks of differential geometry and the literature of automated geometric reasoning to see the effectiveness of differential elimination methods. The following examples (of which the first is taken from [17]) are two of them and serve to illustrate how such theorems can be proved automatically.

**Example 1.** If a surface $\boldsymbol{r} = \boldsymbol{r}(u, v)$ has two unequal constant principal curvatures, then the surface is a cylinder.

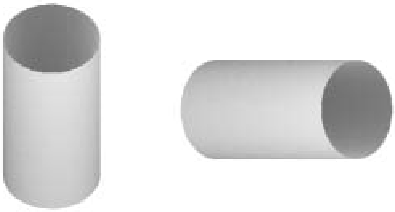


Fig. 5

With the formulation given by Li [17], the hypothesis of this theorem consists of the following seven d-pol equations

$$\begin{aligned}
&p_1 = L_u E - L E_u = 0, \quad p_2 = L_v E - L E_v = 0, \quad p_3 = N_u G - N G_u = 0,\\
&p_4 = N_v G - N G_v = 0, \quad p_5 = 2\, N_u G E - N G_u E - L G_u G = 0,\\
&p_6 = 4\, N L G E + 2\, G_{uu} G E - G_v E_v E - G_u^2 E - G_u G E_u + 2\, G E_{vv} E - G E_v^2 = 0,\\
&p_7 = 8\, L_v L G^2 E - 4\, L^2 G^2 E_v + 2\, G_{uu} G E_v E - G_v E_v^2 E - G_u^2 E_v E - G_u G E_v E_u\\
&\qquad + 2\, G E_{vv} E_v E - G E_v^3 = 0,
\end{aligned}$$

and the inequation $q = EN - GL \neq 0$ (which means that the two principal curvatures are unequal). Let $\mathbb{P} = \{p_1, \ldots, p_7\}$, $\mathbb{Q} = \{E, G, q\}$, and the derivatives of $E \prec G \prec L \prec N$ be ordered according to $\Omega$. Using the refined Seidenberg algorithm, the d-pol system $\langle \mathbb{P}, \mathbb{Q} \rangle$ may be decomposed into two regular d-tri systems $\langle \mathbb{T}_1, \mathbb{U}_1 \rangle$ and $\langle \mathbb{T}_2, \mathbb{U}_2 \rangle$ with

$$\begin{aligned}
\mathbb{T}_1 &= [N, E_v, G_u, E L_v - L E_v, E L_u - L E_u],\\
\mathbb{T}_2 &= [L, G_u, G N_v - N G_v, 2\, E G N_u - E N G_u - G L G_u,\\
&\qquad - 2\, E G E_{vv} + G E_v^2 + E E_v G_v - 4\, G^2 L^2],\\
\mathbb{U}_1 &= \{E, G, L\}, \quad \mathbb{U}_2 = \{E, G, N\}
\end{aligned}$$

such that (6) holds with $e = 2$. Considering the two d-tri systems as e-tri systems and decomposing the first under $N \prec E_v \prec G_u \prec L_v \prec L_u$ (the ordered leads) and the second under $L \prec G_u \prec N_v \prec N_u \prec E_{vv}$, we may get two simple systems $\langle \mathbb{T}_1, \varnothing \rangle$ and $\langle \mathbb{T}_2, \varnothing \rangle$. Therefore, we have

$$\text{d-Zero}([\mathbb{P}] : \mathbb{Q}^\infty) = \text{d-Zero}([\mathbb{T}_1] : \mathbb{V}_1^\infty) \cup \text{d-Zero}([\mathbb{T}_2] : \mathbb{V}_2^\infty), \qquad (9)$$

where $\mathbb{V}_i$ is the set of factors of the d-pols in $\mathrm{si}(\mathbb{T}_i)$: $\mathbb{V}_1 = \{E\}$ and $\mathbb{V}_2 = \{E, G\}$. By means of computing Gröbner bases as proposed by Boulier and others [3,4], one may also get two simple sets

$$\begin{aligned}
\mathbb{T}_1^* &= [N, E_v, G_u, L_v, E L_u - L E_u],\\
\mathbb{T}_2^* &= [L, G_u, G N_v - N G_v, N_u, 2\, E G E_{vv} - G E_v^2 - E E_v G_v]
\end{aligned}$$

such that (9) holds as well when $\mathbb{T}_i$ is substituted by $\mathbb{T}_i^*$ for $i = 1, 2$.

The conclusion of the theorem to be proved may be given by five d-pol equations (see [17]). According to the reduction techniques mentioned in Sect. 3.3, one may easily verify that the conclusion d-pols can be reduced to 0 wrt $\mathbb{T}_1$ or $\mathbb{G}_1$, but not wrt $\mathbb{T}_2$ or $\mathbb{G}_2$. In other words, the theorem is proved to be true for *one* of the two components. The reason is that the conclusion is formulated only for one of two cases shown in Fig. 5. For the other case, we need to swap the positions of $u$ and $v$ in the conclusion d-pols as remarked by Li [17]. The resulting d-pols may be reduced to 0 wrt $\mathbb{T}_2$ or $\mathbb{G}_2$. In any case, the surface is a cylinder under the condition $EG \not= 0$; this condition is already satisfied because $EG > 0$.

**Example 2.** If a surface $\boldsymbol{r} = \boldsymbol{r}(u, v)$ consists entirely of umbilics, then it is planar or spherical.

A point $P$ on the surface is called an *umbilic* if the two principal curvatures at $P$ are equal. At every umbilic we have

$$\frac{L}{E} = \frac{M}{F} = \frac{N}{G}.$$

Take the d-pol equations (1)–(4) together with $EM - FL = 0$ and $EN - GL = 0$ as the hypothesis of the theorem, denote the corresponding set of d-pols by $\mathbb{P}$, and let $\mathbb{Q} = \{E, G, \delta\}$, where $\delta = EG - F^2$ as before. Order the derivatives of $E \prec F \prec G \prec L \prec M \prec N \prec \boldsymbol{r} \prec \boldsymbol{n}$ according to $\Omega$.

If $L = 0$, then it is easy to decompose $\langle \mathbb{P} \cup \{L\}, \mathbb{Q}\rangle$ into one regular d-tri system $\langle \mathbb{T}, \mathbb{U}\rangle$, and one may see that both $\boldsymbol{n}_u$ and $\boldsymbol{n}_v$ are contained in $\mathbb{T}$. Therefore, $\boldsymbol{n}$ is a constant vector and the surface is planar.

Now consider the case $L \not= 0$. Then decomposition of$\langle \mathbb{P}, \mathbb{Q}\rangle$ as of the form (6) yields two regular d-tri systems $\langle \mathbb{T}_1, \mathbb{U}_1\rangle$ and $\langle \mathbb{T}_2, \mathbb{U}_2\rangle$, where

$$\begin{aligned} &\mathbb{T}_1 = \langle T_1, \ldots, T_{10}\rangle, && \mathbb{T}_2 = \langle F, M, T_2, T_3, T_4', T_5, T_6, T_7', T_8', T_9', T_{10}'\rangle,\\ &\mathbb{U}_1 = \mathbb{Q} \cup \{F\}, && \mathbb{U}_2 = \{E, G\}, \end{aligned}$$

and

$$\begin{aligned} T_1 &= EM - FL,\\ T_2 &= EN - GL,\\ T_3 &= EL_v - LE_v,\\ T_4 &= E^2L_v - ELE_v - EFL_u + LFE_u,\\ T_5 &= E\boldsymbol{n}_v + L\boldsymbol{r}_v,\\ T_6 &= E\boldsymbol{n}_u + L\boldsymbol{r}_u,\\ T_7 &= E^2[2\,\delta(G_{uu} - 2\,F_{uv} + E_{vv}) - EG_u^2 - GE_v^2 + 4\,G^2L^2\\ &\quad + (2\,FF_u - GE_u + FE_v)G_u + (2\,EF_u - FE_u - EE_v)G_v\\ &\quad - 2\,(2\,FF_u - GE_u - FE_v)F_v] - 4\,(2\,EG - F^2)F^2L^2,\\ T_8 &= E[2\,\delta\boldsymbol{r}_{vv} + (GG_u + FG_v - 2\,GF_v)\boldsymbol{r}_u - (FG_u + EG_v - 2\,FF_v)\boldsymbol{r}_v]\\ &\quad - 2\,LG\delta\boldsymbol{n}, \end{aligned}$$

$$
\begin{aligned}
T_9 &= E[2\,\delta\boldsymbol{r}_{uv} + (FG_u - GE_v)\boldsymbol{r}_u - (EG_u - FE_v)\boldsymbol{r}_v] - 2\,LF\delta\boldsymbol{n},\\
T_{10} &= 2\,\delta\boldsymbol{r}_{uu} + (2\,FF_u - GE_u - FE_v)\boldsymbol{r}_u - (2\,EF_u - FE_u - EE_v)\boldsymbol{r}_v\\
&\qquad - 2\,L\delta\boldsymbol{n};\\
T_4' &= EL_u - LE_u,\\
T_7' &= 2\,EG(G_{uu} + E_{vv}) - EG_u^2 - GE_v^2 - GE_uG_u - EE_vG_v + 4\,G^2L^2,\\
T_8' &= 2\,EG\boldsymbol{r}_{vv} + GG_u\boldsymbol{r}_u - EG_v\boldsymbol{r}_v - 2\,G^2L\boldsymbol{n},\\
T_9' &= 2\,EG\boldsymbol{r}_{uv} - GE_v\boldsymbol{r}_u - EG_u\boldsymbol{r}_v,\\
T_{10}' &= 2\,EG\boldsymbol{r}_{uu} - GE_u\boldsymbol{r}_u + EE_v\boldsymbol{r}_v - 2\,EGL\boldsymbol{n}.
\end{aligned}
$$

In fact, $\mathbb{T}_1$ is a simple set and $T_4$ may be replaced by $T_4'$. Let

$$k = \frac{L}{E}, \quad \boldsymbol{f} = \frac{\boldsymbol{n}}{k} + \boldsymbol{r}.$$

Then one can easily verify that

$$\text{d-prem}(k_u, \mathbb{T}_i) = \text{d-prem}(k_v, \mathbb{T}_i) = \text{d-prem}(\boldsymbol{f}_u, \mathbb{T}_i) = \text{prem}(\boldsymbol{f}_v, \mathbb{T}_i) = 0$$

for $i = 1, 2$. It follows that $k$ is a nonzero constant and $\boldsymbol{f}$ is equal to a constant vector $\boldsymbol{r}_0$, and thus

$$\boldsymbol{r} - \boldsymbol{r}_0 = -\frac{\boldsymbol{n}}{k}.$$

Therefore, we have

$$|\boldsymbol{r} - \boldsymbol{r}_0| = \frac{|\boldsymbol{n}|}{|k|} = \frac{1}{|k|}.$$

This proves that the surface is spherical.

For differential geometry, the algebraic formulation of problems is not always easy and straightforward. We are looking for more geometric theorems that are nontrivial and instructive, may be formulated in the setting of d-pol equations and inequations, and are computationally tractable. We have also been observing the performance of decomposition methods with different variants and the computational difficulties involved for theorem proving. Not as easy as in elementary geometry, we often encounter very large d-pols when dealing with problems about surfaces. The computation is highly sensitive to the variable and term orderings. How to design more efficient decomposition algorithms and how to apply them effectively to attack problems in the local and global theory of surfaces have become interesting questions for our research.

## Acknowledgements

Part of this work has been supported by the SPACES Project (http://www-spaces.lip6.fr/) and by the Chinese National 973 Project NKBRSF G19980306.

## References

1. Aubry, P. (1999). *Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom.* Ph.D. thesis, Université Paris VI, France.
2. Aubry, P., Lazard, D., Moreno Maza, M. (1999). On the theories of triangular sets. *J. Symb. Comput.* **28**: 105–124.
3. Boulier, F., Lazard, D., Ollivier, F., Petitot, M. (1995). Representation for the radical of a finitely generated differential ideal. In: Levelt, A. H. M. (ed.): *Proc. ISSAC '95*, Montreal, Canada, ACM Press, New York, pp. 158–166.
4. Boulier, F., Lazard, D., Ollivier, F., Petitot, M. (1998). Computing representation for radicals of finitely generated differential ideals. Preprint, LIFL, Université Lille I, France.
5. Boulier, F., Lemaire, F. (2000). Computing canonical representatives of regular differential ideals. In: Traverso, C. (ed.): *Proc. ISSAC '2000*, St. Andrews, Scotland, ACM Press, New York, pp. 38–47.
6. Buchberger, B. (1985). Gröbner bases: An algorithmic method for polynomial ideal theory. In: Bose, N. K. (ed): *Multidimensional Systems Theory*, D. Reidel, Dordrecht, pp. 184–232.
7. Carrà Ferro, G., Gallo, G. (1990). A procedure to prove statements in differential geometry. *J. Automat. Reason.* **6**: 203–209.
8. Chou, S.-C., Gao, X.-S. (1993). Automated reasoning in differential geometry and mechanics using the characteristic set method — Part I. An improved version of Ritt–Wu's decomposition algorithm. Part II. Mechanical theorem proving. *J. Automat. Reason.* **10**: 161–189.
9. Hubert, E. (1999). The *diffalg* package. http://daisy.uwaterloo.ca/~ehubert/Diffalg/.
10. Hubert, E. (2000). Factorization-free decomposition algorithms in differential algebra. *J. Symb. Comput.* **29**: 641–662.
11. Kalkbrener, M. (1991). *Three Contributions to Elimination Theory.* Ph.D. thesis, Johannes Kepler University, Austria.
12. Kalkbrener, M. (1998). Algorithmic properties of polynomial rings. *J. Symb. Comput.* **26**: 525–581.
13. Klingenberg, W. (1978). *A Course in Differential Geometry.* Translated by D. Hoffman. Springer, New York.
14. Kreyszig, E. (1968). *Introduction to Differential Geometry and Riemannian Geometry.* University of Toronto Press, Tornoto.
15. Li, H. (1997). Mechanical theorem proving in differential geometry — Local theory of surfaces. *Sci. China* (Ser. A) **40**: 350–356.
16. Li, H., Cheng, M. (1998). Clifford algebraic reduction method for automated theorem proving in differential geometry. *J. Automat. Reason.* **21**: 1–21.
17. Li, Z. (1995). Mechanical theorem proving in the local theory of surfaces. *Ann. Math. Artif. Intell.* **13**: 25–46.
18. Li, Z., Wang, D. (1999). Coherent, regular and simple systems in zero decompositions of partial differential systems. *Syst. Sci. Math. Sci.* **12** (Suppl.): 43–60.
19. Ritt, J. F. (1950). *Differential Algebra.* Amer. Math. Soc., New York.
20. Rosenfeld, A. (1959). Specializations in differential algebra. *Trans. Amer. Math. Soc.* **90**: 394–407.
21. Seidenberg, A. (1956). An elimination theory for differential algebra. *Univ. California Publ. Math.* (N.S.) **3**(2): 31–66.

22. Wang, D. (1995). A method for proving theorems in differential geometry and mechanics. *J. Univ. Comput. Sci.* **1**: 658–673.
23. Wang, D. (1996). An elimination method for differential polynomial systems I. *Syst. Sci. Math. Sci.* **9**: 216–228.
24. Wang, D. (1998). Decomposing polynomial systems into simple systems. *J. Symb. Comput.* **25**: 295–314.
25. Wang, D. (2000). Automated reasoning about surfaces (progress report). In: Richter-Gebert, J., Wang, D. (eds.): *Proc. ADG 2000*, Zurich, Switzerland, September 25–27, 2000, pp. 183–196.
26. Wu, W.-t. (1979). On the mechanization of theorem-proving in elementary differential geometry (in Chinese). *Sci. Sinica* Special Issue on Math. (I): 94–102.
27. Wu, W.-t. (1987). A constructive theory of differential algebraic geometry based on works of J. F. Ritt with particular applications to mechanical theorem-proving of differential geometries. In: Gu, C., Berger, M., Bryant, R. L. (eds.): *Differential Geometry and Differential Equations*, LNM **1255**, Springer, Berlin, pp. 173–189.
28. Wu, W.-t. (1989). On the foundation of algebraic differential geometry. *Syst. Sci. Math. Sci.* **2**: 289–312.
29. Wu, W.-t. (1991). Mechanical theorem proving of differential geometries and some of its applications in mechanics. *J. Automat. Reason.* **7**: 171–191.

# Effective Methods in Computational Synthetic Geometry

Jürgen Bokowski

Darmstadt University of Technology
Department of Mathematics
D-64289 Darmstadt
juergen@bokowski.de

**Abstract.** We discuss algorithmic steps when dealing with realizability problems in discrete geometry, especially that of finding realizations for a given oriented matroid. After a brief introduction to known methods, we discuss a dynamic inductive realization method, which has proven successful when other methods did not succeed. A useful theorem in this context in the rank 3 case asserts that a one-element extension of a uniform rank 3 oriented matroid depends essentially just on the mutations involving that element. There are problems in computational synthetic geometry of course, where intuition must help. In this context we mention the application of the software *Cinderella* to automated deduction in computational synthetic geometry, when studying face lattices of polytopes.

## 1 Introduction

We start with a motivation for studying oriented matroids. When using an $(n \times r)$-matrix to describe a geometrical object in Euclidean space, we are actually often interested in the equivalence class of all matrices that describe images of our object under rigid motions. Oriented matroids form a natural framework for such equivalence classes. They are even invariants of the corresponding projective space with respect to homeomorphic transformations. For the novice in the theory of oriented matroids, we recommend to think of an oriented matroid with $n$ elements in rank $r$ as an equivalence class of $(n \times r)$-matrices. In each such equivalence class we collect matrices that agree on certain combinatorial properties, about the relative position of the row vectors of the matrix. So, in principle, oriented matroids are purely combinatorial objects that mimic the "relative-position behavior" of vectors. While real matrices provide a kind of realizable paradigm for oriented matroids, there are, moreover, also oriented matroids that do not come from a matrix (still sharing the same abstract combinatorial properties). They are called *non-realizable.* One model of an oriented matroid with $n$ elements in rank $r$ describes it as an abstract sign vector with $\binom{n}{r}$ components. The components are indexed by the $\binom{n}{r}$ $r$-tuples $(\lambda_1, \lambda_2, \ldots, \lambda_r)$ with $1 \leq \lambda_1 < \lambda_2 < \ldots < \lambda_r \leq n$ and in case of an $(n \times r)$-matrix $\mathcal{M}$, $n \geq r$, the

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 175–192, 2001.

component corresponding to $(\lambda_1, \lambda_2, \dots, \lambda_r)$ equals the sign of the determinant of the sub-matrix of $\mathcal{M}$ with rows $\lambda_1, \lambda_2, \dots, \lambda_r$. While all realizable oriented matroids can be generated in this way, a general oriented matroid is such a sign vector that satisfies certain conditions of *local realizability.* We omit the technical description of these criteria here and refer to [3] for an elaborate treatment of oriented matroid axioms.

One essential property of oriented matroids is that many geometric questions that are usually treated on the coordinate level can be dealt with as well on the combinatorial level of a suitably chosen oriented matroid. For instance the convex hull of a point configuration $P$ can be calculated from the oriented matroid of the matrix of the homogeneous coordinates. Similarly this oriented matroid carries enough information to decide whether a given simplicial complex can be embedded without self-intersections on the vertices of $P$. This enables us to break up the process of deciding the realizability of a simplicial complex $\Delta$ into a two-step procedure. First one enumerates (on a combinatorial level) all oriented matroids that are compatible with $\Delta$. Then one tries to realize at least one of them. So, in principle a simplicial complex may be non-realizable for two different reasons. Either no oriented matroid was found (this is a purely combinatorial statement) or the possible oriented matroids turned out to be non-realizable (this is a geometric statement).

The notion *Computational Synthetic Geometry* has been introduced in [4,24]. It deals with realizability problems in discrete geometry, see also [14]. An essential subproblem in this field is that of finding realizations for given oriented matroids (i.e. given the sign vector, find a matrix that has this vector as oriented matroid). By observing that determinants are a special kind of polynomials, the realizability problem turns out to be a semialgebraic problem of finding solutions of a system of real polynomial equations and inequalities. A first naive attempt to the realizability problem would therefore try to apply the more or less standard techniques and algorithms of real algebraic geometry. However the general complexity behavior of this problem is known to be intrinsically hard and a general algorithm is far from being applicable for practical purposes, see [14,3]. One might think that there is some hope that the realizability problem for oriented matroids turns out to be simpler than general semialgebraic problems, since determinants are very special polynomials. However it turns out that this is not the case: the universality theorems of Mnëv [17] show that the realizability problem and the problem of solving systems of polynomial inequalities are essentially equivalent. See also the closely related results about realization spaces of polytopes of Richter-Gebert [21]. On the other hand, several heuristic methods for deciding the realizability of oriented matroids have been applied successfully in the past. Starting with a brief introduction to the known methods, we discuss an additional method, a dynamic inductive realization method, which has proven successful when other methods did not succeed. We provide in particular a useful theorem in this context in the rank 3 case which asserts that the one-element extension of a uniform rank 3 oriented matroid depends essentially just on the *mutations* (see below) involving that element.

There are problems in computational synthetic geometry of course, where no formal heuristics did provide a solution and intuition must help. In this context we mention the application of the software *Cinderella* for automated deduction in computational synthetic geometry, when studying face lattices of polytopes.

We start with a typical example in computational synthetic geometry: the realization problem for a triangulated 2-manifold. We exemplify how to proceed algorithmically. A fundamental subproblem will be that of finding coordinates of a given oriented matroid. As mentioned above we know that a general algorithm for this problem is unsuitable for practical applications. So we have to look at fast heuristic methods instead.

*Example 1.* We consider the following abstract list of triangles, an orientable triangulated 2-manifold:

125 127 137 138 146 148 156 236 238 245 248 267 345 347 356 467

**Input:** *A triangulated combinatorial torus.*

**Problem.** Can we find 8 corresponding points in Euclidean 3-space such that the flat triangles defined via the above list have no self-intersections?

**Step 0.** *Finding an admissible oriented matroid.*

Assume for a moment that we have found such 8 points, and write their homogeneous coordinates as an $8 \times 4$ matrix $\mathcal{M}$. One can compute the signs of all determinants of $4 \times 4$ submatrices of $\mathcal{M}$. The sign of such a quadruple tells us whether the corresponding four points form a left- or a right-handed system in space. When we consider a collection of 5 rows of this matrix $\mathcal{M}$ corresponding to the vertices of a triangle and a line segment of our example, the corresponding 5 signs of determinants of $4 \times 4$ submatrices tell us whether the edge pierces the triangle or not. We ask the reader to confirm this.

Having this is mind, we look for a sign structure that satisfies two properties. On the one hand it should have the chance to be the sign structure of the determinants of $4 \times 4$ submatrices of a matrix $\mathcal{M}$ (i.e. we look for an oriented matroid) and on the other hand, the sign structure should not violate any of the intersection properties that are forced by the simplicial complex. In other words, we determine an admissible oriented matroid in rank 4. We skip here the method for generating them. An effective algorithm can be found in [8]. The output of the algorithm can be one of two possibilities:

**Either:** *The set of admissible oriented matroids is empty.*

If this were the case, it would be impossible to embed the example under consideration.

**Or:** *Admissible oriented matroids were found.*

If this were the case, the embedability of the example depends on the question whether at least one of these oriented matroids is realizable. If there is more than one admissible oriented matroid, then the torus is not embeddable if and only if *all of them* are non-realizable.

In our torus case, we have found the following admissible oriented matroid with 8 elements in rank 4 written in terms of signed bases.

$$\begin{array}{cccccccccc}
-1234 & -1235 & -1236 & -1237 & +1238 & +1245 & +1246 & +1247 & -1248 & +1256 \\
+1257 & -1258 & +1267 & -1268 & -1278 & -1345 & -1346 & -1347 & +1348 & +1356 \\
+1357 & +1358 & +1367 & +1368 & +1378 & -1456 & -1457 & +1458 & -1467 & +1468 \\
-1478 & +1567 & -1568 & -1578 & -1678 & +2345 & +2346 & +2347 & +2348 & +2356 \\
+2357 & +2358 & +2367 & +2368 & +2378 & +2456 & +2457 & -2458 & +2467 & -2468 \\
-2478 & -2567 & -2568 & -2578 & -2678 & +3456 & +3457 & +3458 & +3467 & +3468 \\
+3478 & -3567 & -3568 & +3578 & +3678 & -4567 & +4568 & +4578 & +4678 & -5678
\end{array}$$

Now the problem reduces to finding coordinates of points that generate exactly the sign pattern of the oriented matroid. Any such realization will automatically be an embedding of the torus under consideration.

We use the following method when the problem is small enough (i.e. when we can be optimistic about finding a decision for the forthcoming inequality system). The advantage of the following method is that we do not give up the full generality. Nevertheless, we can also start with any other alternative mentioned later.

**Step 1.** *Careful analysis to determine the unit matrix.*

Since the oriented matroid of our matrix $\mathcal{M}$ is invariant under multiplication by a $4 \times 4$ matrix with positive determinant, we can assume that the rows 1, 3, 5 and 7 form a unit matrix. In our oriented matroid the sign $[1, 3, 5, 7]$ is positive, which is consistent with the sign of our chosen basis. If this were not the case we would have w.l.o.g. to realize the oriented matroid where all signs are reversed.

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ A & B & C & D \\ 0 & 1 & 0 & 0 \\ E & F & G & H \\ 0 & 0 & 1 & 0 \\ I & J & K & L \\ 0 & 0 & 0 & 1 \\ M & N & O & P \end{pmatrix} \qquad \mathcal{M}' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a & b & c & -d \\ 0 & 1 & 0 & 0 \\ -e & -f & -g & h \\ 0 & 0 & 1 & 0 \\ -i & -j & k & l \\ 0 & 0 & 0 & 1 \\ -m & -n & -o & p \end{pmatrix}$$

Having chosen certain rows to be a unit matrix, considerably simplifies the algebraic structure of the system of inequalities. While we before had to deal with a system of many $4 \times 4$ determinants, now (due to all the zeros in the matrix) many determinants can be expressed as $1 \times 1$, $2 \times 2$, or $3 \times 3$ determinants of suitable sub-matrices. In particular the variables $A \ \ldots \ P$ themselves can be expressed as a determinant (e.g. $P = [1, 3, 5, 8]$). By this the oriented matroid itself determines the signs of the variables. The signs of all variables are determined when we know the signs of the following determinants: [2357], [1257], [1237], [1235], [3457], [1457], [1347], [1345], [3567], [1567], [1367], [1356], [3578], [1578], [1378], [1358].

Now we substitute all negative variables $v$ with $-v$ in order to have positive variables in $\mathcal{M}'$ only. In what follows all calculations refer to the matrix $\mathcal{M}'$, in which the variables $a, \ldots, p$ have to be positive in order to get a realization of our oriented matroid.

**Step 2.** *Finding a minimal reduced system.*

The selection of the rows that are chosen to be a basis is not at random. The arguments that lead to this particular choice will be discussed now. See e.g. [5] for a more detailed example.

It is a well known fact that the values of sub-determinants in a matrix are not independent from each other. They adhere to algebraic dependencies; so called Graßmann Plücker relations (for instance we have $[1357][1368] - [1356][1378] + [1358][1367] = 0$, as you can easily verify for the matrix $\mathcal{M}'$). Not all terms in such an equation can be positive at the same time. This implies that already a subset of signs of determinants determines all of them. For a given choice of a unit matrix and a given oriented matroid we can look for a "small" reduced system: a subset of determinants whose signs determine all other signs such that a weighted sum of the degress of the corresponding polynomials is as small as possible. In our example the weights should be chosen in a way such that the $1 \times 1$ submatrices (i.e. the variables) are "for free", $2 \times 2$ are "cheep", $3 \times 3$ are "expensive" and $4 \times 4$ are "very expensive."

We skip arguments in which Graßmann Plücker relations are used for finding a minimal reduced system. We are finally left with the following subsystem of inequalities.

$$\begin{array}{rcll}
cp & < & do & [1238] \\
bh & < & df & [1245] \\
bo & < & cn & [1278] \\
gn & < & fo & [1478] \\
di & < & al & [2356] \\
aj & < & bi & [2567] \\
lm & < & ip & [3568] \\
fi & < & ej & [4567]
\end{array}
\qquad
\begin{array}{lcll}
\begin{vmatrix} f & -g & h \\ j & k & l \\ n & -o & p \end{vmatrix} & < & 0 & [1468] \\[2em]
\begin{vmatrix} a & c & d \\ e & g & h \\ m & o & p \end{vmatrix} & < & 0 & [2348]
\end{array}$$

So far we did not loose any generality. Every solution of the above system of 10 equations together with the positivity restriction for the variables provides a realization of the oriented matroid. The choice of the rows that constitute the unit matrix is such that this system of inequality is as cheap as possible with respect to our weight function.

**Step 3.** *Starting the solvability sequence method* (for details see [13]).

First we observe that we can choose $k$ very large: This variable occurs only in the inequality $[1468] < 0$. This determinant can be written as $[1468] = k \cdot (fp - hn) + \cdots = -k \cdot [1458] + \cdots$ such that $k$ does not occur in the remaining terms. The oriented matroid tells us that $[1458] > 0$, hence the choice $\boxed{k \to \infty}$ implies that the term containing $k$ dominates all others and we get $[1468] < 0$ (under the new side condition $[1458] > 0$).

We eliminate $k$ by this argument and add the euqation $fp < hn$ to our system. Now we only have to consider the $2 \times 2$ determinants and the last $3 \times 3$ determinant. Next we analyse the signs of the gradients for each variable in each inequality. We can do this by similar signature arguments that we used for the elimination of $k$.

$$\begin{array}{llllllll}
a & \downarrow\uparrow\Uparrow & b & \downarrow\downarrow\uparrow & c & \downarrow\uparrow\Uparrow & d & \downarrow\uparrow\uparrow\Downarrow \\
e & \uparrow\Downarrow & f & \downarrow\uparrow\uparrow & g & \downarrow\Downarrow & h & \downarrow\Uparrow \\
i & \downarrow\downarrow\uparrow\uparrow & j & \downarrow\uparrow & k & & l & \downarrow\uparrow \\
m & \downarrow\Downarrow & n & \downarrow\uparrow & o & \downarrow\uparrow\uparrow\Downarrow & p & \downarrow\uparrow\Uparrow
\end{array}$$

Here a $\downarrow$ or $\Downarrow$ means that there is an equation that tells us that the corresponding variable should be chosen "small" in order to satisfy the inequality. (Gradients that come from a $2 \times 2$ matrix are marked by $\downarrow$, and gradients that come form a $3 \times 3$ matrix are marked by $\Downarrow$. Siminarly for $\uparrow$ and $\Uparrow$.)

If we consider the variable $g$ we see that making this value positive and close to zero improves all inequalities in which it is involved. Moreover the equation $gn < fo$ will be automatically satisfied if $g > 0$ is chosen very small for any fixed positive choice of $n$, $f$ and $o$. Thus a choice $\boxed{\quad g \to 0 \quad}$ eliminates this variable and the inequality $[1478] < 0$ from our considerations. Next we can apply the same argument to $m$ and get $m \to 0$. Having eliminated the inequality $[1478] < 0$ variable $n$ only remains in $2 \times 2$ inequalities on the right side. Choosing it large enough we can eliminate $n$ and the corresponding equations. Now we skip all details that help us to proceed iteratively in essentially the same manner. We only list the resulting sequence of variables.

$$\boxed{\begin{array}{l} \frac{bo}{c}, \frac{fp}{h} < n \\ \\ \frac{di}{a} < l \end{array}} \quad \boxed{\begin{array}{l} i = 1 \\ \\ \frac{f}{e} < j < \frac{b}{a} \end{array}} \quad \boxed{\begin{array}{l} b = 1 \\ \\ \frac{h}{d} < f < \frac{e}{a} \end{array}}$$

$$\boxed{a = c = d = e = p = 1} \ \boxed{h = 0.5} \ \boxed{o = 1.2}$$

It is important to mention that the whole procedure we sketched can be defined in a rigurous algorithmic way (see [5]). It can essentially be carried out on a completely combinatorial level. Going all the way back, we find that

$$\mathcal{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 \\ -1 & -0.6 & -0.2 & 0.5 \\ 0 & 0 & 1 & 0 \\ -1 & -0.8 & 2 & 1.5 \\ 0 & 0 & 0 & 1 \\ -0.5 & -2 & -1.2 & 1 \end{pmatrix}$$

is a possible realization of the oriented matroid: *A solution was found.*

*A contradiction would lead to a final polynomial* (see [14]).

What can we do, when we get stuck with our system of inequalities?

**Alternative 1.** *Trying to find a bi-quadratic final polynomial* (see [11]).

**Alternative 2.** *Trying a combinatorial reduction method* (see [19]).

In an investigation of finding the set of all non-realizable uniform 10 element oriented matroids in rank 3 in [9], our Alternative 1 and Alternative 2 was applied successfully but a final set of about 143 different cases remained undecided. This was the birth of Alternative 3 suggested by Bokowski which will be described below. The method started with a rubber band model. It was implemented by K.P.Pock, support was given by J. Richter-Gebert in the rank 3 case. There is also a later implementation by J. Scharnbacher in the rank 4 case.

**Alternative 3.** *Trying the dynamic inductive realization method* (see details in this article).

**Alternative 4.** *There is always hope for some intuition, compare our last section.*

For our dynamic inductive realization method, we have now to introduce more notations.

## 2 Configurations and Arrangements

We use the notation of a recent paper [10] in which a direct proof of the equivalence of the hyperline sequence representation and the Folkman Lawrence representation of an oriented matroid in the rank 3 case was established. Since we use in this article both models and their interaction, it is useful for the reader to have a look at that paper. We include the following definitions especially for the reader not familiar with the theory of oriented matroids.

We introduce several configurations and arrangements representing geometrically a class of matrices. It is useful to think simultaneously of all these models and to pick the most convenient one for a particular application or argument.

We consider a non-degenerate *vector configuration* in $R^3$, i.e. a finite ordered set $V_n = \{v_1, \ldots, v_n\} \subset R^3$, $n \geq 3$, $v_i \not\models 0$, $i = 1, \ldots, n$, such that the one dimensional subspaces generated by $v_i$, $i = 1, \ldots, n$, are pairwise different and such that the corresponding $n \times 3$ matrix $M$ with $v_i$ as its $i$-th row vector has rank 3. The vector configuration will be viewed as a representative of the equivalence class of matrices $cl_n(M) := \{M' \,|\, M' = D\,M,\, D = diag(\lambda_1, \lambda_2, \ldots, \lambda_n), \lambda_i > 0, i = 1, \ldots, n\}$.

A vector configuration $V_n$ induces an *arrangement of oriented central planes* $H_n = \{h_1, \ldots, h_n\}$, via the concept of polar duality. The unoriented plane of $h_i$ is given as the zero space $\{\underline{x} = (x_1, x_2, x_3) \in R^3 \,|\, h_i(\underline{x}) = 0\}$ of a linear form $h_i(\underline{x}) = v_{i_1}x_1 + v_{i_2}x_2 + v_{i_3}x_3$, $v_i = (v_{i_1}, v_{i_2}, v_{i_3}) \not\models 0$. The positive and negative sides of an oriented central plane are the two induced half-spaces $h_i^+ : \{\underline{x} \,|\, h_i(\underline{x}) > 0\}$ and $h_i^- : \{\underline{x} \,|\, h_i(\underline{x}) < 0\}$.

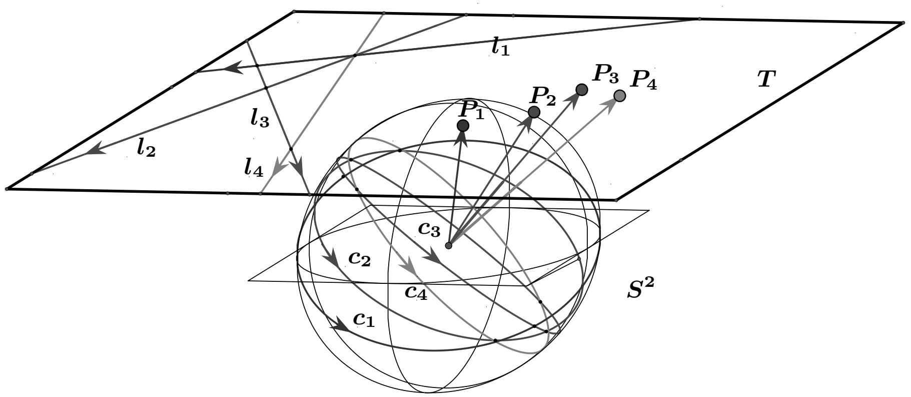


**Fig. 1.** An equivalence class of matrices and geometric representatives.

An arrangement of oriented central planes $H_n$ induces an *arrangement of oriented great circles* $C_n = \{c_1, \ldots, c_n\}$ on the 2-sphere and vice-versa. An oriented central plane cuts the unit sphere $S^2$ in $R^3$ along a great circle which we consider to be parameterized and oriented such that, when looking from outside, the positive half-space lies to its left when the parameter increases.

A vector $v \not= 0, v \in R^3$ induces a directed line $l_v : \{\alpha v | \alpha \in R\}$ through the origin, which intersects the sphere in two antipodal points $s_v$ (in the direction of $v$) and $\overline{s}_v$ (in the opposite direction). A vector configuration $V_n$ induces a *configuration of points on the sphere*, $S_n = \{s_1, s_2, \ldots, s_n\}$, where $s_i = s_{v_i}, i = 1, \ldots, n$. Each point $p$ on the sphere has an associated antipodal point $\overline{p}$.

We carry over the previous polar dual pairs to the affine plane $T$, viewed as a plane tangent to the 2-sphere. We assume that $v_i, i \in \{1, \ldots, n\}$ is neither parallel nor orthogonal to the plane $T$.

The great circle parallel to $T$ defines two open hemispheres. One of them, called the *upper hemisphere*, contains the tangent point of $T$. An oriented great circle $c_i$ induces an oriented half-circle in this upper hemisphere which projects to an oriented straight line $l^T(c_i)$ in the plane $T$ via radial projection, and vice-versa, any oriented straight line in $T$ defines an oriented great circle on $S^2$. An arrangement of oriented great circles induces an *arrangement of oriented lines* $L_n^T = \{l_1, \ldots, l_n\}$, where $l_i := l^T(c_i)$, in the affine plane.

The same transition from the sphere $S^2$ to the plane $T$ leads from a point configuration on the sphere to a signed point configuration in the affine plane. We define $sp^T(s_i)$ to be a pair of a signed index and a point $p_i \in T$ obtained via radial projection from $s_i$, as follows. A point $s_i$ on the upper hemisphere maps to the pair $sp^T(s_i) = (i, p_i)$, $i \in \{1, \ldots, n\}$, and a point $s_i$ on the lower hemisphere maps to a pair $(\overline{i}, p_{\overline{i}})$ and $p_{\overline{i}} := p_i \in T$. We obtain from $S_n = \{s_1, \ldots, s_n\}$ a *signed point configuration* $P_n^T = \{sp_1, \ldots, sp_n\}$, with $sp_i := sp^T(s_i)$, and vice versa.

## 3 Hyperline Sequences of Configurations and Arrangements

We use $E_n = \{1, \ldots, n\}$, endowed with the natural order, to denote the index set of geometric objects like vectors, planes, great circles and points on the sphere, lines and points in the Euclidean plane, or of a finite ordered set of abstract elements. The associated *signed index set* $\overline{E}_n = \{1, \ldots, n, \overline{1}, \ldots, \overline{n}\}$ makes it possible to denote orientations or signs of these elements. The $s \mapsto \overline{s}$ operator is required to be an involution: $\overline{\overline{s}} = s$, $\forall s \in \overline{E}_n$.

All ordered sets $V_n, H_n, C_n, S_n, L_n^T, P_n^T$ above can be viewed as geometric representations of the same equivalence class of matrices $cl_n(M)$. We can reorient the elements. The reorientation classes are the equivalence classes with respect to reorienting subsets such as vector configurations or central plane arrangements, great circle arrangements or pairs of antipodal points on the 2-sphere, line arrangements or point sets in the plane. These reorientation classes are obtained when the numbers $\lambda_i \not= 0$ can be negative as well. The reorientation of a vector $v_i$ is the vector $v_{\overline{i}} = -v_i$ and the reorientation of an oriented central plane is the change of the sign of its normal vector. The reorientation of an oriented great circle or of an oriented line means replacing it by the same object with the reversed orientation. The reorientation of a signed point $(i, p_i), i \in \overline{E}_n$ is the signed point $(\overline{i}, p_{\overline{i}}), p_i = p_{\overline{i}}$. The reorientation of an index $i$ is its replacement with $\overline{i}$. The relabelling of an ordered set is given by a permutation of its elements.
We now extract combinatorial information from all the geometric sets defined above. We will work *only* with signed subsets $q \subset \overline{E}_n$ which do not contain simultaneously both an element $i$ and its negation $\overline{i}$. If $q \subset \overline{E}_n$, we define $\overline{q} = \{\overline{s} | s \in q\}$. The unsigned support $supp(q) \subset E_n$ of $q \subset \overline{E}_n$ is obtained by ignoring all the signs in $q$. A *signed partition* of $E_n$ is a signed set $I = I^+ \cup I^-$ with $I^+, \overline{I^-} \in E_n$, $I^+ \cup \overline{I^-} = E_n$.

**Definition 1.** *A* hyperline sequence $hs_i$ over $\overline{E}_n, i \in \overline{E}_n$ *with half-period length* $k_i$ *is a pair* $hs_i = (i, \pi_i)$*, where* $\pi_i$ *is a double infinite sequence* $\pi_i = (q^i_j)_{j \in Z}$ *with* $q^i_j \subset \overline{E}_n \setminus \{i, \overline{i}\}$, $q^i_j = \overline{q^i_{j+k_i}}$, $\forall j \in Z$, $supp(\bigcup_{j \in Z} q^i_j) = E_n \setminus supp(\{i\})$*, where the unsigned supports of* $q^i_1, \ldots, q^i_{k_i}$ *are mutually disjoint. We consider* $hs_i = (i, \pi_i)$ *and* $hs_{\overline{i}} = (\overline{i}, \pi_{\overline{i}})$ *to be equivalent when* $\pi_i$ *is obtained from* $\pi_{\overline{i}}$ *by reversing the order.*

The name hyperline for a subspace of codimension 2 is justified by the concept in higher dimensions. In the particular case when all the $q^i_j$'s are one-element subsets, the sequence is said to be in *general position*, *simple* or *uniform*, and we replace the sets $q^i_j$ with their elements. In this case, any half period of $\pi_i$ is a signed permutation of $E_n \setminus supp(\{i\})$. In general we have an additional ordered partition into pairwise disjoint subsets of the signed elements. An infinite sequence $\pi_i$ in a hyperline sequence $hs_i = (i, \pi_i)$ can be represented by any half period, i.e. by any $k_i$ consecutive signed sets $q^i_{t+1}, \ldots, q^i_{t+k_i}, q^i_{t+j} \subset \overline{E}_n \setminus \{i, \overline{i}\}, t \in Z$.

*Example 2.* $(\bar{1}, \pi_{\bar{1}}) = (\bar{1}, (\ldots, \{5\}, \{2, \bar{4}\}, \{3\}, \{\bar{5}\}, \{\bar{2}, 4\}, \{\bar{3}\}, \ldots))$ is a hyperline sequence over $\overline{E}_5$, $E_5 = \{1 \ldots 5\}$, with half period length $k_1 = 3$.

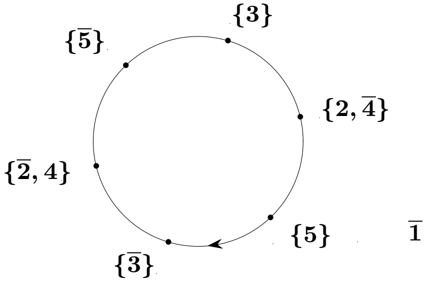


**Fig. 2.** A hyperline sequence over $\overline{E}_5$.

We obtain the *normalized representation* $hs_r = (r, \pi_r)$ of a hyperline sequence $hs_i = (i, \pi_i)$ by first choosing $(r, \pi_r) := (i, \pi_i)$ if $i \in E_n$ or $(r, \pi_r) := (\bar{i}, reverse(\pi_i))$ if $\bar{i} \in E_n$, and afterwards choosing the half period of $\pi_r$ starting with the set $q^r_j \subset \overline{E}_n$ containing the smallest positive element.

*Example 3.* The normalized representation of the hyperline sequence in the previous example is $(1, (\{2, \bar{4}\}, \{5\}, \{\bar{3}\}))$. From now on, we will use the more convenient notation $(1 : \{2, \bar{4}\}, \{5\}, \{\bar{3}\})$.

To a *signed point configuration* $P^T_n = \{(i, p_i) \mid i \in I\}$ (obtained from a vector configuration as described above) we associate a set $HS(P^T_n) = \{hs_1, \ldots, hs_n\}$ of $n$ hyperline sequences $hs_i = (i, \pi_i)$ over $\overline{E}_n$. The sequence $\pi_i$, denoted by a half period $q^i_1, q^i_2, \ldots, q^i_{k_i}$, with $q^i_j \subset \overline{E}_n \setminus \{i, \bar{i}\}$, corresponds to the signed point $(i, p_i) \in P^T_n$. It is obtained by rotating an oriented line in ccw order around $p_i$ if $i \in E_n$ or in cw order around $p_i$ if $\bar{i} \in E_n$ and looking at the successive positions where it coincides with lines defined by pairs of points $(p_i, p_j)$ with $p_j \not= p_i$. When $P^T_n$ is not in general position, several points may become simultaneously collinear with the rotating line, and they are recorded as a set $q^i_k$. If the point $p_j$ of the signed point $(j, p_j)$ is encountered by the rotating line in positive direction from $p_i$, it will be recorded as the index $j$, otherwise as the negated index $\bar{j}$. The whole sequence is recorded in the order induced by the rotating line, and an arbitrary half-period is chosen to represent it.

**Definition 2.** *The rank* 3 oriented matroid induced by hyperline sequences associated to a signed point configuration $P^T_n = \{(i, p_i) | i \in I\}$, *where $I$ is a signed partition of $E_n$, is $HS(P^T_n) = \{hs_i = (i, \pi_i) \mid i \in I\}$ as described above. We identify $HS(P^T_n)$ with $\{(\bar{i}, \pi_i) \mid i \in I\}$.*

Note that if the orientation of the plane $T$ is reversed, all the sequences are reversed. The identification in the previous definition makes the notion of hyperline sequences independent of the chosen orientation of the plane $T$.

**Remark.** When we start with a set of vectors $V_n$ and two admissible tangent planes $T$ and $T'$, by radial projection we obtain two sets of signed planar points $P_n^T$ and $P_n^{T'}$. The reader can verify that our definition ensures that the resulting hyperline sequences $HS(P_n^T)$ and $HS(P_n^{T'})$ will coincide. This allows for a definition of hyperline sequences associated to any of the previously considered geometric ordered sets: vectors, oriented central planes, etc.

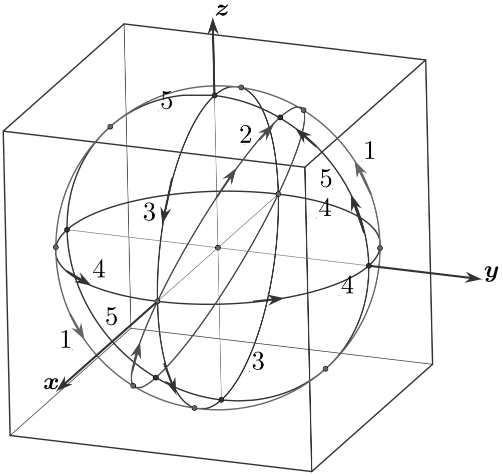


**Fig. 3.** Arrangement $C_5$ of oriented great circles on the 2-sphere.

Consider an arrangement $C = \{c_1, \ldots, c_n\}$ of $n$ oriented great circles on the sphere $S^2$. To each circle $c_i$ associate a hyperline sequence by recording the points of intersection (ordered according to the orientation of the circle $c_i$) with the remaining oriented circles. An index $j$ is recorded as positive (resp. negative) when the circle $c_j$ crosses $c_i$ from left to right (resp., right to left).

An arrangement of oriented lines $L_n^T = \{l_1, \ldots, l_n\}$ induces a set of $n$ hyperline sequences $HS(L_n^T)$: for each line $l_i$, record the points of intersection with the other lines (ordered according to the orientation of the line). Each element $j$ is signed: positive if line $l_j$ crosses $l_i$ from left to right, negative otherwise.

*Example 4.* For the arrangement of oriented great circles in Fig. 3, we have the following induced set of normalized representations of hyperline sequences $HS(C_5)$. We get the same set of normalized representations $HS(M)$ of hyperline sequences for $M$:

$$HS(C_5) = HS(M) = \begin{pmatrix} 1: \{2\}, & \{\bar{3}\}, & \{5\}, & \{\bar{4}\} \\ 2: \{1\}, & \{3,4\}, & \{\bar{5}\} & \\ 3: \{1\}, & \{5\}, & \{\bar{2},\bar{4}\} & \\ 4: \{1\}, & \{\bar{2},3\}, & \{\bar{5}\} & \\ 5: \{1\}, & \{4\}, & \{2\}, & \{\bar{3}\} \end{pmatrix} \qquad M = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -4 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

## 4 Pseudoline Arrangements and Hyperline Sequences

A pseudoline arrangement $\mathcal{A}$ in the projective plane is a set of simple closed curves such that every two curves have precisely one point in common, at which they cross each other. We exclude the case when all curves have a point in common. Let $T$ be the group of homeomorphic transformations of the projective plane. For an arrangement $\mathcal{A}$ we have the equivalence class of arrangements $cl(\mathcal{A}) := \{\mathcal{A}' | \mathcal{A}' = t\mathcal{A}, t \in T\}$. We always consider pseudoline arrangements $\mathcal{A}$ as representatives of their equivalence class $cl(\mathcal{A})$. An arrangement and its mirror image are identified.

To introduce a suitable concept of orientation we single out a "line at infinity" of the projective plane. We embed the usual Euclidean plane in the projective plane with respect to this line at infinity. An oriented pseudoline different from the line at infinity is a simple closed curve which is parameterized and oriented such that the "left side" denotes the side which lies to its left when the parameter increases and the line at infinity has been removed. The projective plane without the line at infinity is the left side, or the right side, of the oriented line at infinity when its orientation is ccw, or cw, respectively.

**Definition 3.** *The* oriented matroid given by an arrangement of $n$ oriented pseudolines *is the equivalence class with respect to homeomorphic transformations of the projective plane of a finite ordered set of $n$ simple oriented closed curves (oriented pseudolines) such that every two curves have precisely one point in common, at which they cross each other. We exclude the case when all curves have a point in common and identify an arrangement with its mirror image.*

The oriented pseudoline arrangement is called *simple*, *uniform* or *in general position*, if no more than two pseudolines cross at a point.

The rule to create a set of hyperline sequences $HS(L_n)$ from an arrangement of oriented lines $L_n = \{l_1, \ldots, l_n\}$ can be carried over in the same way to any arrangement $PL_n = \{pl_1, \ldots, pl_n\}$ of oriented pseudolines. Since there are oriented pseudoline arrangements for which there is no oriented line arrangement within the class of homeomorphic transformations for $n \geq 9$, we get in the pseudoline case a strictly more general concept, $|\{HS(L_n)|L_n$ is an arrangement of lines $\}| < |\{HS(PL_n)|PL_n$ is an arrangement of pseudolines $\}|$ for $n \geq 9$.

We extend the concept of oriented matroids induced by hyperline sequences in another way. Hyperline sequences of configurations and arrangements of the last section store the signs of determinants of $3 \times 3$ submatrices of the matrix $M$ of a corresponding vector configuration $V_n = \{v_1, \ldots, v_n\} \subset R^3$, $n \geq 3$ $v_i \not= 0$, $i = 1, \ldots, n$. This is an invariant for all matrices $M' \in cl_n(M)$. Let $i, j, k$ be three distinct signed indices in $\overline{E}_n$. Let [i,j,k] be the determinant of the submatrix of $M$ with row vectors $v_i, v_j, v_k$. If $j$ and $k$ appear within the same set $q^i_k$ of $\pi_i$, we have sign $[i, j, k] = 0$. If $j$ and $k$ occur in this order in some half-period of $\pi_i$, we have sign $[i, j, k] = +1$, and sign $[i, j, k] = -1$ otherwise. The sign of the determinant $\chi(ijk) :=$ sign $[i, j, k]$ is independent of the chosen half periods and compatible by alternation $\chi(ijk) = \chi(jki) = \chi(kij) = -\chi(ikj) = -\chi(kji) = -\chi(jik)$ and anti-symmetry $\chi(\bar{i}jk) = -\chi(ijk)$.

Given an abstract set of hyperline sequences, let us choose its corresponding normalized form and define $\chi : \overline{E}_n^3 \to \{-1, 0, +1\}$, (partially) by: $\chi(ijk) := 0$, if $j$ and $k$ appear within the same set $q_s$ of $\pi_i$, for $i$ in $E_n$, $j, k$ in $\overline{E}_n$, $j \not= k$, $\chi(ijk) := +1$, if $j$ and $k$ occur in this order in $\pi_i$, and $\chi(ijk) := -1$, if $j$ and $k$ occur in the reversed order in $\pi_i$.

Extending this partial definition of $\chi$ by alternation and anti-symmetry, the value of $\chi(ijk)$ for $0 < i < j < k$ is obtained either directly, by the above rule applied to each of the three hyperline sequences, or via alternation and anti-symmetry. When these three values for $\chi(ijk)$ are compatible in all cases, we say that *the set of hyperline sequences admit an abstract sign of determinant function.*

**Definition 4.** *A* rank 3 oriented matroid with $n$ elements given by hyperline sequences *is a set of hyperline sequences* $\{(i, \pi_i) \mid i \in I\}$ *over* $\overline{E}_n$ *which admit an abstract sign of the determinant function. The oriented matroid is uniform when all hyperline sequences are uniform.*

**Theorem 1.** *The hyperline sequences $HS(PL_n)$ of an oriented pseudoline arrangement $PL_n$ admit an abstract sign of the determinant function.*

For each rank 3 oriented matroid given by hyperline sequences $HS$ we can find an oriented pseudoline arrangement $PL_n$ which induces it, $HS = HS(PL_n)$. For a direct proof of this theorem see [10].

While the pseudoline arrangement of an oriented matroid shows that we are dealing with a topological invariant, the hyperline sequences show that this information can be stored effectively and that the complete generation of these objects for given number of elements and given rank becomes possible. The enumeration of corresponding possible types of matrices is available only by extending this problem to the concept of oriented matroids and deciding later which of them are realizable.

We again see that it is an essential problem to find out for a given oriented matroid whether it belongs to the non-realizable ones or to the realizable ones.

## 5 Extension Determined by Mutations in Rank 3

We consider the set $E_n = \{1, 2, \ldots, n\}$. We start with a uniform oriented matroid $\chi = \chi_{E_n}$ in rank 3 with $n$ elements and its set of mutations $\mathbf{Mut}_\chi = \{(i, j, k) \mid i < j < k\}$. The triple $(i, j, k)$ is a mutation when changing its sign leads again to an oriented matroid. By deleting the $n$-th element we obtain the oriented matroid $\chi_{E_{n-1}}$.

**Theorem 2.** *The extension of an oriented matroid $\chi = \chi_{E_{n-1}}$ in rank* 3 *with* $n - 1$ *elements by an additional element* $n$ *is uniquely determined by the signs of brackets* $(x, y, n) \in \mathbf{Mut}_\chi$ *together with an additional sign of a bracket* $(x, y, n) \not\in \mathbf{Mut}$.

**Remarks.** (i) This theorem was first proved by Bokowski and Scharnbacher [23]. Their proof is unpublished. It was originally conjectured by Richter-Gebert (private communication). We present a new proof here.

(ii) Figure 2 shows that the additional assumption is essential in many instances.

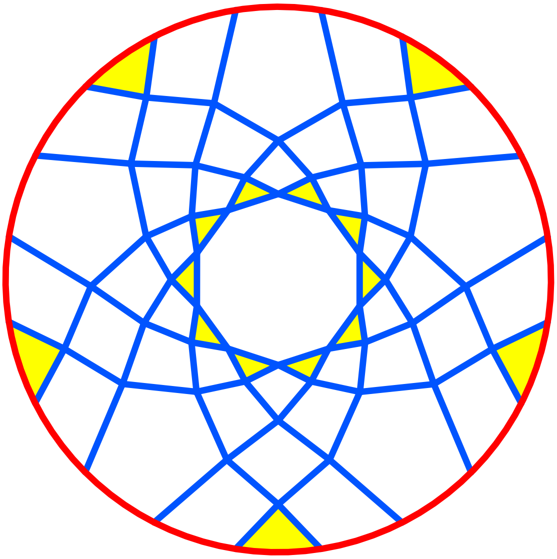


**Fig. 4.** Changing all mutations at the line at infinity simultaneously leads again to an oriented matroid.

*Proof.* W.l.o.g. we assume that $(1, n-1, n)$ is a mutation and that the sign of $(1, n-2, n)$ which is not a mutation is given. We look at the normalized representation of the hyperline sequence. We see immediately that the hyperline sequence of element 1 is uniquely defined. As a consequence, the sign of element $n$ in the $i$-th hyperline sequence $(i > 1)$ is uniquely defined.

Again as a consequence, the position of the element $n$ is uniquely defined in all hyperlines which are contained in a mutation of $n$. We collect all these hyperline indices in a set $E_1$.

Now we consider the oriented matroid $\chi_{E_1 \cup \{n\}}$ obtained by deleting the elements not in $E_1 \setminus \{n\}$ from $\chi_E$. The Folkman Lawrence representation of $\chi_E$ defines an additional cell decomposition within the Folkman Lawrence representation of $\chi_{E_1 \cup \{n\}}$. On all rank 2 contractions of elements in $E_1$, we have a unique ordering of the cocircuits of $\chi_E$. Now we look at a cell $C$ within the Folkman Lawrence representation of $\chi_{E_1 \cup \{n\}}$ which contains a pseudoline segment of pseudoline $n$. Because of the known ordering of cocircuits of $\chi_E$ on the boundary of $C$, we can split the set of elements not in $E_1 \cup \{n\}$ into those which cut the pseudoline segment of pseudoline $n$, say $\{c_1, \ldots, c_k\}$ and into those which do not cut, say $\{p_1, \ldots, p_l\}$. A latter *parallel* element $p_m$ decomposes $C$ into two discs. We denote with $C_{p_m}$ the corresponding disc which contains the pseudoline segment of pseudoline $n$. Now consider $C' := int(C \cap C_{p_1} \cap C_{p_2} \cap \ldots \cap C_{p_m})$. When $C'$ contains no cocircuits of $\chi_E$, we are done. In this case the insertion segments for the $n$-th element is uniquely defined.

But if a cocircuit exists in $C'$, there is one which is adjacent to the $n$-th pseudoline segment. The two elements defining that cocircuit cut the $n$-th pseudoline segment within $C'$ and define together with $n$ a triangle in $C'$. If this triangle does not contain a cocircuit in its interior, we find a mutation in it which was excluded. But in the other case, we find a smaller triangle completely included in the former one, again having one cocircuit adjacent to the $n$-th element. After a finite number of steps we arrive at a contradiction.

## 6 Dynamic Inductive Realization

When we apply the dynamic inductive realization method, we start with a realized oriented matroid with $k-1$ elements. We consider having $k-1$ great circles on a 2-sphere. We add the $k$-th pseudo great circle on the upper half of the 2-sphere as a rubber band (we identify antipodal points on the equator by a diameter which can rotate around the midpoint). We mark the mutations with pins which guarantee that the rubber band lies always on the proper side with respect to a mutation. When the rubber band can be straightened to a half circle not violating the mutation conditions, we have found according to our last theorem a realization of the oriented matroid with $k$ elements.

When this great half circle cannot be rectified, we can change a former element within the realization space of the former oriented matroid such that inserting the new element becomes *easier*. Doing this systematically for all foregoing elements was very effective in many instances.

This idea of the author was implemented in the rank 3 case by K. P. Pock in his Diplom thesis [18], support was given also by J. Richter-Gebert. The corresponding idea worked also in the rank 4 case. But the corresponding theorem about the mutations does not hold. Results can be found in the articles [1,2]. There is an implementation in the rank 4 case by J. Scharnbacher [23].

## 7 Cinderella Playing with Polytopes

The first impression when checking what can be done with the software Cinderella by J. Richter-Gebert and U. Kortenkamp [22], or with any other similar dynamic program, these are applications to planar drawings and planar theorems. But when the applications for architects come to your mind (see e.g. `http://juergen.bokowski.de` and `http://homepages.tu-darmstadt.de/~raiwi`) projections from higher dimensions to the plane appear. Dynamic geometry programs can be used to visualize, study and modify these projections and by this obtaining new insights on the situation in high dimensions.

When we start drawing the projections of the unit vectors of a basis in $R^n$, we can insert e.g. projections of all vertices of a polytope. The latter change of the positions of all projections of the unit vectors of the basis generate all possible linear projections of the polytope onto the plane. The edge graph can easily be drawn. The face lattice of the polytope can be visualized and studied this way, facets can appear as line segments. A first application can be seen in [7].

For another interesting 3-sphere with 16 vertices , and 50 simplicial facets:

$1234$ $1235$ $1246$ $1256$ $1347$ $1357$ $1467$ $1567$ $2348$ $2389$
$2395$ $2480$ $2406$ $2809$ $209a$ $20a6$ $29a5$ $2a56$ $34b8$ $34b7$
$3b8c$ $3bc7$ $38c9$ $3c95$ $3c57$ $4bd8$ $4bd7$ $4d80$ $4d06$ $4d67$
$bd8c$ $bdc7$ $d0e6$ $dce7$ $de67$ $0e9a$ $0ea6$ $ce95$ $ce57$ $e9a5$
$ea56$ $e567$ $fd80$ $fd8c$ $fd0e$ $fdce$ $f809$ $f8c9$ $f0e9$ $fce9$

B. Sturmfels and A. Zelevinsky asked whether it forms the face lattice of a 4-polytope. We use this 3-sphere here as an example to show another method which was not applied in this context before. A realization was found by Bokowski by drawing the edge graph with its symmetry. Of course some intuition was essential to find the realization. But in order to confirm the lattice structure of the boundary of the polytope you can look at the Cinderella program output at `http://juergen.bokowski.de`

With the generators of the symmetry group

$$(2)(5)(6)(8)(c)(d)(1,a)(b,f)(3,9)(4,0)(7,c)$$
$$(3)(4)(7)(9)(0)(c)(1,b)(a,f)(5,9c(2,8)(6,d)$$
$$(1)(a)(b)(f)(2,6,5)(9,0,e)(8,d,c)(4,7,3)$$

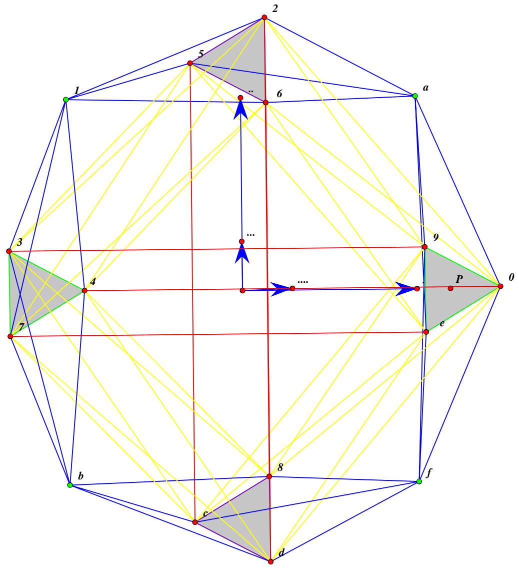


**Fig. 5.** A projection of the corresponding 4-polytope.

we can determine the orbits of the facets. By checking one facet in each orbit and by using the connectedness of the facets, we can confirm that the face lattice is that of a convex polytope. This is done for each facet by changing the projection contineously until a position is reached in which the vertices of that facet of the sphere lies on a supporting line of the projection of the polytope.

## References

1. A. Altshuler, J. Bokowski, and P. Schuchert. *Spatial polyhedra without diagonals.* Israel J. Math. 86, 373–396, 1994.
2. A. Altshuler, J. Bokowski, and P. Schuchert. *Sphere systems and neighborly spatial polyhedra with 10 vertices.* Suppl. Rend. Circ. Mat. Palermo, II. Ser. 35, 15–28, 1994.
3. A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler. *Oriented Matroids.* Cambridge University Press, Cambridge, 1993.
4. J. Bokowski. *Aspects of computational synthetic geometry II: Combinatorial complexes and their geometric realization — An algorithmic approach.* Proceedings of the INRIA Workshop on Computer-Aided Geometric Reasoning (H. Crapo, ed.), Antibes, France, 1987.
5. J. Bokowski. *On the geometric flat embedding of abstract complexes with symmetries.* Symmetry of Discrete Mathematical Structures and Their Symmetry Groups: A Collection of Essays (K. H. Hofmann and R. Wille, eds.), Research and Exposition in Mathematics 15, 1–48, Heldermann, Berlin, 1991.
6. J. Bokowski. *Handbook of Convex Geometry*, Chapter on Oriented Matroids (P. Gruber and J. M. Wills, eds.). Elsevier, North-Holland, Netherlands, 1992.
7. J. Bokowski, P. Cara, and S. Mock. *On a self dual 3-sphere of Peter McMullen.* Periodica Mathematica Hungarica 39, 17–32, 1999.
8. J. Bokowski and A. Guedes de Oliveira. *On the generation of oriented matroids.* Discrete Comput. Geom. 24, 197–208, 2000.
9. J. Bokowski, G. Lafaille, and J. Richter-Gebert. *Classification of non-stretchable pseudoline arrangements and related properties.* Manuscript, 1991.
10. J. Bokowski, S. Mock, and I. Streinu. *The Folkman-Lawrence topological representation theorem: A direct proof in the rank 3 case.* Proceedings of the CIRM Conference "Géométries combinatoires: Matroïdes orientés, matroïdes et applications", Luminy, France, 1999.
11. J. Bokowski and J. Richter. *On the finding of final polynomials.* Eur. J. Comb. 11, 21–34, 1990.
12. J. Bokowski and J. Richter-Gebert. *Reduction theorems for oriented matroids.* Manuscript, 1990.
13. J. Bokowski and B. Sturmfels. *On the coordinatization of oriented matroids.* Discrete Comput. Geom. 1, 293–306, 1986.
14. J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry.* Lecture Notes in Mathematics 1399, Springer, Berlin, 1989.
15. B. Grünbaum. *Arrangements and Spreads.* Regional Conf. Ser. Math. 10, Amer. Math. Soc., Providence, RI, 1972.
16. D. Ljubic, J.-P. Roudneff, and B. Sturmfels. *Arrangements of lines and pseudolines without adjacent triangles.* J. Comb. Theory, Ser. A 50, 24–32, 1989.

17. N. E. Mnëv. *The universitality theorems on the classification problem of configuration varieties and convex polytope varieties.* Topology and Geometry — Rohlin Seminar (O. Ya. Viro, ed.), Lecture Notes in Mathematics 1346, 527–544, Springer, Berlin, 1988.
18. K. P. Pock. *Entscheidungsmethoden zur Realisierung orientierter Matroide.* Diplom thesis, TH Darmstadt, 1991.
19. J. Richter. *Kombinatorische Realisierbarkeitskriterien für orientierte Matroide.* Mitteilungen aus dem Math. Sem. Gießen 194, 1–113, 1989. Diplom thesis, TH Darmstadt, 1988.
20. J. Richter-Gebert. *On the Realizability Problem of Combinatorial Geometries — Decision Methods.* Ph.D. thesis, Darmstadt, 1992.
21. J. Richter-Gebert. *Realization Spaces of Polytopes.* Lecture Notes in Mathematics 1643, Springer, Berlin, 1996.
22. J. Richter-Gebert and U. Kortenkamp. *The Interactive Geometry Software Cinderella.* Springer, Berlin, 1999.
23. J. Scharnbacher. *Zur Realisation simplizialer orientierter Matroide.* Diplom thesis, TH Darmstadt, 1993.
24. B. Sturmfels. *Aspects of computational synthetic geometry I: Algortihmic coordinatization of matroids.* Proceedings of the INRIA Workshop on Computer-Aided Geometric Reasoning (H. Crapo, ed.), Antibes, France, 1987.

# Decision Complexity in Dynamic Geometry

Ulrich Kortenkamp[1] and Jürgen Richter-Gebert[2]

[1] Institut für Informatik, Freie Universität Berlin, Takustr. 9, D-14195 Berlin, Germany kortenkamp@inf.fu-berlin.de
[2] Zentrum Mathematik, SB4, Technische Universität München, D-80290 München, Germany richter@ma.tum.de

**Abstract.** Geometric straight-line programs [5,9] can be used to model geometric constructions and their implicit ambiguities. In this paper we discuss the complexity of deciding whether two instances of the same geometric straight-line program are connected by a continuous path, the *Complex Reachability Problem*.

## 1 Introduction

Straight-line programs and randomized techniques for proving their equivalence did find their application in geometric theorem proving. Using estimates for the degrees of the variables of a multivariate polynomial given by a straight-line program and evaluations for some random samples, we can prove geometric theorems with much less computational effort than usual [2,14], for example compared to symbolic methods using Gröbner bases.

An apparent drawback of polynomials is that we have to refer to systems of polynomial equations as soon as we want to describe theorems involving circles or conics. Although there are very powerful methods to do theorem proving in these contexts (e.g. Wu's method, see [13,12]), it is desirable to have a concept like straight-line programs that is able to describe constructive theorems, and is able to model the dynamic aspects of theorems as they occur in dynamic geometry systems. The implementation of one dynamic geometry system [8,7] caused the definition of *geometric straight-line programs*, which are one way to approach the above issues.

One question that must be settled before we could use techniques similar to the methods of Schwartz and Zippel [10,6] to prove geometric theorems is the question of (complex) reachability: Can we move one instance of a geometric theorem continuously into another instance? This paper describes first results on the algorithmic complexity of this question.

## 2 Geometric Straight-Line Programs

Geometric straight-line programs extend the concept of straight-line programs (see the book of Bürgisser et al. [1] for a detailed discussion of straight-line programs). Informally, a straight-line program (SLP) is a sequence of operations

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 193–198, 2001.

(usually addition, multiplication, subtraction, and sometimes division) that operate on a certain input (usually values of some algebra $A$) or intermediate results from previous operations.

Straight-line programs are important due to the fact that they provide a very compact description of multivariate polynomials (or rational functions, if we allow divisions). The degree of the polynomials can be much higher than the length of the straight-line program (up to exponential).

In [5] it is shown that geometric constructions using points and lines as objects, and meets and joins as operations, are equivalent to straight-line programs over $\mathbb{R}$ or $\mathbb{C}$. In a way this is a consequence of von-Staudt's approach, who has shown that there is a coordinate-free description of projective geometry [3].

As soon as we want to describe constructions that involve ambiguous operations (like Intersection of Circle and Line, Intersection of Circle and Circle, or Angular Bisector of two lines) the concept of straight-line programs fails. Better said, it is not possible to describe constructions with varying input parameters that behave *continuously* using straight-line programs.

*Geometric straight-line programs (GSPs)* are a way to keep a concise algebraic description even for constructions involving ambiguous operations. The operations of a straight-line program are replaced by relations from a suitable *relational instruction set (RIS).* The objects can be choosen arbitrarily, as long as they match the relations. In this paper we will deal with the complex numbers $\mathbb{C}$ as objects and the RIS $R := \{+, -, *, \pm\sqrt{\cdot}\}$ only, and we will emphasize this sometimes by calling them *complex GSPs.*

Again, we refer to [5] for a more formal and detailed description. Here we rely on the readers‘ intuition and introduce geometric straight-line programs using an example.

*Example 1 (A GSP on $(\mathbb{C}, R)$).* Here is a GSP encoding the expression $\pm\sqrt{{z_1}^2 + {z_2}^2}$, with two input variables. The negative indices denote input variables, the other ones index the intermediate results. All statements refer to the indices of previous results or input variables.

| Index | Statement | Remark |
|---|---|---|
| $-2$ | $z_2$ | Input |
| $-1$ | $z_1$ | Input |
| 0 | $*(-1,-1)$ | ${z_1}^2$ |
| 1 | $*(-2,-2)$ | ${z_2}^2$ |
| 2 | $+(0,1)$ | ${z_1}^2 + {z_2}^2$ |
| 3 | $\pm\sqrt{\cdot}(2)$ | $\pm\sqrt{{z_1}^2 + {z_2}^2}$ |

A fundamental difference between ordinary straight-line programs and GSPs is that we cannot just "run through" the statements of a GSP in order to calculate the expression for a given input. This is due to the fact that the relations can have different valid outputs for the same input. This gives rise to the notion of an *instance* of a GSP, an assignment of the input parameters and all intermediate results that is compatible with the relations.

*Example 2 (Instance of a GSP).* An instance for the GSP above is given by

| Index | Value | Remark |
|---|---|---|
| $-2$ | 3 | Input $z_2$ |
| $-1$ | 4 | Input $z_1$ |
| 0 | 16 | ${z_1}^2$ |
| 1 | 9 | ${z_2}^2$ |
| 2 | 25 | ${z_1}^2 + {z_2}^2$ |
| 3 | $-5$ | $\pm\sqrt{{z_1}^2 + {z_2}^2}$ |

Observe that all but the last value are determined by the input, and there is only one other instance with the same input (where the last value is 5).

## Moving GSPs

For polynomials, or straight-line programs, it is easy to speak about dynamic changes of the input parameters. Since the value of all intermediate results of an SLP is determined by the input, we can vary the input parameters and recalculate the polynomial. Of course, the intermediate results *are* polynomials in the input variables, and as such they are analytic functions, in particular *continuous*.

If we want to do the same with GSPs we must specify how to resolve ambiguities. A natural requirement would be that the intermediate results should be continuous functions in the input parameters. A direct consequence is that the intermediate results must be *analytic* [5] in the following way: Let $U := (u_1, \ldots, u_n), V := (v_1, \ldots, v_n) \in \mathbb{C}^n$ be two inputs for a complex GSP, and let $\gamma\colon [0,1] \mapsto \mathbb{C}^n$ be a path from $\gamma(0) = U$ to $\gamma(1) = V$. If we can find instances of the GSP for every $\lambda \in [0,1]$ such that every intermediate result is an analytic function in $\lambda$ for $\lambda \in (0,1)$ and a continuous function for $\lambda \in [0,1]$, then these instances form an analytic path.

Here are two examples showing the subtilities of analytic paths:

*Example 3 (Square Root).* Take the complex GSP with one input that has the $\pm\sqrt{\cdot}$-Relation as the one and only statement, and consider the path

$$\gamma\colon [0,1] \mapsto \mathbb{C}$$
$$\gamma(\lambda) = e^{2i\pi\lambda}$$

For each of the two possible choices at $\lambda = 0$ there is a unique assignment of instances for $\lambda \in (0,1]$ to form an analytic path, which is the proper branch of the complex square root function. The value of the square root at $\lambda = 1$ will be the negative of the value at $\lambda = 0$.

We can find this path by doing analytic continuations along $\gamma$, and here in this example it is clear that we can do this for all paths avoiding 0 for $\lambda \in (0,1)$, and only these.

*Example 4 (Roots of squares).* Take the complex GSP with one input $z$ and with two statements, first multiplying the input with itself and then the $\pm\sqrt{\cdot}$-Relation. The first intermediate result, the square of the input, is determined by the input, and since it is a polynomial, it is analytic in the input $z$, so it is analytic for any analytic function $\gamma$.

The second relation can be simplified to either $+z$ or $-z$, but not to the absolute value function $|x|$, since this would destroy analyticity. We do not have to consider a special path to observe this, it holds for any path.

In the second example there is not always a need to avoid the 0 for the square root function, for example for the path $\gamma(\lambda) = 2\lambda - 1$ there are instances that make it analytic. However, in most considerations it will be a good idea to avoid any zeros of square roots, since these are the critical points where singularities can occur.

## 3 Complex Reachability and Testing of Polynomials

A problem in straight-line program analysis is to decide whether a given straight-line program is equivalent to another one, i.e. whether it describes the same polynomial (or rational function). The algorithmic complexity of this decision problem is unknown, but there exist polynomial-time randomized algorithms [10]. The main obstacle is that we can neither handle the full, symbolic expression for the polynomial, since the coefficients and the degree of the polynomial can be large, nor the evaluation of the straight-line program for sufficiently large numbers, since the coding length for the intermediate results becomes too large.

If we could find an algorithm to test equivalence of straight-line programs efficiently, then their range of application could be extended to efficient encodings of large numbers. It would also be possible to derive efficient deterministic algorithms to prove geometric theorems.

We will now formulate a version of this decision problem which is equivalent to the equivalence testing problem.

> [SLP zero testing] Given a division-free straight-line program $\Gamma$ over $\mathbb{Q}$ with one input variable. Is the polynomial $p$ encoded by $\Gamma$ the zero polynomial?

We will show that this problem is at most as hard as deciding whether we can move analytically from one instance of a GSP to another instance of the same GSP that is different at exactly one intermediate result by giving a polynomial transformation from [SLP zero testing] to the following decision problem:

> [Complex Reachability Problem] Given two instances of a complex GSP with one input variable that differ in exactly one intermediate result. Is it possible to move analytically from the first instance to the second?

We will prove the following theorem, with this corollary as an easy consequence:

**Corollary 1.** *The* [Complex Reachability Problem] *is algorithmically at least as hard as* [SLP zero testing]*.*

**Theorem 1.** *There is a polynomial transformation of* [SLP zero testing] *to the* [Complex Reachability Problem]*, i.e. we can answer an instance of* [SLP zero testing] *by transforming it to an instance of the* [Complex Reachability Problem] *and answering this.*

*Proof.* First, we have to clarify how we specify an instance of a complex GSP in a polynomial size of the encoding length of the GSP (where the encoding length of the GSP is the number of bits needed to write down all statements of the GSP). We will deal with GSP inputs that have polynomial encoding length, and then we just have to specify for each $\pm\sqrt{\cdot}$-statement which solution we choose. This can be done using one bit for each decision, saying to choose the solution with the smaller or equal angle in the polar coordinate representation of the two possibilities. We denote an instance by writing down all values of the input variables and a $+$ or $-$ for each decision bit.

Observe that there is no need to evaluate the GSP; indeed, we *must not* evaluate the GSP since this could take exponential time.

Having done this, we assume to have an SLP $\Gamma$ of length $n$ with one input variable $z$ and want to know whether it describes the zero polynomial. Let us refer to the last result, the polynomial, by $p(z)$.

Let $M$ be the largest constant that can be created using a straight-line program $\Gamma_M$ of length $n$ and with encoding length less or equal to the encoding length of $\Gamma$. Using one additional statement we can write a straight-line program $\Gamma_{2M}$ that evaluates to $2M$. Thus we can transform $\Gamma$ in polynomial time and space to $\Gamma'$ which evaluates $p(z) + 2M$. Due to the construction of $\Gamma'$ the value at $z = 0$ of $\Gamma'$ cannot be 0.

Now we add one additional statement to $\Gamma'$ in order to evaluate $\pm\sqrt{p(z) + 2M}$. Let the new GSP be $\Gamma_1$. In a similar way, we also create a GSP $\Gamma_2$ that evaluates $\pm\sqrt{zp(z) + 2M}$. This requires just one additional statement compared to $\Gamma_1$

These two GSPs can be used to decide the zeroness of $p(z)$ using the complex reachability decision. Let $(z = 0, +)$ be the start instance, and $(z = 0, -)$ the end instance for both $\Gamma_1$ and $\Gamma_2$.

Now we claim that the reachability decision will be "not reachable" for both $\Gamma_1$ and $\Gamma_2$ if and only if $p(z)$ is the zero polynomial. For the $\Leftarrow$ direction we observe that $p(z) + 2M = 2M$ and $zp(z) + 2M = 2M$, i.e. the arguments of the $\pm\sqrt{\cdot}$-statement are constant and non-zero. So they can never change continuously from one sign decision to the other.

For the $\Rightarrow$ direction we note that $p(z) + 2M$ and $zp(z) + 2M$ are two polynomials of even and odd resp. odd and even degree if $p(z) \not\equiv 0$. This means that at least one of them has a root of odd multiplicity at, say, $z_0$. But this means that we can change the sign of the square root by following a path from $z = 0$ to $z = z_0 + \varepsilon$, cycling once around $z_0$ and going back from $z = z_0 + \varepsilon$ to $z = 0$. So for at least one of $\Gamma_1$ and $\Gamma_2$ the reachability decision will be "reachable." □

## 4 Remarks

The paper "Randomized Zero Testing of Radical Expressions and Elementary Geometry Theorem Proving" by Daniela Tulone, Chee Yap and Chen Li, that was also presented at ADG 2000, also introduces square roots for straight-line programs. The main difference between our two approaches is that we rely on the implicit sign decision for our notion of geometric theorems, which is different from the usual notion of theorems given by polynomial equations for hypothesis, nondegeneracies and conclusions. Also, since we only work with complex numbers, we cannot state theorems that are given by semi-algebraic varieties.

Nevertheless, it seems that both the results of both papers can be combined in one or the other way, which we will try to do in our further investigations.

Kurt Mehlhorn pointed out that our transformation shows not only that the complex reachability problem is as hard as to find out whether a polynomial is the zero polynomial, but also as hard as to find out whether a polynomial has at least one root of odd degree.

## References

1. Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *A Series of Comprehensive Studies in Mathematics*, chapter 4, pages 103–124. Springer-Verlag, Berlin Heidelberg New York, 1997.
2. Mike Deng. The parallel numerical method of proving the constructive geometric theorem. *Chinese Science Bulletin*, 34:1066–1070, 1989.
3. Hans Freudenthal. The impact of von Staudt's foundations of geometry. In R. S. Cohen, J. J. Stachel, and M. W. Wartofsky, editors, *For Dirk Struik*, pages 189–200. D. Reidel, Dordrecht-Holland, 1974. An article emphasizing the foundation-laying contribution (in terms of purely algebraic description) of von Staudt to projective geometry.
4. Erich Kaltofen. Greatest common divisors of polynomials given by straight-line programs. *Journal of the Association for Computing Machinery*, 35(1):231–264, January 1988.
5. Ulrich Kortenkamp. *Foundations of Dynamic Geometry.* Dissertation, ETH Zürich, October 1999.
6. Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*, chapter 7. Cambridge University Press, Cambridge, 1995.
7. Jürgen Richter-Gebert and Ulrich Kortenkamp. *Die interaktive Geometriesoftware Cinderella.* Book & CD-ROM, HEUREKA-Klett Softwareverlag, Stuttgart, 1999.
8. Jürgen Richter-Gebert and Ulrich Kortenkamp. *The Interactive Geometry Software Cinderella.* Book & CD-ROM, Springer-Verlag, Berlin Heidelberg New York, 1999.
9. Jürgen Richter-Gebert and Ulrich Kortenkamp. Complexity issues in Dynamic Geometry. Submitted to *Proceedings of the Smale Fest 2000*, Hongkong, 2000.
10. Jacob T. Schwartz. Probabilistic algorithms for verification of polynomial identities. In *Symbolic and Algebraic Computation*, EUROSAM '79, Int. Symp., Marseille 1979, Lect. Notes Comput. Sci. 72, pages 200–215. Springer-Verlag, Berlin Heidelberg New York, 1979.
11. Volker Strassen. Berechnung und Programm I. *Acta Informatica*, 1:320–335, 1972.
12. Wen-tsün Wu. On the decision problem and the mechanization of theorem-proving in elementary geometry. In *Contemp. Math.*, volume 29, pages 213–234. AMS, Providence, 1984.
13. Wen-tsün Wu. *Mechanical Theorem Proving in Geometries. Basic Principles.* Transl. from the Chinese by Xiaofan Jin and Dongming Wang. Texts and Monographs in Symbolic Computation. Springer-Verlag, Wien, 1994.
14. Jingzhong Zhang, Lu Yang, and Mike Deng. The parallel numerical method of mechanical theorem proving. *Theoretical Computer Science*, 74:253–271, 1990.

# Automated Theorem Proving in Incidence Geometry – A Bracket Algebra Based Elimination Method

Hongbo Li and Yihong Wu

Academy of Mathematics and System Sciences
Chinese Academy of Sciences
Beijing 100080, P. R. China
{hli, yhwu}@mmrc.iss.ac.cn

**Abstract.** In this paper we propose a bracket algebra based elimination method for automated generation of readable proofs for theorems in incidence geometry. This method features three techniques, the first being heuristic automated reordering of geometric constructions for the purpose of producing shorter proofs, the second being some heuristic elimination rules which improve the performance of the area method of Zhang and others without introducing signed length ratios, the third being a simplification technique called contraction, which reduces the size of bracket polynomials. More than twenty theorems in incidence geometry have been proved, for which short proofs can be produced very fast, together with the corresponding nondegeneracy conditions. An interesting phenomenon is that a proof composed of polynomials of at most two terms can always be found for any of these theorems, similar to that by the biquadratic final polynomial method of Richter-Gebert.

## 1 Introduction

According to the first fundamental theorem of invariant theory [17], brackets are the fundamental invariants under projective transformations. From an invariant theoretic point of view, the ring of brackets forms a suitable algebraic setting to deal with projective configurations [19,12]. Bracket algebra is the most general structure in which projective properties can be expressed in a coordinate-free way.

Let $\mathcal{V}^{n+1}$ be an $(n+1)$-dimensional vector space. For a sequence of $n+1$ vectors $\mathbf{A}_1, \ldots, \mathbf{A}_{n+1} \in \mathcal{V}^{n+1}$, the corresponding *bracket* is defined by

$$[\mathbf{A}_1 \cdots \mathbf{A}_{n+1}] = \det(\mathbf{A}_1 \cdots \mathbf{A}_{n+1}). \tag{1}$$

Let $\mathbf{A}_1, \ldots, \mathbf{A}_m$ be indeterminates (vectors) in $\mathcal{V}^{n+1}$, $m > n$. The *bracket algebra* generated by them is the polynomial algebra $\mathcal{R}([\mathbf{A}_{i_1} \cdots \mathbf{A}_{i_{n+1}}] | 1 \leq i_j \leq m)$ generated by all possible brackets of the indeterminates modulo the ideal generated

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 199–227, 2001.

by the following *Grassmann-Plücker polynomials*:

$$\mathcal{GP} = \left\{ \sum_{k=1}^{n+2} (-1)^k [\mathbf{A}_{i_1} \cdots \mathbf{A}_{i_n} \mathbf{A}_{j_k}][\mathbf{A}_{j_1} \cdots \mathbf{A}_{j_{k-1}} \mathbf{A}_{j_{k+1}} \cdots \mathbf{A}_{j_{n+2}}] \;\middle|\; 1 \leq i_1 < \cdots < i_n \leq m,\ 1 \leq j_1 < \cdots < j_{n+2} \leq m \right\}. \tag{2}$$

On the level of bracket algebra, a geometric theorem prover can be implemented using the straightening algorithm [22,5]. The main idea behind this approach is to rewrite the projective incidence statement as a term in Grassmann algebra which vanishes if and only if the statement is true. After this, the Grassmann algebra term is expanded into a bracket algebra term. If this term vanishes modulo the ideal generated by the Grassmann-Plücker polynomials, then the theorem is proved. It is shown by Sturmfels and White [18] that the straightening algorithm can be considered as a special kind of Gröbner bases algorithm for bracket polynomials. The algorithm works in full generality, but requires over-exponential CPU time.

The prover proposed by Richter-Gebert [15] is based on the biquadratic final polynomial method (see also [1,16,4]). A proof produced by this prover is extremely short and geometrically meaningful. In particular, every polynomial which occurs in the proof is composed of two terms. Although the algorithm does not work in general, it could manage almost all projective incidence theorems.

Another prover is proposed by Chou and others [3] and is based on the area method. This is an elimination method whose rules are derived from properties of signed areas, or brackets in 2-dimensional projective space. This method is complete when area coordinates are used. When the coordinates are avoided, proofs produced by the prover are often short and readable.

Our work is inspired both by the area method and by the final polynomial method. First, we propose a set of heuristic elimination rules to improve the performance of the area method by producing shorter proofs (for elimination methods, see also [21,20,8,7]). Second, we propose a new technique for bracket polynomial simplification, of which a special case is used as the foundation for setting up biquadratic equations in the final polynomial method. Third, as an optional technique, we use the heuristic method in [9] for automated reordering of geometric constructions in case a polynomial in the proof has more than two terms. We build up a prover based on the three techniques.

The performance of the prover is very satisfactory: more than twenty incidence theorems have been tested, which covers all the 2-dimensional incidence theorems in [3,15]. For every theorem, a proof composed of polynomials of at most two terms can be produced very fast. Furthermore, every proof finishes before any free point in the plane is eliminated, and in some cases, even before some semifree points on lines are eliminated.

The prover is complete for 2-dimensional incidence theorems of the following constructive types.

**Constructive type 1**. Take a free point in the plane.

**Constructive type 2**. Take a semifree point on a line.

**Constructive type 3**. Take the intersection of two lines.

## 2 Algorithm

The following is an algorithm which can produce a proof in the form of brackets for a theorem whose conclusion is either an equality or an inequality of negative type.

**Input:**

- A set of constructions of points.
- An order for eliminations of points.
- A conclusion $conc = 0$ or $conc$ /= 0, where$conc$ is a polynomial of brackets.

**Preprocess.** Change every collinearity constraint in the constructions into a rule for computing brackets. (Optional) reorder the points together with their geometric constructions.

**Step 1. Eliminate constrained points and semifree points.** First, assume that point $\mathbf{X}$ is the intersection of lines $\mathbf{AB}$ and $\mathbf{CD}$. To eliminate $\mathbf{X}$ from a bracket $[\mathbf{XPQ}]$, there are three formulas available:

$$[\mathbf{XPQ}] = \mathbf{X} \vee (\mathbf{P} \wedge \mathbf{Q}) = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{C} \wedge \mathbf{D}) \vee (\mathbf{P} \wedge \mathbf{Q})$$

$$= \begin{cases} [\mathbf{ABD}][\mathbf{CPQ}] - [\mathbf{ABC}][\mathbf{DPQ}] & (3.1) \\ [\mathbf{ACD}][\mathbf{BPQ}] - [\mathbf{BCD}][\mathbf{APQ}] & (3.2) \\ [\mathbf{ABP}][\mathbf{CDQ}] - [\mathbf{ABQ}][\mathbf{CDP}] & (3.3) \end{cases} \qquad (3)$$

**Rule 1 (same as in the area method).** If a bracket in (3) equals zero, use the corresponding formula.

**Rule 2 (heuristic).** In general, use the formula which separates into different brackets the pair of points in $(\mathbf{A}, \mathbf{B}), (\mathbf{C}, \mathbf{D}), (\mathbf{P}, \mathbf{Q})$ having the largest number of concurrent lines.

In the area method, (3.2) is generally adopted.

Second, assume that point $\mathbf{X}$ is on line $\mathbf{AB}$. Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be linearly independent vectors. To eliminate $\mathbf{X}$ from a bracket polynomial $p$, first contract $p$ (see Step 2), then for each $[\mathbf{XPQ}]$ in $p$, apply the following formula:

$$[\mathbf{ABC}][\mathbf{XPQ}] = [\mathbf{XBC}][\mathbf{APQ}] - [\mathbf{XAC}][\mathbf{BPQ}], \qquad (4)$$

which is a Grassmann-Plücker relation in the case $[\mathbf{XAB}] = 0$.

**Rule 3 (heuristic).** In general, choose $\mathbf{C}$ to be the free point outside line $\mathbf{AB}$ that has the largest number of occurrences in $p$. The nondegeneracy condition is $[\mathbf{ABC}]$ /= 0 if$[\mathbf{ABC}]$ occurs in the denominator of $p$.

**Step 2. Simplification by contraction.** For any vectors $\mathbf{A}_1, \ldots, \mathbf{A}_5$ in $\mathcal{R}^3$,

$$[\mathbf{A}_1\mathbf{A}_2\mathbf{A}_5][\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5] + [\mathbf{A}_1\mathbf{A}_3\mathbf{A}_5][\mathbf{A}_4\mathbf{A}_2\mathbf{A}_5] = [\mathbf{A}_1\mathbf{A}_4\mathbf{A}_5][\mathbf{A}_3\mathbf{A}_2\mathbf{A}_5].$$

This is a Grassmann-Plücker relation.

Let $p$ be a bracket polynomial of two terms. If $p$ is reduced to a single monomial by the above identity, this reduction is called a *contraction*. It can be extended to any bracket polynomial.

It can be proved that a polynomial is reduced to zero modulo the ideal generated by the Grassmann-Plücker polynomials if and only if when multiplied by some bracket polynomial, it is reduced to zero through contractions. As a result, the outcome of the contraction is always zero for $conc = 0$, and nonzero for $conc \, /\!= 0$.

**Output:** The proving procedure and the nondegeneracy conditions.

**Remark 1.** In earlier papers [11,10], we addressed the problem of theorem proving in projective geometry involving both lines and conics. The current algorithm has the function of reordering geometric constructions, i.e., reformulation of geometric theorems. At this moment it works only for incidence geometry.

**Remark 2.** The heuristic rules 2 and 3 can contribute to obtaining short proofs. When searching for a proof composed of polynomials of at most two terms, these rules can serve as guidelines for setting up precedence tables.

**Remark 3.** The method is between the biquadratic final polynomial method [1] and the reduction method in classical invariant theory [2].

**Remark 4.** To improve the performance of the algorithm for $conc = 0$, after each elimination we can delete the common bracket factors in *conc*. These factors are **not** nondegeneracy conditions.

## 3 Examples

Below is a collection of 23 examples and their machine generated proofs composed of polynomials of at most two terms. The program is written in Maple V.4 and runs on an IBM compatible Pentium II/366 with Microsoft Windows 98. The generation of each proof is very fast. The nondegeneracy conditions are generated at the same time.

For theorems of equality type, common bracket factors (underlined) are found out in each step and are deleted before the next step starts.

**Example 1** (See also [3], Example 6.203).

Free points: **1, 2, 3, 4**.
Intersections:

$$\begin{array}{lll} \mathbf{5} = \mathbf{12} \cap \mathbf{34}, & \mathbf{6} = \mathbf{13} \cap \mathbf{24}, & \mathbf{7} = \mathbf{23} \cap \mathbf{14}, \\ \mathbf{8} = \mathbf{23} \cap \mathbf{56}, & \mathbf{9} = \mathbf{24} \cap \mathbf{57}, & \mathbf{0} = \mathbf{34} \cap \mathbf{67}. \end{array}$$

Conclusion: **8, 9, 0** are collinear.

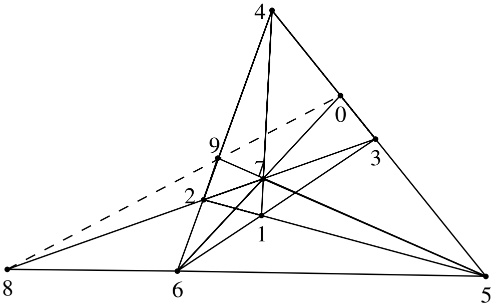


**Fig. 1.** Example 1.

Proof:

| Rules | $[\mathbf{890}]$ |
|---|---|
| | $\overset{\mathbf{0}}{=} [\mathbf{347}][\mathbf{689}]-[\mathbf{346}][\mathbf{789}]$ |
| $[\mathbf{689}] = [\mathbf{248}][\mathbf{567}]$<br>$[\mathbf{789}] = [\mathbf{247}][\mathbf{578}]$ | $\overset{\mathbf{9}}{=} [\mathbf{248}][\mathbf{347}][\mathbf{567}]-[\mathbf{247}][\mathbf{346}][\mathbf{578}]$ |
| $[\mathbf{248}] = [\mathbf{236}][\mathbf{245}]$<br>$[\mathbf{578}] = [\mathbf{235}][\mathbf{567}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{567}]}[\mathbf{236}][\mathbf{245}][\mathbf{347}]-\underline{[\mathbf{567}]}[\mathbf{235}][\mathbf{247}][\mathbf{346}]$ |
| $[\mathbf{347}] = [\mathbf{134}][\mathbf{234}]$<br>$[\mathbf{247}] = [\mathbf{124}][\mathbf{234}]$ | $\overset{\mathbf{7}}{=} \underline{[\mathbf{234}]}[\mathbf{134}][\mathbf{245}][\mathbf{236}]-\underline{[\mathbf{234}]}[\mathbf{124}][\mathbf{235}][\mathbf{346}]$ |
| $[\mathbf{236}] = [\mathbf{123}][\mathbf{234}]$<br>$[\mathbf{346}] = [\mathbf{134}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} \underline{[\mathbf{134}][\mathbf{234}]}([\mathbf{123}][\mathbf{245}]-[\mathbf{124}][\mathbf{235}])$ |
| $[\mathbf{245}] = -[\mathbf{124}][\mathbf{234}]$<br>$[\mathbf{235}] = -[\mathbf{123}][\mathbf{234}]$ | $\overset{\mathbf{5}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 2** (See also [6], Proposition 5.8).

Free points: **1, 2, 3, 4**.
Intersections:

$$\mathbf{5} = \mathbf{12} \cap \mathbf{34},\ \mathbf{6} = \mathbf{13} \cap \mathbf{24},\ \mathbf{7} = \mathbf{23} \cap \mathbf{14},$$
$$\mathbf{8} = \mathbf{13} \cap \mathbf{57},\ \mathbf{9} = \mathbf{67} \cap \mathbf{48},\ \mathbf{0} = \mathbf{24} \cap \mathbf{57}.$$

Conclusion: **3, 9, 0** are collinear.

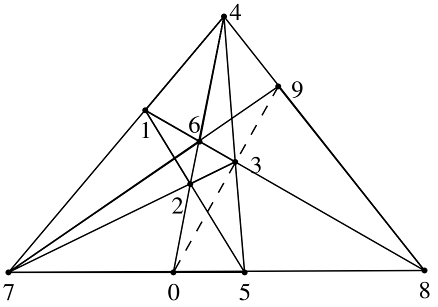


**Fig. 2.** Example 2.

Proof:

| Rules | $[\mathbf{390}]$ |
|---|---|
| | $\overset{\mathbf{0}}{=} -[\mathbf{257}][\mathbf{349}]-[\mathbf{239}][\mathbf{457}]$ |
| $[\mathbf{349}] = -[\mathbf{348}][\mathbf{467}]$<br>$[\mathbf{239}] = \phantom{-}[\mathbf{236}][\mathbf{478}]$ | $\overset{\mathbf{9}}{=} [\mathbf{257}][\mathbf{348}][\mathbf{467}]-[\mathbf{236}][\mathbf{457}][\mathbf{478}]$ |
| $[\mathbf{348}] = -[\mathbf{134}][\mathbf{357}]$<br>$[\mathbf{478}] = -[\mathbf{137}][\mathbf{457}]$ | $\overset{\mathbf{8}}{=} -[\mathbf{134}][\mathbf{257}][\mathbf{357}][\mathbf{467}]+[\mathbf{137}][\mathbf{236}][\mathbf{457}]^2$ |
| $[\mathbf{257}] = \phantom{-}[\mathbf{124}][\mathbf{235}]$<br>$[\mathbf{467}] = \phantom{-}[\mathbf{146}][\mathbf{234}]$<br>$[\mathbf{357}] = \phantom{-}[\mathbf{134}][\mathbf{235}]$<br>$[\mathbf{457}] = \phantom{-}[\mathbf{145}][\mathbf{234}]$<br>$[\mathbf{137}] = -[\mathbf{123}][\mathbf{134}]$ | $\overset{\mathbf{7}}{=} \underline{[\mathbf{134}][\mathbf{234}]}(-[\mathbf{124}][\mathbf{134}][\mathbf{146}][\mathbf{235}]^2-[\mathbf{123}][\mathbf{145}]^2[\mathbf{234}][\mathbf{236}])$ |
| $[\mathbf{146}] = -[\mathbf{124}][\mathbf{134}]$<br>$[\mathbf{236}] = \phantom{-}[\mathbf{123}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} [\mathbf{124}]^2[\mathbf{134}]^2[\mathbf{235}]^2-[\mathbf{123}]^2[\mathbf{145}]^2[\mathbf{234}]^2$ |
| $[\mathbf{235}] = -[\mathbf{123}][\mathbf{234}]$<br>$[\mathbf{145}] = -[\mathbf{124}][\mathbf{134}]$ | $\overset{\mathbf{5}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 3** (See also [14], p. 63).

Free points: **1, 2, 3, 4**.
Intersections:

$$\mathbf{5} = \mathbf{12} \cap \mathbf{34},\ \mathbf{6} = \mathbf{13} \cap \mathbf{24},\ \mathbf{7} = \mathbf{23} \cap \mathbf{14},$$
$$\mathbf{8} = \mathbf{13} \cap \mathbf{57},\ \mathbf{9} = \mathbf{14} \cap \mathbf{56},\ \mathbf{0} = \mathbf{34} \cap \mathbf{67}.$$

Conclusion: **8, 9, 0** are collinear.

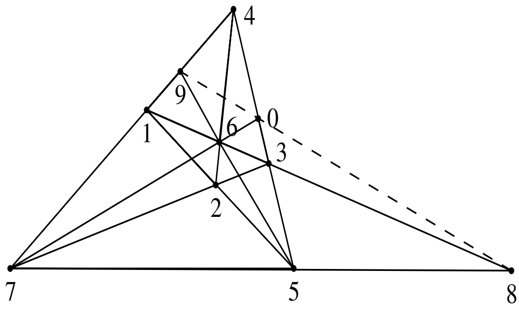


**Fig. 3.** Example 3.

Proof:

| Rules | $[\mathbf{890}]$ |
|---|---|
| | $\overset{\mathbf{0}}{=} [\mathbf{347}][\mathbf{689}]-[\mathbf{346}][\mathbf{789}]$ |
| $[\mathbf{689}] = \quad [\mathbf{156}][\mathbf{468}]$<br>$[\mathbf{789}] = -[\mathbf{148}][\mathbf{567}]$ | $\overset{\mathbf{9}}{=} [\mathbf{156}][\mathbf{347}][\mathbf{468}]+[\mathbf{148}][\mathbf{346}][\mathbf{567}]$ |
| $[\mathbf{468}] = -[\mathbf{134}][\mathbf{567}]$<br>$[\mathbf{148}] = -[\mathbf{134}][\mathbf{157}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{134}][\mathbf{567}]}(-[\mathbf{156}][\mathbf{347}]-[\mathbf{157}][\mathbf{346}])$ |
| $[\mathbf{347}] = [\mathbf{134}][\mathbf{234}]$<br>$[\mathbf{157}] = [\mathbf{123}][\mathbf{145}]$ | $\overset{\mathbf{7}}{=} -[\mathbf{134}][\mathbf{156}][\mathbf{234}]-[\mathbf{123}][\mathbf{145}][\mathbf{346}]$ |
| $[\mathbf{156}] = -[\mathbf{124}][\mathbf{135}]$<br>$[\mathbf{346}] = \quad [\mathbf{134}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} \underline{[\mathbf{134}][\mathbf{234}]}([\mathbf{124}][\mathbf{135}]-[\mathbf{123}][\mathbf{145}])$ |
| $[\mathbf{135}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{145}] = -[\mathbf{124}][\mathbf{134}]$ | $\overset{\mathbf{5}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 4** (See also [3], Example 6.32).

Free points: **1, 2, 3, 4, 5**.

Intersections:

$$\mathbf{6} = \mathbf{12} \cap \mathbf{34},\ \mathbf{7} = \mathbf{13} \cap \mathbf{24},\ \ \mathbf{8} = \mathbf{23} \cap \mathbf{14},\ \ \mathbf{9} = \mathbf{56} \cap \mathbf{78},$$
$$\mathbf{0} = \mathbf{57} \cap \mathbf{68},\ \mathbf{A} = \mathbf{39} \cap \mathbf{20},\ \mathbf{B} = \mathbf{67} \cap \mathbf{58}.$$

Conclusion: **1, A, B** are collinear.

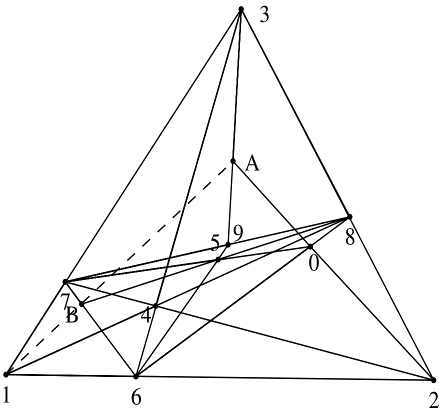


**Fig. 4.** Example 4.

Proof:

| Rules | $[\mathbf{1AB}]$ |
|---|---|
| | $\overset{\mathbf{B}}{=} [\mathbf{17A}][\mathbf{568}]-[\mathbf{16A}][\mathbf{578}]$ |
| $[\mathbf{17A}] = -[\mathbf{179}][\mathbf{230}]$<br>$[\mathbf{16A}] = -[\mathbf{160}][\mathbf{239}]$ | $\overset{\mathbf{A}}{=} -[\mathbf{179}][\mathbf{230}][\mathbf{568}]+[\mathbf{160}][\mathbf{239}][\mathbf{578}]$ |
| $[\mathbf{230}] = [\mathbf{236}][\mathbf{578}]$<br>$[\mathbf{160}] = [\mathbf{168}][\mathbf{567}]$ | $\overset{\mathbf{0}}{=} -\underline{[\mathbf{578}]}[\mathbf{179}][\mathbf{236}][\mathbf{568}]+\underline{[\mathbf{578}]}[\mathbf{168}][\mathbf{239}][\mathbf{567}]$ |
| $[\mathbf{179}] = -[\mathbf{178}][\mathbf{567}]$<br>$[\mathbf{239}] = \quad [\mathbf{237}][\mathbf{568}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{567}][\mathbf{568}]}([\mathbf{178}][\mathbf{236}]+[\mathbf{168}][\mathbf{237}])$ |
| $[\mathbf{178}] = [\mathbf{123}][\mathbf{147}]$<br>$[\mathbf{168}] = [\mathbf{123}][\mathbf{146}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{123}]}[\mathbf{147}][\mathbf{236}]+\underline{[\mathbf{123}]}[\mathbf{146}][\mathbf{237}]$ |
| $[\mathbf{147}] = -[\mathbf{124}][\mathbf{134}]$<br>$[\mathbf{237}] = \quad [\mathbf{123}][\mathbf{234}]$ | $\overset{\mathbf{7}}{=} -[\mathbf{124}][\mathbf{134}][\mathbf{236}]+[\mathbf{123}][\mathbf{146}][\mathbf{234}]$ |
| $[\mathbf{236}] = -[\mathbf{123}][\mathbf{234}]$<br>$[\mathbf{146}] = -[\mathbf{124}][\mathbf{134}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 5** (Pappus point theorem, see also [3], Example 6.22).

Free points: **1, 2, 3, 4, 5**.
Intersections:

$$\mathbf{6} = \mathbf{13} \cap \mathbf{24},\ \mathbf{7} = \mathbf{23} \cap \mathbf{56},\ \ \mathbf{8} = \mathbf{25} \cap \mathbf{34},\ \ \ \mathbf{9} = \mathbf{12} \cap \mathbf{68},$$
$$\mathbf{0} = \mathbf{79} \cap \mathbf{24},\ \mathbf{A} = \mathbf{39} \cap \mathbf{67},\ \mathbf{B} = \mathbf{15} \cap \mathbf{4A},\ \mathbf{C} = \mathbf{28} \cap \mathbf{39}.$$

Conclusion: **0, B, C** are collinear.

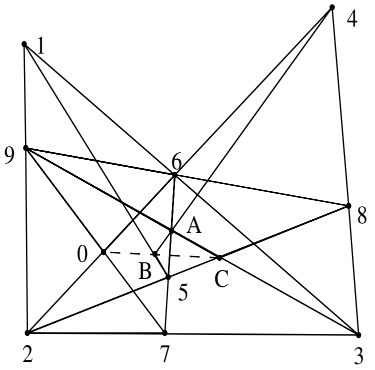


**Fig. 5.** Example 5.

Proof:

| Rules | $[\mathbf{0BC}]$ |
|---|---|
| | $\overset{\mathbf{C}}{=} [\mathbf{280}][\mathbf{39B}]-[\mathbf{28B}][\mathbf{390}]$ |
| $[\mathbf{39B}] = -[\mathbf{15A}][\mathbf{349}]$<br>$[\mathbf{28B}] = [\mathbf{128}][\mathbf{45A}]$ | $\overset{\mathbf{B}}{=} -[\mathbf{15A}][\mathbf{280}][\mathbf{349}]-[\mathbf{128}][\mathbf{390}][\mathbf{45A}]$ |
| $[\mathbf{15A}] = [\mathbf{167}][\mathbf{359}]$<br>$[\mathbf{45A}] = [\mathbf{359}][\mathbf{467}]$ | $\overset{\mathbf{A}}{=} -\underline{[\mathbf{359}]}[\mathbf{167}][\mathbf{280}][\mathbf{349}]-\underline{[\mathbf{359}]}[\mathbf{128}][\mathbf{390}][\mathbf{467}]$ |
| $[\mathbf{280}] = [\mathbf{248}][\mathbf{279}]$<br>$[\mathbf{390}] = [\mathbf{249}][\mathbf{379}]$ | $\overset{\mathbf{0}}{=} -[\mathbf{167}][\mathbf{248}][\mathbf{279}][\mathbf{349}]-[\mathbf{128}][\mathbf{249}][\mathbf{379}][\mathbf{467}]$ |
| $[\mathbf{279}] = -[\mathbf{127}][\mathbf{268}]$<br>$[\mathbf{349}] = [\mathbf{128}][\mathbf{346}]$<br>$[\mathbf{379}] = -[\mathbf{137}][\mathbf{268}]$<br>$[\mathbf{249}] = -[\mathbf{124}][\mathbf{268}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{128}][\mathbf{268}]}([\mathbf{127}][\mathbf{167}][\mathbf{248}][\mathbf{346}]-[\mathbf{124}][\mathbf{137}][\mathbf{268}][\mathbf{467}])$ |
| $[\mathbf{248}] = [\mathbf{234}][\mathbf{245}]$<br>$[\mathbf{268}] = -[\mathbf{234}][\mathbf{256}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{234}]}[\mathbf{127}][\mathbf{167}][\mathbf{245}][\mathbf{346}]+\underline{[\mathbf{234}]}[\mathbf{124}][\mathbf{137}][\mathbf{256}][\mathbf{467}]$ |
| $[\mathbf{127}] = [\mathbf{123}][\mathbf{256}]$<br>$[\mathbf{167}] = -[\mathbf{156}][\mathbf{236}]$<br>$[\mathbf{467}] = -[\mathbf{236}][\mathbf{456}]$<br>$[\mathbf{137}] = [\mathbf{123}][\mathbf{356}]$ | $\overset{\mathbf{7}}{=} \underline{[\mathbf{123}][\mathbf{236}][\mathbf{256}]}(-[\mathbf{156}][\mathbf{245}][\mathbf{346}]-[\mathbf{124}][\mathbf{356}][\mathbf{456}])$ |
| $[\mathbf{156}] = -[\mathbf{124}][\mathbf{135}]$<br>$[\mathbf{346}] = [\mathbf{134}][\mathbf{234}]$<br>$[\mathbf{456}] = [\mathbf{134}][\mathbf{245}]$<br>$[\mathbf{356}] = [\mathbf{135}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 6** (Pappus' theorem, see also [3], Example 6.20).

Free points: $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}$.
Semifree points: $\mathbf{5}$ on $\mathbf{12}$, $\mathbf{6}$ on $\mathbf{34}$.
Intersections: $\mathbf{7} = \mathbf{23} \cap \mathbf{14}$, $\mathbf{8} = \mathbf{35} \cap \mathbf{16}$, $\mathbf{9} = \mathbf{45} \cap \mathbf{26}$.
Conclusion: **7, 8, 9** are collinear.

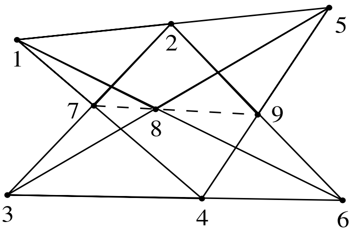


**Fig. 6.** Example 6.

Proof:

| Rules | |
|---|---|
| | $[\mathbf{789}]$ |
| | $\overset{\mathbf{9}}{=} [\mathbf{278}][\mathbf{456}]-[\mathbf{245}][\mathbf{678}]$ |
| $[\mathbf{278}] = [\mathbf{136}][\mathbf{257}]$<br>$[\mathbf{678}] = [\mathbf{167}][\mathbf{356}]$ | $\overset{\mathbf{8}}{=} [\mathbf{136}][\mathbf{257}][\mathbf{456}]-[\mathbf{167}][\mathbf{245}][\mathbf{356}]$ |
| $[\mathbf{257}] = [\mathbf{124}][\mathbf{235}]$<br>$[\mathbf{167}] = [\mathbf{123}][\mathbf{146}]$ | $\overset{\mathbf{7}}{=} [\mathbf{124}][\mathbf{136}][\mathbf{235}][\mathbf{456}]-[\mathbf{123}][\mathbf{146}][\mathbf{245}][\mathbf{356}]$ |
| $[\mathbf{134}][\mathbf{456}] = -[\mathbf{146}][\mathbf{345}]$<br>$[\mathbf{134}][\mathbf{356}] = -[\mathbf{136}][\mathbf{345}]$ | $\overset{\mathbf{6}}{=} \frac{\underline{[\mathbf{136}][\mathbf{146}][\mathbf{345}]}}{\underline{[\mathbf{134}]}}(-[\mathbf{124}][\mathbf{235}]+[\mathbf{123}][\mathbf{245}])$ |
| | $= 0.$ |

Nondegeneracy condition: $[\mathbf{134}]$ /= 0.

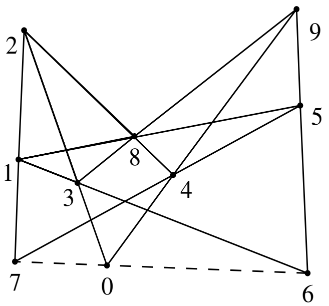


**Fig. 7.** Example 7.

**Example 7** (Desargues' theorem, see also [3], Example 6.24).

Free points: $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}$.
Semifree point: $\mathbf{6}$ on $\mathbf{13}$.
Intersections: $\mathbf{7} = \mathbf{12} \cap \mathbf{45}$, $\mathbf{8} = \mathbf{15} \cap \mathbf{24}$, $\mathbf{9} = \mathbf{38} \cap \mathbf{56}$, $\mathbf{0} = \mathbf{23} \cap \mathbf{49}$.
Conclusion: **6, 7, 0** are collinear.
Proof:

| Rules | |
|---|---|
| | $[\mathbf{670}]$ |
| | $\overset{\mathbf{0}}{=} [\mathbf{239}][\mathbf{467}]-[\mathbf{234}][\mathbf{679}]$ |
| $[\mathbf{239}] = -[\mathbf{238}][\mathbf{356}]$<br>$[\mathbf{679}] = [\mathbf{368}][\mathbf{567}]$ | $\overset{\mathbf{9}}{=} -[\mathbf{238}][\mathbf{356}][\mathbf{467}]-[\mathbf{234}][\mathbf{368}][\mathbf{567}]$ |
| $[\mathbf{238}] = -[\mathbf{125}][\mathbf{234}]$<br>$[\mathbf{368}] = [\mathbf{124}][\mathbf{356}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{234}][\mathbf{356}]}([\mathbf{125}][\mathbf{467}]-[\mathbf{124}][\mathbf{567}])$ |
| $[\mathbf{467}] = [\mathbf{124}][\mathbf{456}]$<br>$[\mathbf{567}] = [\mathbf{125}][\mathbf{456}]$ | $\overset{\mathbf{7}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 8** (See also [3], Example 6.34).

Free points: **1, 2, 3**.
Semifree points: **4** on **12**, **5** on **12**, **6** on **13**, **7** on **23**.
Intersections:

$$\mathbf{8} = \mathbf{23} \cap \mathbf{46}, \quad \mathbf{9} = \mathbf{23} \cap \mathbf{56}, \quad \mathbf{0} = \mathbf{13} \cap \mathbf{57},$$
$$\mathbf{A} = \mathbf{13} \cap \mathbf{47}, \quad \mathbf{B} = \mathbf{12} \cap \mathbf{80}.$$

Conclusion: **9, A, B** are collinear.

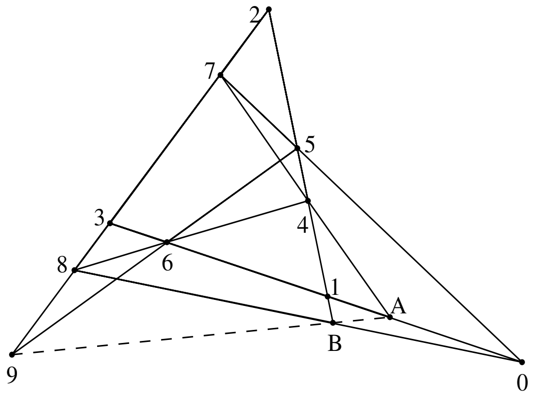


**Fig. 8.** Example 8.

Proof:

| Rules | $[\mathbf{9AB}]$ |
|---|---|
| | $\overset{\mathbf{B}}{=} [\mathbf{120}][\mathbf{89A}]+[\mathbf{128}][\mathbf{90A}]$ |
| $[\mathbf{89A}] = [\mathbf{137}][\mathbf{489}]$<br>$[\mathbf{90A}] = [\mathbf{139}][\mathbf{470}]$ | $\overset{\mathbf{A}}{=} [\mathbf{120}][\mathbf{137}][\mathbf{489}]+[\mathbf{128}][\mathbf{139}][\mathbf{470}]$ |
| $[\mathbf{120}] = [\mathbf{123}][\mathbf{157}]$<br>$[\mathbf{470}] = -[\mathbf{137}][\mathbf{457}]$ | $\overset{\mathbf{0}}{=} \underline{[\mathbf{137}]}[\mathbf{123}][\mathbf{157}][\mathbf{489}]-\underline{[\mathbf{137}]}[\mathbf{128}][\mathbf{139}][\mathbf{457}]$ |
| $[\mathbf{489}] = -[\mathbf{236}][\mathbf{458}]$<br>$[\mathbf{139}] = [\mathbf{123}][\mathbf{356}]$ | $\overset{\mathbf{9}}{=} -\underline{[\mathbf{123}]}[\mathbf{157}][\mathbf{236}][\mathbf{458}]-\underline{[\mathbf{123}]}[\mathbf{128}][\mathbf{356}][\mathbf{457}]$ |
| $[\mathbf{458}] = -[\mathbf{234}][\mathbf{456}]$<br>$[\mathbf{128}] = [\mathbf{123}][\mathbf{246}]$ | $\overset{\mathbf{8}}{=} [\mathbf{157}][\mathbf{234}][\mathbf{236}][\mathbf{456}]-[\mathbf{123}][\mathbf{246}][\mathbf{356}][\mathbf{457}]$ |
| $[\mathbf{123}][\mathbf{157}] = -[\mathbf{127}][\mathbf{135}]$<br>$[\mathbf{123}][\mathbf{457}] = [\mathbf{127}][\mathbf{345}]$ | $\overset{\mathbf{7}}{=} \frac{\underline{[\mathbf{127}]}}{\underline{[\mathbf{123}]}}(-[\mathbf{135}][\mathbf{234}][\mathbf{236}][\mathbf{456}]-[\mathbf{123}][\mathbf{246}][\mathbf{345}][\mathbf{356}])$ |
| $[\mathbf{123}][\mathbf{456}] = [\mathbf{126}][\mathbf{345}]$<br>$[\mathbf{123}][\mathbf{356}] = [\mathbf{135}][\mathbf{236}]$<br>$[\mathbf{123}][\mathbf{246}] = -[\mathbf{126}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: $[\mathbf{123}]$ /= 0.

**Example 9** (See also [3], Example 6.38).

Free points: **1, 2, 3, 4**.
Semifree point: **5** on **12**.
Intersections:

$$\mathbf{6} = \mathbf{12} \cap \mathbf{34},\ \mathbf{7} = \mathbf{13} \cap \mathbf{24},\ \mathbf{8} = \mathbf{23} \cap \mathbf{14},\ \mathbf{9} = \mathbf{13} \cap \mathbf{45},\ \mathbf{0} = \mathbf{23} \cap \mathbf{45},$$
$$\mathbf{A} = \mathbf{14} \cap \mathbf{35},\ \mathbf{B} = \mathbf{24} \cap \mathbf{35},\ \mathbf{C} = \mathbf{12} \cap \mathbf{89},\ \mathbf{D} = \mathbf{12} \cap \mathbf{70},\ \mathbf{E} = \mathbf{12} \cap \mathbf{0A}.$$

Conclusions: (1) **7, A, C** are collinear; (2) **8**, **B**, **D** are collinear; (3) **9**, **B**, **E** are collinear.

Proof: (1)

| Rules | $[\mathbf{7AC}]$ |
|---|---|
| | $\overset{\mathbf{C}}{=} [\mathbf{189}][\mathbf{27A}]-[\mathbf{17A}][\mathbf{289}]$ |
| $[\mathbf{27A}] = [\mathbf{127}][\mathbf{345}]$<br>$[\mathbf{17A}] = -[\mathbf{135}][\mathbf{147}]$ | $\overset{\mathbf{A}}{=} [\mathbf{127}][\mathbf{189}][\mathbf{345}]+[\mathbf{135}][\mathbf{147}][\mathbf{289}]$ |
| $[\mathbf{189}] = -[\mathbf{138}][\mathbf{145}]$<br>$[\mathbf{289}] = -[\mathbf{128}][\mathbf{345}]$ | $\overset{\mathbf{9}}{=} -\underline{[\mathbf{345}]}[\mathbf{127}][\mathbf{138}][\mathbf{145}]-\underline{[\mathbf{345}]}[\mathbf{128}][\mathbf{135}][\mathbf{147}]$ |
| $[\mathbf{138}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{128}] = -[\mathbf{123}][\mathbf{124}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{123}]}[\mathbf{127}][\mathbf{134}][\mathbf{145}]+\underline{[\mathbf{123}]}[\mathbf{124}][\mathbf{135}][\mathbf{147}]$ |

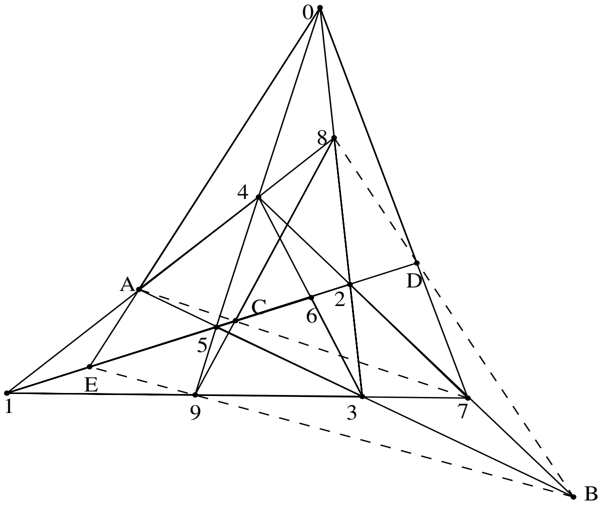


**Fig. 9.** Example 9.

$$\begin{array}{ll} \boxed{\begin{array}{l} [\mathbf{127}] = \phantom{-}[\mathbf{123}][\mathbf{124}] \\ [\mathbf{147}] = -[\mathbf{124}][\mathbf{134}] \end{array}} & \overset{\underline{\mathbf{7}}}{=} \underline{[\mathbf{124}][\mathbf{134}]}([\mathbf{123}][\mathbf{145}]-[\mathbf{124}][\mathbf{135}]) \\ & = 0. \end{array}$$

(2)

$$\begin{array}{ll} \text{Rules} & [\mathbf{8BD}] \\ & \overset{\underline{\mathbf{D}}}{=} [\mathbf{170}][\mathbf{28B}]-[\mathbf{18B}][\mathbf{270}] \\ \boxed{\begin{array}{l} [\mathbf{28B}] = -[\mathbf{235}][\mathbf{248}] \\ [\mathbf{18B}] = -[\mathbf{128}][\mathbf{345}] \end{array}} & \overset{\underline{\mathbf{B}}}{=} -[\mathbf{170}][\mathbf{235}][\mathbf{248}]+[\mathbf{128}][\mathbf{270}][\mathbf{345}] \\ \boxed{\begin{array}{l} [\mathbf{170}] = \phantom{-}[\mathbf{127}][\mathbf{345}] \\ [\mathbf{270}] = -[\mathbf{237}][\mathbf{245}] \end{array}} & \overset{\underline{\mathbf{0}}}{=} -\underline{[\mathbf{345}]}[\mathbf{127}][\mathbf{235}][\mathbf{248}]-\underline{[\mathbf{345}]}[\mathbf{128}][\mathbf{237}][\mathbf{245}] \\ \boxed{\begin{array}{l} [\mathbf{248}] = -[\mathbf{124}][\mathbf{234}] \\ [\mathbf{128}] = -[\mathbf{123}][\mathbf{124}] \end{array}} & \overset{\underline{\mathbf{8}}}{=} \underline{[\mathbf{124}]}[\mathbf{127}][\mathbf{234}][\mathbf{235}]+\underline{[\mathbf{124}]}[\mathbf{123}][\mathbf{237}][\mathbf{245}] \\ \boxed{\begin{array}{l} [\mathbf{127}] = -[\mathbf{123}][\mathbf{124}] \\ [\mathbf{237}] = \phantom{-}[\mathbf{123}][\mathbf{234}] \end{array}} & \overset{\underline{\mathbf{7}}}{=} \underline{[\mathbf{123}][\mathbf{234}]}(-[\mathbf{124}][\mathbf{235}]+[\mathbf{123}][\mathbf{245}]) \\ & = 0. \end{array}$$

(3)

$$\begin{array}{ll} \text{Rules} & [\mathbf{9BE}] \\ & \overset{\mathbf{E}}{=} [\mathbf{10A}][\mathbf{29B}]-[\mathbf{19B}][\mathbf{20A}] \\ \boxed{\begin{array}{l} [\mathbf{29B}] = -[\mathbf{235}][\mathbf{249}] \\ [\mathbf{19B}] = -[\mathbf{159}][\mathbf{234}] \end{array}} & \overset{\underline{\mathbf{B}}}{=} -[\mathbf{10A}][\mathbf{235}][\mathbf{249}]+[\mathbf{159}][\mathbf{234}][\mathbf{20A}] \end{array}$$

| | |
|---|---|
| $[\mathbf{10A}] = -[\mathbf{135}][\mathbf{140}]$<br>$[\mathbf{20A}] = -[\mathbf{134}][\mathbf{250}]$ | $\overset{\mathbf{A}}{=} [\mathbf{135}][\mathbf{140}][\mathbf{235}][\mathbf{249}]-[\mathbf{134}][\mathbf{159}][\mathbf{234}][\mathbf{250}]$ |
| $[\mathbf{140}] = -[\mathbf{145}][\mathbf{234}]$<br>$[\mathbf{250}] = -[\mathbf{235}][\mathbf{245}]$ | $\overset{\mathbf{0}}{=} \underline{[\mathbf{234}][\mathbf{235}]}(-[\mathbf{135}][\mathbf{145}][\mathbf{249}]+[\mathbf{134}][\mathbf{159}][\mathbf{245}])$ |
| $[\mathbf{249}] = -[\mathbf{134}][\mathbf{245}]$<br>$[\mathbf{159}] = -[\mathbf{145}][\mathbf{135}]$ | $\overset{\mathbf{9}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 10** (See also [3], Example 6.208).

Free points: **1, 2, 3, 4**.
Semifree point: **5** on **12**.
Intersections:

$$\mathbf{6} = \mathbf{12} \cap \mathbf{34},\ \mathbf{7} = \mathbf{13} \cap \mathbf{24},\ \ \mathbf{8} = \mathbf{13} \cap \mathbf{45},\ \ \mathbf{9} = \mathbf{23} \cap \mathbf{67},$$
$$\mathbf{0} = \mathbf{24} \cap \mathbf{19},\ \mathbf{A} = \mathbf{34} \cap \mathbf{19},\ \mathbf{B} = \mathbf{23} \cap \mathbf{80},\ \mathbf{C} = \mathbf{49} \cap \mathbf{30}.$$

Conclusions: (1) **5, A, B** are collinear; (2) **7**, **A**, **C** are collinear.

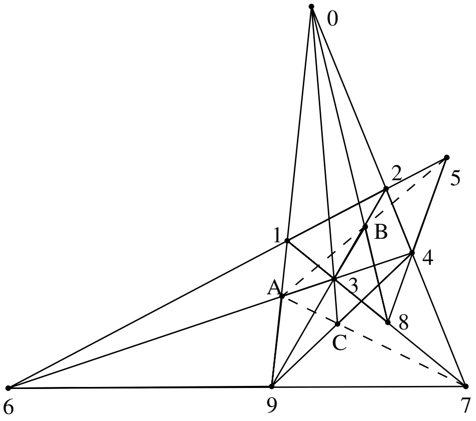


**Fig. 10.** Example 10.

Proof: (1)

| Rules | |
|---|---|
| | $[\mathbf{5AB}]$ |
| | $\overset{\mathbf{B}}{=} [\mathbf{280}][\mathbf{35A}]-[\mathbf{25A}][\mathbf{380}]$ |
| $[\mathbf{35A}] = \ \ [\mathbf{139}][\mathbf{345}]$<br>$[\mathbf{25A}] = -[\mathbf{134}][\mathbf{259}]$ | $\overset{\mathbf{A}}{=} [\mathbf{139}][\mathbf{280}][\mathbf{345}]+[\mathbf{134}][\mathbf{259}][\mathbf{380}]$ |
| $[\mathbf{280}] = \ \ [\mathbf{129}][\mathbf{248}]$<br>$[\mathbf{380}] = -[\mathbf{124}][\mathbf{389}]$ | $\overset{\mathbf{0}}{=} [\mathbf{129}][\mathbf{139}][\mathbf{248}][\mathbf{345}]-[\mathbf{124}][\mathbf{134}][\mathbf{259}][\mathbf{389}]$ |

| | |
|---|---|
| $[\mathbf{129}] = [\mathbf{123}][\mathbf{267}]$<br>$[\mathbf{139}] = [\mathbf{123}][\mathbf{367}]$<br>$[\mathbf{389}] = -[\mathbf{238}][\mathbf{367}]$<br>$[\mathbf{259}] = -[\mathbf{235}][\mathbf{267}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{267}][\mathbf{367}]}([\mathbf{123}]^2[\mathbf{248}][\mathbf{345}]-[\mathbf{124}][\mathbf{134}][\mathbf{235}][\mathbf{238}])$ |
| $[\mathbf{248}] = -[\mathbf{134}][\mathbf{245}]$<br>$[\mathbf{238}] = -[\mathbf{123}][\mathbf{345}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{123}][\mathbf{134}][\mathbf{345}]}(-[\mathbf{123}][\mathbf{245}]+[\mathbf{124}][\mathbf{235}])$ |
| | $= 0.$ |

(2)

| Rules | |
|---|---|
| | $[\mathbf{7AC}]$ |
| | $\overset{\mathbf{C}}{=} [\mathbf{340}][\mathbf{79A}]+[\mathbf{390}][\mathbf{47A}]$ |
| $[\mathbf{79A}] = [\mathbf{179}][\mathbf{349}]$<br>$[\mathbf{47A}] = [\mathbf{149}][\mathbf{347}]$ | $\overset{\mathbf{A}}{=} [\mathbf{179}][\mathbf{349}][\mathbf{340}]+[\mathbf{149}][\mathbf{347}][\mathbf{390}]$ |
| $[\mathbf{340}] = [\mathbf{149}][\mathbf{234}]$<br>$[\mathbf{390}] = [\mathbf{139}][\mathbf{249}]$ | $\overset{\mathbf{0}}{=} \underline{[\mathbf{149}]}[\mathbf{179}][\mathbf{234}][\mathbf{349}]+\underline{[\mathbf{149}]}[\mathbf{139}][\mathbf{249}][\mathbf{347}]$ |
| $[\mathbf{179}] = -[\mathbf{167}][\mathbf{237}]$<br>$[\mathbf{349}] = -[\mathbf{234}][\mathbf{367}]$<br>$[\mathbf{249}] = -[\mathbf{234}][\mathbf{267}]$<br>$[\mathbf{139}] = [\mathbf{123}][\mathbf{367}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{234}][\mathbf{367}]}([\mathbf{167}][\mathbf{234}][\mathbf{237}]-[\mathbf{123}][\mathbf{267}][\mathbf{347}])$ |
| $[\mathbf{237}] = [\mathbf{123}][\mathbf{234}]$<br>$[\mathbf{167}] = -[\mathbf{124}][\mathbf{136}]$<br>$[\mathbf{347}] = [\mathbf{134}][\mathbf{234}]$<br>$[\mathbf{267}] = -[\mathbf{123}][\mathbf{246}]$ | $\overset{\mathbf{7}}{=} \underline{[\mathbf{123}][\mathbf{234}]}(-[\mathbf{124}][\mathbf{136}][\mathbf{234}]+[\mathbf{123}][\mathbf{134}][\mathbf{246}])$ |
| $[\mathbf{136}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{246}] = -[\mathbf{124}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 11** (Nehring's theorem, see also [3], Example 6.27).

Free points: **1, 2, 3, 4**.
Semifree point: **5** on **12**.
Intersections:

$$\begin{array}{l} \mathbf{6} = \mathbf{12} \cap \mathbf{34},\ \mathbf{7} = \mathbf{13} \cap \mathbf{24},\ \ \mathbf{8} = \mathbf{23} \cap \mathbf{14},\ \ \ \mathbf{9} = \mathbf{13} \cap \mathbf{58}, \\ \mathbf{0} = \mathbf{23} \cap \mathbf{69},\ \mathbf{A} = \mathbf{12} \cap \mathbf{70},\ \mathbf{B} = \mathbf{13} \cap \mathbf{8A},\ \mathbf{C} = \mathbf{23} \cap \mathbf{6B}. \end{array}$$

Conclusion: **5, 7, C** are collinear.

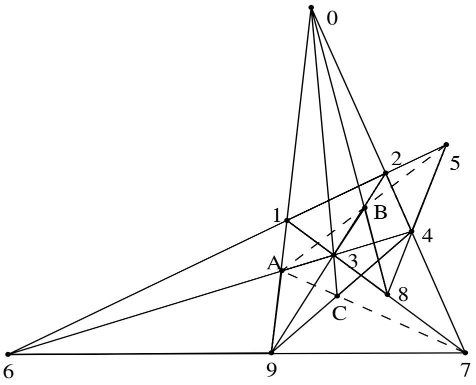


**Fig. 11.** Example 11.

Proof:

| Rules | |
|---|---|
| | $[\mathbf{57C}]$ |
| | $\overset{\mathbf{C}}{=} -[\mathbf{235}][\mathbf{67B}]-[\mathbf{237}][\mathbf{56B}]$ |
| $[\mathbf{67B}] = [\mathbf{136}][\mathbf{78A}]$<br>$[\mathbf{56B}] = [\mathbf{13A}][\mathbf{568}]$ | $\overset{\mathbf{B}}{=} -[\mathbf{136}][\mathbf{235}][\mathbf{78A}]-[\mathbf{13A}][\mathbf{237}][\mathbf{568}]$ |
| $[\mathbf{78A}] = -[\mathbf{127}][\mathbf{780}]$<br>$[\mathbf{13A}] = -[\mathbf{123}][\mathbf{170}]$ | $\overset{\mathbf{A}}{=} [\mathbf{127}][\mathbf{136}][\mathbf{235}][\mathbf{780}]+[\mathbf{123}][\mathbf{170}][\mathbf{237}][\mathbf{568}]$ |
| $[\mathbf{780}] = -[\mathbf{237}][\mathbf{689}]$<br>$[\mathbf{170}] = [\mathbf{127}][\mathbf{369}]$ | $\overset{\mathbf{0}}{=} \underline{[\mathbf{127}][\mathbf{237}]}(-[\mathbf{136}][\mathbf{235}][\mathbf{689}]+[\mathbf{123}][\mathbf{369}][\mathbf{568}])$ |
| $[\mathbf{689}] = [\mathbf{138}][\mathbf{568}]$<br>$[\mathbf{369}] = -[\mathbf{136}][\mathbf{358}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{136}][\mathbf{568}]}(-[\mathbf{138}][\mathbf{235}]-[\mathbf{123}][\mathbf{358}])$ |
| $[\mathbf{138}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{358}] = [\mathbf{134}][\mathbf{235}]$ | $\overset{\mathbf{8}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 12** (See also [15], Example 7).

Free points: **1, 2, 3, 4, 5, 6, 7, 8, 9**.
Semifree point: **0** on **19**.
Intersections:

$$\begin{array}{llll}
\mathbf{A} = \mathbf{13} \cap \mathbf{24}, & \mathbf{B} = \mathbf{24} \cap \mathbf{35}, & \mathbf{C} = \mathbf{35} \cap \mathbf{46}, & \mathbf{D} = \mathbf{46} \cap \mathbf{57}, \\
\mathbf{E} = \mathbf{57} \cap \mathbf{68}, & \mathbf{F} = \mathbf{68} \cap \mathbf{17}, & \mathbf{G} = \mathbf{17} \cap \mathbf{28}, & \mathbf{H} = \mathbf{28} \cap \mathbf{13}, \\
\mathbf{A_1} = \mathbf{29} \cap \mathbf{0H}, & \mathbf{B_1} = \mathbf{39} \cap \mathbf{AA_1}, & \mathbf{C_1} = \mathbf{49} \cap \mathbf{BB_1}, & \mathbf{D_1} = \mathbf{59} \cap \mathbf{CC_1}, \\
\mathbf{E_1} = \mathbf{69} \cap \mathbf{DD_1}, & \mathbf{F_1} = \mathbf{79} \cap \mathbf{EE_1}, & \mathbf{G_1} = \mathbf{89} \cap \mathbf{FF_1}. &
\end{array}$$

Conclusion: **0**, **G**, $\mathbf{G}_1$ are collinear.

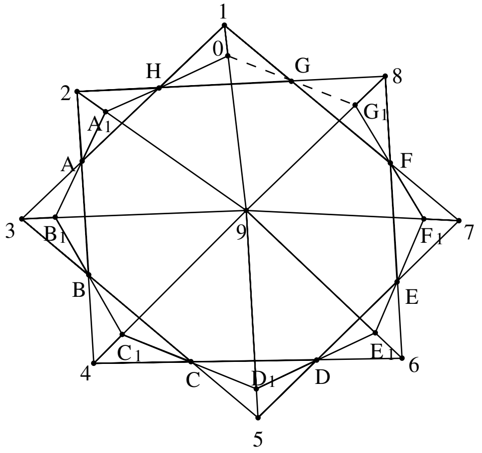


**Fig. 12.** Example 12.

Proof:

| Rules | | $\mathbf{[0GG_1]}$ |
|---|---|---|
| | $\overset{\mathbf{G_1}}{=}$ | $\mathbf{[8FF_1][90G]-[80G][9FF_1]}$ |
| $\mathbf{[8FF_1]} = -\mathbf{[79E][8FE_1]}$<br>$\mathbf{[9FF_1]} = -\mathbf{[79F][9EE_1]}$ | $\overset{\mathbf{F_1}}{=}$ | $-\mathbf{[79E][8FE_1][90G]+[79F][80G][9EE_1]}$ |
| $\mathbf{[8FE_1]} = -\mathbf{[6DD_1][89F]}$<br>$\mathbf{[9EE_1]} = -\mathbf{[69E][9DD_1]}$ | $\overset{\mathbf{E_1}}{=}$ | $\mathbf{[6DD_1][79E][89F][90G]-[69D][79F][80G][9DD_1]}$ |
| $\mathbf{[6DD_1]} = -\mathbf{[59C][6DC_1]}$<br>$\mathbf{[9DD_1]} = -\mathbf{[59D][9CC_1]}$ | $\overset{\mathbf{D_1}}{=}$ | $-\mathbf{[59C][6DC_1][79E][89F][90G]+[59D][69E][79F][80G][9CC_1]}$ |
| $\mathbf{[6DC_1]} = -\mathbf{[4BB_1][69D]}$<br>$\mathbf{[9CC_1]} = -\mathbf{[49C][9BB_1]}$ | $\overset{\mathbf{C_1}}{=}$ | $\mathbf{[4BB_1][59C][69D][79E][89F][90G]}$<br>$-\mathbf{[49C][59D][69E][79F][9BB_1][0G8]}$ |
| $\mathbf{[4BB_1]} = -\mathbf{[39A][4BA_1]}$<br>$\mathbf{[9BB_1]} = -\mathbf{[39B][9AA_1]}$ | $\overset{\mathbf{B_1}}{=}$ | $-\mathbf{[39A][4BA_1][59C][69D][79E][89F][90G]}$<br>$+\mathbf{[39B][49C][59D][69E][79F][80G][9AA_1]}$ |
| $\mathbf{[4BA_1]} = -\mathbf{[20H][49B]}$<br>$\mathbf{[9AA_1]} = -\mathbf{[29A][90H]}$ | $\overset{\mathbf{A_1}}{=}$ | $\mathbf{[20H][39A][49B][59C][69D][79E][89F][90G]}$<br>$-\mathbf{[29A][39B][49C][59D][69E][79F][80G][90H]}$ |
| $\mathbf{[20H]} = \mathbf{[123][280]}$<br>$\mathbf{[90H]} = -\mathbf{[128][390]}$ | $\overset{\mathbf{H}}{=}$ | $\mathbf{[123][280][39A][49B][59C][69D][79E][89F][90G]}$<br>$+\mathbf{[128][29A][390][39B][49C][59D][69E][79F][80G]}$ |
| $\mathbf{[90G]} = \mathbf{[128][790]}$<br>$\mathbf{[80G]} = \mathbf{[178][280]}$ | $\overset{\mathbf{G}}{=}$ | $\underline{\mathbf{[128][280]}}\mathbf{[123][39A][49B][59C][69D][790][79E][89F]}$<br>$+\underline{\mathbf{[128][280]}}\mathbf{[178][29A][390][39B][49C][59D][69E][79F]}$ |
| $\mathbf{[89F]} = -\mathbf{[178][689]}$<br>$\mathbf{[79F]} = -\mathbf{[179][678]}$ | $\overset{\mathbf{F}}{=}$ | $-\underline{\mathbf{[178]}}\mathbf{[123][39A][49B][59C][689][69D][790][79E]}$<br>$-\underline{\mathbf{[178]}}\mathbf{[179][29A][390][39B][49C][59D][678][69E]}$ |

$$\boxed{\begin{array}{rl}[\mathbf{79E}] = & [\mathbf{579}][\mathbf{678}] \\ [\mathbf{69E}] = & -[\mathbf{567}][\mathbf{689}]\end{array}}$$

$$\begin{aligned}\overset{\underline{\mathbf{E}}}{=} & -\underline{[\mathbf{678}][\mathbf{689}]}[\mathbf{123}][\mathbf{39A}][\mathbf{49B}][\mathbf{579}][\mathbf{59C}][\mathbf{69D}][\mathbf{790}] \\ & +\underline{[\mathbf{678}][\mathbf{689}]}[\mathbf{179}][\mathbf{29A}][\mathbf{390}][\mathbf{39B}][\mathbf{49C}][\mathbf{567}][\mathbf{59D}]\end{aligned}$$

$$\boxed{\begin{array}{rl}[\mathbf{69D}] = & [\mathbf{469}][\mathbf{567}] \\ [\mathbf{59D}] = & -[\mathbf{456}][\mathbf{579}]\end{array}}$$

$$\begin{aligned}\overset{\underline{\mathbf{D}}}{=} & -\underline{[\mathbf{567}][\mathbf{579}]}[\mathbf{123}][\mathbf{39A}][\mathbf{469}][\mathbf{49B}][\mathbf{59C}][\mathbf{790}] \\ & -\underline{[\mathbf{567}][\mathbf{579}]}[\mathbf{179}][\mathbf{29A}][\mathbf{390}][\mathbf{39B}][\mathbf{456}][\mathbf{49C}]\end{aligned}$$

$$\boxed{\begin{array}{rl}[\mathbf{59C}] = & [\mathbf{359}][\mathbf{456}] \\ [\mathbf{49C}] = & -[\mathbf{345}][\mathbf{469}]\end{array}}$$

$$\begin{aligned}\overset{\underline{\mathbf{C}}}{=} & -\underline{[\mathbf{456}][\mathbf{469}]}[\mathbf{123}][\mathbf{359}][\mathbf{39A}][\mathbf{49B}][\mathbf{790}] \\ & +\underline{[\mathbf{456}][\mathbf{469}]}[\mathbf{179}][\mathbf{29A}][\mathbf{345}][\mathbf{390}][\mathbf{39B}]\end{aligned}$$

$$\boxed{\begin{array}{rl}[\mathbf{49B}] = & [\mathbf{249}][\mathbf{345}] \\ [\mathbf{39B}] = & -[\mathbf{234}][\mathbf{359}]\end{array}}$$

$$\overset{\underline{\mathbf{B}}}{=} \underline{[\mathbf{345}][\mathbf{359}]}(-[\mathbf{123}][\mathbf{249}][\mathbf{39A}][\mathbf{790}]-[\mathbf{179}][\mathbf{234}][\mathbf{29A}][\mathbf{390}])$$

$$\boxed{\begin{array}{rl}[\mathbf{39A}] = & [\mathbf{139}][\mathbf{234}] \\ [\mathbf{29A}] = & -[\mathbf{123}][\mathbf{249}]\end{array}}$$

$$\overset{\underline{\mathbf{A}}}{=} \underline{[\mathbf{123}][\mathbf{234}][\mathbf{249}]}(-[\mathbf{139}][\mathbf{790}]+[\mathbf{179}][\mathbf{390}])$$

$$= 0.$$

Nondegeneracy condition: none.

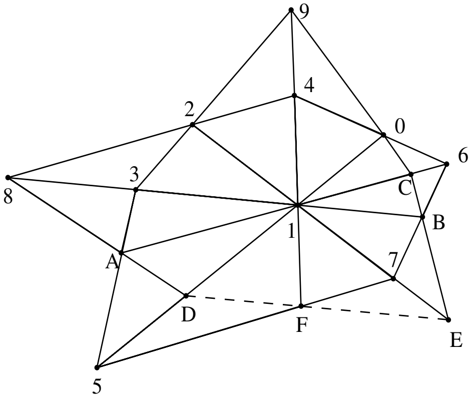


**Fig. 13.** Example 13.

**Example 13** (Saam's theorem, see also [15], Example 6).

Free points: **1, 2, 3, 4, 5, 6**.
Semifree point: **7** on **12**.
Intersections:

$$\begin{array}{lll}\mathbf{8} = \mathbf{13} \cap \mathbf{24}, & \mathbf{9} = \mathbf{23} \cap \mathbf{14}, & \mathbf{0} = \mathbf{15} \cap \mathbf{46}, \\ \mathbf{A} = \mathbf{35} \cap \mathbf{16}, & \mathbf{B} = \mathbf{13} \cap \mathbf{67}, & \mathbf{C} = \mathbf{16} \cap \mathbf{90}, \\ \mathbf{D} = \mathbf{15} \cap \mathbf{8A}, & \mathbf{E} = \mathbf{12} \cap \mathbf{BC}, & \mathbf{F} = \mathbf{57} \cap \mathbf{14}.\end{array}$$

Conclusion: **D, E, F** are collinear.

Proof:

| Rules | $[\mathbf{DEF}]$ |
|---|---|
| | $\overset{\mathbf{F}}{=} [\mathbf{145}][\mathbf{7DE}]-[\mathbf{147}][\mathbf{5DE}]$ |
| $[\mathbf{7DE}] = -[\mathbf{12D}][\mathbf{7BC}]$<br>$[\mathbf{5DE}] = [\mathbf{1BC}][\mathbf{25D}]$ | $\overset{\mathbf{E}}{=} -[\mathbf{12D}][\mathbf{145}][\mathbf{7BC}]-[\mathbf{147}][\mathbf{1BC}][\mathbf{25D}]$ |
| $[\mathbf{12D}] = [\mathbf{125}][\mathbf{18A}]$<br>$[\mathbf{25D}] = -[\mathbf{125}][\mathbf{58A}]$ | $\overset{\mathbf{D}}{=} -\underline{[\mathbf{125}]}[\mathbf{145}][\mathbf{18A}][\mathbf{7BC}]+\underline{[\mathbf{125}]}[\mathbf{147}][\mathbf{1BC}][\mathbf{58A}]$ |
| $[\mathbf{7BC}] = -[\mathbf{17B}][\mathbf{690}]$<br>$[\mathbf{1BC}] = -[\mathbf{16B}][\mathbf{190}]$ | $\overset{\mathbf{C}}{=} [\mathbf{145}][\mathbf{17B}][\mathbf{18A}][\mathbf{690}]-[\mathbf{147}][\mathbf{16B}][\mathbf{190}][\mathbf{58A}]$ |
| $[\mathbf{17B}] = -[\mathbf{137}][\mathbf{167}]$<br>$[\mathbf{16B}] = -[\mathbf{136}][\mathbf{167}]$ | $\overset{\mathbf{B}}{=} -\underline{[\mathbf{167}]}[\mathbf{137}][\mathbf{145}][\mathbf{18A}][\mathbf{690}]+\underline{[\mathbf{167}]}[\mathbf{136}][\mathbf{147}][\mathbf{190}][\mathbf{58A}]$ |
| $[\mathbf{18A}] = [\mathbf{135}][\mathbf{168}]$<br>$[\mathbf{58A}] = [\mathbf{156}][\mathbf{358}]$ | $\overset{\mathbf{A}}{=} -[\mathbf{135}][\mathbf{137}][\mathbf{145}][\mathbf{168}][\mathbf{690}]+[\mathbf{136}][\mathbf{147}][\mathbf{156}][\mathbf{190}][\mathbf{358}]$ |
| $[\mathbf{690}] = [\mathbf{156}][\mathbf{469}]$<br>$[\mathbf{190}] = -[\mathbf{146}][\mathbf{159}]$ | $\overset{\mathbf{0}}{=} -\underline{[\mathbf{156}]}[\mathbf{135}][\mathbf{137}][\mathbf{145}][\mathbf{168}][\mathbf{469}]-\underline{[\mathbf{156}]}[\mathbf{136}][\mathbf{146}][\mathbf{147}][\mathbf{159}][\mathbf{358}]$ |
| $[\mathbf{469}] = [\mathbf{146}][\mathbf{234}]$<br>$[\mathbf{159}] = [\mathbf{123}][\mathbf{145}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{145}][\mathbf{146}]}(-[\mathbf{135}][\mathbf{137}][\mathbf{168}][\mathbf{234}]-[\mathbf{123}][\mathbf{136}][\mathbf{147}][\mathbf{358}])$ |
| $[\mathbf{168}] = -[\mathbf{124}][\mathbf{136}]$<br>$[\mathbf{358}] = [\mathbf{135}][\mathbf{234}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{135}][\mathbf{136}][\mathbf{234}]}([\mathbf{124}][\mathbf{137}]-[\mathbf{123}][\mathbf{147}])$ |
| | $= 0.$ |

Nondegeneracy condition: none.

**Example 14** (See also [3], Example 6.190). Two doubly perspective triangles are also triply perspective.

Free points: **1, 2, 3, 4, 5**.
Intersections: $\mathbf{6} = \mathbf{12} \cap \mathbf{34}$, $\mathbf{7} = \mathbf{24} \cap \mathbf{15}$, $\mathbf{8} = \mathbf{13} \cap \mathbf{45}$, $\mathbf{A} = \mathbf{56} \cap \mathbf{37}$.
Conclusion: **2, 8, 9** are collinear.

Proof:

| Rules | $[\mathbf{289}]$ |
|---|---|
| | $\overset{\mathbf{9}}{=} [\mathbf{268}][\mathbf{357}]-[\mathbf{258}][\mathbf{367}]$ |
| $[\mathbf{268}] = -[\mathbf{145}][\mathbf{236}]$<br>$[\mathbf{258}] = -[\mathbf{135}][\mathbf{245}]$ | $\overset{\mathbf{8}}{=} -[\mathbf{145}][\mathbf{236}][\mathbf{357}]+[\mathbf{135}][\mathbf{245}][\mathbf{367}]$ |
| $[\mathbf{357}] = [\mathbf{135}][\mathbf{245}]$<br>$[\mathbf{367}] = [\mathbf{145}][\mathbf{236}]$ | $\overset{\mathbf{7}}{=} 0.$ |

Nondegeneracy condition: none.

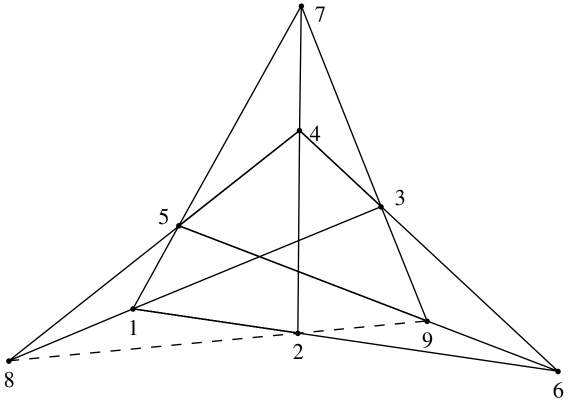


**Fig. 14.** Example 14.

**Example 15** (See also [3], Example 6.26). In a hexagon whose vertices are **1, 2, 3, 4, 5, 9**, if both **39**, **12**, **45** and **19**, **34**, **25** are concurrent, then **14**, **59**, **23** are concurrent.

Free points: $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}$.
Intersections: $\mathbf{6} = \mathbf{23} \cap \mathbf{14}$, $\mathbf{7} = \mathbf{12} \cap \mathbf{45}$, $\mathbf{8} = \mathbf{34} \cap \mathbf{25}$, $\mathbf{9} = \mathbf{37} \cap \mathbf{18}$.
Conclusion: **5, 6, 9** are collinear.

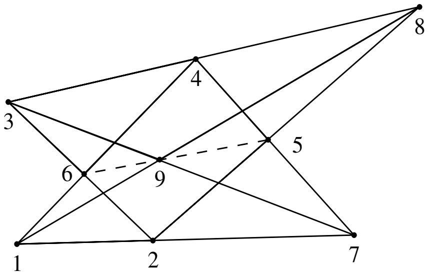


**Fig. 15.** Example 15.

Proof:

| Rules | |
|---|---|
| | $[\mathbf{569}]$ |
| | $\overset{\mathbf{9}}{=} [\mathbf{178}][\mathbf{356}] - [\mathbf{138}][\mathbf{567}]$ |
| $[\mathbf{178}] = [\mathbf{157}][\mathbf{234}]$ <br> $[\mathbf{138}] = -[\mathbf{134}][\mathbf{235}]$ | $\overset{\mathbf{8}}{=} [\mathbf{157}][\mathbf{234}][\mathbf{356}] + [\mathbf{134}][\mathbf{235}][\mathbf{567}]$ |
| $[\mathbf{157}] = -[\mathbf{125}][\mathbf{145}]$ <br> $[\mathbf{567}] = [\mathbf{125}][\mathbf{456}]$ | $\overset{\mathbf{7}}{=} -\underline{[\mathbf{125}]}[\mathbf{145}][\mathbf{234}][\mathbf{356}] + \underline{[\mathbf{125}]}[\mathbf{134}][\mathbf{235}][\mathbf{456}]$ |
| $[\mathbf{356}] = [\mathbf{134}][\mathbf{235}]$ <br> $[\mathbf{456}] = [\mathbf{145}][\mathbf{234}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: none.

**Example 16** (Permutation theorem, see also [15], Example 3). If **6**, **7**, **8**, **9** are collinear, then there exits a projectivity between $(\mathbf{8}, \mathbf{9}, \mathbf{7}, \mathbf{6})$ and $(\mathbf{6}, \mathbf{7}, \mathbf{9}, \mathbf{8})$.

Free points: **1, 2, 3, 4**.
Semifree point: **5** on **23**.
Intersections:

$$\mathbf{6} = \mathbf{12} \cap \mathbf{34}, \quad \mathbf{7} = \mathbf{13} \cap \mathbf{24}, \quad \mathbf{8} = \mathbf{15} \cap \mathbf{67}, \quad \mathbf{9} = \mathbf{45} \cap \mathbf{67}, \quad \mathbf{0} = \mathbf{23} \cap \mathbf{48}.$$

Conclusion: **1, 9, 0** are collinear.

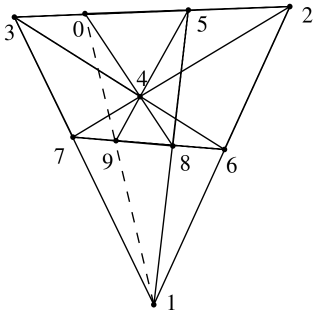


**Fig. 16.** Example 16.

Proof:

| Rules |
|---|
| $[\mathbf{189}] = [\mathbf{148}][\mathbf{567}]$<br>$[\mathbf{149}] = [\mathbf{145}][\mathbf{467}]$ |
| $[\mathbf{148}] = [\mathbf{145}][\mathbf{167}]$<br>$[\mathbf{238}] = -[\mathbf{123}][\mathbf{567}]$ |
| $[\mathbf{167}] = -[\mathbf{124}][\mathbf{136}]$<br>$[\mathbf{467}] = [\mathbf{134}][\mathbf{246}]$ |
| $[\mathbf{136}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{246}] = -[\mathbf{124}][\mathbf{234}]$ |

$$\begin{aligned}
&[\mathbf{190}] \\
\overset{\mathbf{0}}{=}\; &[\mathbf{189}][\mathbf{234}]-[\mathbf{149}][\mathbf{238}] \\
\overset{\mathbf{9}}{=}\; &[\mathbf{148}][\mathbf{234}][\mathbf{567}]-[\mathbf{145}][\mathbf{238}][\mathbf{467}] \\
\overset{\mathbf{8}}{=}\; &\underline{[\mathbf{145}][\mathbf{567}]}([\mathbf{167}][\mathbf{234}]+[\mathbf{123}][\mathbf{467}]) \\
\overset{\mathbf{7}}{=}\; &-[\mathbf{124}][\mathbf{136}][\mathbf{234}]+[\mathbf{123}][\mathbf{134}][\mathbf{246}] \\
\overset{\mathbf{6}}{=}\; &0.
\end{aligned}$$

Nondegeneracy condition: none.

**Example 17** (Harmonic points, see also [3], Example 6.236, and [15], Example 4). If **6**, **7**, **8**, **B** is a harmonic quadruple of points, then **B** is uniquely determined by **6**, **7**, **8**.

Free points: **1, 2, 3, 4, 5**.
Semifree point: **9** on **58**.

Intersections:

$$\mathbf{6} = \mathbf{12} \cap \mathbf{34},\ \mathbf{7} = \mathbf{23} \cap \mathbf{14},\ \ \mathbf{8} = \mathbf{67} \cap \mathbf{13},$$
$$\mathbf{0} = \mathbf{79} \cap \mathbf{56},\ \mathbf{A} = \mathbf{69} \cap \mathbf{57},\ \mathbf{B} = \mathbf{67} \cap \mathbf{24}.$$

Conclusion: **0, A, B** are collinear.

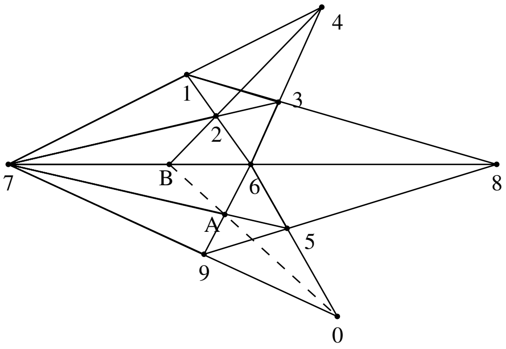


**Fig. 17.** Example 17.

Proof:

| Rules | $[\mathbf{0AB}]$ |
|---|---|
| | $\overset{\mathbf{B}}{=} [\mathbf{246}][\mathbf{70A}]-[\mathbf{247}][\mathbf{60A}]$ |
| $[\mathbf{70A}] = -[\mathbf{570}][\mathbf{679}]$<br>$[\mathbf{60A}] = \ \ [\mathbf{567}][\mathbf{690}]$ | $\overset{\mathbf{A}}{=} -[\mathbf{246}][\mathbf{570}][\mathbf{679}]-[\mathbf{247}][\mathbf{567}][\mathbf{690}]$ |
| $[\mathbf{570}] = [\mathbf{567}][\mathbf{579}]$<br>$[\mathbf{690}] = [\mathbf{569}][\mathbf{679}]$ | $\overset{\mathbf{0}}{=} \underline{[\mathbf{567}][\mathbf{679}]}(-[\mathbf{246}][\mathbf{579}]-[\mathbf{247}][\mathbf{569}])$ |
| $[\mathbf{158}][\mathbf{579}] = [\mathbf{159}][\mathbf{578}]$<br>$[\mathbf{158}][\mathbf{569}] = [\mathbf{159}][\mathbf{568}]$ | $\overset{\mathbf{9}}{=} \frac{\underline{[\mathbf{159}]}}{\underline{[\mathbf{158}]}}(-[\mathbf{246}][\mathbf{578}]-[\mathbf{247}][\mathbf{568}])$ |
| $[\mathbf{578}] = [\mathbf{137}][\mathbf{567}]$<br>$[\mathbf{568}] = [\mathbf{136}][\mathbf{567}]$ | $\overset{\mathbf{8}}{=} -\underline{[\mathbf{567}]}[\mathbf{137}][\mathbf{246}]-\underline{[\mathbf{567}]}[\mathbf{136}][\mathbf{247}]$ |
| $[\mathbf{137}] = -[\mathbf{123}][\mathbf{134}]$<br>$[\mathbf{247}] = \ \ [\mathbf{124}][\mathbf{234}]$ | $\overset{\mathbf{7}}{=} [\mathbf{123}][\mathbf{134}][\mathbf{246}]-[\mathbf{124}][\mathbf{136}][\mathbf{234}]$ |
| $[\mathbf{246}] = -[\mathbf{124}][\mathbf{234}]$<br>$[\mathbf{136}] = -[\mathbf{123}][\mathbf{134}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition: $[\mathbf{158}]$ /= 0.

**Example 18** (See also [3], Example 6.237, and [15], Example 5). If the intersections of five correponding sides of two complete quadrilaterals are on the same line $l$, then the remaining sides also meet in $l$.

Free points: **1, 2, 3, 4, 5, 6**.
Semifree point: **7** on **12**.
Intersections:

$$\mathbf{8} = \mathbf{23} \cap \mathbf{56},\ \mathbf{9} = \mathbf{13} \cap \mathbf{78},\ \mathbf{0} = \mathbf{14} \cap \mathbf{78},\quad \mathbf{A} = \mathbf{24} \cap \mathbf{78},$$
$$\mathbf{B} = \mathbf{34} \cap \mathbf{78},\ \mathbf{C} = \mathbf{57} \cap \mathbf{69},\ \mathbf{D} = \mathbf{5A} \cap \mathbf{6B}.$$

Conclusion: **0, C, D** are collinear.

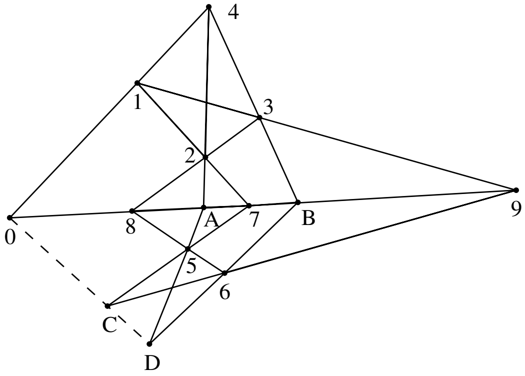


**Fig. 18.** Example 18.

Proof:

| Rules | $[\mathbf{0CD}]$ |
|---|---|
| | $\overset{\mathbf{D}}{=} [\mathbf{50C}][\mathbf{6AB}]-[\mathbf{56B}][\mathbf{0AC}]$ |
| $[\mathbf{50C}] = [\mathbf{569}][\mathbf{570}]$<br>$[\mathbf{0AC}] = -[\mathbf{579}][\mathbf{60A}]$ | $\overset{\mathbf{C}}{=} [\mathbf{569}][\mathbf{570}][\mathbf{6AB}]+[\mathbf{56B}][\mathbf{579}][\mathbf{60A}]$ |
| $[\mathbf{6AB}] = -[\mathbf{34A}][\mathbf{678}]$<br>$[\mathbf{56B}] = [\mathbf{348}][\mathbf{567}]$ | $\overset{\mathbf{B}}{=} -[\mathbf{34A}][\mathbf{569}][\mathbf{570}][\mathbf{678}]+[\mathbf{348}][\mathbf{567}][\mathbf{579}][\mathbf{60A}]$ |
| $[\mathbf{34A}] = -[\mathbf{234}][\mathbf{478}]$<br>$[\mathbf{60A}] = -[\mathbf{240}][\mathbf{678}]$ | $\overset{\mathbf{A}}{=} \underline{[\mathbf{678}]}[\mathbf{234}][\mathbf{478}][\mathbf{569}][\mathbf{570}]-\underline{[\mathbf{678}]}[\mathbf{240}][\mathbf{348}][\mathbf{567}][\mathbf{579}]$ |
| $[\mathbf{570}] = -[\mathbf{147}][\mathbf{578}]$<br>$[\mathbf{240}] = -[\mathbf{124}][\mathbf{478}]$ | $\overset{\mathbf{0}}{=} -\underline{[\mathbf{478}]}[\mathbf{147}][\mathbf{234}][\mathbf{569}][\mathbf{578}]+\underline{[\mathbf{478}]}[\mathbf{124}][\mathbf{348}][\mathbf{567}][\mathbf{579}]$ |
| $[\mathbf{569}] = [\mathbf{138}][\mathbf{567}]$<br>$[\mathbf{579}] = -[\mathbf{137}][\mathbf{578}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{567}][\mathbf{578}]}(-[\mathbf{138}][\mathbf{147}][\mathbf{234}]-[\mathbf{124}][\mathbf{137}][\mathbf{348}])$ |
| $[\mathbf{138}] = [\mathbf{123}][\mathbf{356}]$<br>$[\mathbf{348}] = -[\mathbf{234}][\mathbf{356}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{234}][\mathbf{356}]}(-[\mathbf{123}][\mathbf{147}]+[\mathbf{124}][\mathbf{137}])$ |
| | $= 0.$ |

Nondegeneracy condition: none.

**Example 19** (Pascal's theorem, see also [3], Example 6.390).

Free points: $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}$.
Semifree point: $\mathbf{6}$ on $\mathbf{12}$.
Intersections:

$$\begin{aligned} &\mathbf{7} = \mathbf{34} \cap \mathbf{15},\ \ \mathbf{8} = \mathbf{46} \cap \mathbf{59},\ \ \mathbf{2} = \mathbf{16} \cap \mathbf{39},\\ &\mathbf{A} = \mathbf{36} \cap \mathbf{15},\ \mathbf{B} = \mathbf{45} \cap \mathbf{69},\ \mathbf{0} = \mathbf{34} \cap \mathbf{19}. \end{aligned}$$

Conclusion: If $\mathbf{2}, \mathbf{7}, \mathbf{8}$ are collinear, so are $\mathbf{0}, \mathbf{A}, \mathbf{B}$.

Reformulation of the theorem:

Free points: **1, 2, 3, 4, 5**.
Semifree point: **6** on **12**.
Intersections:

$$\begin{aligned} &\mathbf{7} = \mathbf{15} \cap \mathbf{34},\ \mathbf{8} = \mathbf{27} \cap \mathbf{46},\ \ \mathbf{9} = \mathbf{58} \cap \mathbf{23},\\ &\mathbf{0} = \mathbf{19} \cap \mathbf{34},\ \mathbf{A} = \mathbf{36} \cap \mathbf{15},\ \mathbf{B} = \mathbf{69} \cap \mathbf{45}. \end{aligned}$$

Conclusion: **0, A, B** are collinear.

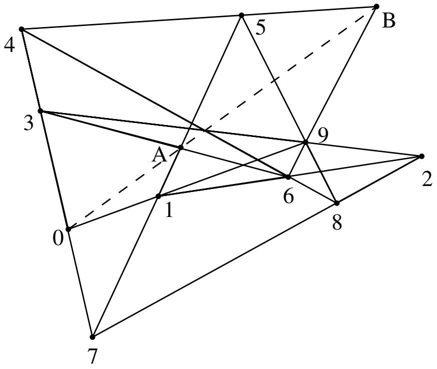


**Fig. 19.** Example 19.

Proof:

| Rules | $[\mathbf{0AB}]$ |
|---|---|
| | $\overset{\mathbf{B}}{=} [\mathbf{456}][\mathbf{90A}] - [\mathbf{459}][\mathbf{60A}]$ |
| $[\mathbf{90A}] = -[\mathbf{136}][\mathbf{590}]$<br>$[\mathbf{60A}] = -[\mathbf{156}][\mathbf{360}]$ | $\overset{\mathbf{A}}{=} -[\mathbf{136}][\mathbf{456}][\mathbf{590}] + [\mathbf{156}][\mathbf{360}][\mathbf{459}]$ |
| $[\mathbf{590}] = -[\mathbf{159}][\mathbf{349}]$<br>$[\mathbf{360}] = -[\mathbf{139}][\mathbf{346}]$ | $\overset{\mathbf{0}}{=} [\mathbf{136}][\mathbf{159}][\mathbf{349}][\mathbf{456}] - [\mathbf{139}][\mathbf{156}][\mathbf{346}][\mathbf{459}]$ |
| $[\mathbf{159}] = \ \ [\mathbf{158}][\mathbf{235}]$<br>$[\mathbf{349}] = \ \ [\mathbf{234}][\mathbf{358}]$<br>$[\mathbf{459}] = \ \ [\mathbf{235}][\mathbf{458}]$<br>$[\mathbf{139}] = -[\mathbf{123}][\mathbf{358}]$ | $\overset{\mathbf{9}}{=} \underline{[\mathbf{235}][\mathbf{358}]}([\mathbf{136}][\mathbf{158}][\mathbf{234}][\mathbf{456}] + [\mathbf{123}][\mathbf{156}][\mathbf{346}][\mathbf{458}])$ |

| | |
|---|---|
| $[\mathbf{158}] = [\mathbf{125}][\mathbf{467}]$<br>$[\mathbf{458}] = [\mathbf{247}][\mathbf{456}]$ | $\overset{\mathbf{8}}{=} \underline{[\mathbf{456}]}[\mathbf{125}][\mathbf{136}][\mathbf{234}][\mathbf{467}]+\underline{[\mathbf{456}]}[\mathbf{123}][\mathbf{156}][\mathbf{247}][\mathbf{346}]$ |
| $[\mathbf{467}] = -[\mathbf{145}][\mathbf{346}]$<br>$[\mathbf{247}] = \ \ [\mathbf{145}][\mathbf{234}]$ | $\overset{\mathbf{7}}{=} \underline{[\mathbf{145}][\mathbf{234}][\mathbf{346}]}(-[\mathbf{125}][\mathbf{136}]+[\mathbf{123}][\mathbf{156}])$ |
| | $= 0.$ |

Nondegeneracy condition in the proof of the reformulated theorem: none.

**Example 20** (See also [3], Example 6.28).

Free points: **1, 2, 3, 4, 5, 6**.
Semifree points: **7** on **12**, **8** on **13**.
Intersections:

$$\begin{array}{l}\mathbf{9} = \mathbf{14} \cap \mathbf{56},\ \ \mathbf{0} = \mathbf{15} \cap \mathbf{46},\ \ \mathbf{A} = \mathbf{37} \cap \mathbf{28}, \\ \mathbf{B} = \mathbf{34} \cap \mathbf{89},\ \mathbf{C} = \mathbf{25} \cap \mathbf{70},\ \mathbf{D} = \mathbf{58} \cap \mathbf{30}.\end{array}$$

Conclusion: **A, C, D** are collinear.

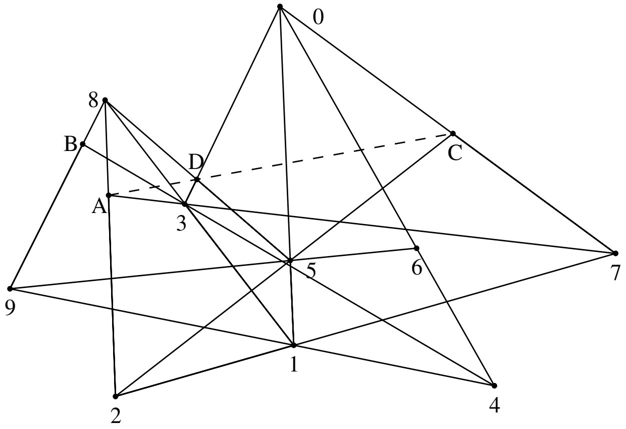


**Fig. 20.** Example 20.

Proof:

| Rules | $[\mathbf{ACD}]$ |
|---|---|
| | $\overset{\mathbf{D}}{=} [\mathbf{380}][\mathbf{5AC}]-[\mathbf{350}][\mathbf{8AC}]$ |
| $[\mathbf{5AC}] = -[\mathbf{25A}][\mathbf{570}]$<br>$[\mathbf{8AC}] = \ \ [\mathbf{270}][\mathbf{58A}]$ | $\overset{\mathbf{C}}{=} -[\mathbf{25A}][\mathbf{380}][\mathbf{570}]-[\mathbf{270}][\mathbf{350}][\mathbf{58A}]$ |
| $[\mathbf{25A}] = -[\mathbf{237}][\mathbf{258}]$<br>$[\mathbf{58A}] = \ \ [\mathbf{258}][\mathbf{378}]$ | $\overset{\mathbf{A}}{=} \underline{[\mathbf{258}]}[\mathbf{237}][\mathbf{380}][\mathbf{570}]-\underline{[\mathbf{258}]}[\mathbf{270}][\mathbf{350}][\mathbf{378}]$ |

$$\begin{array}{|l|l}
\hline
[\mathbf{380}] = -[\mathbf{146}][\mathbf{358}] & \\
[\mathbf{570}] = \quad [\mathbf{157}][\mathbf{456}] & \overset{\mathbf{0}}{=} -\underline{[\mathbf{146}][\mathbf{456}]}[\mathbf{157}][\mathbf{237}][\mathbf{358}]+\underline{[\mathbf{146}][\mathbf{456}]}[\mathbf{135}][\mathbf{257}][\mathbf{378}] \\
[\mathbf{350}] = \quad [\mathbf{135}][\mathbf{456}] & \\
[\mathbf{270}] = -[\mathbf{146}][\mathbf{257}] & \\
\hline
[\mathbf{135}][\mathbf{378}] = [\mathbf{137}][\mathbf{358}] & \overset{\mathbf{8}}{=} \underline{[\mathbf{358}]}(-[\mathbf{157}][\mathbf{237}]+[\mathbf{257}][\mathbf{137}]) \\
\hline
 & = 0.
\end{array}$$

Nondegeneracy condition: $[\mathbf{135}]$ /= 0.

**Example 21** (See also [3], Example 6.33).

Free points: **3, 4, 6, 7**.
Intersections: $\mathbf{1} = \mathbf{36} \cap \mathbf{47}$, $\mathbf{2} = \mathbf{46} \cap \mathbf{37}$, $\mathbf{8} = \mathbf{67} \cap \mathbf{34}$.
Semifree points: **9** on **18**, **0** on **12**.
Intersections: $\mathbf{A} = \mathbf{28} \cap \mathbf{90}$, $\mathbf{B} = \mathbf{36} \cap \mathbf{7A}$, $\mathbf{C} = \mathbf{67} \cap \mathbf{39}$, $\mathbf{5} = \mathbf{37} \cap \mathbf{60}$.
Conclusion: $\mathbf{B}, \mathbf{C}, \mathbf{5}$ are collinear.

Reformulation of the theorem:

Free points: **1, 2, 3, 4**.
Semifree points: **5** on **23**, **9** on **12**.
Intersections:

$$\mathbf{6} = \mathbf{13} \cap \mathbf{24},\ \mathbf{7} = \mathbf{23} \cap \mathbf{14},\ \mathbf{8} = \mathbf{34} \cap \mathbf{67},\ \mathbf{0} = \mathbf{56} \cap \mathbf{18},$$
$$\mathbf{A} = \mathbf{28} \cap \mathbf{90}, \mathbf{B} = \mathbf{7A} \cap \mathbf{13}, \mathbf{C} = \mathbf{39} \cap \mathbf{67}.$$

Conclusion: **5, B, C** are collinear.

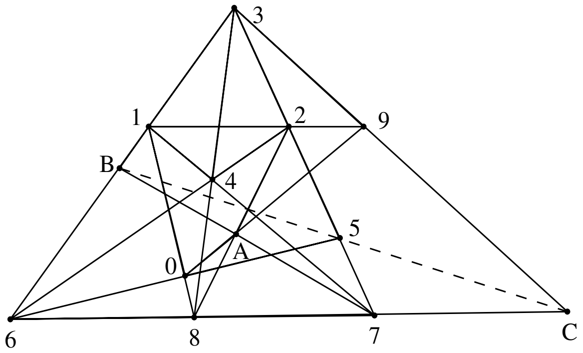


**Fig. 21.** Example 21.

Proof:

| Rules | |
|---|---|
| | $[\mathbf{5BC}]$ |
| | $\overset{\mathbf{C}}{=} [\mathbf{379}][\mathbf{56B}]-[\mathbf{369}][\mathbf{57B}]$ |
| $[\mathbf{56B}] = -[\mathbf{135}][\mathbf{67A}]$<br>$[\mathbf{57B}] = [\mathbf{137}][\mathbf{57A}]$ | $\overset{\mathbf{B}}{=} -[\mathbf{135}][\mathbf{379}][\mathbf{67A}]-[\mathbf{137}][\mathbf{369}][\mathbf{57A}]$ |
| $[\mathbf{67A}] = -[\mathbf{267}][\mathbf{890}]$<br>$[\mathbf{57A}] = [\mathbf{290}][\mathbf{578}]$ | $\overset{\mathbf{A}}{=} [\mathbf{135}][\mathbf{267}][\mathbf{379}][\mathbf{890}]-[\mathbf{137}][\mathbf{290}][\mathbf{369}][\mathbf{578}]$ |
| $[\mathbf{890}] = [\mathbf{189}][\mathbf{568}]$<br>$[\mathbf{290}] = [\mathbf{156}][\mathbf{289}]$ | $\overset{\mathbf{0}}{=} [\mathbf{135}][\mathbf{189}][\mathbf{267}][\mathbf{379}][\mathbf{568}]-[\mathbf{137}][\mathbf{156}][\mathbf{289}][\mathbf{369}][\mathbf{578}]$ |
| $[\mathbf{123}][\mathbf{379}] = [\mathbf{137}][\mathbf{239}]$<br>$[\mathbf{123}][\mathbf{189}] = [\mathbf{128}][\mathbf{139}]$<br>$[\mathbf{123}][\mathbf{369}] = -[\mathbf{139}][\mathbf{236}]$<br>$[\mathbf{123}][\mathbf{289}] = [\mathbf{128}][\mathbf{239}]$ | $\overset{\mathbf{9}}{=} \dfrac{\underline{[\mathbf{128}][\mathbf{137}][\mathbf{139}][\mathbf{239}]}}{\underline{[\mathbf{123}]^2}}([\mathbf{135}][\mathbf{267}][\mathbf{568}]+[\mathbf{156}][\mathbf{236}][\mathbf{578}])$ |
| $[\mathbf{568}] = -[\mathbf{346}][\mathbf{567}]$<br>$[\mathbf{578}] = -[\mathbf{347}][\mathbf{567}]$ | $\overset{\mathbf{8}}{=} -\underline{[\mathbf{567}]}[\mathbf{135}][\mathbf{267}][\mathbf{346}]-\underline{[\mathbf{567}]}[\mathbf{156}][\mathbf{236}][\mathbf{347}]$ |
| $[\mathbf{267}] = [\mathbf{124}][\mathbf{236}]$<br>$[\mathbf{347}] = [\mathbf{134}][\mathbf{234}]$ | $\overset{\mathbf{7}}{=} -\underline{[\mathbf{236}]}[\mathbf{124}][\mathbf{135}][\mathbf{346}]-\underline{[\mathbf{236}]}[\mathbf{134}][\mathbf{156}][\mathbf{234}]$ |
| $[\mathbf{346}] = [\mathbf{134}][\mathbf{234}]$<br>$[\mathbf{156}] = -[\mathbf{124}][\mathbf{135}]$ | $\overset{\mathbf{6}}{=} 0.$ |

Nondegeneracy condition in the proof of the reformulated theorem: $[\mathbf{123}] \;/\!= 0$.

**Example 22** (Non-realizable $10_3$-configuration, see also [15], Example 9).

Free points: **1, 2, 3, 4, 5**.
Semifree point: **6** on **12**.
Intersections: $\mathbf{7} = \mathbf{23} \cap \mathbf{14}$, $\mathbf{8} = \mathbf{15} \cap \mathbf{46}$, $\mathbf{9} = \mathbf{25} \cap \mathbf{36}$, $\mathbf{0} = \mathbf{34} \cap \mathbf{57}$.
Conclusion: **8, 9, 0** are not collinear.

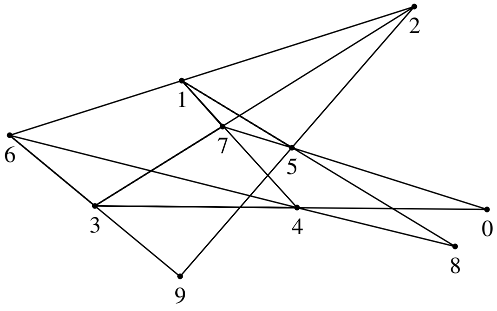


**Fig. 22.** Example 22.

Proof:

| Rules |
|---|
| $[\mathbf{489}] = [\mathbf{256}][\mathbf{348}]$<br>$[\mathbf{389}] = -[\mathbf{235}][\mathbf{368}]$ |
| $[\mathbf{348}] = [\mathbf{145}][\mathbf{346}]$<br>$[\mathbf{368}] = -[\mathbf{156}][\mathbf{346}]$ |
| $[\mathbf{357}] = [\mathbf{134}][\mathbf{235}]$<br>$[\mathbf{457}] = [\mathbf{145}][\mathbf{234}]$ |

$$\begin{aligned}
&[\mathbf{890}]\\
\overset{\mathbf{0}}{=}\ &[\mathbf{357}][\mathbf{489}]-[\mathbf{389}][\mathbf{457}]\\
\overset{\mathbf{9}}{=}\ &[\mathbf{256}][\mathbf{348}][\mathbf{357}]+[\mathbf{235}][\mathbf{368}][\mathbf{457}]\\
\overset{\mathbf{8}}{=}\ &\underline{[\mathbf{346}]}[\mathbf{145}][\mathbf{256}][\mathbf{357}]-\underline{[\mathbf{346}]}[\mathbf{156}][\mathbf{235}][\mathbf{457}]\\
\overset{\mathbf{7}}{=}\ &\underline{[\mathbf{145}][\mathbf{235}][\mathbf{346}]}([\mathbf{134}][\mathbf{256}]-[\mathbf{156}][\mathbf{234}])\\
=\ &\underline{[\mathbf{125}][\mathbf{145}][\mathbf{235}][\mathbf{346}]^2}.
\end{aligned}$$

Nondegeneracy conditions: $[\mathbf{125}], [\mathbf{145}], [\mathbf{235}], [\mathbf{346}]$ /= 0.

**Example 23** (Fano's axiom, see also [6], p. 46, and [13], p. 66). There is no complete quadrilateral whose three diagonal points are collinear.

Free points: $\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}$.
Intersections: $\mathbf{5} = \mathbf{12} \cap \mathbf{34}$, $\mathbf{6} = \mathbf{23} \cap \mathbf{14}$, $\mathbf{7} = \mathbf{13} \cap \mathbf{24}$.
Conclusion : $\mathbf{5}$, $\mathbf{6}$, $\mathbf{7}$ are not collinear.

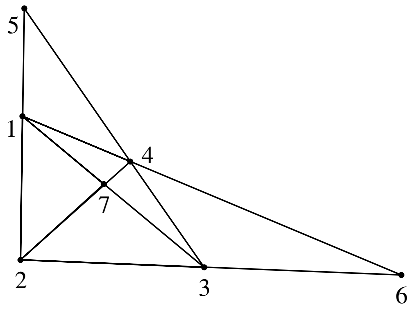


**Fig. 23.** Example 23.

Proof:

| Rules |
|---|
| $[\mathbf{356}] = [\mathbf{134}][\mathbf{235}]$<br>$[\mathbf{156}] = [\mathbf{123}][\mathbf{145}]$ |
| $[\mathbf{235}] = -[\mathbf{123}][\mathbf{234}]$<br>$[\mathbf{145}] = -[\mathbf{124}][\mathbf{134}]$ |

$$\begin{aligned}
&[\mathbf{567}]\\
\overset{\mathbf{7}}{=}\ &[\mathbf{124}][\mathbf{356}]+[\mathbf{156}][\mathbf{234}]\\
\overset{\mathbf{6}}{=}\ &[\mathbf{124}][\mathbf{134}][\mathbf{235}]+[\mathbf{123}][\mathbf{145}][\mathbf{234}]\\
\overset{\mathbf{5}}{=}\ &-2\underline{[\mathbf{123}][\mathbf{124}][\mathbf{134}][\mathbf{234}]}.
\end{aligned}$$

Nondegeneracy conditions: $[\mathbf{123}], [\mathbf{124}], [\mathbf{134}], [\mathbf{234}]$ /= 0.

## References

1. J. Bokowski and J. Richter-Gebert. On the finding of final polynomials. *Europ. J. Combinatorics* **11**, 21–34, 1990.
2. J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry.* LNM **1355**, Springer, Berlin Heidelberg, 1989.
3. S.-C. Chou, X.-S. Gao and J.-Z. Zhang. *Machine Proofs in Geometry — Automated Production of Readable Proofs for Geometric Theorems.* World Scientific, Singapore, 1994.
4. H. Crapo and J. Richter-Gebert. Automatic proving of geometric theorems. In: *Invariant Methods in Discrete and Computational Geometry* (N. White, ed.), pp. 107–139. Kluwer, Dordrecht, 1994.
5. P. Doubilet, G. C. Rota and J. Stein. On the foundations of combinatorial theory IX: Combinatorial methods in invariant theory. *Stud. Appl. Math.* **57**, 185–216, 1974.
6. L. Kadison and M. T. Kromann. *Projective Geometry and Modern Algebra.* Birkhäuser, Boston, 1996.
7. H. Li. Vectorial equations solving for mechanical geometry theorem proving. *J. Automated Reasoning* **25**, 83–121, 2000.
8. H. Li and M.-T. Cheng. Proving theorems in elementary geometry with Clifford algebraic method. *Chinese Math. Progress* **26**(4), 357–371, 1997.
9. H. Li and M.-T. Cheng. Automated ordering for automated theorem proving in elementary geometry — Degree of freedom analysis method. MM Research Preprints **18**, 84–98, 1999.
10. H. Li and Y. Wu. Outer product factorization in Clifford algebra. In: *Proc. ATCM 99* (Guangzhou, December 17–21, 1999), pp. 255–264.
11. H. Li and Y. Wu. Mechanical theorem proving in projective geometry with bracket algebra. In: *Computer Mathematics* (X.-S. Gao and D. Wang, eds.), pp. 120–129. World Scientific, Singapore, 2000.
12. B. Mourrain and N. Stolfi. Computational symbolic geometry. In: *Invariant Methods in Discrete and Computational Geometry* (N. White, ed.), pp. 107–139. Kluwer, Dordrecht, 1994.
13. C. W. O'hara, S. J. and D. R. Ward, S. J. *An Introduction to Projective Geometry.* Oxford University Press, London, 1936.
14. D. Pedoe. *An Introduction to Projective Geometry.* Pergamon Press, Oxford, 1963.
15. J. Richter-Gebert. Mechanical theorem proving in projective geometry. *Ann. Math. Artif. Intell.* **13**, 159–171, 1995.
16. B. Sturmfels. Computing final polynomials and final syzygies using Buchberger's Gröbner bases method. *Result. Math.* **15**, 351–360, 1989.
17. B. Sturmfels. *Algorithms in Invariant Theory.* Springer, Wien New York, 1993.
18. B. Sturmfels and N. White. Gröbner bases and invariant theory. *Adv. Math.* **76**, 245–259, 1989.
19. B. Sturmfels and W. Whiteley. On the synthetic factorization of homogeneous invariants. *J. Symbolic Computation* **11**, 439–454. 1991.
20. D. Wang. Elimination procedures for mechanical theorem proving in geometry. *Ann. Math. Artif. Intell.* **13**, 1–24, 1995.
21. W.-T. Wu. On the decision problem and the mechanization of theorem proving in elementary geometry, In: *Contemp. Math.* **29**, pp. 213–234. AMS, Providence, 1984.
22. A. Young. On quantative substitutionals analysis (3rd paper), *Proc. London Math. Soc.*, Ser. 2, **28**, 255–292, 1928.

# Qubit Logic, Algebra and Geometry

Timothy F. Havel

MIT (NW14-2218), 150 Albany St., Cambridge, MA 02139, USA
tfhavel@mit.edu

**Abstract.** A qubit is a two-state quantum system, in which one bit of binary information can be stored and recovered. A qubit differs from an ordinary bit in that it can exist in a complex linear combination of its two basis states, where combinations differing by a factor are identified. This projective line, in turn, can be regarded as an entity within a Clifford or *geometric* algebra, which endows it with both an algebraic structure and an interpretation as a Euclidean unit 2-sphere. Testing a qubit to see if it is in a basis state generally yields a random result, and attempts to explain this in terms of random variables parametrized by the points of the spheres of the *individual* qubits lead to contradictions. Geometric reasoning forces one to the conclusion that the parameter space is a tensor product of projective lines, and it is shown how this structure is contained in the tensor product of their geometric algebras.

## 1 Introduction

At its most fundamental level, quantum mechanics is pure geometry. In the non-relativistic case of interest here, it is in fact "just" three-dimensional Euclidean geometry. Although scientists in the field are basically aware of this fact, it is not widely known even by geometers. The following are four reasons for this state of affairs.

The first is that the representations of the group of Euclidean motions with which quantum mechanics deals are not those that have been studied in "classical" geometry; indeed, they are often over infinite dimensional complex function spaces! Nevertheless, as Felix Klein and Hermann Weyl have taught us, the elements of the carrier spaces of these representations are every bit as much a part of Euclidean geometry as lines and planes, even if they may be a great deal more difficult to visualize. It is in the *interpretation* of such mathematical structures, for example by means of suitable low-dimensional models, that geometric concepts and methods can make important contributions to quantum mechanics.

The second is the way in which these representations are combined when two or more quantum systems are merged into one, namely as the *direct product* of their constituent representations, rather than as the direct sum. Of course this is perfectly standard in mathematics today, but such composite representations were seldom considered throughout the long history of geometry leading up to Klein, and they continue to receive relatively little attention from the

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 228–245, 2001.

modern-day mathematicians working on classical geometry, probably because these representations become physically relevant (so far as is known) only at (sub)atomic scales.

The third is that the actual transformations studied in quantum mechanics are motivated by physical considerations, and described in physical terms, which lie outside the purview of many mathematicians. This is true particularly of the *interactions* between different quantum systems, which induce nonclassical correlations between them and produce an “entangled” joint state that must be described by a nonfactorizable tensor. In addition to this language barrier, the notation used in quantum mechanics, especially its use of an uninterpreted imaginary unit, tends to obscure the underlying geometry.

The fourth is the intrinsically stochastic nature of quantum *measurements*. Such measurements involve amplification of information about the state of a quantum system from the submicroscopic to the macroscopic level, a task not unlike (but considerably more drastic than) trying to pick up a live ant with a bulldozer. In general, such measurements destroy all correlations among the constituent parts of the system, which must therefore be inferred indirectly from the results of repeated measurements on ensembles of identically prepared systems. The nondeterministic, and hence “nongeometric”, nature of this process is why Einstein so disliked quantum mechanics, and it is safe to say that more than half a century later nobody really understands “how it can be like that”. This is, once again, a matter of interpreting the mathematics, which nevertheless describes reality extremely well.

The main message of this paper is that the computer-aided geometric reasoning community is missing out on a great deal of excitement, and a potentially very appreciative audience, by limiting themselves largely to geometry as inspired by the space in which classical physics is perceived to take place.[1] It will seek to illustrate this in the context of a particularly simple area of quantum mechanics which is nonetheless of great contemporary interest, not just to physicists but also to computer scientists and growing number of mathematicians, namely “quantum computing” or (more generally) *quantum information processing*. The simplicity of this area is derived from that of the type of quantum systems it usually deals with, which are merely two-dimensional systems or *qubits*.

In order to illuminate the geometric nature of qubits, the notation and language of the paper is derived from the most powerful tool for bridging the gap between the analytic and the synthetic that has been bequeathed to us by the grand masters of geometry from the 19th century, namely Grassmann’s “Calculus of Extension” as enlarged and streamlined by Hamilton, Clifford, Peano,

[1] Note that “classical” physics is usually considered to include special relativity — itself a very geometric subject with connections to non-Euclidean geometry. The book by Albert [1] contains a wonderful nontechnical introduction to the apparent paradoxes of quantum mechanics, whereas that by Peres [2] provides a more technical and detailed account of how these paradoxes arise in trying to apply classical geometric reasoning to quantum systems. For an interesting interpretation of quantum mechanics within complex projective metric geometry, the reader may wish to consult [3,4].

Cayley, Gibbs, Lipschitz, Chevalley, Cartan and Riesz together with (in more recent times) Hestenes, Sobczyk, Lounesto, and many others. Today it is most appropriately called (with apologies to E. Artin) *geometric algebra*. Among its many uses, it has become the basis of a number of symbolic algebra programs (as reviewed in [5]) which, though intended primarily for either physical or abstract mathematical investigations, also have the potential to serve as potent aids to reasoning in the classical geometries. This potential has been discussed and illustrated in e.g. [6,7,8,9,10,11,12,13], and further progress along these lines is expected to be presented in this proceedings, as well.

## 2 Quantum Information Processing

We begin with a general, but necessarily rather brief, introduction to quantum information processing, using the language and notation of conventional quantum mechanics. Further details will be provided in a more geometric form over the following sections, and longer and gentler introductions are available in e.g. [14,15,16].

Information, however it is conceived, exists only by virtue of being contained in the state of some physical system. Thus, although the information can have more than one meaning, it always has something to say about the state of the system it is stored in. Any $N$ bits of binary information, for example, can be regarded as an abstract vector in $\mathbb{Z}_2^N$, though it may actually be the beads on an abacus. Similarly, *quantum information* can be regarded as a vector $[\psi_1, \psi_2, \ldots, \psi_N] \in \mathbb{C}^N$, or as the state of a given multiparticle quantum system. In either case, it is the physics of its embodiment which determines what can be done with the information.

In order to discuss these issues more concretely, a generic but simple model system is needed. This is a *quantum computer*, which encodes binary information in an ordered array of two-state quantum systems, or "qubits". This encoding is obtained by choosing a fixed pair of orthonormal vectors in the two-dimensional state space of each qubit, which are written in Dirac notation as $|\,0\,\rangle \equiv [1,0], |\,1\,\rangle \equiv [0,1]$ and identified with binary 0, 1 respectively. The conjugate transposes are denoted by $\langle\,0\,|, \langle\,1\,|$, respectively, and hence orthonormality means $\langle\,0\,|\,0\,\rangle = \langle\,1\,|\,1\,\rangle = 1$ and $\langle\,0\,|\,1\,\rangle = \langle\,1\,|\,0\,\rangle = 0$. By the previously mentioned rules for merging multiple quantum systems into one, the state space of an $N$-qubit quantum computer is the $(2^N)$-dimensional tensor product of those of its qubits, which is spanned by the induced orthonormal basis

$$|\,\delta^1\,\rangle \otimes \cdots \otimes |\,\delta^N\,\rangle \;\equiv\; |\,\delta^1\,\rangle \cdots |\,\delta^N\,\rangle \;\equiv\; |\,\delta^1 \cdots \delta^N\,\rangle \tag{1}$$

$(\delta^1, \ldots, \delta^N \in \{0,1\})$. These $N$-qubit basis states may also be labeled by the integers $k = 0, \ldots, 2^N - 1$ via their binary expansions, and denoted by $|\,k\,\rangle$. It should be understood that, although a classical analog computer can be built whose bits could be in arbitrary *superpositions* (linear combinations) $\psi_0|\,0\,\rangle + \psi_1|\,1\,\rangle$ $(\psi_0, \psi_1 \in \mathbb{C})$, its state space would be only $2N$-dimensional over $\mathbb{C}$. Thus it is the exponential growth in the dimension of their state space which really distinguishes quantum computers from classical.

The extra dimensions are due to the existence of nonfactorizable or *entangled* states, which exhibit a number of nonintuitive properties (cf. [1,2]). These in turn are closely tied to the stochastic nature of quantum mechanical measurements. These measurements are *quantized*, so that measuring a qubit always yields one of the two possible outcomes 0 or 1, and leaves the system in the corresponding basis state $|\,0\,\rangle$ or $|\,1\,\rangle$. Which outcome will be obtained can be predicted only if the system is already in one of the two basis states, whereas measurements on superpositions yield random outcomes. In particular, the information in entangled states is contained in the *correlations* between the outcomes of measurements on different qubits. As a result of this intrinsic randomness, even though it takes an exponentially increasing amount of classical information to specify the state of an $N$-qubit quantum computer to any fixed precision, at most $N$ bits of classical information can be stored in and recovered from it (hence the name, "qubit")

Having discussed the nonintuitive features of a quantum state, we now turn to the operations which can be performed on it, again in the context of an ideal quantum computer. According to Schrödinger's equation, isolated quantum systems evolve in time by unitary transformations. The subgroup of the unitary group $\mathsf{U}(2^N)$ consisting of all permutations of the basis states $|\,k\,\rangle$ corresponds to the set of all *invertible* boolean transformations of $\mathbb{Z}_2^N$. It is known that, with at most a polynomial number of constant inputs and unused outputs, any boolean transformation can be embedded in an invertible one [17], and hence a quantum computer is computationally universal. In fact, the entire unitary group is generated by *local* operations acting on at most two qubits at a time [18], which is convenient because all elementary physical forces are two-particle interactions.

Let us illustrate this with some specific "quantum logic gates". The simplest is just the $\mathsf{NOT}$ of a qubit, which has the following matrix representation in the computational basis:

$$\underline{\boldsymbol{N}}\,|\,0\,\rangle \;=\; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \;=\; \begin{bmatrix} 0 \\ 1 \end{bmatrix} \;=\; |\,1\,\rangle \tag{2}$$

A one-bit gate without a classical counterpart is the Hadamard gate, which maps basis states to superpositions:

$$\underline{\boldsymbol{H}}\,|\,0\,\rangle \;=\; \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \;=\; \begin{bmatrix} 2^{-1/2} \\ 2^{-1/2} \end{bmatrix} \;=\; (|\,0\,\rangle + |\,1\,\rangle)/\sqrt{2} \tag{3}$$

$$\underline{\boldsymbol{H}}\,|\,1\,\rangle \;=\; \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \;=\; \begin{bmatrix} 2^{-1/2} \\ -2^{-1/2} \end{bmatrix} \;=\; (|\,0\,\rangle - |\,1\,\rangle)/\sqrt{2} \tag{4}$$

Note that both these operations are involutions, in that $\underline{\boldsymbol{N}}^2 = \underline{\boldsymbol{H}}^2 = \underline{\mathbf{1}}$.

Interesting calculations require that operations on one qubit be *conditional* on the state of one or more others. Since we cannot read a qubit in a superposition without destroying that superposition, this conditionality must be built into a unitary transformation which acts simultaneously on all the qubits. Clearly conventional logic gates like the $\mathsf{AND}$ are not possible, since these are not invertible (two inputs but only one output!). The most common example of a two-bit gate is the controlled $\mathsf{NOT}$, or $\mathsf{c\text{-}NOT}$, gate, which has the matrix representation

$$\underline{\boldsymbol{S}}^{2|1} \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} , \tag{5}$$

and so acts upon the computational basis as follows:

$$\begin{aligned} \underline{\boldsymbol{S}}^{2|1}\,|\,00\,\rangle \;=\; |\,00\,\rangle, \quad \underline{\boldsymbol{S}}^{2|1}\,|\,01\,\rangle \;=\; |\,01\,\rangle, \\ \underline{\boldsymbol{S}}^{2|1}\,|\,10\,\rangle \;=\; |\,11\,\rangle, \quad \underline{\boldsymbol{S}}^{2|1}\,|\,11\,\rangle \;=\; |\,10\,\rangle \end{aligned} \tag{6}$$

Thus the $\boldsymbol{S}^{2|1}$ flips the second (right) qubit whenever the first is 1. This single two-bit gate, together with all possible one-bit unitary operations, is known to generate the entire unitary group $\mathsf{U}(2^N)$ [18]. The number of operations required, however, can be exponential in $N$.

Much of the current interest in quantum computers is due to the spectacular discovery of a polynomial-time quantum algorithm for the integer factorization problem [19]. Since every known classical algorithm is exponential, this implies that quantum computers violate that cornerstone of computational complexity known as the *strong Church-Turing thesis*, which states that the distinction between exponential and polynomial-time algorithms does not depend on the underlying hardware. Such speed-ups rely upon the linearity of quantum evolution, so that a unitary operation on the basis states operates simultaneously on all terms of a superposition, as follows:

$$\underline{\boldsymbol{U}} \textstyle\sum_{k=0}^{2^N-1} c_k\,|\,k\,\rangle \;=\; \textstyle\sum_{k=0}^{2^N-1} c_k\,\underline{\boldsymbol{U}}\,|\,k\,\rangle \tag{7}$$

This shows that a quantum computer offers a degree of parallelism which potentially grows exponentially with the size of the problem. The catch is that the above-mentioned limitations on the amount of information available from quantum measurements prevents us from simply reading out the results (which would require exponential time and memory anyway). The trick, therefore, is to find an additional quantum operation that maps the final superposition back to a basis state which reveals the desired answer, without knowledge of the superposition. Such operations, generally some form of Fourier transformation, have been discovered for only a very few problems to date, and in particular, quantum computers are unlikely to be a panacea for $\mathsf{NP}$-complete problems [20].

Having described the operational features of quantum computers, we turn our attention to their underlying geometric structure.

## 3 The Pauli Algebra

The state space of a single qubit, being a two-dimensional vector space over $\mathbb{C}$, appears very different from physical space. A relation between the two is obtained

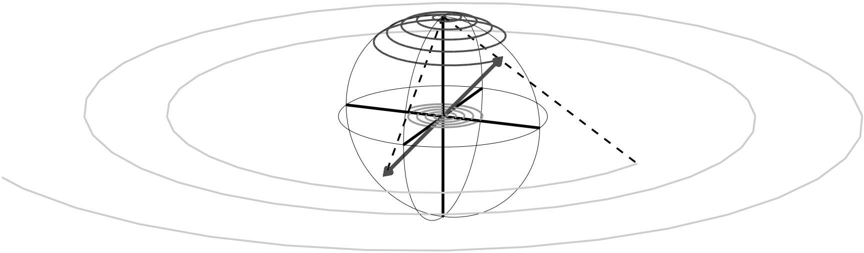


**Fig. 1.** The Bloch vector of a qubit being subjected to a continuous rotation on the Riemann (unit) sphere, starting from its reference position along the vertical $\mathsf{z}$-axis. The spirals in the complex $\mathsf{xy}$-plane are its stereographic projection (outside the sphere) together with that of its antipode (inside), as indicated by the dashed lines from the north pole. This action of $\mathsf{SO}(3)$ on the Bloch vector is induced under stereographic projection by certain Möbius transformations (or homographies) of the complex plane, which in turn are induced by the standard representation of the special unitary group $\mathsf{SU}(2)$ under taking ratios of the complex coordinates $\psi_1/\psi_0$ of qubit state vectors. This mapping is a two-to-one homomorphism of $\mathsf{SU}(2)$ onto $\mathsf{SO}(3)$, in accord with the well-known Lie algebra relationship between these groups: $\mathsf{SU}(2)$ is the universal (i.e. simply connected) covering group of $\mathsf{SO}(3)$. On the projective line given by the ratio $\psi_1/\psi_0$, the antipode corresponds to the harmonic conjugate, with respect to the projective points of the north and south poles, of the Bloch vector's projective point.

by viewing the components of a general state vector $|\,\psi\,\rangle = \psi_0|\,0\,\rangle + \psi_1|\,1\,\rangle$ as the *Cayley-Klein parameters* for a three-dimensional Euclidean rotation [22], which are given in terms of the Euler angles by

$$\psi_0 \;=\; \cos(\vartheta/2)\, e^{\boldsymbol{\iota}(\zeta-\varphi)/2} \;, \quad \psi_1 \;=\; -\,\boldsymbol{\iota}\sin(\vartheta/2)\, e^{\boldsymbol{\iota}(\zeta+\varphi)/2} \tag{8}$$

(wherein $\boldsymbol{\iota}^2 = -1$). These parameters determine an element of the two-fold universal covering group $\mathsf{SU}(2)$ of $\mathsf{SO}(3)$, namely

$$\underline{\boldsymbol{\Psi}} \;\equiv\; \begin{bmatrix} \psi_0 & -\psi_1^* \\ \psi_1 & \psi_0^* \end{bmatrix} \;\in\; \mathsf{SU}(2)\;, \tag{9}$$

and hence are determined by the rotation up to sign. This identification makes it possible to map the state of the qubit to a three-dimensional unit vector, which is obtained from a given reference vector by applying the rotation to it. Under this mapping, the states $|\,0\,\rangle$ and $|\,1\,\rangle$ coincide with the reference vector and its negative, while the physically irrelevant net phase $\exp(\boldsymbol{\iota}\zeta)$ corresponds to the angle of rotation about the reference vector.

Assuming that the reference is a unit vector along the $\mathsf{z}$-coordinate axis, the transformed vector, called the *Bloch vector*, is given in terms of an orthonormal basis $\boldsymbol{\sigma}_{\mathsf{x}}\,, \boldsymbol{\sigma}_{\mathsf{y}}\,, \boldsymbol{\sigma}_{\mathsf{z}}$ by

$$\sin(\vartheta)\cos(\varphi)\boldsymbol{\sigma}_{\mathsf{x}} + \sin(\vartheta)\sin(\varphi)\boldsymbol{\sigma}_{\mathsf{y}} + \cos(\vartheta)\boldsymbol{\sigma}_{\mathsf{z}}\;, \tag{10}$$

so its polar coordinates are Euler angles. This "classical" model of a qubit's state which corresponds to the stereographic projection of $\psi_1/\psi_0$ onto the Riemann sphere (see **Fig. 1**)[2], has a number of interesting implications. First, it associates the computational basis $|\,0\,\rangle$, $|\,1\,\rangle$ with a direction in physical space, which (in any physical realization of qubits) is determined by the measurement apparatus. Because physical rotations of the apparatus induce passive transformations of the underlying state space, the question of whether a qubit is in a superposition state or a basis state is geometrically meaningless; whether or not they are *entangled*, however, turns out to be basis independent. More importantly for the present purposes, the model enables the standard matrix formulation of quantum mechanics to be embedded in the natural algebraic structure of a metric vector space, namely its Clifford or *geometric algebra* (cf. [23,24,25,22]).

The geometric algebra of a three-dimensional Euclidean vector space, also called the *Pauli algebra* $\mathcal{G}_3$, is the associative algebra generated by a basis subject to the relations $\boldsymbol{\sigma}_\mu\boldsymbol{\sigma}_\nu + \boldsymbol{\sigma}_\nu\boldsymbol{\sigma}_\mu = 2\,\delta_{\mu\nu}$ $(\mu,\nu \in \{\mathsf{x},\mathsf{y},\mathsf{z}\})$. These relations readily imply that the Euclidean inner product is given by the symmetric part of the geometric product, $\boldsymbol{a}\cdot\boldsymbol{b} = (\boldsymbol{ab}+\boldsymbol{ba})/2$, while the antisymmetric part corresponds to Grassmann's *outer product* $\boldsymbol{a}\wedge\boldsymbol{b} = (\boldsymbol{ab}-\boldsymbol{ba})/2$. A linear basis for this eight-dimensional algebra is given by:

$$\begin{array}{rl} 1 & \text{(scalars, 0-dimensional)} \\ \boldsymbol{\sigma}_\mathsf{x},\ \boldsymbol{\sigma}_\mathsf{y},\ \boldsymbol{\sigma}_\mathsf{z} & \text{(vectors, 1-dimensional)} \\ \boldsymbol{\sigma}_\mathsf{x}\boldsymbol{\sigma}_\mathsf{y},\ \boldsymbol{\sigma}_\mathsf{z}\boldsymbol{\sigma}_\mathsf{x},\ \boldsymbol{\sigma}_\mathsf{y}\boldsymbol{\sigma}_\mathsf{z} & \text{(bivectors, 2-dimensional)} \\ \boldsymbol{\sigma}_\mathsf{x}\boldsymbol{\sigma}_\mathsf{y}\boldsymbol{\sigma}_\mathsf{z} & \text{(pseudo-scalars, 3-dimensional)} \end{array} \tag{11}$$

In particular, the unit pseudo-scalar $\boldsymbol{\iota} \equiv \boldsymbol{\sigma}_\mathsf{x}\boldsymbol{\sigma}_\mathsf{y}\boldsymbol{\sigma}_\mathsf{z}$ is easily seen to square to $-1$ and commute with everything in the algebra, thereby providing a geometric interpretation for the imaginary unit of qubits (and many other physical entities). Just as a real number can be interpreted as either a magnitude (e.g. length = 2) or as an operator (e.g. scale by 2), $\boldsymbol{\iota}$ can be interpreted as either the *oriented volume element* of space, or as an operator that maps vectors to their orthogonal bivectors. The latter shows that the outer product of two vectors $\boldsymbol{a}$, $\boldsymbol{b}$ is related to their cross product by:

$$\boldsymbol{a}\wedge\boldsymbol{b} \;=\; \boldsymbol{\iota}(\boldsymbol{a}\times\boldsymbol{b}) \tag{12}$$

In addition to its many applications to classical geometry and physics [26], the Pauli algebra provides a concise mathematical encoding of the physics of qubits (be they nuclear spins, photon polarizations, electronic states of atoms, etc.), as follows. The *even subalgebra* $\mathcal{G}_3^+$ is that generated by the products of even numbers of vectors, namely by the identity together with the basis bivectors. Its multiplicative subgroup of unit norm elements is isomorphic to $\mathsf{SU}(2)$, which in turn is isomorphic to the multiplicative group of unit quaternions. A time-

[2] For many further illustrations of these beautiful morphisms between seemingly different geometries, the reader is referred to the delightful book "Visual Complex Analysis" by T. Needham [21].

dependent rotation of a vector $\boldsymbol{v}$ is given by its conjugate $\boldsymbol{U}\boldsymbol{v}\boldsymbol{U}^{-1}$ with a one-parameter subgroup of $\mathcal{G}_3^+$, namely

$$\boldsymbol{U} \;=\; \exp(-\boldsymbol{\iota}\omega t\boldsymbol{u}/2) \;=\; \cos(\omega t/2) - \boldsymbol{\iota u}\sin(\omega t/2)\;, \tag{13}$$

where $\omega$ is the angular velocity and $\boldsymbol{u}$ a unit vector on the axis of rotation. The inverse $\boldsymbol{U}^{-1} = \exp(\boldsymbol{\iota}\omega t\boldsymbol{u}/2) = \tilde{\boldsymbol{U}}$ is obtained by *reversing* the order of the factors in the terms of an expansion of $\boldsymbol{U}$ relative to the above basis, thereby changing the sign of its bivector part. In most physical realizations of qubits, $\boldsymbol{H} \equiv \omega\boldsymbol{u}$ is the Hamiltonian "operator" for the interaction of the qubit with a constant external field which induces the rotation.

Elementary idempotents of the form $\boldsymbol{E_v} \equiv (1+\boldsymbol{v})/2$ with $\boldsymbol{v}^2 = 1$ play several important roles in the algebra and the physics. First, these idempotents describe a qubit's state as the sum of a scalar (1/2) and the vector ($\boldsymbol{v}/2$) obtained by applying a rotation $\boldsymbol{\varPsi} \in \mathcal{G}_3^+$ to the reference vector (conventionally taken to be $\boldsymbol{\sigma}_\mathsf{z}/2$),

$$\boldsymbol{E_v} \;=\; \tfrac{1}{2}(1+\boldsymbol{v}) \;=\; \boldsymbol{\varPsi}\,\tfrac{1}{2}(1+\boldsymbol{\sigma}_\mathsf{z})\,\tilde{\boldsymbol{\varPsi}} \;\equiv\; \boldsymbol{\varPsi}\,\boldsymbol{E}_+\,\tilde{\boldsymbol{\varPsi}}\;. \tag{14}$$

The reference idempotent and its complement will henceforth be written as $\boldsymbol{E}_\pm \equiv \boldsymbol{E}_{\pm\boldsymbol{\sigma}_\mathsf{z}}$, so that $\boldsymbol{E}_+ \leftrightarrow |\,0\,\rangle\langle\,0\,|$ and $\boldsymbol{E}_- \leftrightarrow |\,1\,\rangle\langle\,1\,|$ (where ↔ indicates a correspondence between geometric and coordinate-based objects). Although the scalar part may seem superfluous, it will be seen shortly that it provides a normalization factor needed to describe the statistics of ensembles of qubits.

Second, right-multiplication by the reference idempotent projects the algebra onto a left-ideal with half as many (i.e. four) degrees of freedom, on which the even subalgebra acts from the left to give a representation of $\mathbb{R}^* \oplus \mathsf{SU}(2)$. This allows the elements of the left-ideal to be interpreted as qubit state vectors, since they transform in the same way. Explicitly, if we choose a representation in which $\boldsymbol{\sigma}_\mathsf{z}$ and hence $\boldsymbol{E}_+$ is diagonal, we have:

$$\underline{\boldsymbol{\varPsi}}\,\underline{\boldsymbol{E}}_+ \;=\; \begin{bmatrix}\psi_0 & -\psi_1^* \\ \psi_1 & \psi_0^*\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix} \;=\; \begin{bmatrix}\psi_0 & 0 \\ \psi_1 & 0\end{bmatrix} \;=\; (\psi_0|\,0\,\rangle + \psi_1|\,1\,\rangle)\langle\,0\,| \tag{15}$$

It follows that $\boldsymbol{\varPsi E}_+\tilde{\boldsymbol{\varPsi}}\boldsymbol{E}_+ = \boldsymbol{\varPsi E}_+\psi_0^*$ is the inverse of our mapping from qubit state vectors to spatial vectors (up to phase and normalization).

Third, $\boldsymbol{E_v}$ is the operator for a measurement that yields 1 if the qubit is in the state $\boldsymbol{E_v}$ and 0 if it is in the orthogonal state $\boldsymbol{E}_{-\boldsymbol{v}}$. In accord with the properties of quantum measurements introduced previously, when applied to a state $\boldsymbol{E_w}$ this measurement irreversibly "collapses" it onto one of the two states $\boldsymbol{E}_{\pm\boldsymbol{v}}$, each with probability equal to

$$2\langle\,\boldsymbol{E_w}\boldsymbol{E}_{\pm\boldsymbol{v}}\,\rangle_0 \;=\; \tfrac{1}{2}(1 \pm \boldsymbol{w}\cdot\boldsymbol{v})\;, \tag{16}$$

where $\langle\,\_\,\rangle_0$ denotes the scalar part (or half the trace in the above matrix representation). Thus we see that, curiously enough, the geometric relations among Euclidean vectors representing states and operators are what determines the statistics of quantum measurements. This equivalence will now be extended to *ensembles* of qubits, wherein the qubit states' statistical frequencies represent a probability density function over the state space.

In the matrix formulation of a qubit mechanics [2], the expectation value of a measurement is given by a quadratic form in the state vector components, e.g.

$$\langle\, \psi\, | \underline{\boldsymbol{E}}_{\boldsymbol{v}} |\, \psi\, \rangle \;=\; \mathrm{tr}(\underline{\boldsymbol{E}}_{\boldsymbol{v}}\, |\, \psi\, \rangle\langle\, \psi\, |) \;\leftrightarrow\; 2\langle\, \boldsymbol{E}_{\boldsymbol{v}} \boldsymbol{E}_{\boldsymbol{w}}\, \rangle_0 \;=\; \tfrac{1}{2}(1 + \boldsymbol{v} \cdot \boldsymbol{w})\ , \tag{17}$$

which is an *affine* form in the Bloch vector $\boldsymbol{w}$. This allows us to write the average of the expectation value over a probabilistic ensemble $\{\, p_i\, ,\ \, |\, \psi_i\, \rangle\, \}$ ($p_i \geq 0$, $\sum_i p_i = 1$) of qubits as

$$\textstyle\sum_i p_i\, \langle\, \psi_i\, | \underline{\boldsymbol{E}}_{\boldsymbol{v}} |\, \psi_i\, \rangle \;=\; 2\, \langle\, \boldsymbol{E}_{\boldsymbol{v}} \sum_i p_i \boldsymbol{E}_{\boldsymbol{w}_i}\, \rangle_0 \;=\; \tfrac{1}{2}\, (1 + \boldsymbol{v} \cdot \sum_i p_i \boldsymbol{w}_i)\ . \tag{18}$$

It follows that the average of any test over the ensemble is determined by a Bloch vector $\eta \boldsymbol{w} \equiv \sum_i p_i \boldsymbol{w}_i$ of length $\eta \leq 1$, or equivalently, by the corresponding *density operator* $\boldsymbol{\rho} \equiv \frac{1}{2}(1 + \eta \boldsymbol{w})$. The "coherence" of the ensemble $\eta$ is related to the average of the density operator versus itself by $\eta^2 = 4\langle\, \boldsymbol{\rho}^2\, \rangle_0 - 1$. An ensemble is called *pure* when $\eta = 1$, meaning that $\boldsymbol{\rho} \leftrightarrow |\, \psi\, \rangle\langle\, \psi\, |$ for a single state vector $|\, \psi\, \rangle$, and *mixed* otherwise.

## 4 Multi-qubit Systems

Extending these geometric interpretations to multi-qubit systems necessitates taking the $N$-fold tensor power of $\mathcal{G}_3$.[3] The subalgebra generated by the operators for any one qubit is thus a copy of the Pauli algebra as above, while the operators for different qubits commute freely. Thus an arbitrary element of the algebra, denoted by $\mathcal{G}_3^{\otimes N}$, can be written as a linear combination of *product operators*,

$$\sum_{\mu \in \{0,\mathrm{x},\mathrm{y},\mathrm{z}\}^N} \alpha_\mu\, \boldsymbol{\sigma}^1_{\mu^1} \cdots \boldsymbol{\sigma}^N_{\mu^N}\ , \tag{19}$$

where the superscripts are qubit indices (written here arbitrarily in increasing order), and for notational convenience $\boldsymbol{\sigma}_0 \equiv 1$. The coefficients $\alpha_\mu$ are hypercomplex numbers in the center of the algebra of the form

$$\sum_{\nu \subseteq \{1,\ldots,N\}} \alpha^\nu_\mu\, \boldsymbol{\iota}^{\nu_1} \cdots \boldsymbol{\iota}^{\nu_M}\ , \tag{20}$$

where $\alpha^\nu_\mu \in \mathbb{R}$ and $\boldsymbol{\iota}^q$ is the unit pseudo-scalar of the $q$-th qubit ($q = 1, \ldots, N$).

It follows that $\mathcal{G}_3^{\otimes N}$ has dimension $2^{3N}$ as a linear space over $\mathbb{R}$. In the conventional quantum mechanics of qubits, however, one works in the algebra of $2^N \times 2^N$ matrices over $\mathbb{C}$, which has real dimension $2^{2N+1}$. These apparently superfluous degrees of freedom arise from the fact that $\mathcal{G}_3^{\otimes N}$ contains a different

[3] A physical justification for this may be found in the fact that this algebra is obtained from the geometric algebra of a direct sum of $N$ copies of Minkowski space-time by choosing an inertial frame in each [22,24].

unit pseudo-scalar $\boldsymbol{\iota}^q$ for each qubit $q = 1, \ldots, N$. They can be projected out by working in the principal ideal generated by the so-called *correlator* idempotent:

$$\boldsymbol{C} \;\equiv\; \tfrac{1}{2}(1 - \boldsymbol{\iota}^1\boldsymbol{\iota}^2)\tfrac{1}{2}(1 - \boldsymbol{\iota}^1\boldsymbol{\iota}^3)\cdots\tfrac{1}{2}(1 - \boldsymbol{\iota}^1\boldsymbol{\iota}^N) \tag{21}$$

It is easily seen that $\boldsymbol{\iota}^p\boldsymbol{\iota}^q\boldsymbol{C} = -\boldsymbol{C}$ for all $p, q \in \{1, \ldots, N\}$, which enables us to simply drop the qubit labels on the $\boldsymbol{\iota}$'s and use a single imaginary unit as in conventional quantum mechanics. Also, since $\boldsymbol{C}$ commutes with the entire algebra, it too can be dropped for most purposes, as will be done from here on.

The resulting algebra $\mathcal{G}_3^{\otimes N}(\boldsymbol{C})$ is isomorphic to the algebra of all $2^N \times 2^N$ matrices over $\mathbb{C}$, but the entities within it no longer depend on an arbitrary choice of coordinate frames. As with a single qubit, there are two ways to view the states of an $N$-qubit system [22]. The first is as a left-ideal in the product of the even subalgebras $(\mathcal{G}_3^+)^{\otimes N}(\boldsymbol{C})$, namely $\boldsymbol{\Psi}\boldsymbol{E}_+$ where

$$\boldsymbol{\Psi} \in (\mathcal{G}_3^+)^{\otimes N} \;, \quad \|\boldsymbol{\Psi}\| \;=\; 1 \;, \quad \boldsymbol{E}_+ \;\equiv\; \boldsymbol{E}_+^1 \cdots \boldsymbol{E}_+^N \;, \quad \boldsymbol{E}_+^q \;\equiv\; \tfrac{1}{2}(1 + \boldsymbol{\sigma}_{\mathsf{z}}^q) \tag{22}$$

($q \in \{1, \ldots, N\}$). The second is as the corresponding density operator:

$$\boldsymbol{\psi} \;\equiv\; \boldsymbol{\Psi}\boldsymbol{E}_+\tilde{\boldsymbol{\Psi}} \tag{23}$$

The density operator formalism has the advantages that it is closer in most respects to conventional quantum mechanics, that it explicitly exhibits the duality between states and their transformations (namely multiplication by $\boldsymbol{\iota}$), and that it can be used to describe statistical ensembles as well as subsystems of entangled quantum systems (see below). For these reasons we will concentrate on the density operator formulation in the following.

Let us now consider how to describe logical operations in the algebra, and their geometric interpretations (cf. [27]). The simplest such gate is the $\mathsf{NOT}$ of a single qubit $\boldsymbol{N} \equiv -\boldsymbol{\iota}\boldsymbol{\sigma}_{\mathsf{x}}$, which is just a rotation by $\pi$ about $\mathsf{x}$. This acts on the initial density operator $\boldsymbol{E}_+ \leftrightarrow |\,0\,\rangle\langle\,0\,|$ as:

$$-\boldsymbol{\iota}\boldsymbol{\sigma}_{\mathsf{x}}\boldsymbol{E}_+\boldsymbol{\iota}\boldsymbol{\sigma}_{\mathsf{x}} \;=\; \tfrac{1}{2}(1 + \boldsymbol{\sigma}_{\mathsf{x}}\boldsymbol{\sigma}_{\mathsf{z}}\boldsymbol{\sigma}_{\mathsf{x}}) \;=\; \tfrac{1}{2}(1 - (\boldsymbol{\sigma}_{\mathsf{x}})^2\boldsymbol{\sigma}_{\mathsf{z}}) \;=\; \boldsymbol{E}_- \;\leftrightarrow\; |\,1\,\rangle\langle\,1\,| \tag{24}$$

That is to say, $-\boldsymbol{\iota}\boldsymbol{\sigma}_{\mathsf{x}} = \exp(-\boldsymbol{\iota}\boldsymbol{\sigma}_{\mathsf{x}}\pi/2)$ both generates and is itself the desired logic gate. Similarly, the Hadamard gate is a rotation by $\pi$ about the axis $(\boldsymbol{\sigma}_{\mathsf{x}} + \boldsymbol{\sigma}_{\mathsf{z}})/\sqrt{2}$:

$$\boldsymbol{H} \;=\; \exp\left(-\boldsymbol{\iota}\pi(\boldsymbol{\sigma}_{\mathsf{x}} + \boldsymbol{\sigma}_{\mathsf{z}})/\sqrt{8}\right) \;=\; -\,\boldsymbol{\iota}(\boldsymbol{\sigma}_{\mathsf{x}} + \boldsymbol{\sigma}_{\mathsf{z}})/\sqrt{2} \tag{25}$$

We leave it as an exercise to show that this acts on the basis states $\boldsymbol{E}_\pm$ as previously described using matrices.

Finally, the c-NOT gate is given in coordinate-free form as[4]:

$$\boldsymbol{S}^{2|1} \;\equiv\; \boldsymbol{E}_{+}^{1} \,+\, \boldsymbol{E}_{-}^{1}\,\boldsymbol{\iota\sigma}_{\mathsf{x}}^{2} \tag{26}$$

This can also be written in exponential form as $\boldsymbol{S}^{2|1} = \exp(-\boldsymbol{\iota E}_{-}^{1}\boldsymbol{\sigma}_{\mathsf{x}}^{2}\pi/2)$. It is easily shown that it acts on the basis states as

$$\boldsymbol{S}^{2|1}\boldsymbol{E}_{\epsilon^1}^{1}\boldsymbol{E}_{\epsilon^2}^{2}\,\tilde{\boldsymbol{S}}^{2|1} \;=\; \boldsymbol{E}_{\epsilon^1}^{1}\boldsymbol{E}_{\epsilon^1\epsilon^2}^{2} \qquad (\epsilon^1, \epsilon^2 \in \{\pm 1\})\ , \tag{27}$$

so it takes the NOT of the second qubit whenever the first is $\boldsymbol{E}_{-}^{1}$. The c-NOT gate thus illustrates yet another use for the elementary idempotents: they describe the conditionality of operations. Alternatively, from a geometric standpoint, a c-NOT gate is a rotation by $\pi$ in the left-ideal defined by $\boldsymbol{E}_{-}^{1}$ only.

The following are some physically significant properties of multi-qubit density operators:

- A density operator $\boldsymbol{\rho}$ is *pure* if it is idempotent, meaning $(\boldsymbol{\rho})^2 = \boldsymbol{\rho}$, and *mixed* otherwise.
- It is *factorizable* if $\boldsymbol{\rho} = \boldsymbol{\rho}^1 \cdots \boldsymbol{\rho}^N$, where each $\boldsymbol{\rho}^q$ $(q = 1, \ldots, N)$ can be expressed in terms of operators acting on just the $q$-th qubit, and *correlated* if it is not factorizable.
- It is *separable* if there exists an ensemble for it whose individual states are factorizable (unentangled), meaning

$$\begin{aligned} \boldsymbol{\rho} \;&=\; \textstyle\sum_k p_k \,|\, \psi_k^1 \ldots \psi_k^N \,\rangle\langle\, \psi_k^1 \ldots \psi_k^N \,| \\ &=\; \textstyle\sum_k p_k \,|\, \psi_k^1 \,\rangle\langle\, \psi_k^1 \,| \cdots |\, \psi_k^N \,\rangle\langle\, \psi_k^N \,|\ . \end{aligned} \tag{28}$$

Note that a nonseparable density operator (or ensemble) is necessarily correlated, and that the converse holds if $\boldsymbol{\rho}$ is pure. It is in general difficult to determine if a given density operator is factorizable or not, and even more difficult to determine if it is separable. These problems have been completely solved only in the case of two qubits, thanks in large part to the *Schmidt decomposition* of general bipartite quantum systems. The Schmidt decomposition of a two-qubit pure state is

$$|\,\chi\,\rangle \;=\; \varsigma_1 |\,\psi_1^1\,\rangle|\,\phi_1^2\,\rangle + \varsigma_2 |\,\psi_2^1\,\rangle|\,\phi_2^2\,\rangle\ , \tag{29}$$

and is easily derived from the singular value decomposition of the $2 \times 2$ matrix obtained from the four entries of the vector $|\,\chi\,\rangle$. A pure state is unentangled if and only if one of the two real coefficients $\varsigma_1^2 + \varsigma_2^2 = 1$ vanishes.

[4] The astute reader will have observed that the phase of the states transformed by this operator differs from that of the matrix Eq. (5); since this has no effect upon the basis states, the c-NOT given here is an equally valid extension of the classical gate to superpositions.

The problem of interpreting multi-qubit density operators geometrically is perhaps even less well-understood, so we restrict ourselves to the two qubit case, and consider the product operator expansion of a general density operator:

$$\begin{aligned}\boldsymbol{\rho} \;=\; & \tfrac{1}{4} + \textstyle\sum_{\mu\in\{\mathsf{x,y,z}\}} \alpha_\mu^1 \boldsymbol{\sigma}_\mu^1 + \sum_{\nu\in\{\mathsf{x,y,z}\}} \alpha_\nu^2 \boldsymbol{\sigma}_\nu^2 \\ & + \textstyle\sum_{\mu\in\{\mathsf{x,y,z}\}} \sum_{\nu\in\{\mathsf{x,y,z}\}} \beta_{\mu\nu} \boldsymbol{\sigma}_\mu^1 \boldsymbol{\sigma}_\nu^2\end{aligned} \tag{30}$$

The orthogonality of the product operators implies that coefficients in this expansion *are* the ensemble average expectation values of the corresponding product operator. The first term, of course, transforms as a scalar, while each of the summations over the $\alpha$'s transform as vectors. The $\beta$'s are the correlations among the components of these vectors. The summation over the $\beta$'s transforms as a rank 2 tensor, which decomposes into three orbits under the simultaneous action of the rotation group on both qubits together:

a scalar, $\frac{1}{3}(\beta_{\mathsf{xx}} + \beta_{\mathsf{yy}} + \beta_{\mathsf{zz}})(\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{x}^2 + \boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{y}^2 + \boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{z}^2)$ ;

a vector, $\frac{1}{2}(\beta_{\mathsf{yx}} - \beta_{\mathsf{xy}})(\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{y}^2 - \boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{x}^2) + \frac{1}{2}(\beta_{\mathsf{xz}} - \beta_{\mathsf{zx}})(\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{z}^2 - \boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{x}^2)$
$+ \frac{1}{2}(\beta_{\mathsf{zy}} - \beta_{\mathsf{yz}})(\boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{z}^2 - \boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{y}^2)$

(since this does not change sign under inversion in the origin, it would be more accurate to say it transforms as a *bi*vector!);

and a symmetric traceless rank 2 tensor,

$$\begin{aligned}&(\tfrac{2}{3}\beta_{\mathsf{xx}} - \tfrac{1}{3}\beta_{\mathsf{yy}} - \tfrac{1}{3}\beta_{\mathsf{zz}})\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{x}^2 + (-\tfrac{1}{3}\beta_{\mathsf{xx}} + \tfrac{2}{3}\beta_{\mathsf{yy}} - \tfrac{1}{3}\beta_{\mathsf{zz}})\boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{y}^2 \\ &+ (-\tfrac{1}{3}\beta_{\mathsf{xx}} - \tfrac{1}{3}\beta_{\mathsf{yy}} + \tfrac{2}{3}\beta_{\mathsf{zz}})\boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{z}^2 + \tfrac{1}{2}(\beta_{\mathsf{xy}} + \beta_{\mathsf{yx}})(\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{y}^2 + \boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{x}^2) \\ &+ \tfrac{1}{2}(\beta_{\mathsf{xz}} + \beta_{\mathsf{zx}})(\boldsymbol{\sigma}_\mathsf{x}^1\boldsymbol{\sigma}_\mathsf{z}^2 + \boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{x}^2) + \tfrac{1}{2}(\beta_{\mathsf{yz}} + \beta_{\mathsf{zy}})(\boldsymbol{\sigma}_\mathsf{y}^1\boldsymbol{\sigma}_\mathsf{z}^2 + \boldsymbol{\sigma}_\mathsf{z}^1\boldsymbol{\sigma}_\mathsf{y}^2).\end{aligned}$$

The vanishing of the trace of this tensor implies that the corresponding quadratic form is a second-order solid harmonic. A basis for such an irreducible representation of the rotation group acting identically on all qubits together constitutes a *system of covariants* for the rotation group; an *invariant* is a system consisting of a single covariant.

The group of interest in qubit geometry is actually $\mathsf{SU}(2^N) \supset (\mathsf{SU}(2))^{\otimes N}$, the $(2^N)$-fold cover of $(\mathsf{SO}(3))^{\otimes N}$. This is because entanglement can only be created by rotating two subsystems with respect to one another in the larger space that is allowed by entanglement, whereas *local* operations in $(\mathsf{SU}(2))^{\otimes N}$ are generated by the rotations of single qubits. In any physical realization of qubits, local operations correspond to interactions of the qubits with external fields, whereas entangling operations involve interactions *between* qubits. The entangling operations convert the representations of $(\mathsf{SO}(3))^{\otimes N}$ on density operators into an infinite family of similar representations, which can in principle be classified by the values of the invariants. It is known, for example, that the invariants of a two-qubit density operator are generated algebraically by only ten fundamental invariants, and although such a minimal system has yet to be exhibited, 21 linearly independent invariants have been found using computer algebra methods [28]. Also in the case of two qubits, a geometric interpretation of the Schmidt decomposition has been given [22].

Up to this point, we have assumed we are dealing with a quantum computer that is perfectly isolated from its environment, so that a pure ensemble evolves *coherently* into another pure ensemble. The main reason why a quantum computer of any substantial complexity has yet to be built is that it is very difficult to achieve a sufficient degree of isolation in the laboratory, while still being able to interact with the system as needed to perform logical operations on its state. The interactions between a system and its environment generally result in their mutual entanglement, and since the environment is (by definition) unobservable, this in turn results in a loss of accessible information on the state of the system, which is otherwise known as *decoherence.* The system may then be described conceptually by an ensemble of quantum states which reproduces the statistics of measurements on it, or more compactly, by the density operator of such an ensemble.

To illustrate this, suppose that the "environment" itself consists of a single qubit, which interacts with a second "system" qubit so as to produce one of the so-called *Bell states*:

$$\begin{aligned}\underline{\boldsymbol{S}}^{2|1} e^{\boldsymbol{\iota}\underline{\boldsymbol{\sigma}}_{\mathsf{x}}^{1}\pi/4} |\,00\,\rangle \;&=\; \underline{\boldsymbol{S}}^{2|1}(|\,00\,\rangle - |\,10\,\rangle)/\sqrt{2} \\ &=\; (|\,00\,\rangle - |\,11\,\rangle)/\sqrt{2} \;\equiv\; |\,\phi_-\,\rangle\ .\end{aligned} \tag{31}$$

The corresponding density operator is

$$|\,\phi_-\,\rangle\langle\,\phi_-\,| \;\leftrightarrow\; \boldsymbol{\rho} \;\equiv\; \tfrac{1}{4}\left(1 - \boldsymbol{\sigma}_{\mathsf{x}}^{1}\boldsymbol{\sigma}_{\mathsf{x}}^{2} + \boldsymbol{\sigma}_{\mathsf{y}}^{1}\boldsymbol{\sigma}_{\mathsf{y}}^{2} + \boldsymbol{\sigma}_{\mathsf{z}}^{1}\boldsymbol{\sigma}_{\mathsf{z}}^{2}\right)\ , \tag{32}$$

which exhibits the correlations between the qubits explicitly. None of the terms of this expression can be observed without performing simultaneous measurements on both qubits save for the identity 1, and hence the density operator of the system qubit #2 alone is simply

$$\boldsymbol{\rho}^{2} \;=\; \tfrac{1}{2} \;\leftrightarrow\; \tfrac{1}{2}(|\,0\,\rangle\langle\,0\,| + |\,1\,\rangle\langle\,1\,|)\ . \tag{33}$$

This in turn corresponds to a completely mixed ensemble wherein half the qubits are in each of the states $|\,0\,\rangle$ and $|\,1\,\rangle$, meaning that all information about the state of the system has been lost to the environment. The corresponding mathematical operation is called the *partial trace* over the first qubit, which means dropping all terms from the product operator expansion of $\boldsymbol{\rho}$ depending on that qubit, and renormalizing by multiplying the remainder by two. It should be noted, however, that even if the environment qubit is also measured, the state $|\,\phi_-\,\rangle$ cannot be distinguished from its complementary Bell state $|\,\phi_+\,\rangle \equiv (|\,00\,\rangle + |\,11\,\rangle)/\sqrt{2}$. In other words, one of the two bits of information contained in this system is fundamentally hidden by its entanglement, and can be recovered only by disentangling them.

## 5 An Entanglement Paradox

On adding a third qubit to the Bell state $|\,\phi_-\,\rangle = |\,\phi_-^{12}\,\rangle$ above and performing another c-NOT on it, we get:

$$\begin{aligned} \underline{\boldsymbol{S}}^{3|1}|\,\phi_-^{12}\,\rangle|\,0\,\rangle &= \underline{\boldsymbol{S}}^{3|1}(|\,000\,\rangle - |\,110\,\rangle)/\sqrt{2} \\ &= (|\,000\,\rangle - |\,111\,\rangle)/\sqrt{2} \;\equiv\; |\,\mathrm{GHZ}\,\rangle \end{aligned} \tag{34}$$

This is the highly entangled *Greenberger-Horne-Zeilinger state* [29], the density operator of which is $|\,\mathrm{GHZ}\,\rangle\langle\,\mathrm{GHZ}\,| \leftrightarrow \boldsymbol{\rho}_{\mathrm{GHZ}} =$

$$\tfrac{1}{8}\,(1 + \boldsymbol{\sigma}_{\mathsf{z}}^1\boldsymbol{\sigma}_{\mathsf{z}}^2 + \boldsymbol{\sigma}_{\mathsf{z}}^1\boldsymbol{\sigma}_{\mathsf{z}}^3 + \boldsymbol{\sigma}_{\mathsf{z}}^2\boldsymbol{\sigma}_{\mathsf{z}}^3 - \boldsymbol{\sigma}_{\mathsf{x}}^1\boldsymbol{\sigma}_{\mathsf{x}}^2\boldsymbol{\sigma}_{\mathsf{x}}^3 + \boldsymbol{\sigma}_{\mathsf{y}}^1\boldsymbol{\sigma}_{\mathsf{y}}^2\boldsymbol{\sigma}_{\mathsf{x}}^3 + \boldsymbol{\sigma}_{\mathsf{y}}^1\boldsymbol{\sigma}_{\mathsf{x}}^2\boldsymbol{\sigma}_{\mathsf{y}}^3 + \boldsymbol{\sigma}_{\mathsf{x}}^1\boldsymbol{\sigma}_{\mathsf{y}}^2\boldsymbol{\sigma}_{\mathsf{y}}^3)\;. \tag{35}$$

It will now be shown that the statistics of measurements performed on an ensemble of three-qubit systems, all prepared in the GHZ state as described above, cannot be reproduced by any probabilistic ensemble of systems wherein the outcomes of the measurements are determined by the prior states of the *individual* qubits [30]. This shows, in turn, that these statistics must be determined by parameters which are pertinent to more than one qubit. If one accepts that the Bloch vector fully characterizes the statistics of measurements on single qubits and seeks to maintain the linearity of quantum mechanics, then the most natural such parameter space is the full tensor product of the Bloch vectors of the three qubits. Empirically, it has been found that these parameters are also sufficient to fully characterize the statistics.

First, let us consider the results of measuring the individual qubits along $\mathsf{z}$, namely $\boldsymbol{\sigma}_{\mathsf{z}}^1, \boldsymbol{\sigma}_{\mathsf{z}}^2, \boldsymbol{\sigma}_{\mathsf{z}}^3$. These form what is called a *complete system of commuting observables*, meaning that their simultaneous eigenstates are uniquely determined by the outcomes of measuring them (i.e. by their eigenvalues $\pm 1$). Hence knowledge of these outcomes is sufficient to predict the result of measuring any other commuting observable. Since taking the partial trace over any two of the qubits in $\boldsymbol{\rho}_{\mathrm{GHZ}}$ yields the density operator $1/2$, the outcomes of these measurements are all completely random (i.e. have a 50% chance of being either $\pm 1$). Now consider the three product observables $\boldsymbol{\sigma}_{\mathsf{z}}^{q_1}\boldsymbol{\sigma}_{\mathsf{z}}^{q_2}$ ($q_1 \neq q_2 \in \{1,2,3\}$), the outcomes of which are 1 when both Bloch vectors come out parallel along $\mathsf{z}$, and $-1$ if they come out antiparallel. Since the terms $\boldsymbol{\sigma}_{\mathsf{z}}^{q_1}\boldsymbol{\sigma}_{\mathsf{z}}^{q_2}$ all occur in the above expansion of $\boldsymbol{\rho}_{\mathrm{GHZ}}$ with a coefficient of $1/8$, the expectation values are $2^3\,\langle\,\boldsymbol{\rho}_{\mathrm{GHZ}}\,\boldsymbol{\sigma}_{\mathsf{z}}^{q_1}\boldsymbol{\sigma}_{\mathsf{z}}^{q_2}\,\rangle_0 = 1$ for all these observables, which (since their outcomes must be $\pm 1$) shows that their outcomes on any given system are 1 with probability one. Thus any two qubits must always yield the same value when measured, and this further implies that the outcomes are also the same when all three qubits are measured. It follows that the outcomes of measurements of the qubits in the $\boldsymbol{\sigma}_{\mathsf{z}}$ computational basis are compatible with a “classical” ensemble, which consists of an equal mixture of systems in each of which all three qubits are either $|\,0\,\rangle$ or $|\,1\,\rangle$. This might lead one to believe that the state $|\,\mathrm{GHZ}\,\rangle$ is unentangled, but nothing could be further from the truth!

Now let us consider the results of measurements along the $\mathsf{x}$ and $\mathsf{y}$ axes, noting that any three of the observables $\boldsymbol{\sigma}_{\mathsf{x}}^{q}$ or $\boldsymbol{\sigma}_{\mathsf{y}}^{q}$ with distinct qubit indices $q = 1, 2, 3$ form a complete system of commuting observables. Once again, the result of measuring any one of these observables is completely random, but now the result of measuring any two of them is likewise completely random (i.e. all four possibilities occur with equal frequency). To show that certain combinations of all three outcomes are nonrandom, we observe that the three product operators

$$\boldsymbol{A} \;\equiv\; \boldsymbol{\sigma}_{\mathsf{x}}^{1}\boldsymbol{\sigma}_{\mathsf{y}}^{2}\boldsymbol{\sigma}_{\mathsf{y}}^{3}\ , \quad \boldsymbol{B} \;\equiv\; \boldsymbol{\sigma}_{\mathsf{y}}^{1}\boldsymbol{\sigma}_{\mathsf{x}}^{2}\boldsymbol{\sigma}_{\mathsf{y}}^{3}\ , \quad \boldsymbol{C} \;\equiv\; \boldsymbol{\sigma}_{\mathsf{y}}^{1}\boldsymbol{\sigma}_{\mathsf{y}}^{2}\boldsymbol{\sigma}_{\mathsf{x}}^{3} \tag{36}$$

form a complete system of commuting observables, so that the fourth commuting observable

$$\boldsymbol{D} \;\equiv\; \boldsymbol{\sigma}_{\mathsf{x}}^{1}\boldsymbol{\sigma}_{\mathsf{x}}^{2}\boldsymbol{\sigma}_{\mathsf{x}}^{3} \tag{37}$$

can be predicted when the outcomes of the first three are known.

Now, it follows from their coefficients in the above expansion of $\boldsymbol{\rho}_{\mathrm{GHZ}}$ that the expectation values of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are 1, while that of $\boldsymbol{D}$ is $-1$. Therefore, if we measure three commuting observables of the form $\boldsymbol{\sigma}_{\mathsf{x}}^{q_1}$, $\boldsymbol{\sigma}_{\mathsf{y}}^{q_2}$, $\boldsymbol{\sigma}_{\mathsf{y}}^{q_3}$ $(q_1 \not= q_2 \not= q_3 \not= q_1)$, we can be sure that either all three outcomes are 1, or else exactly two are $-1$. If $M_{\mathsf{x}}^{1}, M_{\mathsf{y}}^{1}, \ldots, M_{\mathsf{y}}^{3}$ are the predetermined outcomes of the individual measurements on some arbitrary system in the GHZ state, the expectation values of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ show that

$$M_{\mathsf{x}}^{1}M_{\mathsf{y}}^{2}M_{\mathsf{y}}^{3} \;=\; M_{\mathsf{y}}^{1}M_{\mathsf{x}}^{2}M_{\mathsf{y}}^{3} \;=\; M_{\mathsf{y}}^{1}M_{\mathsf{y}}^{2}M_{\mathsf{x}}^{3} \;=\; 1\ . \tag{38}$$

Thus the product of all three triples of outcomes is likewise

$$1 \;=\; (M_{\mathsf{x}}^{1}M_{\mathsf{y}}^{2}M_{\mathsf{y}}^{3})(M_{\mathsf{y}}^{1}M_{\mathsf{x}}^{2}M_{\mathsf{y}}^{3})(M_{\mathsf{y}}^{1}M_{\mathsf{y}}^{2}M_{\mathsf{x}}^{3}) \;=\; M_{\mathsf{x}}^{1}M_{\mathsf{x}}^{2}M_{\mathsf{x}}^{3}\ , \tag{39}$$

since $(M_{\mathsf{y}}^{q})^{2} = (\pm 1)^{2} = 1$ for $q = 1, 2, 3$. But this contradicts the fact that the expectation value of $\boldsymbol{D}$ is $-1$, since that implies the product of the three $M_{\mathsf{x}}^{q}$ should be $-1$ with probability one. For those who would like a more detailed proof, the following table shows that all combinations of signs for the $\mathsf{x}$ (rows) and $\mathsf{y}$ (columns) outcomes are *excluded* by the expectation values of at least one of the observables $\boldsymbol{A}$ through $\boldsymbol{D}$.

At this point, one might begin to doubt that quantum mechanics is a self-consistent theory. The thing that saves it is the Heisenberg uncertainty principle, which forbids us from making simultaneous measurements of noncommuting operators such as $\boldsymbol{\sigma}_{\mathsf{x}}^{q}$ and $\boldsymbol{\sigma}_{\mathsf{y}}^{q}$. In the above we start from an ensemble of triples of qubits, each of which is in the quantum state $|\,\mathrm{GHZ}\,\rangle$, choose one of the four triples of measurements given in Eqs. (36), (37) at random, and perform those measurements on the qubits one-at-a-time (as must be done if they are widely separated in space). Although each qubit measurement produces a random outcome, they are perfectly correlated in the sense that the products of the three outcomes for $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are always 1 while that for $\boldsymbol{D}$ is always $-1$. Because of the Heisenberg uncertainty principle, we are never able to actually measure more than one of these four triples on any single system, but since the correlations for

Combinations of Outcomes of x & y Measurements on Three Qubits in a GHZ State Excluded by the Expectation Values of the Operators ***A***, ***B***, ***C*** & ***D***

| y 3 | + | + | + | + | − | − | − | − |
|---|---|---|---|---|---|---|---|---|
| x 2 | + | + | − | − | + | + | − | − |
| 3 2 1 | + | − | + | − | + | − | + | − |
| + + + | D | BCD | ACD | ABD | ABD | ACD | BCD | D |
| + + − | A | ABC | C | B | B | C | ABC | A |
| + − + | B | C | ABC | A | A | ABC | C | B |
| + − − | ABD | ACD | BCD | D | D | BCD | ACD | ABD |
| − + + | C | B | A | ABC | ABC | A | B | C |
| − + − | ACD | ABD | D | BCD | BCD | D | ABD | ACD |
| − − + | BCD | D | ABD | ACD | ACD | ABD | D | BCD |
| − − − | ABC | A | B | C | C | B | A | ABC |

any given triple are always the same, no matter which member of the ensemble we take, we assume (quite reasonably) that it does not matter. In other words, we assume that if we *had* measured ***D*** rather than ***A*** (say) in any given case, we would have seen the same correlations as we always had before (of course we could measure the ***A*** triple, and then the ***D***, but then the outcome of ***D*** would show no correlations, according to Heisenberg). This is known as *contrafactual reasoning*. The amazing thing is that this "cosmic censorship" corresponds to the orthogonality of two vectors $\boldsymbol{\sigma}_{\mathsf{x}}^{q}$ and $\boldsymbol{\sigma}_{\mathsf{y}}^{q}$ in the space of a single qubit!

This paradox, which has been tested experimentally [31,32], indicates that qubit geometry must be based on a tensor product space, rather than a direct sum of the underlying state spaces as in the geometry of classical physics, because only a tensor product space is big enough to independently specify all the correlations. The study of such spaces provides geometers with fertile new territory for their methods, which despite considerable study among physicists has not yet been explored with anything approaching the thoroughness of classical geometry. Geometric algebra is applicable in both domains, and computational methods for geometric reasoning based on geometric algebra (see Introduction) should be extended to the study of multi-qubit geometry, with the goal of gaining deeper insight into the structure of entanglement [33,34].

Finally, the construction of large-scale quantum computers is currently a matter of intensive research in many laboratories [16,35]. Although the primary motivation for this work is their unique digital information processing capabilities, they will also be extremely useful as analog devices which can simulate other quantum systems (see e.g. [36,37,38]). This also makes it possible, however, for a quantum computer to operate directly on tensor products of geometric algebras, as well as (by computing in suitable homomorphic images) many other algebraic structures. This may enable quantum computers (when one is eventually built!) to solve algebraic and geometric problems that are far beyond the reach of any possible computer based on classical physics alone.

**Acknowledgements**

The author thanks David Cory (MIT) and Chris Doran (Cambridge) for useful discussions on the topics covered herein. This work was supported by the U. S. Army Research Office under grant DAAG 55-97-1-0342 from DARPA/MTO.

## References

1. D. Z. Albert. *Quantum Mechanics and Experience.* Harvard Univ. Press (Cambridge, MA), 1992.
2. A. Peres. *Quantum Theory: Concepts and Methods.* Kluwer Academic (Amsterdam, NL), 1993.
3. D. C. Brody and L. P. Hughston. Statistical geometry in quantum mechanics. *Proc. R. Soc. Lond. A*, 454:2445–2475, 1998.
4. D. C. Brody and L. P. Hughston. Information content for quantum states. *J. Math. Phys.*, 41:2586–2592, 2000.
5. R. Ablamowicz, P. Lounesto, and J. M. Parra, eds. *Clifford Algebras with Numeric and Symbolic Computations.* Birkhäuser (Boston, MA), 1996.
6. T. F. Havel, B. Sturmfels, and N. White. Proposal for a geometric algebra software package. *SIGSAM*, 23(1):13–15, Jan. 1989.
7. T. F. Havel and I. Najfeld. A new system of equations, based on geometric algebra, for ring closure in cyclic molecules. In J. Fleischer, J. Grabmeier, F. W. Hehl, and W. Küchlin, eds., *Computer Algebra in Science and Engineering*, pp. 243–259. World Scientific (Singapore; River Edge, NJ; London, UK; Hong Kong), 1995.
8. T. F. Havel. Computational synthetic geometry with Clifford algebra. In D. Wang, ed., *Automated Deduction in Geometry (ADG'96)*, vol. 1360 of *Lect. Notes in Artif. Intel.*, pp. 102–114. Springer-Verlag (Berlin & Heidelberg, D), 1997.
9. D. Wang. Clifford algebraic calculus for geometric reasoning. In D. Wang, ed., *Automated Deduction in Geometry (ADG'96)*, vol. 1360 of *Lect. Notes in Artif. Intel.*, pp. 115–140. Springer-Verlag (Berlin & Heidelberg, D), 1997.
10. T. Boy de la Tour, S. Fèvre, and D. Wang. Clifford term rewriting for geometric reasoning in 3D. In X.-S. Gao, D. Wang, and L. Yang, eds., *Automated Deduction in Geometry (ADG'98)*, vol. 1669 of *Lect. Notes Artif. Intel.*, pp. 130–155. Springer-Verlag (Berlin & Heidelberg, D), 1999.
11. H. Li. Hyperbolic geometry with Clifford algebra. *Acta Appl. Math.*, 48:317–358, 1997.
12. H. Li. Some applications of Clifford algebra to geometries. In X.-S. Gao, D. Wang, and L. Yang, eds., *Automated Deduction in Geometry (ADG'98)*, vol. 1669 of *Lect. Notes in Artif. Intel.*, pp. 156–179. Springer-Verlag (Berlin & Heidelberg, D), 1999.
13. H. Li. Doing geometric research with Clifford algebra. In R. Ablamowicz and B. Fauser, eds., *Clifford Algebras and their Applications to Mathematical Physics*, vol. 18 of *Prog. Math. Phys.*, pp. 195–218. Birkhäuser (Boston, MA), 2000.
14. C. P. Williams and S. H. Clearwater. *Ultimate Zero and One.* Springer-Verlag (New York, NY), 1999.
15. T. F. Havel, S. S. Somaroo, C.-H. Tseng, and D. G. Cory. Principles and demonstrations of quantum information processing by NMR spectroscopy. *Appl. Algebra Eng. Commun. Comput.*, 10:339–374, 2000. In T. Beth and M. Grassl, eds., *Special Issue: Quantum Computing* (see also LANL preprint `quant-ph/9812086`).
16. C. H. Bennett and D. P. DiVincenzo. Quantum information and computation. *Nature*, 404:247–255, 2000.

17. C. H. Bennett. The thermodynamics of computation: A review. *Intnl. J. Theor. Phys.*, 21:905–940, 1982.
18. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, 1995.
19. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.
20. C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26:1510–1523, 1997.
21. T. Needham. *Visual Complex Analysis.* Clarendon Press (Oxford, UK), 2000.
22. T. F. Havel and C. J. L. Doran. Geometric algebra in quantum information processing. *Contemporary Math. Series*, Am. Math. Soc. (Providence, RI), in press, 2001 (see LANL preprint `quant-ph/0004031`).
23. D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus.* D. Reidel (Dordrecht, NL), 1984.
24. C. J. L. Doran, A. N. Lasenby, and S. F. Gull. States and operators in the spacetime algebra. *Found. Phys.*, 23:1239–1264, 1993.
25. P. Lounesto. *Clifford Algebras and Spinors.* London Math. Soc. Lect. Notes 239. Cambridge Univ. Press (Cambridge, UK), 1997.
26. D. Hestenes. *New Foundations for Classical Mechanics* (2nd ed.). Kluwer Academic (Amsterdam, NL), 1999.
27. S. S. Somaroo, D. G. Cory, and T. F. Havel. Expressing the operations of quantum computing in multiparticle geometric algebra. *Phys. Lett. A*, 240:1–7, 1998.
28. M. Grassl, M. Rötteler, and T. Beth. Computing local invariants of qubit systems. *Phys. Rev. A*, 58:1833–1839, 1998.
29. D. M. Greenberger, M. A. Horne, and A. Zeilinger. Multiparticle interferometry and the superposition principle. *Physics Today*, 46:22–29, Aug. 1993.
30. N. D. Mermin. Quantum mysteries refined. *Am. J. Phys.*, 62:880–887, 1994.
31. J.-W. Pan, D. Bouwmeester, M. Daniell, H. Weinfurter, and A. Zeilinger. Experimental test of quantum nonlocality in three-photon Greenberger-Horne-Zeilinger entanglement. *Nature*, 403:515–519, 2000.
32. R. J. Nelson, D. G. Cory, and S. Lloyd. Experimental demonstration of Greenberger-Horne-Zeilinger correlations using nuclear magnetic resonance. *Phys. Rev. A*, 61:002106, 2000.
33. C. H. Bennett and P. W. Shor. Quantum information theory. *IEEE Trans. Info. Th.*, 44:2724–2742, 1998.
34. S. L. Lomonaco, Jr. An entangled tale of quantum entanglement. *Contemporary Math. Series*, Am. Math. Soc. (Providence, RI), in press, 2001 (see LANL preprint `quant-ph/0101120`).
35. D. G. Cory, R. Laflamme, E. Knill, L. Viola, T. F. Havel, N. Boulant, G. Boutis, E. Fortunato, S. Lloyd, R. Martinez, C. Negrevergne, M. Pravia, Y. Sharf, G. Teklemarian, Y. S. Weinstein, and Z. H. Zurek. NMR based quantum information processing. *Prog. Phys.*, 48:875–907, 2000.
36. R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467–488, 1982.
37. S. Lloyd. Universal quantum simulator. *Science*, 273:1073–1078, 1996.
38. S. S. Somaroo, C.-H. Tseng, T. F. Havel, R. Laflamme, and D. G. Cory. Quantum simulations on a quantum computer. *Phys. Rev. Lett.*, 82:5381–5384, 1999.

# Nonstandard Geometric Proofs

Jacques D. Fleuriot

Division of Informatics – University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN
jdf@dai.ed.ac.uk

**Abstract.** This paper describes ongoing work in our formal investigation of some of the concepts and properties that arise when infinitesimal notions are introduced in a geometry theory. An algebraic geometry theory is developed in the theorem prover Isabelle using hyperreal vectors. We follow a strictly definitional approach and build our theory of vectors within the nonstandard analysis (NSA) framework developed in Isabelle. We show how this theory can be used to give intuitive, yet rigorous, nonstandard proofs of standard geometric theorems through the use of infinitesimal and infinite geometric quantities.

## 1 Introduction

In our previous work on the mechanization of Newton's *Principia*, we introduced, through a combination of techniques from geometry theorem proving (GTP) and nonstandard analysis (NSA), the notion of an infinitesimal geometry in which quantities can be infinitely small [11,12]. The main aim was to capture and mechanize the limit or *ultimate* notions used by Newton in his proofs, while respecting as much as possible his original geometric arguments.

Our formalization task, within the interactive framework of Isabelle, was made possible through the use of concepts from powerful— yet geometrically intuitive— GTP techniques known as the signed area and full-angle methods [5,6]. These methods were highly adequate to our goals as they provided us with lemmas powerful enough to prove the results we wanted but also used geometric notions such as areas and ratios of segments that were directly relevant to Newton's proofs.

In the current work, however, we depart to some extent from the framework already established in Isabelle for geometry. Our aim, now, is to *formally* explore the properties of the infinitesimal geometry theory developed in Isabelle. To this end, we formulate an alternative treatment of geometry based on the notions of hyperreal vectors. We want to provide a rigorous yet powerful theory that can capture formally the properties of our geometry, as well as provide a secure foundations for our previous work.

Moreover, the approach we describe in this paper also differs from that previously adopted in that it is fully definitional. In other words, we now formally define and derive *all* mathematical notions rather than postulate any of them. This approach guarantees consistency, which cannot be ensured when axioms are introduced (see Section 3.1 for a brief overview of this methodology).

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 246–267, 2001.

In what follows, we first offer a brief review of some basic notions from nonstandard analysis that will prove useful to our discussion (Section 2). Next, we give an overview of the vector theory developed in Isabelle (Section 3). In particular, we briefly review the vector algebra, the vectorial definitions used for familiar geometric properties, and some of the infinitesimal geometry theorems that follow. We then outline some of the novel infinitesimal geometric concepts formalized in the work so far (Section 4). We then describe a new approach, based on nonstandard methods, that can be used for proving familiar (standard) geometry theorems (Section 5). Finally, we outline some of the further work currently in the pipeline (Section 6) and share some of the conclusions we have reached so far (Section 7).

## 2 A Few Concepts from Nonstandard Analysis

The definitions below describe the various types of numbers that exist in the nonstandard universe introduced by nonstandard analysis. They provide some of the basic NSA concepts needed to follow this paper.

**Definition 1.** *In an ordered field extension* $\mathbb{R}^* \supseteq \mathbb{R}$, *an element* $x \in \mathbb{R}^*$ *is said to be an infinitesimal if* $|x| < r$ *for all positive* $r \in \mathbb{R}$*; finite if* $|x| < r$ *for some* $r \in \mathbb{R}$*; infinite if* $|x| > r$ *for all* $r \in \mathbb{R}$.

The extended, richer number system $\mathbb{R}^*$ is known as the *hyperreals* [20] and has been developed in Isabelle through purely definitional means using the so-called *ultrapower* construction [13].

**Definition 2.** $x, y \in \mathbb{R}^*$ *are said to be infinitely close,* $x \approx y$ *if* $x - y$ *is infinitesimal.*

This is an important equivalence relation that is crucial to NSA and also to both our past and current work. We present an extension of this relation to hyperreal vectors in Section 3.4 and use it to define the various properties investigated by this work.

## 3 A Mechanized Theory of Hyperreal Vectors

Apart from using an interactive (hence slower) approach to GTP, the current work also differs from the traditional automated approach by residing within the higher-order logic framework of Isabelle/HOL [19]. One of the main reasons for choosing Isabelle/HOL is that it provides a rigorous approach to the formalization of the infinitesimal— a notoriously difficult task. The suitability of Isabelle/HOL for our formalization stems mostly from the benefits gained by adopting the so-called HOL methodology. This is briefly examined next.

### 3.1 The HOL Methodology

The HOL methodology, which derives from work done by Gordon in the HOL88 theorem prover [14], admits only conservative extensions to a theory. This means, as we already mentioned in the introduction, defining and deriving the required mathematical notions rather than postulating them. The definitional approach of HOL requires that assertions are proved about some model instead of being postulated. Such a rigorous definitional extension guarantees consistency, which cannot be ensured when axioms are introduced. As pointed out by Harrison [15], such an approach provides a simple logical basis that can be seen to be correct once and for all. With regards to the foundations of infinitesimals, the definitional approach is certainly advisable when one considers the numerous inconsistent axiomatizations that have been proposed in the past [7]. Of course, care still needs to be exercised, as a wrong definition will almost certainly yield the wrong properties.

The way to proceed is thus very much in the spirit of Hilbert's *Grundlagen*, namely to show that there is a number system (say a field such as the hyperreals) associated with the geometry and reducing consistency of Isabelle's geometric theory to that of hyperreal arithmetic. This is achieved when working within the context of Isabelle/HOL, by developing a geometry theory according to the HOL-methodology i.e. strictly through definitions that capture the notions (points, lines, signed areas, etc.) that are being dealt with and then prove that the various properties follow.

To carry out this task, the hyperreal theories of Isabelle are extended with the notions of hyperreal vectors. In essence, this is an algebraic approach which develops geometric objects and relations between these objects in the Cartesian product $\mathbb{R}^{*n}$ of the field of hyperreals, where $n = 3$. We develop a theory of vectors in three dimensions, although we are mainly interested in plane properties since this has an algebra rich enough to capture the various notions we want to deal with. Thus, it also has more scope for future use. The hyperreals are chosen rather than the reals since we can then express infinitesimal geometric notions as well. The definitions that are used in the theories are given next. One theory introduces the algebraic operations on vectors while the other deals with the development of simple analytic geometry.

### 3.2 Hyperreal Vector Space

In general, the simplest definition for a real vector in $n$ dimensions is as an $n$-tuple of real numbers, $(r_1, \ldots, r_n)$. However, a more geometric definition can be provided that suits our purpose well.

**Definition 3.** *Given two points $P = (x_1, y_1, z_1)$ and $Q = (x_2, y_2, z_2)$ in $\mathbb{R}^{*3}$, the vector $Q - P$ is called the directed line segment from $P$ to $Q$. The components of the directed line segment are the terms in the 3-tuple $(x_2 - x_1, y_2 - y_1, z_2 - z_1)$.*

In this definition, we implicitly assume that the origin is given by the hyperreal coordinates $(0, 0, 0)$ and hence that a particular point is specified by the vector whose components correspond to its Cartesian coordinates. In Isabelle, we formulate a theory of three-dimensional vectors by first introducing vectors as a new type corresponding to a triple of hyperreal numbers:[1]

$$\texttt{hypvec} \equiv \{\texttt{p} :: (\texttt{hypreal} * (\texttt{hypreal} * \texttt{hypreal})).\ \texttt{True}\}$$

Once a new type has been introduced successfully, Isabelle provides coercion functions— the abstraction and representation functions— that enable us to define basic operations on the new type. Thus, in this particular example, the functions

$$\begin{array}{l}\texttt{Abs\_hypvec} :: (\texttt{hypreal} * \texttt{hypreal} * \texttt{hypreal}) \Rightarrow \texttt{hypvec}\\ \texttt{Rep\_hypvec} :: \texttt{hypvec} \Rightarrow (\texttt{hypreal} * \texttt{hypreal} * \texttt{hypreal})\end{array}$$

are added to the theory such that `hypvec` and $\{\texttt{p}.\ \texttt{True}\}$ are isomorphic by `Rep_hypvec` and its inverse `Abs_hypvec`. On a more intuitive level, one may simply read `Abs_hypvec` as:

$$\texttt{Abs\_hypvec}\ (x, y, z) \equiv \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

in what follows.

We can then define the various operations on the new type. For example, the *inner product* or *dot product* of two vectors $P$ and $Q$ is defined, using tuples as patterns in abstractions [19], by:[2]

$$\begin{array}{rl} P \cdot Q \equiv & (\lambda((x_1, y_1, z_1), (x_2, y_2, z_2)).\\ & \quad x_1x_2 + y_1y_2 + z_1z_2)\\ & \quad (\texttt{Rep\_hypvec}\ P, \texttt{Rep\_hypvec}\ Q)\end{array}$$

This definition is slightly more complicated than the usual textbook one since it uses an explicit $\lambda$-abstraction and the representation function. However, we prove theorems that capture the more familiar definitions and which can then be fed to Isabelle's simplifier for rewriting. So for the dot product, we have the expected:

$$\texttt{Abs\_hypvec}\ (x_1, y_1, z_1) \cdot \texttt{Abs\_hypvec}\ (x_2, y_2, z_2) = x_1x_2 + y_1y_2 + z_1z_2$$

Similarly, we also define other important operations, such as *cross product* and *scalar* multiplication ($\cdot_s$). For clarity, we give their definitions as the simplification theorems proved in Isabelle rather than the actual definitions in terms of `Rep_hypvec` and $\lambda$-abstractions. The Isabelle definitions unfortunately tend to be slightly cluttered and become somewhat hard to read, especially in the case

[1] The Isabelle notation $a::\tau$ denotes that $a$ is of type $\tau$.

[2] In what follows, the multiplication sign $(\cdot)$ between hyperreal variables is omitted whenever no ambiguity is likely to result.

of the cross product. So, for cross and scalar products we prove the following rules:

$$\begin{aligned}&\texttt{Abs\_hypvec}\ (x_1, y_1, z_1) \times \texttt{Abs\_hypvec}\ (x_2, y_2, z_2) = \\ &\quad \texttt{Abs\_hypvec}\ (y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)\end{aligned}$$

$$a \cdot_s \texttt{Abs\_hypvec}\ (x, y, z) = \texttt{Abs\_hypvec}\ (ax, ay, az)$$

For any two vectors $P$ and $Q$, the cross product can be viewed as defining the vector area of a parallelogram, with the vectors as two of the sides of the parallelogram and $P \times Q$ perpendicular to the plane containing $P$ and $Q$. With this nice geometric interpretation in mind, the next step involves proving various properties of the cross product. These will be needed for our investigation and will also enable to capture useful notions such as the signed area. The following theorem, which shows that the cross product is not commutative, is thus proved:

$$P \times Q = (-Q) \times P$$

Geometrically, this means a change in the direction of the vector while its magnitude remains unaffected. The negation of a vector $P$, for its part, is defined by negating its various components. In Isabelle:

$$-P \equiv (\lambda(x_1, x_2, x_3).\ \texttt{Abs\_hypvec}\ (-x_1, -x_2, -x_3))(\texttt{Rep\_hypvec}\ P)$$

In the next section, the definition of signed area of a triangle follows directly from the geometric interpretation and algebraic behaviour associated with the cross product.

Various other algebraic properties of the operations introduced so far are proved in Isabelle. A few straightforward ones that are useful to the development are as follows:

- $P \cdot Q = Q \cdot P$
- $P \cdot (Q + R) = P \cdot Q + P \cdot R$
- $(a \cdot_s P) \cdot (b \cdot_s Q) = ab \cdot_s (P \cdot Q)$

- $P \times (Q + R) = P \times Q + P \times R$
- $-(P \times Q) = (-P) \times Q$
- $-(P \times Q) = P \times (-Q)$
- $-(P \times Q) = Q \times P$
- $(a \cdot_s P) \times (b \cdot_s Q) = ab \cdot_s (P \times Q)$
- $P \times P = 0$

- $P \cdot (P \times R) = 0$
- $P \times (Q \times R) = (P \cdot R) \cdot_s Q - (P \cdot Q) \cdot_s R$
- $(a \cdot_s P + b \cdot_s Q) \times R = a \cdot_s (P \times R) + b \cdot_s (Q \times R)$

In these theorems, the *zero vector* is defined, as expected, by

$$0 \equiv \texttt{Abs\_hypvec}\ (0, 0, 0)$$

Another important concept that has not yet been introduced is that of the *length* or *norm* of a vector. For a vector $P$, this is usually denoted by $|P|$ and defined by taking the square root of the dot product $P \cdot P$. In Isabelle:

$$\texttt{hvlen}\ P = \texttt{hsqrt}\ (P \cdot P)$$

The square root operation over the hyperreals, denoted by `hsqrt` in Isabelle, is defined as the nonstandard extension of the square root operation (`sqrt`) over the reals. Details of these nonstandard concepts are given elsewhere [13] and are not especially important to the current exposition. It is sufficient for our purpose to regard taking the square root of a hyperreal as a well-defined operation with the usual properties. Other important theorems proved in the theory include:[3]

- Cauchy-Schwarz inequality: $\texttt{abs}\ (u \cdot v) \leq \texttt{hvlen}\ u \cdot \texttt{hvlen}\ v$
- Minkowski inequality: $\texttt{hvlen}\ (u + v) \leq \texttt{hvlen}\ u + \texttt{hvlen}\ v$

After proving some further results of vector algebra, we develop a simple geometry theory based on the geometric interpretation of vectors and their operations. In the next sections, the definitions and results of the vector geometry development, as it currently stands, are outlined.

### 3.3 Hyperreal Vector Geometry

Chou, Gao, and Zhang have also used vector calculations in automated geometry theorem proving [4]. They assert a set of basic rules about the operations that can be carried out on vectors. Theorems are then derived using these basic axioms of the theory. The algorithm used by Chou et al. is nice and relatively simple: given a construction sequence for a geometric configuration, the points (i.e. vector variables) are eliminated one at a time from the vector expression standing for the conclusion, until only independent vector variables are left. The conclusion that results is then tested to see if it is identically zero.

In contrast to the above approach, we proceed by means of definitions only and having introduced hyperreal vectors and defined the operations on them, there is enough algebraic power for the theories to express geometric concepts: orthogonality and parallelism, signed areas, congruence of angles, infinitesimal geometric notions and much more. Moreover, we proceed mostly through simplification and substitution steps that are be applied to both the conclusion and premises of the current goal. That is, the proof steps in Isabelle are not limited to point elimination only.

We first introduce as basic geometric objects the notions of points and lines by defining the following types in Isabelle:

$$\begin{aligned}&\texttt{pt} \equiv \{\texttt{p} :: \texttt{hypvec}.\ \texttt{True}\}\\&\texttt{line} \equiv \{\texttt{l} :: (\texttt{pt} * \texttt{pt}).\ \texttt{True}\}\end{aligned}$$

From these definitions, a point is therefore specified by a position vector and a (directed) line given by a pair of vectors representing its end-points. We note that

[3] In Isabelle, $\texttt{abs}\ x$ denotes $|x|$.

it is possible for a line to have its two end-points the same— this is not a problem as we can still prove all the expected properties. However, with hindsight, we should probably have ruled out this type of degenerate lines as this would have removed side conditions from several of our theorems.

Notwithstanding the last remark, these definitions do give the theory a separate, nicer geometric interpretation in which geometric objects (points and lines) are dealt with rather than vectors of hyperreal numbers. The abstraction and representation functions of Isabelle enable us to deal with the underlying vector theory to prove basic properties of parallelism, perpendicularity, collinearity etc. Once this is done, we can hope to work at a higher abstract level which deals with geometric relations and interact rather minimally with the underlying vector constructions. This is similar in spirit with our construction of numbers, say the reals by Dedekind cuts, where initially for each operation we have to prove cut properties but as more theorems are proved, we deal less and less with the actual cuts and more with the algebra of the reals.

However, in the subsequent exposition we shall regard position vectors and points as being interchangeable when giving the definitions and describing properties proved. This abuse of notation is simply to make the definitions more readable on paper since it avoids the use of the coercion functions. We will show the definitions or theorems as actually formulated if the need ever arises. We also note that the notation $A -\!\!- B$, used in Isabelle for a line from point $A$ to point $B$, is syntactic sugar for $\texttt{Abs\_line}(A, B)$. Therefore, for each geometric condition, we have the corresponding vector definition:

1. That $A$, $B$, and $C$ are collinear:
$$\texttt{coll}\; C\, A\, B \equiv (C - A) \times (B - A) = 0$$
2. That $AB$ is parallel to $CD$:
$$A -\!\!- B \parallel C -\!\!- D \equiv (B - A) \times (D - C) = 0$$
3. That $AB$ is perpendicular to $CD$:
$$A -\!\!- B \perp C -\!\!- D \equiv (B - A) \cdot (D - C) = 0$$
4. The length of a line $AB$:
$$\texttt{len}\; (A -\!\!- B) \equiv \texttt{hvlen}\; (B - A)$$
5. The signed vector area of triangle $ABC$:
$$\texttt{s\_delta}\; A\; B\; C \equiv 1/2 \cdot_s (A - B) \times (C - B)$$
6. The angle between $AB$ and $CD$:
$$\langle A -\!\!- B, B -\!\!- C\rangle \equiv \texttt{arcos}\; (\texttt{unitvec}\; (A - B) \cdot \texttt{unitvec}\; (C - B))$$
where
$$\texttt{unitvec}\; P = (1/\texttt{hvlen}\; P) \cdot_s P$$

> The definition of the angle relies on the theory of transcendental functions developed in Isabelle. In our work on the formalization of analysis, the various trigonometric functions are defined over the reals through their power series expansions, and then extended to the hyperreals [10].

With these definitions set up, we prove that the basic properties of signed areas actually hold and justify the statements of geometric relations that were made in terms of them. The rules about the sign of the area depending on the ordering of the vertices of the triangle are all proved without any problems since our definition makes them direct consequences of the algebraic properties of the cross product. Consider, for example:

$$\begin{aligned} -\texttt{s\_delta}\; c\; b\; a &= -1/2 \cdot_s (c-b) \times (a-b) \\ &= -1/2 \cdot_s (-(a-b)) \times (c-b) \\ &= - - 1/2 \cdot_s (a-b) \times (c-b) \\ &= \texttt{s\_delta}\; a\; b\; c \end{aligned}$$

This and similar rules are proved with the help of Isabelle's automatic tactic and added to the simplifier. The definition of parallelism in terms of signed areas is also easily verified:

$$a -\!\!- b \parallel c -\!\!- d \Longleftrightarrow (\texttt{s\_delta}\; a\; b\; c = \texttt{s\_delta}\; a\; b\; d)$$

and the following theorem defines incidence (or collinearity) in terms of signed area:

$$\texttt{coll}\; a\, b\, c \Longleftrightarrow (\texttt{s\_delta}\; a\; b\; c = 0) \tag{1}$$

We also extend the definition of incidence to that of a set of points incident on a line, thereby enabling us to prove some more theorems. We can deal with the ratios of oriented lines by proving theorems such as these:

- $A -\!\!- B \parallel C -\!\!- D$:

$$C \not= D \Longrightarrow \frac{\texttt{len}\,(A -\!\!- B)}{\texttt{len}\,(C -\!\!- D)} = \frac{(B-A)\cdot(D-C)}{(D-C)\cdot(D-C)}$$

- if $R$ is the foot of the perpendicular from point $A$ to line $PQ$:

$$P \not= Q \Longrightarrow \frac{\texttt{len}\,(P -\!\!- R)}{\texttt{len}\,(P -\!\!- Q)} = \frac{(A-P)\cdot(Q-P)}{\texttt{len}\,(P -\!\!- Q)^2}$$

- if two non-parallel lines intersect at a point $R$:

$$\begin{aligned} &\texttt{len}\,(P -\!\!- R) \cdot_s (Q-P) \times (V-U) = \\ &\texttt{len}\,(P -\!\!- Q) \cdot_s (U-P) \times (V-U) \end{aligned}$$

Some of the results above are high level lemmas stated by Chou et al. as being used in their automated GTP method based on vectors [4]. We verify all of them in Isabelle and store them as lemmas that become valuable when proving complicated geometry theorems. This verification of lemmas used in the area method is not a mere exercise as it consolidates the axiomatic geometry that we previously used in Isabelle. Moreover, since we are able to prove the expected geometric properties in the formalization, this gives us a relatively high degree of assurance that we are using the right definitions for various concepts.

### 3.4 Introducing the Infinitesimal Geometry

We start by extending some of the definitions used for the hyperreals (see Section 2) to their vectors.

**Definition 4.** *A hyperreal vector $P$ is said to be infinitesimal, finite, or infinite if its length $|P|$ is infinitesimal, finite or infinite respectively. Moreover, $P$ is infinitely close to $Q$ ($P \approx_v Q$) if and only if $Q - P$ is infinitesimal.*

With this definition formalized in Isabelle, the following equivalence theorem about infinitely close vectors is proved:

$$\begin{gathered}\texttt{Abs\_hypvec}\ (x_1, y_1, z_1) \approx_v \texttt{Abs\_hypvec}\ (x_2, y_2, z_2) \\ \Longleftrightarrow x_1 \approx x_2 \wedge y_1 \approx y_2 \wedge z_1 \approx z_2\end{gathered}$$

In other words, two hyperreal vectors are infinitely close if and only if their components in corresponding positions are infinitely close to one another. This is a useful theorem that can be used in many cases to reduce infinitesimal reasoning involving vectors to similar reasoning over the hyperreals. We also prove the following theorems about the different types of vectors:

1. $P$ is infinitesimal if and only if all its components are infinitesimal.
2. $P$ is finite if and only if all its components are finite.
3. $P$ is infinite if and only if at least one of its components is infinite.

and many other interesting theorems about the algebra of the operations and relations on them, such as:[4]

$$[|a \in I\!R; a \not= 0|] \Longrightarrow (a \cdot_s w \approx_v a \cdot_s v) = (w \approx_v v) \tag{2}$$

$$u \approx_v 0 \Longrightarrow u \cdot u \approx 0 \tag{3}$$

$$[|u \approx_v 0; w \in \texttt{VFinite}|] \Longrightarrow u \times w \approx_v 0 \tag{4}$$

$$[|u \approx_v w; w \in \texttt{VFinite}|] \Longrightarrow u \times w \approx_v 0 \tag{5}$$

$$(x \approx_v y) \Longleftrightarrow (\texttt{hvlen}\ (y - x) \approx 0) \tag{6}$$

$$x \approx_v y \Longrightarrow \texttt{hvlen}\ x \approx \texttt{hvlen}\ y \tag{7}$$

$$[|a \approx_v b; c \in \texttt{VFinite}|] \Longrightarrow a \cdot c \approx b \cdot c \tag{8}$$

$$x \cdot x \in \texttt{Infinitesimal} \Longleftrightarrow x \in \texttt{VInfinitesimal} \tag{9}$$

$$[|x \in \texttt{VInfinitesimal}; y \in \texttt{VFinite}|] \Longrightarrow x \cdot y \in \texttt{Infinitesimal} \tag{10}$$

$$u \in \texttt{VFinite} - \texttt{VInfinitesimal} \Longrightarrow ((u \times v \approx_v 0) \Longleftrightarrow (\exists k.\ v \approx_v k \cdot_v u)) \tag{11}$$

where `VInfinitesimal` and `VFinite` denote the sets of infinitesimal and finite vectors respectively. Most of the theorems proved, we believe, have clear geometric readings and formalize the intuitive behaviour one would expect. Theorem (7), for example, can be used directly to prove an intuitive theorem about a shrinking triangle in which one of the sides is infinitesimal. In Fig. 1, for example, one can intuitively see that as the length of $bc$ becomes smaller, the

[4] The Isabelle notation $[|\phi_1, \ldots, \phi_n|] \Longrightarrow \psi$ can be read as **if** $\phi_1 \wedge \ldots \wedge \phi_n$ **then** $\psi$.

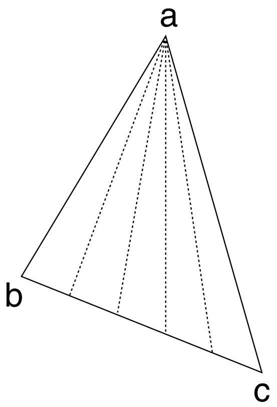


**Fig. 1.** A “shrinking” triangle

lengths of *ab* and *ac* approach each other, until they are infinitely close when *bc* is infinitesimal. This is captured by the following Isabelle theorem:

$$\texttt{len}\ (b -\!\!- c) \approx 0 \Longrightarrow \texttt{len}\ (a -\!\!- b) \approx \texttt{len}\ (a -\!\!- c)$$

Interestingly, if the lengths of the sides *ab* and *bc* are *real* valued, then they have to be *equal* (i.e. triangle *abc* is an isosceles) when *bc* is infinitesimal:

$$\begin{array}{l}[|\texttt{len}\ (a -\!\!- b) \in I\!R; \texttt{len}\ (b -\!\!- c) \in I\!R; \texttt{len}\ (a -\!\!- c) \approx 0|] \\ \Longrightarrow \texttt{len}\ (a -\!\!- b) = \texttt{len}\ (b -\!\!- c)\end{array}$$

This is because of a theorem stating that two real numbers that are infinitely close to one another are effectively equal. We also formally derive, for example, theorems such as:

$$[|\texttt{len}\ (a -\!\!- b) \in \texttt{Finite}; \texttt{len}\ (b -\!\!- c) \in \texttt{Infinitesimal}|] \Longrightarrow \texttt{s\_delta}\ a\ b\ c \approx_v 0$$

and

$$[|\texttt{coll}\ a\, b\, c; \texttt{s\_delta}\ p\ b\ c \approx_v 0|] \Longrightarrow \texttt{s\_delta}\ p\ a\ c \approx_v \texttt{s\_delta}\ p\ a\ b \qquad (12)$$

The latter (see Fig. 2) is proved using the cancellation theorem (2), as well as various others involving associativity and commutativity of vector addition to perform AC-rewriting. These are just a few of the infinitesimal geometry theorems involving familiar geometric concepts. We next introduce a number of basic concepts systematically defined using the various notions from our nonstandard vector theory.

## 4 Some Infinitesimal Geometric Notions

Each of the new definitions can be viewed as weakening of the more familiar ones. We start with a nonstandard formulation of parallelism and orthogonality.

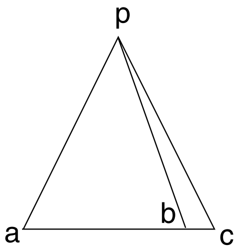


**Fig. 2.** Infinitely close areas

*Almost parallel and almost perpendicular*

Just as the concept of two lines being parallel was introduced, using hyperreal vectors the weaker notion of two lines being *almost parallel* is defined (with $A \not= B$ and $C \not= D$):

$$\begin{aligned} A -\!\!- B \parallel_a C -\!\!- D \equiv\ & \mathtt{unitvec}\ (B - A) \approx_v \mathtt{unitvec}\ (D - C) \vee \\ & \mathtt{unitvec}\ (B - A) \approx_v -\mathtt{unitvec}\ (D - C) \end{aligned}$$

We trivially prove that this is an equivalence relation. More importantly, the relation between this definition and that of parallel lines (see Section 3.3) is highlighted by the following theorem, which is also proved in Isabelle:[5]

$$\begin{aligned} & [|D - C \in \mathtt{VFinite} - \mathtt{VInfinitesimal}; \\ & \ B - A \in \mathtt{VFinite} - \mathtt{VInfinitesimal}|] \\ & \Longrightarrow A -\!\!- B \parallel_a C -\!\!- D \Longleftrightarrow (B - A) \times (D - C) \approx_v 0 \end{aligned} \tag{13}$$

The theorem expresses the almost parallel property in a form similar to that of ordinary parallelism, with equality replaced by the infinitely close relation. However, there is a notable difference which is shown as an additional conditions on the two lines. Without the conditions, (13) above is not a theorem as the cross product of an infinitesimal and infinite vector is not necessarily infinitely close to zero. Also, in terms of area, justifying a more geometrically intuitive definition based on signed areas, we have:

$$\begin{aligned} & [|\mathtt{len}\ (C -\!\!- D) \in \mathtt{Finite} - \mathtt{Infinitesimal}; \\ & \ \mathtt{len}\ (A -\!\!- B) \in \mathtt{Finite} - \mathtt{Infinitesimal}|] \\ & \Longrightarrow A -\!\!- B \parallel_a C -\!\!- D \Longleftrightarrow (\mathtt{s\_delta}\ a\ c\ d \approx_v \mathtt{s\_delta}\ b\ c\ d) \end{aligned}$$

We also define the notion of two lines being almost perpendicular. Once again, we make use of the notion of unit vector to get a suitable definition. Lines

$$A -\!\!- B \perp_a C -\!\!- D \equiv \mathtt{unitvec}\ (B - A) \cdot \mathtt{unitvec}\ (D - C) \approx 0$$

[5] We wish to thank one of the referees for pointing out an omission in our initial statement of the theorem.

We note that since the dot product produces a hyperreal, we use the infinitely close relation $\approx$ over these numbers rather than $\approx_v$ which is defined over hyperreal vectors.

*Almost collinear*

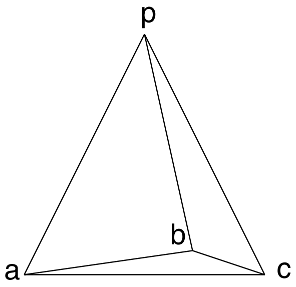


**Fig. 3.** Infinitely close areas

We next introduce the notion of three points being *almost collinear*. Intuitively, one might expect three points $a$, $b$, and $c$ to be almost collinear (denoted by `acoll` $a\,b\,c$ in Isabelle) if and only if the signed area `s_delta` $a\ b\ c$ is infinitely close to zero. Such a definition would be very similar in spirit to the equivalence theorem (1). However, since our geometry allows both infinitesimal and infinite quantities, this definition is inadequate: it does not hold in the case where two of the points concerned, say $b$ and $c$, are infinitely far apart and the third one, say $a$, is infinitely close to the line $bc$. This is because the cross product $(c-b) \times (a-b)$ is not necessarily infinitely close to zero in this case as well. Instead, we define the property as follows:

$$\texttt{acoll}\ a\,b\,c \equiv (b-a) \parallel_a (b-c)$$

and prove a number of theorems involving it such as the variant of (12), shown in Fig. 3:

$$[|\texttt{acoll}\ a\,b\,c; \texttt{s\_delta}\ p\ b\ c \approx_v 0|] \Longrightarrow \texttt{s\_delta}\ p\ a\ c \approx_v \texttt{s\_delta}\ p\ a\ b$$

*Infinitesimal angles*

Our NSA theory is powerful enough to prove theorems involving the trigonometric functions and infinitesimal angles. For example, we can formally formulate and prove assertions such as

$$\sin(\theta) = \theta \text{ and } \cos(\theta) = 1 \ \ \textit{where } \theta \textit{ is infinitely small}$$

that one often sees in textbooks. These are rarely given any further justification: the reader needs to rely on her knowledge of trigonometric functions and on her intuition about what infinitely small means to see that the statements are indeed plausible. Such assertions can be formalized in NSA, however, by making $\theta$ an infinitesimal and replacing equality by the infinitely close relation $\approx$. The proofs are intuitive, yet rigorous, and relatively easy to mechanize. We give, as an example, a brief proof of the statement $\sin(\theta) \approx \theta$.

In the NSA theory [13] of Isabelle/HOL, the formal nonstandard definition of the derivative of a function $f$ at $x$ ($\mathsf{DERIV}$) is given by:

$$\mathsf{DERIV}(x)\ f :> d \equiv \forall h \in \mathsf{Infinitesimal} - \{0\}.\ \frac{f(x+h) - f(x)}{h} \approx d$$

This is simply saying that the derivative of $f$ at $x$ is $d$ if $\frac{\Delta f}{\Delta x}$ is *infinitely close* to $d$. With this, and assuming the standard results (proved in Isabelle) that

$$\cos(0) = 1, \sin(0) = 0,$$

and

$$\mathsf{DERIV}(x)\ (\lambda x.\ \sin(x)) :> \cos(x),$$

we can easily prove that $\cos(\theta) \approx \theta$ for all infinitesimal $\theta$.

**Proof:**

if $\theta = 0$: This is trivial since $\approx$ is reflexive.
else if $\theta \not= 0$: Since$\mathsf{DERIV}(x)\ (\lambda x.\ \sin(x)) :> \cos(x)$, for all $x$, we have that

$$\mathsf{DERIV}(0)\ (\lambda x.\ \sin(x)) :> \cos(0)$$

$$\Rightarrow \forall h \in \mathsf{Infinitesimal} - \{0\}.\ (\sin(0+h) - \sin(0))/h \approx 1$$

$$\Rightarrow (\sin(0+\theta) - \sin(0))/\theta \approx 1$$

$$\Rightarrow \sin(\theta)/\theta \approx 1$$

$$\Rightarrow \sin(\theta) \approx \theta$$

One important point to note is that we made use of the following theorem to reach the final step:

$$\frac{a \approx b \qquad c \in \mathsf{Finite}}{a \cdot c \approx b \cdot c}$$

where $\mathsf{Finite}$ is the set of finite numbers and $\mathsf{Infinitesimal} \subseteq \mathsf{Finite}$ [13]. In a similar fashion, we also prove that $\cos(\theta) \approx 1$ and, interestingly, that $\tan(\pi/2 + \theta) \in \texttt{Infinite}$, for all infinitesimal $\theta$. We expect such results involving angles and trigonometry will to prove useful in the further development of the geometry.

In addition, we also prove that the angle between two lines which are almost perpendicular is infinitely close to $\pi/2$, i.e.,

$$a \mathbin{—\!—} b \perp_a c \mathbin{—\!—} d \Longleftrightarrow \langle a \mathbin{—\!—} b, c \mathbin{—\!—} d\rangle \approx \pi/2$$

*Almost similar triangles*

This is basically the notion of *ultimately similar* triangles that we have described and used a number of times before [11,12]. We briefly recall its definition here:

$$\begin{aligned}\texttt{USIM}\ a\ b\ c\ a'\ b'\ c' \equiv\ & \langle b \;\text{---}\; a, a \;\text{---}\; c\rangle \approx \langle b' \;\text{---}\; a', a' \;\text{---}\; c'\rangle \ \wedge \\ & \langle a \;\text{---}\; c, c \;\text{---}\; b\rangle \approx \langle a' \;\text{---}\; c', c' \;\text{---}\; b'\rangle \ \wedge \\ & \langle c \;\text{---}\; b, b \;\text{---}\; a\rangle \approx \langle c' \;\text{---}\; b', b' \;\text{---}\; a'\rangle\end{aligned}$$

We are still formally investigating the properties of this concept. We have already reproduced in our new setting most of the theorems described in previous work [12]. Similarly, we have defined the notion of two triangles being almost congruent.

## 5 Nonstandard Proofs of Standard Geometry Theorems

Our nonstandard techniques are strong enough to produce nice proofs of traditional geometry theorems. The proofs can be viewed as moving into the hyperreal space, just as it is possible to move into complex space when dealing with proofs in analytic geometry. In what follows, we illustrate our nonstandard methods by considering infinite polygonal approximations of the circle.

### 5.1 Polygonal Area Approximation

We first consider a nonstandard proof that the area of a circle of radius $r$ is $\pi r^2$. The area of the circle will be shown to be infinitely close, hence equal, to the area of infinitely many enclosed (inscribed) polygons.

In Fig. 4, the area of the closed polygon $P \equiv A_1 \dots A_n$ is defined by the formula:

$$\text{area } A_1 \dots A_n \equiv OA_1A_2 + OA_2A_3 + \dots + OA_{n-1}A_n + OA_nA_1$$

where $OA_1A_2$, for example, represents the area of triangle $OA_1A_2$ (which we re-define below in terms of the vector outer-product in the plane). The value of the polygonal area is independent of $O$ but depends on $OA_i$, the radius vector to the $i$th point. The definition of polygonal area looks recursive except for the last area term ($OA_nA_1$). In Isabelle, this motivates the following definition for the area of the polygon with the zero vector as $O$ (and the polygonal points numbered from 0 rather than 1):

$$\texttt{polyArea}\ P\ n\ r \equiv \texttt{pArea}\ P\ n\ r + \texttt{area}\ 0\ (P\,n\,r)\ (P\,0\,r)$$

where `pArea`, the area of the open polygon, is inductively defined as

$$\begin{aligned}&\texttt{pArea}\ P\ 0\ r = 0\\ &\texttt{pArea}\ P\ n\ r = \texttt{pArea}\ P\ (n-1)\ r + \texttt{area}\ 0\ (P\,(n-1)\,r)\ (P\,n\,r)\end{aligned}$$

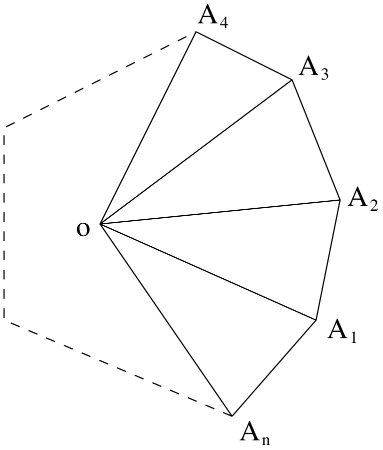


**Fig. 4.** A closed polygon

and
$$\texttt{area}\ a\ b\ c \equiv 1/2[(b-a)(c-a)]$$
$$[\texttt{Abs\_hypvec}\ (x_1, y_1)\texttt{Abs\_hypvec}\ (x_2, y_2)] = x_1 y_2 - y_1 x_2 \tag{14}$$

We note that a parameter $r$, which at first sight might appear superfluous, is included in the definitions of both `polyArea` and `pArea`. This is because often the radius vectors $OA_i$ does not depend on just $i$ but also on some other quantity such as an angle. The two parameters (e.g. multiplied) together enable us to progress along the curve being approximated. In an initial formalization, we omitted the extra parameter but then found that we could not adequately represent the inscribed polygon. This lead to the revised definition presented in this section.

Now, if $C$ is a circle of radius 1, we can inscribe a polygon $A_1 \cdots A_n$ by choosing points $A_1$, $A_2$,..., $A_n$ in order along it. If $n$ is an infinite hypernatural number then the points $A_i$ crowd one another, and we expect to arrive at the formula for the area enclosed by $C$.

In our mechanized proof, we first consider the unit semi-circle $ABC$ (see Fig. 6). Using the angle $\theta$ between successive radius vectors as parameter, the polygon can be defined by the following sequence:

$$\lambda k\, \theta.\ \texttt{Abs\_hypvec}\ (\cos k\theta, \sin k\theta) \tag{15}$$

where $k$ denotes the $k$-th point of the polygon. Hence, given that $n$ points are inscribed in the semi-circle, the angle between the radius vectors is $\pi/n$ and so the polygonal area is:

$$\texttt{polyArea}\ (\lambda k\, \theta.\ \texttt{Abs\_hypvec}\ (\cos k\theta, \sin k\theta))\ n\ (\pi/n)$$

We then easily prove by induction and with the help of the mechanized lemma:

$$\sin(x-y) = \cos y \sin x - \sin y \cos x \tag{16}$$

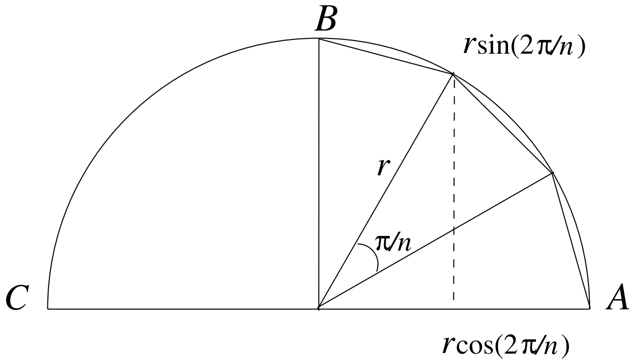


**Fig. 5.** Inscribing a polygon of $n$ sides in a semi-circle

supplied to Isabelle's simplifier that the following theorem holds:

$$\texttt{polyArea}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (\cos k\theta, \sin k\theta))\ n\ (\pi/n) = 1/2 n \sin(\pi/n) \tag{17}$$

We use the fact that:

$$\texttt{area}\ 0\ (c \cdot_s x)\ (c \cdot_s y) = c^2 \cdot \texttt{area}\ 0\ x\ y$$

to prove by induction the following property of polygonal areas:

$$\texttt{polyArea}\ (\lambda n r.\ c \cdot_s P\ n\ r)\ N\ R = c^2 \cdot \texttt{polyArea}\ P\ N\ R \tag{18}$$

which means that for a semi-circle of radius $r$, we have:

$$\texttt{polyArea}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (\pi/n) = 1/2 r^2 n \sin(\pi/n) \tag{19}$$

Given that $n$, the number of inscribed points, is an infinite hypernatural number, we have that $\pi/n$ is infinitesimal. But, from the result in the previous section about infinitesimal angles, we also know that

$$\frac{\sin(\pi/n)}{(\pi/n)} \approx 1$$

and hence that

$$n \sin(\pi/n) \approx \pi \tag{20}$$

This result, with (19) above, allows us to prove that:

$$\texttt{polyArea}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (\pi/n) \approx 1/2 \pi r^2$$

from which we deduce that for a circle, with the angle between successive radius vectors given by $2\pi/n$, the following holds:

$$\texttt{polyArea}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (2\pi/n) \approx \pi r^2 \tag{21}$$

Hence, by "exhausting" the circle with a inscribed polygon of infinite number of sides, we have formalized a simple and relatively intuitive proof that the area of the circle of radius $r$ is infinitely close to $\pi r^2$. If we assume that the area of the circle is a real quantity then we can deduce that it is equal to $\pi r^2$, as one would expect.

### 5.2 Polygonal Length Approximation

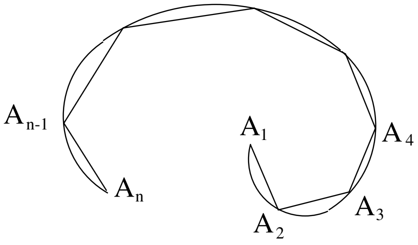


**Fig. 6.** Approximating the length of a curve using an inscribed polygon

Our technique for determining areas is easily adapted to determining the length of a curvilinear arc. Geometrically, the length of a polygon $A_1 ... A_n$ inscribed along some arc is given by $\sum_{i=1}^{n} |A_i - A_{i-1}|$ (see Fig. 6). In Isabelle, this is a direct recursive definition (with the points numbered from 0 rather than 1):

$$\texttt{pLength}\ P\ 0\ r = 0$$
$$\texttt{pLength}\ P\ n\ r = \texttt{pLength}\ P\ (n-1)\ r + \texttt{hvlen}\ ((P\, n\, r) - (P\, (n-1)\, r))$$

This definition is further refined to the case when we are dealing with a closed curve such as a circle. We then need to close the polygon by adding the length of the vector (segment) from the last to the first point of the figure. In Isabelle,

$$\begin{aligned}\texttt{polyLength}\ P\ n\ r \equiv\ & \texttt{if}\ n = 1\ \texttt{then}\ \texttt{hvlen}\ ((P\, 1\, r) - (P\, 0\, r)) \\ & \texttt{else}\ \texttt{pLength}\ P\ n\ r + \texttt{hvlen}\ ((P\, n\, r) - (P\, 0\, r))\end{aligned}$$

The definition is conditional to prevent the length of a degenerate closed polygon with only two points $A_0$ and $A_1$ (i.e. a line) from being defined as $|A_0 A_1| + |A_1 A_0|$. With these concepts defined, we prove a theorem about polygonal length analogous to theorem (18) about area:

$$\texttt{polyLength}\ (\lambda nr.\ c \cdot_s P\ n\ r)\ N\ R = \texttt{abs}\ c \cdot \texttt{polyLength}\ P\ N\ R$$

This property holds because of the following theorem about lengths of vectors:

$$\texttt{hvlen}\ (c \cdot_s x - c \cdot_s y) = \texttt{abs}\ c \cdot \texttt{hvlen}\ (x - y)$$

We continue with our case study involving the circle and outline how to approximate its circumference using an infinite polygonal approximation. The mechanization follows a similar approach to that of the previous section and uses the parametric definition (15) for the inscribed polygon. This time we consider $n$ points being inscribed in a circle of radius $r$, which means that the angle between each radius vector is $2\pi/n$. The polygonal length is now given by the theorem:

$$\begin{aligned}&\texttt{polyLength}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (2\pi/n)\\ &= \texttt{abs}\ r\cdot \texttt{polyLength}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (\cos k\theta, \sin k\theta))\ n\ (2\pi/n)\\ &= \texttt{abs}\ r\cdot n\sqrt{2-2\cos(2\pi/n)}\end{aligned} \tag{22}$$

proved by induction on $n$ followed by simplification. Using the following lemmas (all proved in Isabelle) as rewrite rules:

$$\begin{aligned}\cos 2x &= \cos^2 x - \sin^2 x\\ \cos^2 x &= 1 - \sin^2 x\\ 2 - 2\cos 2x &= 4\sin^2 x\\ \sqrt{x^2} &= \texttt{abs}\ x\end{aligned}$$

theorem (22) automatically simplifies to

$$\begin{aligned}&\texttt{polyLength}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (2\pi/n)\\ &= \texttt{abs}\ r\cdot n\cdot \texttt{abs}\ (2\sin(\pi/n))\end{aligned} \tag{23}$$

With this result set up, we are almost done, since using theorem (20), we now have that:

$$n\cdot \texttt{abs}\ (2sin(\pi/n)) = \texttt{abs}\ (2n\sin(\pi/n)) \approx \texttt{abs}\ (2n(\pi/n)) = 2\pi$$

and hence that:

$$\texttt{polyLength}\ (\lambda k\,\theta.\ \texttt{Abs\_hypvec}\ (r\cos k\theta, r\sin k\theta))\ n\ (2\pi/n) \approx 2\pi\cdot \texttt{abs}\ r \tag{24}$$

Once again, our infinite approximation has provided a reasonably intuitive proof of a familiar geometric result. All our proofs proceed with a relatively high degree of automation since much of the simplification work can be done automatically by Isabelle's rewriter.

Our definitions for polygonal area and polygonal length are generic and can thus be used to approximate the areas and lengths of other figures provided these can be defined formally. This also means that the techniques described in this section are general ones that, we believe, provide new methods for mechanical theorem proving in geometry.

### 5.3 Brief Remarks on the Mechanization

We now make a few further remarks on the proof formalization just considered. In particular, we give some indication of the amount of work involved in completing some of these proofs, and discuss how the proof process might be (completely) automated.

As we have already mentioned, Isabelle can provide a relatively high degree of proof automation in many cases (e.g. through the use of built-in decision procedures and automatic tactics). However, since the system is a proof assistant rather than an automatic theorem prover (like Otter [18], say) the user is expected to play an active role in the proof-finding process. Indeed, during proofs the user often interacts with the system and guides it by indicating which tactic and rules to apply at each step.

The proof of lemma (17), for example, requires 5 steps and takes less than half a second. Four of these steps are rewrites carried out by supplying rules such as (14) and (16) to Isabelle's automatic tactic `auto_tac`. The only different step is a case-split: this requires an explicit intervention in which we consider, using Isabelle's `case_tac` tactic, the cases $n = 0$ and $n \neq 0$ as two separate subgoals. As mentioned in Section 5.2, theorem (21) then easily follows from this lemma and result (20). This last proof is automatic: it only requires one application of `auto_tac` with these theorems supplied as rewrite rules and takes negligible time to complete.

Another important aspect of many of our proofs, which directly affects their possible automation, is the need for mathematical induction. As our definitions for the area and boundary of a polygon are both recursive, the use of induction in proving their properties is to be expected. Thus, in the case of theorem (18), for instance, we have to explicitly specify on which variable Isabelle is to perform the induction ($N$ in that case). More importantly, the use of an inductive theory for expressing our geometric notions means that, in general, we cannot hope for full automation (due to results like Gödel's first incompleteness theorem).

However, these negative aspects do not mean that we have to give up on trying to automate these proofs altogether. In fact, there are has been extensive work on the automation of inductive proofs in the past by Boyer and Moore [2], Bundy [3], Kapur [16], and many others. The proof-planning approach of Bundy is especially relevant to us, as we are currently involved in integrating it with Isabelle. This successful AI-style planning technique which guides inductive (and other types of) proofs through the use of powerful heuristics should, we hope, provide a clear path to the automation of our own geometric proofs.

## 6 Further Work

As mentioned already, this paper describes work currently in progress. We still have much of the geometry to explore. One currently unproved conjecture, for example, is that two (co-planar) lines which are almost parallel do *meet* at a point infinitely far away i.e., we expect to have a well-defined, non-degenerate solution to the problem.

We now have a relatively well developed vector theory. This contains many of the familiar theorems about vector operations as well as the new theorems involving the infinitely close relation, infinitesimal and infinite vectors, and other nonstandard notions. As the work proceeds, we expect to add more theorems to provide a theory that can be useful for other purposes (e.g. proofs in mechanics that often involve vectors and infinitesimals).

We will be introducing and investigating other, perhaps less obvious, *almost relations*. For example, we have recently mechanized notions of approximate geometric objects. Using this notion, an ellipse with infinitely close foci can be regarded as being almost (but not quite) a circle. Other notions include "almost betweenness", approximate point inclusion in a triangle, and "almost a tangent" to a circle, for example.

Another aim will be to mechanize more geometric proofs that use infinitesimal and infinite quantities to reach infinitely accurate approximation results. We have formalized the useful notion of a polygon with an infinite number of sides which can be used to approximate any closed figure (curve). We have shown, in details, how this can be used to derive simple and intuitive proofs about the area and circumference of the circle. We will mechanize other proofs that use Archimedes so-called "Method of Exhaustion" in which one figure is approximated more and more accurately by another one in order to compute geometric quantities such as areas and volumes. We believe that these are proofs not currently captured by existing mechanical theorem proving methods. The work of Baron [1], for example, provides a wealth of such proofs throughout the centuries for us to work with and mechanize.

Finally, as mentioned in the previous section, we hope to make automation an important feature of our approach by using proof-planning, rather that human intervention, to guide Isabelle in finding geometric proofs of conjectures. This goal should create a nice link between our geometry work and some of our other interests.

## 7 Concluding Remarks

In this paper, we have formally introduced the notion of an infinitesimal geometry based on hyperreal vectors. Various theorems have been proved that have no direct counterparts in Euclidean geometry since the latter only deals with real numbers.

Vector algebra offers an attractive approach to mechanical geometry theorem proving. There is much active research going on using the related field of Clifford algebra, which is generally regarded as being more expressive [21,9]. In our case, since we are doing interactive rather than automatic theorem proving, vectors provide a simple and adequate approach to analytic geometry. Also, as was shown by Dieudonné, inner (dot) and cross products of vectors are sufficient to develop elementary geometry [8].

As far as we are aware, this is the first mechanization of a theory of hyperreal vectors. Moreover, Keisler's textbook is, to our knowledge, the only work to give a brief exposition of a vector theory [17]. As a result, most of the theorems mechanized in Isabelle have been proved independently of any previous work or textbooks. We have shown that these vectors obey the usual algebraic rules for vectors since they form an inner product space over the field $I\!R^*$. By using the extended vectors instead of real vectors, it is possible to describe, in addition to ordinary geometric concepts, the novel notions of infinitesimal geometry presented in this paper.

The analytic geometry development was carried out to provide a rigorous definitional approach in which to investigate our infinitesimal geometry. By following the HOL methodology, we have the guarantee that our formalization is consistent and that all results proved are actual theorems about the geometry we have developed. In addition, it provides support for our previous work by giving rigorous proofs of many of the basic rules from the GTP methods of Chou et al. that were previously asserted as axioms in Isabelle.

As a final note, we remark on an important realisation emphasized by the current work: the inclusion of infinitesimals and other nonstandard concepts in geometry introduces subtle issues that can easily lead to inadequate definitions. Indeed, it can be problematic to formulate concepts that rely on some form of product (cross, dot, multiplication etc.) as the operation can be ill-defined whenever it involves both an infinitesimal and an infinite quantity. We became especially aware of the subtlety involved when our initial definition for almost parallel lines (we used the equivalence theorem (13) without the associated conditions) proved inadequate. We could not prove some of the properties we felt should hold since we were implicitly ruling out an infinitesimal line and an infinite line being almost parallel.

The realisation came after some experimentation with the framework and did force us to exercise much more care. However, the fact that we encountered such a problem is probably unsurprising. After all, the flaw that we found in one of the famous proofs of the great Newton was also of this nature [12]; it involved taking the ill-defined product of an infinitesimal and an infinite quantity. However, this is a useful experience that will help us as we explore more challenging concepts in this geometry.

### Acknowledgement

This research was funded by EPSRC grant GR/M45030 ‘Computational Modelling of Mathematical Reasoning’. I would like to thank the anonymous referees for their insightful comments.

## References

1. M. E. Baron. *The Origins of the Infinitesimal Calculus.* Pergammon Press, 1969.
2. R. S. Boyer and J. S. Moore. *A Computational Logic.* ACM Monograph Series. ACM Press, 1979.
3. A. Bundy. The use of explicit plans to guide inductive proofs. In R. Lusk and R. Overbeek, editors, *9th International Conference on Automated Deduction – CADE-9*, volume 310 of *Lecture Notes in Computer Science*, pages 111–120. Springer-Verlag, May 1988.

4. S. C. Chou, X. S. Gao, and J. Z. Zhang. Automated geometry theorem proving by vector calculation. In *ACM-ISSAC*, Kiev, Ukraine, July 1993, pages 284–291. ACM Press, 1993.
5. S. C. Chou, X. S. Gao, and J. Z. Zhang. Automated generation of readable proofs with geometric invariants, I. Multiple and shortest proof generation. *Journal of Automated Reasoning*, 17:325–347, 1996.
6. S. C. Chou, X. S. Gao, and J. Z. Zhang. Automated generation of readable proofs with geometric invariants, II. Theorem proving with full-angles. *Journal of Automated Reasoning*, 17:349–370, 1996.
7. P. J. Davis and R. Hersh. *The Mathematical Experience.* Harmondsworth, Penguin, 1983.
8. J. Dieudonné. *Linear Algebra and Geometry.* Hermann, 1969. Translated from the original French text *Algèbre linéaire et géométrie élémentaire.*
9. S. Fevre and D. Wang. Proving geometric theorems using Clifford algebra and rewrite rules. In C. Kirchner and H. Kirchner, editors, *Automated Deduction – CADE-15*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 17–32. Springer-Verlag, July 1998.
10. J. D. Fleuriot. On the mechanization of real analysis in Isabelle/HOL. In J. Harrison and M. Aagaard, editors, *Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000*, volume 1869 of *Lecture Notes in Computer Science*, pages 146–162. Springer-Verlag, 2000.
11. J. D. Fleuriot and L. C. Paulson. A combination of geometry theorem proving and nonstandard analysis, with application to Newton's *Principia.* In C. Kirchner and H. Kirchner, editors, *Automated Deduction – CADE-15*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 3–16. Springer-Verlag, July 1998.
12. J. D. Fleuriot and L. C. Paulson. Proving Newton's Propositio Kepleriana using geometry and nonstandard analysis in Isabelle. In X.-S. Gao, D. Wang, and L. Yang, editors, *Automated Deduction in Geometry*, volume 1669 of *Lecture Notes in Artificial Intelligence*, pages 47–66. Springer-Verlag, 1999.
13. J. D. Fleuriot and L. C. Paulson. Mechanizing nonstandard real analysis. *LMS Journal of Computation and Mathematics*, 3:140–190, 2000.
14. M. Gordon and T. Melham. *Introduction to HOL: A theorem proving environment for Higher Order Logic.* Cambridge University Press, 1993.
15. John Harrison. *Theorem Proving with the Real Numbers.* Springer-Verlag, 1998. Also published as technical report 408 of the Computer Laboratory, University of Cambridge, 1996.
16. D. Kapur and M. Subramaniam. Lemma discovery in automating induction. In M. A. McRobbie and J. K. Slaney, editors, *Automated Deduction – CADE-13*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 538–552. Springer-Verlag, August 1996.
17. H. J. Keisler. *Foundations of Infinitesimal Calculus.* Prindle, Weber & Schmidt, 1976.
18. W. McCune. OTTER 3.0 reference manual and guide. Technical Report ANL-94/6, Argonne National Laboratory, 1994.
19. L. C. Paulson. Isabelle's object-logics. Technical Report 286, Computer Laboratory, University of Cambridge, February 1998.
20. A. Robinson. *Non-standard Analysis.* North-Holland, 1980.
21. D. Wang. Clifford algebraic calculus for geometric reasoning, with application to computer vision. In D. Wang, R. Caferra, L. Fariñas del Cerro, and H. Shi, editors, *Automated Deduction in Geometry, ADG'96*, volume 1360 of *Lecture Notes in Artificial Intelligence*, pages 115–140. Springer-Verlag, 1997.

# Emphasizing Human Techniques in Automated Geometry Theorem Proving: A Practical Realization

Ricardo Caferra, Nicolas Peltier, and François Puitg

Laboratoire LEIBNIZ-IMAG
46, Avenue Félix Viallet - 38031 Grenoble Cedex - France
{Ricardo.Caferra,Nicolas.Peltier,Francois.Puitg}@imag.fr
Phone: (33) 4 76 57 46 59

**Abstract.** The underlying principles and main original techniques used in a running generic logic-based theorem prover are presented. The system (a prototype) is called $\text{HOARD}_{\text{ATINF}}$ (**H**uman **O**riented **A**utomated **R**easoning on your **D**esk) and has been specialized in this work to proof learning through geometry. It is based on a new calculus, particularly suited to the class of problems we deal with. The calculus allows treatment of equality and automatic model building. $\text{HOARD}_{\text{ATINF}}$ has some other original characteristics such as proving by analogy (using matching techniques), some possibilities of discovering lemmata (using diagrams), handling standard theories in geometry such as commutativity and symmetry (by encoding them in the unification algorithm used by the calculus), and proof verification in a rather large sense (by using capabilities of the calculus).

As this work is intended to set theoretical bases of a new logic-based approach to geometry theorem proving, a comparison of *features* of our system with respect to those of other important, representative logic-based systems is given. Some running examples give a good taste of the $\text{HOARD}_{\text{ATINF}}$ capabilities. One of these examples allows us to compare qualitatively our approach with that of a powerful prover described in a recent paper [8]. Some directions of future research are mentioned.

**Keywords.** Automated geometric reasoning, analogy, model (counter-example) building, proof structuring with diagrams, computer assisted learning.

## 1 Introduction

For many years we have worked with the aim of improving general theorem provers from a **qualitative** point of view in the framework of a project called ATINF (French acronym for “ATelier d’INFérence”, meaning “Inference Laboratory”; see [5,4] for example). We oppose qualitative to (only) “fast and cleverly”. More precisely, we would like general theorem provers not only to prove theorems, but also to help users to analyze and present proofs, to find analogies with other proofs, to build counter-examples (models), etc.

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 268–305, 2001.

Of course our aims are not original and since the very beginning of automated theorem proving some pioneer work pursued similar goals (see e.g. [15] and the work of Bledsoe's group already in the seventies). This early nice work is characterized by *ad hoc* approaches to particular classes of problems. We contend that a general approach towards a formalization of powerful human techniques of reasoning is possible (and valuable). This generality is not contradictory with a specialization a posteriori to particular classes of problems.

The present work is an application of our views to a particular domain: *learning of proof through geometry.* To reach this goal we specialize some techniques presented elsewhere (see for example [5,7,3,13,12]) and well-known such as $E$-unification (i.e. unification modulo equational theories) or hyper-resolution and paramodulation. The latter are the two components of a new calculus that is at the very basis of the system. We also use diagrams in a more human-oriented way, i.e. not only to prune the search space by filtering but also in order to suggest lemmata allowing to plan and to structure proofs (the use of diagrams not only as a filter but also as a guide is suggested too in [10] and in [30] but in a much more informal and embryonic way).

Albeit some of our proposals are only partially developed, these are not only theoretical propositions or general ideas about the subject: a running prototype based on this approach is presented in detail and some running examples are shown.

As it is well known, there exist some extremely efficient, powerful systems for geometrical theorem proving (see e.g. [16]). Such systems are based on powerful algebraic methods such as Wu's. The basic principle is to translate the considered problem into an algebraic language, and then solve the considered set of equations, using specialized algorithms (for example the "characteristic set" or Gröbner basis method, see e.g. [9]). The powerful methods of Tarski, Collins and Wu have on one side limited range of applications (see e.g. [18]) and on the other side — and much more important for our present purpose — they do not provide *readable proofs.*

There exist a few systems based on purely logical approaches to geometrical theorem proving (see e.g. [10,28]) and the most powerful (general-purpose) existing theorem provers such as OTTER, SPASS, and SETHEO are not successful when trying to prove geometry theorems. It is commonly admitted that algebraic approaches are the most powerful and efficient but there are several strong reasons to justify further investigation of logical approaches (see e.g. [8]).

Our main goal is *not* to program a new logic-based geometry theorem prover, but to show that it is possible to incorporate to the provers formalizations of very useful human techniques that increase their power and make their interaction with users much more natural.

Learning (teaching) of the notion of proof through geometry seems to us both a challenge and a way to show evidence of the usefulness of our approach.

This work presents part of the automated reasoning module of a project on computer assisted learning named **Baghera**, under development at the LEIBNIZ-IMAG laboratory. One of the main aims of Baghera is to fill a gap in existing tutorial systems (see e.g. [1]): these systems neither perform really automated reasoning (i.e. proof verification, proposition of alternative proofs, ...) nor offer techniques for the analysis or presentation of proofs. Consequently, we have tried to incorporate these features in our theorem prover HOARD$_{\text{ATINF}}$ (**H**uman **O**riented **A**utomated **R**easoning on your **D**esk). For reasons both historical and of closeness we started from a tutorial systems called CABRI-Euclide (see [20,21], and references therein) and in a similar way we did with our system GLEF$_{\text{ATINF}}$ (see [4]), and we fixed a list of requirements to reach ("ideal" specification).

The system should be able to *verify* proofs (and to *complete* them if necessary: human beings never write fully formalized proofs); to build *counter-examples (models)*; to verify and to suggest *analogies* between proofs and/or formulas; to do *abduction* — i.e. to identify (supplementary) hypothesis that could allow to prove (or to "explain") a given assertion; to perform some form of *proof planning*, suggesting a guideline for the stuck student.

Obviously the system should be efficient enough to react to requests in "reasonably" short time. It should also be *generic* because it is intended to be used by students of very different levels of knowledge. Therefore the underlying geometric theory can change. The system must be able to adapt to these changes (i.e. to deal with different theories), in order to construct proofs *understandable* by the user.[1] Therefore, no general axiomatization (such as Hilbert's or Tarski's) can be used. This need of genericity prevents us from using very specialized and specific approaches such as expert systems, and makes the choice of a *logical* approach very natural.

### Organization of the Paper

The rest of the paper is organized as follows: In Section 2, we describe the fragment of first-order logic used to encode geometric problems submitted to HOARD$_{\text{ATINF}}$, and we present the proof calculus that is used to obtain proofs. This calculus is a refinement of the ordered hyper-resolution + ordered positive paramodulation rules and is particularly well adapted to human oriented proof presentation (see Annex A). We particularly emphasize its specificity w.r.t. existing similar calculi.

In Section 3, we describe the additional features that have been added into the system, such as proof analysis, proof generalization, analogy detection, and counter-example (model) construction.

Section 4 particularly emphasizes one important semantic aspect of the system: the ability to use diagrams for guiding and structuring proofs, in particular to generate lemmata.

[1] Of course, another solution would have been to compute the proof in some fixed axiomatization of geometry, and then to translate it into the desired axiomatization. But designing such a translation algorithm would be a difficult — and possibly infeasible — task.

Sections 3 and 4 are those that will be studied more extensively in our future research. Nevertheless we consider the already obtained results as very promising, as hopefully shown by the examples.

Section 6 gives a short conclusion and main lines of future work.

## 2 The Theorem Prover HOARD$_{\text{ATINF}}$: Principles and Main Features

In this section, we describe the theoretical bases of HOARD$_{\text{ATINF}}$. The prover is implemented in Prolog (for the sake of fast prototyping and portability). HOARD$_{\text{ATINF}}$ is still a prototype and a lot of implementational work remains to be done before putting it in the public domain (see Section 5).

*Remark 1.* It is neither possible nor useful to recall here all the standard notions of automated reasoning used in this work. The interested reader can consult [36,19].

### 2.1 A Restricted First-Order Language

HOARD$_{\text{ATINF}}$ uses a subclass of first-order logic. We recall below some basic definitions concerning the language.

**Definition 1.** *Let $\Sigma$ be a set of functional symbols, let $\mathcal{X}$ be a set of variables. Let arity be a function mapping each element of $\Sigma$ into an element of* $\mathbb{N}$. *The* set of terms *$T(\Sigma, \mathcal{X})$ is the least set that satisfies the following properties:*

- *$\mathcal{X} \subseteq T(\Sigma, \mathcal{X})$;*
- *If $f \in \Sigma, arity(f) = 0$ (i.e. the constants) then $f \in T(\Sigma, \mathcal{X})$;*
- *If $f \in \Sigma, arity(f) = n, n > 0, (t_1, \ldots, t_n)^n \in T(\Sigma, \mathcal{X})^n$ then $f(t_1, \ldots, t_n) \in T(\Sigma, \mathcal{X})$.*

The set of terms containing no variables are called *ground* and noted $T(\Sigma)$

**Definition 2.** *An* atom *is of the form $t = s$, where $t$ and $s$ are two terms. An atom is said to be* non-equational *iff it is of the form $t = true$ or $true = s$,* equational *otherwise (i.e. if it is of the form $t = s$ where $t$ and $s$ are not syntactically identical to true).*

*Remark 2.* In this paper, we only consider atoms of the form $t = s$. No predicate symbols other can "=" are allowed. It is well known that this restriction does not entail any loss of generality, because any atom of the form $P(t_1, \ldots, t_n)$ where $P$ is a predicate symbol may be replaced by an equational atom of the form $p(t_1, \ldots, t_n) = true$, where *true* is a special term and p is a function symbol of arity $n$.

**Definition 3.** *A* literal *is either an atom (*positive *literal) or the negation of an atom (*negative literal*). A literal is said to be equational iff the corresponding atom is equational, non-equational otherwise.*

*A* clause *is a finite set of literals (interpreted as a disjunction). A clause of the form* $\{\neg L_1, \ldots, \neg L_n, L'_1, \ldots, L'_m\}$ *where* $L_1, \ldots, L_n, L'_1, \ldots, L'_m$ *are atoms is often denoted (following the usual sequent-like notation) as a rule:* $L_1 \wedge \ldots \wedge L_n \rightarrow L'_1 \vee \ldots \vee L'_m$.

We also require that each of the clauses be *range-restricted* i.e. all the variables occurring in the positive *or* equational literal of a clause $C$ must also occur in a negative non-equational literal in $C$ (this is an generalization of the standard notion of range-restricted clauses).

Range-restricted clauses are expressive enough to state all the geometric axioms and theorems that we have to deal with in practice. Moreover it is well known that any set of clauses can be transformed into an equivalent set of range-restricted clauses (provided some new predicate symbols — the so called *domain predicates* — are added to the signature).

More formally:

**Notation.** Let $C$ be a clause. We denote by $Var(C)$ the set of variables occurring in $C$.

**Definition 4.** *Let* $C$ *be a clause. We denote by* $C^-$ *(resp.* $C^+$*) the set of negative (resp. positive) literals occurring in* $C$*. We denote by* $C^E$ *the set of equational literals in* $C$ *and by* $C^{NE}$ *the set of non-equational literals in* $C$*.* ${C^-}^E({C^-}^{NE})$ *and* ${C^+}^E({C^+}^{NE})$ *have the obvious meaning.*

*A clause* $C$ *is said to be* positive *(resp.* negative*) iff* $C^+ = C$ *(resp.* $C^- = C$*).*

*A clause* $C$ *is said to be* range-restricted *iff* $Var(C^+) \subseteq Var(C^-)$ *and if* $Var({C^-}^E) \subseteq Var({C^-}^{NE})$*.*

As we shall see, the restriction to range-restricted clauses allows to simplify the proof process especially in conjunction with the use of (positive) hyper-resolution. It is interesting to remark, for example, that any positive range-restricted clause must be *ground* (indeed, if $C$ is positive, then $C^-$ is empty hence $Var(C^-) = \emptyset$ therefore we must have $Var(C^+) = \emptyset$, i.e. $Var(C) = \emptyset$). Moreover, since positive hyper-resolution only produces positive clauses, it implies that only ground clauses will be generated by the application of the hyper-resolution rule.

### 2.2 The Calculus

In this section, we describe the calculus used by HOARD$_{\text{ATINF}}$. It is based on forward chaining (bottom up reasoning), using a variant of the hyper-resolution + positive ordered paramodulation rules.

We introduce a new inference rule, called $E$-hyper-resolution, that combines hyper-resolution with equality reasoning and rewriting techniques. $E$-hyper-resolution can be seen as a *macro inference rule* combining in a single inference step, several applications of the positive ordered paramodulation and positive resolution rules (as defined for example in [17]). This rule has been designed in order to deal with range-restricted problems, especially if the equational part is not very important, relatively to the non-equational part (which is the case for many of the problem we have to treat).

Let us first recall the definition of the (well-known) ordered paramodulation rule. As usual $L[t]_p$ denotes a literal obtained from $L$ by replacing the term at position $p$ by $t$. We assume an order (noted $<$) defined on (ground) terms and (ground) literals.

**Positive Ordered (PO-) Paramodulation:**

$$\frac{\{t = s\} \cup R \qquad\qquad \{L[t]_p\} \cup R'}{\{L[s]_p\} \cup R \cup R'}$$

where $R, R', L[t]_p$ are positive, $t = s > R$, $L[t]_p > R'$ and $s < t$.

*Remark 3.* Notice that unlike the standard definitions (see for example [32]), this rule is applied *only* on positive clauses, hence (since the clauses are range-restricted, see Definition 4) only on ground clauses. Therefore, *no unification is needed* as in the general definition of paramodulation.

In order to define the $E$-hyper-resolution rule, we must first define the following *EC-unification* procedure. It performs a (restricted) form of (conditional) $E$-unification. Given a set of clauses $S$ and two terms $t, s$, it tries to compute a clause $C$ and a substitution $\theta$ such that $S \models (\neg C \rightarrow t\theta = s)$. $\neg C$ can be seen as a *condition* which is *sufficient* to prove that $S \models t\theta = s$. Not all solutions are computed (this would be obviously not possible in general) but the given set of solutions will be still sufficient to ensure the refutational completeness of the whole calculus. The *EC-unification* procedure is given in Figure 1.

*Remark 4.* The tests "$\neg E, \mathcal{C}$ is unsatisfiable" and "$\neg E, \neg R, t = f(s_1, \ldots, s_n), \mathcal{C}$ is satisfiable" are propositional satisfaisability tests.

Since the *EC-unification* procedure is nondeterministic, we must specify how this nondeterminism is handled. Actually, the choice of the pair $(t, s)$ and that of the application of $t', t'', R$ can be done *arbitrarily*, i.e. using a "don't care" nondeterminism. *No backtracking is needed.* Some heuristic are used to prune the search space (for example, equations with the smallest number of variables, or equations that are not likely to be rewrited, are chosen first). In contrast, the choice in line 25 must be done using a "don't know" nondeterminism, which requires backtracking and will therefore provide a set of solutions rather than a unique solution.

Remark that the instruction in line 34 corresponds exactly to the usual decomposition rule of the unification algorithm, i.e.:

$$f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n) \rightarrow \bigwedge_{i=1}^{n} t_i = t'_i$$

```
**Procedure** *EC-unification*
**INPUT** :
  A set of clauses $S$
  A set of equations $S'$
**OUTPUT** :
  A substitution $\theta$ of $Var(t)$ and a positive ground clause $E$
  or a flag "no solution".
**Begin**
  $P := S'$
  $\theta := \emptyset$
  $E := \bot$
  $\mathcal{C} := \emptyset$ % $\mathcal{C}$ is a set of conditions, initially empty
    % Remark: the purpose of $\mathcal{C}$ is only to reduce search space by avoiding
    % irrelevant computations
  **While** $P \not\models \emptyset$
    **If** $\neg E, \mathcal{C}$ is unsatisfiable **Then Return**(**"no solution"**)
    **Choose** $(t = s) \in P$
    **If** $t$ is a variable **Then** $P := P \setminus (t = s), \theta := \theta \cup \{t \rightarrow s\}, P := P\theta, \mathcal{C} := \mathcal{C}\theta$
      % Replacement rule
    **Else If** $s$ is a variable **Then** $P := P \setminus (t = s), \theta := \theta \cup \{s \rightarrow t\}, P := P\theta, \mathcal{C} := \mathcal{C}\theta$
      % Replacement rule
    **Else** % $t$ is of the form $f(t_1, \ldots, t_n)$
    **If** $\exists f(s_1, \ldots, s_n) = t'' \cup R \in S$ such that $f(s_1, \ldots, t_n) = t'' > R, f(s_1, \ldots, s_n) > t''$
           and $\neg E, \neg R, t = f(s_1, \ldots, s_n), \mathcal{C}$ is satisfiable
    **Then** % Conditional narrowing rule
      **Or Begin**
        $P := P \setminus (t = s) \cup \{t'' = s\} \cup \bigcup_{i=1}^{n} \{t_i = s_i\}$
        $E := E \vee R$
        $\mathcal{C} := \mathcal{C} \cup \{f(s_1, \ldots, s_n) = t\}$
      **End**
      **Or**
        $\mathcal{C} := \mathcal{C} \cup \{R \vee \bigvee_{i=1}^{n} s_i \not\models t_i\}$
    **Else If** $t = f(t_1, \ldots, t_n)$ and $s = f(t'_1, \ldots, t'_n)$
    **Then** $P := P \setminus (t = s) \cup \{t_i = t'_i \mid i \in [1..n]\}$ % Decomposition rule
    **Else If** $t = f(t_1, \ldots, t_n)$ and $s = g(t'_1, \ldots, t'_m)$ and $f \not\models g$
    **Then Return**(**"no solution"**) % Clash rule
  **EndWhile**
  **Return**$(\theta, E)$
**End**
```

**Fig. 1.** The *EC-unification* procedure

whereas the instruction in lines 18–20 and 35 corresponds respectively to the replacement and clash rules.

Finally, the instructions in lines 23–30 correspond to an application of a conditional narrowing rule, i.e. a term $t$ is transformed into a term $t''$ if there exists a clause $t' = t'' \vee R$ and a substitution $\theta$ such that $t\theta = t'$ **and** provided that the condition $R$ is added to the resolvent.

Now, we have all what we need to define the $E$-hyper-resolution rule.

$$\frac{\bigvee_{i=1}^{n} t_i = s_i \to C \qquad\qquad S}{C\theta \cup E}$$

where $(\theta, E) := EC\text{-}unification(S, \{t_1 = x_1, s_1 = x_1, \ldots, t_n = x_n, s_n = x_n\})$.

The following example illustrates the use of the $E$-hyper-resolution rule.

*Example 1.* We consider the following set of clauses:

1 $\{\neg(p(f(x)) = true), r(x) = true\}$
2 $\{f(a) = b\} \cup R$
3 $p(b) = true$

Let us apply the $E$-hyper-resolution on clause 1. First, the *EC-unification* procedure is called on the equation $p(f(x)) = true$. Here the conditional narrowing rule can be applied on $p(f(x))$ using clause 3. We obtain two distinct problems:

- $\{true = true, f(x) = b\}$;
- or $\{p(f(x)) = true\}$ with the condition $f(x) \not= b$.

The second problem can be deleted, since narrowing cannot be applied anymore at root position in the term $p(f(x))$, and since $p(f(x))$ is not unifiable with *true* (lines 23 and 35 of the *EC-unification* procedure). Hence, it only remains to solve the first problem. The first equation is trivial and can be deleted immediately (a particular case of the decomposition rule, lines 33–34) . For solving the equation $f(x) = b$, we can again apply the narrowing rule on $f(x) = b$ and clause 2. We obtain the following problem (the other branch can be immediately deleted since $f(x)$ is not unifiable with $b$):

$$\{b = b, x = a\}, \text{ with the condition } \{R\}.$$

$b = b$ is valid, and $x = a$ can be solved immediately by replacing $x$ by $a$ (replacement rule, lines 18–20), hence we obtain the following solution: $\{x \to a\}$, with the condition $R$.

Therefore, the $E$-hyper-resolution rule can be applied and gives the clause:

$$r(a) \vee R$$

Notice that this clause may have been generated by 2 steps of paramodulation into clause 1, followed by one step of hyper-resolution between the obtained clause and the reflexivity axiom $x = x$ (needed when paramodulation is used). The $E$-hyper-resolution allows to merge these 3 steps in a single rule. More important, it also allows to restrict the search space: if we replace for example the clause $p(b) = \mathit{true}$ by $p(f(a)) = \mathit{true}$, then the $E$-hyper-resolution would *not* be applicable any more. Indeed, the reader can easily check that the instance $p(f(a))$ of $p(f(x))$ would still be rewritten as $p(b)$ hence would *not* be unifiable with $p(f(a))$. This does not threaten refutational completeness, since $p(b) \vee R$ may actually be generated from $p(f(a))$ and $f(a) = b \vee R$ by using PO-paramodulation.

**Theorem 1.** *The calculus defined by the E-hyper-resolution (with the EC-unification procedure) and positive ordered paramodulation is* sound *and* refutationally complete*, i.e. for all sets of clauses S, S is unsatisfiable iff there exists a refutation from S using only the E-hyper-resolution and positive ordered paramodulation rules.*

*Proof.* **Soundness.** The soundness of the positive ordered paramodulation rule is well known. We only have to prove that the $E$-hyper-resolution is sound. To this purpose, it suffices to show that for all set of equations $P = \{t_1 = x_1, s_1 = x_1, \ldots, t_n = x_n, s_n = x_n\}$, if $\mathit{EC\text{-}unification}(S, P) = (\theta, E)$ then we have $S, \neg E \models P\theta$. Indeed, in this case, the clause $C\theta \cup E$ must be a logical consequence of the clause $S \cup \{t_1 = s_1, \ldots, t_n = s_n \rightarrow C\}$.

We denote by $P_n$ the set of equations obtained at set $n$, by $E_n$ the set of atoms, by $\mathcal{C}_n$ the corresponding set of conditions, and by $\theta_n$ the corresponding substitutions. We show by induction on the number of iterations, that at each step, and for any model $\mathcal{I}$ of $S \cup \neg E_n$, we must have $\mathcal{I} \models S'\theta_n$ if $\mathcal{I} \models P\theta_n$. When the procedure terminates, we must have $P \equiv \top$ hence $\mathcal{I}$ must be a model of $S'\theta_n$.

- *Base case.* The proof is immediate, since $P_0 \equiv S'$.
- *Inductive case.* Assume that the property holds at step $n$. We show that it holds at step $n + 1$. It suffices to show that $S, \neg E_{n+1} \models (P_n\theta_{n+1} \Leftarrow P_{n+1}\theta_{n+1})$. Let $t = s$ be the equation selected by the procedure at step $n + 1$. We distinguish several case, according to the form of $t, s$.
  - If $t$ (resp. $s$) is a variable, then we have, by definition, $\theta_{n+1} = \theta_n \cup \{t \rightarrow s\}$ and $P_{n+1} = P_n \setminus \{t = s\}$. Hence the proof is immediate.
  - If $t \equiv f(t_1, \ldots, t_n)$ and there exists a clause $f(s_1, \ldots, s_n) = t'' \cup R \in S$ such that $f(s_1, \ldots, s_n) = t'' > R$, $f(s_1, \ldots, s_n) > t''$ then we distinguish two cases according to the chosen OR branch.
    1. *Rewriting branch.* We have $\theta_{n+1} = \theta_n$, $P_{n+1} = P_n \setminus \{(t, s)\} \cup \{t'', s\} \cup \bigcup_{i=1}^{n} \{t_i = s_i\}$ and $E_{n+1} = E_n \cup R$. Let $\sigma$ be a ground substitution of $P_{n+1}\theta_{n+1}$ and let $\mathcal{I}$ be a model of $S, \neg E_{n+1}, P_{n+1}\theta_{n+1}$. We must have $\mathcal{I} \models \neg R$ and $\mathcal{I} \models f(s_1, \ldots, s_n) = t'' \cup R$ hence $\mathcal{I} \models f(s_1, \ldots, s_n) = t''$. Moreover, we have $\forall i \in [1..n], \mathcal{I} \models (t_i = s_i)\theta_{n+1}\sigma$. Hence $\mathcal{I} \models f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\theta_{n+1}\sigma$. Therefore $\mathcal{I} \models (t = t'')\theta_{n+1}\sigma$. Since we must have $\mathcal{I} \models (t'' = s)\theta_{n+1}\sigma$, we deduce that $\mathcal{I} \models (t = s)\theta_{n+1}\sigma$ hence that $\mathcal{I} \models P_n\theta_{n+1}\sigma$. Therefore, we have $S, \neg E_{n+1} \models (P_n\theta_{n+1} \Leftarrow P_{n+1}\theta_{n+1})$.

  2. *Non-rewriting branch.* The proof is immediate, since $P_{n+1} = P_n$ and $\theta_{n+1} = \theta_n$.

- If $t = f(t_1, \dots, t_n)$ and $s = f(s_1, \dots, s_n)$. The proof is immediate, by soundness of the decomposition rule.
- If $t = f(t_1, \dots, t_n)$ and $s = g(s_1, \dots, s_m)$. The proof is immediate, since there is no solutions.

**Refutational completeness.** We first have to show that the *EC-unification* procedure terminates. By definition of *EC-unification*, any term occurring in $P_n$ at any step $n$ is either a subterm of a term occurring in $S'$ or a ground term occurring in $S$ (since any positive clause in $S$ is ground). Therefore, the number of distinct clauses of the form $R \vee \bigvee_{i=1}^{n} t_i \not\models s_i$ such that $f(s_1, \dots, s_n) = t'' \vee R \in S$ and $f(t_1, \dots, t_n)$ occurs in $P_n$ is finite. Therefore, we may assume, w.l.o.g. that no new clause is added into $\mathcal{C}$.

Then, we consider the following measure on the set of equation $P_n$:

$$\mathcal{I}(P) = \{(v(t = s), t = s) \mid (t, s) \in P\}$$

where $s(t = s)$ denotes the number of variables in $t = s$. Equations are ordered using the ordering $<$ and $\mathcal{I}(P)$ is ordered using lexicographic and multiset extensions of the orderings on $s(t = s)$ and $t = s$.

Then at each step, it is easy to see that either $\mathcal{I}(P_n)$ decrease strictly. Indeed, either a variable is instanciated and $v$ decreases strictly, or the value of the terms in $t = s$ decreases strictly w.r.t. $<$.

Now, we prove that the $E$-hyper-resolution and ordered positive paramodulation rules are refutationally complete, using the technique introduced in [17], namely the transfinite tree method.

Let $\mathcal{A}$ be the set of ground literals built on the signature $\Sigma, =$. Clearly, $(\mathcal{A}, <)$ is isomorphic to an ordinal $\omega$. For any ordinal $\alpha < \omega$, we denote by $\mathcal{A}_\alpha$ the literal corresponding to $\alpha$.

We build — by transfinite induction — a sequence of partial interpretations $\mathcal{I}_\alpha$ (i.e. of partial functions from $\mathcal{A}$ into $\{\mathit{true}, \mathit{false}\}$) as follows:

– Base case. $\mathcal{I}_0 = \emptyset$.
– Limit ordinal. $\mathcal{I}_\alpha = \bigcup_{\beta<\alpha} \mathcal{I}_\beta$ if $\alpha$ is a limit ordinal.
– Successor case. $\mathcal{I}_{\alpha+1}$ is defined as follows.
  - If $\mathcal{A}_\alpha = (t = t)$ then $\mathcal{I}_{\alpha+1} = \mathcal{I}_\alpha \cup \{(t = t) \to \mathit{true}\}$.
  - If $\mathcal{A}_\alpha = L[t]_p$ and there exists a literal $t = s$ such that $\mathcal{I}_\alpha(t = s) = \mathit{true}$ and $t > s$, then $\mathcal{I}_{\alpha+1}(L[t]_p) = \mathcal{I}_\alpha(L[s]_p)$.
  - If $\mathcal{A}_\alpha = L$ and there exists a positive clause $L' \vee R$ and a ground substitution $\sigma$ such that $L = L'\sigma$ and $\mathcal{I}_\alpha \models \neg R$ then $\mathcal{I}_{\alpha+1}(L) = \mathit{true}$.
  - Else $\mathcal{I}_\alpha(L) = \mathit{false}$.

*Remark 5.* The reader should note that the sequence $(\mathcal{I}_\alpha)_{\alpha<\omega}$ is monotonic i.e. for all $\beta \leq \alpha$ $\mathcal{I}_\alpha$ is an extension of $\mathcal{I}_\beta$.

Assume that $\Box \not\in S$ and that $S$ is irreducible by $E$-hyper-resolution and ordered positive resolution. Then, we are going to show that $\mathcal{I}_\omega \models S$. We denote by $S_g$ be the set of ground instances of clauses in $S$.

Assume that $\mathcal{I}_\omega \not\models S$. Then, there exists a clause $C$ and a ground substitution $\sigma$ such that $\mathcal{I}_\omega \models \neg C\sigma$. W.l.o.g. we assume that $C, \sigma$ are chosen in such a way that $C\sigma$ is the smallest clause (w.r.t. $<$) having this property.

- Assume that $C$ is positive. $C\sigma$ is of the form $L \vee R$ where $\mathcal{I}_\omega \models \neg R$. By construction of $\mathcal{I}_\omega$, there must exists a term $t'$ occurring at position $p$ in $L$ such that $\mathcal{I}_\omega$ validates an equation $t' = s'$ (else $\mathcal{I}_\omega$ would necessarily validate the maximal literal in $C\sigma$ which is impossible). Therefore, there must exist a term $t'$ and a clause $t' = s' \vee R' \in S_g$ such that $\mathcal{I}_\omega \models (t' = s') \vee R'$, $t' > s'$, $(t' = s') > R'$ and $t'$ occurs at a position $p$ in $L$. By irreducibility w.r.t. the ordered positive paramodulation rule, this implies that $S_g$ contains the clause $L[s']_p \vee R \vee R'$. But we have $\mathcal{I}_\omega \not\models L[s']_p \vee R \vee R'$ and $L[s']_p \vee R \vee R' < C\sigma$. Hence this is impossible.
- Therefore, $C$ must contain at least a negative literal. $C$ is of the form $\bigvee_{i=1}^{n} t_i \not= s_i \vee R$ where $R$ is positive, $t_1, s_1, \ldots, t_n, s_n$ are terms, $n \geq 1$.
  We are going to show that the application of the $E$-hyper-resolution rule must yield a clause of the form $R\sigma \cup E$, where $\mathcal{I}_\omega \not\models E$.
  By definition of the $E$-hyper-resolution rule, we have to show that the application of the *EC-unification* procedure on the input $S, \{t_1 = x_1, s_1 = x_1, \ldots, t_n = x_n, s_n = x_n\}$ yields the output $(\sigma', E)$, where $\sigma' = \sigma \cup \{x_i \to t_i\sigma\}$ is an extension of $\sigma$ and $\mathcal{I}_\omega \not\models E$.
  We define the following strategy for the choice of the OR-branch. The first branch is chosen iff $R$ is falsified by $\mathcal{I}_\omega$ and $\mathcal{I}_\omega \models f(t_1, \ldots, t_n)\sigma = f(s_1, \ldots, s_n)$. Else the second branch is chosen.
  Then, by definition, at each step, any atom in $E$ must be false in $\mathcal{I}_\omega$. Moreover, we prove (by induction on $n$) that at each step $n$, we have: $\mathcal{I}_\omega \models \mathcal{C}_n\sigma'$ and $\mathcal{I}_\omega \models P_n\theta_{n+1}\sigma' \Rightarrow P_{n+1}\theta_{n+1}\sigma'$.
  Let $t = s$ be the equation considered at step $n$. We distinguish several cases, according to the form of $t = s$.
  - If $t$ or $s$ is a variable then the proof is immediate.
  - If $t \equiv f(t_1, \ldots, t_n)$ and there exists a clause $f(s_1, \ldots, s_n) = t'' \cup R \in S$ such that $f(s_1, \ldots, s_n) = t'' > R$, $f(s_1, \ldots, s_n) > t''$, $\neg E_n, \neg R, \mathcal{C}_n \not\models \bigwedge_{i=1}^{n} t_i = s_i$. Since $f(s_1, \ldots, s_n) = t'' \vee R$ is positive, it must be validated by $\mathcal{I}_\omega$.
    1. If the first OR-branch is chosen, then this means — by definition on the chosen strategy — that we must have $\mathcal{I}_\omega \models f(s_1, \ldots, s_n) = t''$ (since $\mathcal{I}_\omega \models \neg R$ and $\mathcal{I}_\omega \models f(s_1, \ldots, s_n) = t'' \vee R$) and $\mathcal{I}_\omega \models (t\sigma' = f(s_1, \ldots, s_n))$. Hence if $\mathcal{I}_\omega(t = s)\sigma'$ then $\mathcal{I}_\omega \models (t'' = s) \wedge (\bigwedge_{i=1}^{n} t_i\sigma' = s_i)$. Therefore $\mathcal{I}_\omega \models (P_n\theta_{n+1} \Rightarrow P_{n+1}\theta_{n+1})$.
    2. Else, we must have either $\mathcal{I}_\omega \models R$ or $\mathcal{I}_\omega \models f(s_1, \ldots, s_n) \not= t\sigma'$, hence $\mathcal{I}_\omega \models \mathcal{C}_{n+1}\sigma'$.
  - If $t \equiv f(t_1, \ldots, t_n)$ and $s \equiv f(s_1, \ldots, s_n)$ and we are not in the previous case. Assume that there exists a clause $f(s'_1, \ldots, s'_n) = t'' \cup R \in S$

such that $f(s'_1, \ldots, s'_n) = t'' > R$, $f(s'_1, \ldots, s'_n) > t''$ and $\mathcal{I}_\omega \models f(s'_1, \ldots, s'_n) = t''$ and $\mathcal{I}_\omega \models (t\sigma' = f(s'_1, \ldots, s'_n))$. Then, by definition, we must have $\neg E_n, \neg R, \mathcal{C}_n \models t \not\models f(s'_1, \ldots, s'_n)$. But this is impossible, since we must have $\mathcal{I}_\omega \models \neg E_n, \neg R, \mathcal{C}_n$ and $\mathcal{I}_\omega \models t\sigma' = f(s'_1, \ldots, s'_n)$. Therefore, there is no clause in $S$ satisfying the above property. By definition of $\mathcal{I}_\omega$, this implies that $\mathcal{I}_\omega \models (t = s)\sigma'$ iff $\forall i \in [1..n], \mathcal{I}_\omega \models t_i\sigma' = s_i$. Hence we must have $\mathcal{I}_\omega \models (P_n\theta_{n+1} \Rightarrow P_{n+1}\theta_{n+1})$.

- If $t \equiv f(t_1, \ldots, t_n)$ and $s \equiv g(s_1, \ldots, s_m)$ and if there is no clause in $S$ satisfying the above property. The proof is similar to the previous case.

Therefore, since $\mathcal{I}_\omega \models (\bigcup_{i=1}^{n}(t_i = x_i \wedge s_i = x_i)\sigma'$, we deduce that *EC-unification* must have a solution $(\theta, E)$ and that $\mathcal{I}_\omega \not\models E$. Then, by irreducibility w.r.t. the $E$-hyper-resolution rule, this implies that $R\sigma \cup E \in S$ hence (since $R$ and $E$ are positive and $\mathcal{I}_\alpha \not\models E$) that $\mathcal{I}_\omega \models R\sigma$. Since $\mathcal{I}_\omega \not\models C\sigma$ this is impossible.

Therefore, the $E$-hyper-resolution rule is refutationally complete.

Using range-restricted clauses together with $E$-hyper-resolution and ordered paramodulation has important consequences: in particular, only ground positive clauses may be deduced, hence the efficiency of the usual algorithms for indexing terms and clauses, and checking redundancies can be improved drastically. For example, checking forward and backward subsumption (which is well known as a difficult and costly problem in the general case) becomes very easy: it suffices to compare the two lists of literals. If the clauses are sorted (w.r.t. $<$) before being stored into the database (which can be done in $n \times ln(n)$ where $n$ is the length — i.e. the number of literals — of the clause), subsumption tests may be done in linear time.

$E$-hyper-resolution is especially useful when the number of equational literals is not very important w.r.t. the size of the clause set, which is the case in most of the problems we have to deal with.

*Remark 6.* The fact that the clauses are range-restricted is essential for the termination of the *EC-unification* procedure, hence for the refutational completeness of the method.

### 2.3 Mixing Up Backward and Forward Proof Search

HOARD$_{\text{ATINF}}$ has been enriched with a special feature allowing to combine $E$-hyper-resolution with a form of top-down reasoning (as it is performed for example by Model-Elimination-based theorem provers or by SLD-resolution and by Prolog).

This is done by adding a special kind of (equational) constraints to the clauses. We consider constrained clauses of the form $[C \mid \mathcal{X}]$ where $C$ is a clause (in the standard sense) and $\mathcal{X}$ a conjunction of equations of the form $t = s$. $C$ is the *clausal part* of $[C \mid \mathcal{X}]$ and $\mathcal{X}$ is the *constraint part.*

The PO-paramodulation rule can easily be extended to constrained clauses, as follows.

**Constrained Positive Ordered paramodulation:**

$$\frac{[\{t = s\} \cup R \mid \mathcal{X}] \qquad\qquad [\{L[t]_p\} \cup R' \mid \mathcal{Y}]}{[\{L[s]_p\} \cup R \cup R' \mid \mathcal{X} \wedge \mathcal{Y}]}$$

where $R, R', L[t]_p$ are positive, $t > s$, $t = s > R$, $L[t]_p > R'$, $u[t]_p > v$.

From a semantic point of view, a constrained clause of the form $[C \mid \bigvee_{i=1}^{n} t_i = s_i]$ is equivalent to the clause $\bigvee_{i=1}^{n} t_i \not\models s_i \vee C$. However, the constraint will not be handled in the same way than the clausal part by the *EC-unification* procedure. More precisely, instead of including as usual this condition into $E$, the procedure will include it into the set of considered equations. Therefore, the *EC-unification* procedure will evaluate all the equations belonging to the constraint part of the clause in order to find solutions to these conditions, before deducing the corresponding $E$-hyper-resolvent. This is done recursively, i.e. the evaluation of the constraint may lead to the consideration of other constrained clauses, hence may entail another application of the *EC-unification* procedure.

For this purpose, the condition "**If** $\exists f(s_1, \ldots, s_n) = t'' \cup R \in S$" in the *EC-unification* procedure (line 23) must be replaced by "**If** $\exists [f(s_1, \ldots, s_n) = t'' \cup R \mid \mathcal{X}] \in S$" and the following additional instruction has to be added to the *EC-unification* procedure, just before **EndWhile** (line 37), in order to deal with the constraint part of the clause:

$\mathcal{P} := \mathcal{P} \cup \mathcal{X}$

This technique allows to integrate in a very natural way useful features of backward reasoning (in particular the building of goal-oriented derivations) into the *EC-unification* procedure. This strongly reduces the number of generated clauses, with reasonable computation cost.

A drawback of this technique is that the *EC-unification* procedure may not terminate, hence refutational completeness may be lost. In order to avoid non-termination in some particular cases, the system keeps track of the goals previously considered in order to avoid entering into infinite loops. Obviously, this cannot prevent non-termination in the general case. It is up to the user to choose carefully which equational conditions should be treated by a backward search. Currently, we use it only on particular axioms for which termination is guaranteed. More flexible strategies, including automatic analysis of the clause sets will be considered in the future.

There are two points for which the use of backward reasoning has proven to be particularly useful.

- **Definitions of geometric objects.** Constrained clauses are especially useful for defining objects. For example the definition: "a parallelogram is a quadrilateral $(A, B, C, D)$ such that $(A, B) \parallel (C, D)$ and $(B, C) \parallel (A, D)$" can be translated into the following (non range-restricted!) clause: $[\{\mathit{parallelogram}((A, B, C, D)) = \mathit{true}\} \mid \mathit{parallel}((A, D), (B, C)) = \mathit{true} \wedge \mathit{parallel}((A, B), (C, D)) = \mathit{true}]$ Then, the mechanism described above will enforce the prover to dynamically replace, during proof search, each atom of the form $\mathit{parallelogram}((A, B, C, D)) = \mathit{true}$ by the conjunction *parallel*$((A, D), (B, C)) =$ *true* $\wedge$ *parallel*$((A, B), (C, D)) =$ *true* (this technique can be seen as an extension of demodulation for non equational literals).

- **Prolog call.** A special predicate "prolog_call" has been introduced: it allows to take advantage of all built-in Prolog predicates (for example for arithmetic computations). Unifying $prolog\_call(G)$ and *true* during the unification procedure will simply cause the evaluation of the goal $G$ using Prolog engine. Thus, adding the constraint $prolog\_call(G) = true$ to a clause, will make the system evaluate the goal $G$ *before* applying the $E$-hyper-resolution rule on this clause.

*Remark 7.* The prolog_call predicate is especially useful for including numerical computations into the calculus, or for *combining* $HOARD_{\text{ATINF}}$ *with existing systems* (see Section 6).

### 2.4 Handling Theories

We analyze in this section some theories usually underlying geometric reasoning. In order to efficiently obtain user-oriented proofs they must be incorporated into the proof steps. As expected, experiments have confirmed that symmetry is one of the most important among these theories. These examples show the way of incorporating new theories in HOARD$_{\text{ATINF}}$.

Equational theories such as commutativity (i.e. $[A, B] = [B, A]$ or $(A, B) = (B, A)$) or circularity (i.e. $(A, B, C, D) = (B, C, D, A)$) are encoded in the core of the unification algorithm. More precisely, the decomposition rule is replaced by the following ones, that are specific to some particular functional symbols. The *EC-unification* procedure is modified accordingly:

$$\begin{array}{ll} f(t_1, t_2) = f(t'_1, t'_2) & \rightarrow (t_1 = t'_1 \wedge t_2 = t'_2) \vee (t_2 = t'_1 \vee t_1 = t'_2) \\ & \quad \text{if } f \text{ is commutative} \\ f(t_1, t_2, t_3, t_4) = f(t'_1, t'_2, t'_3, t'_4) & \rightarrow \bigvee_{i=1}^{n} (t_i = t'_1 \wedge t_{i+1} = t'_2 \wedge t_{i+2} = t'_3 \wedge t_{i+3} = t'_4) \\ & \quad \text{if } f \text{ is circular and with } t_j = t_{j-4} \text{ if } j > 4 \end{array}$$

When a clause is generated, the system computes the *minimal* (w.r.t. $<$) representative of its equivalence class according to the above properties (commutativity and circularity), and only this particular representative (rather than the clause itself) is stored into the database (*this is possible since all generated clauses are ground*). This mechanism significantly speeds up redundancy checks (subsumption tests can still be performed in linear time).

Moreover, special deletion rules are also added, in order to remove clauses that do not carry any geometric information, for example clauses containing terms of the form $[t, s]$ or $(t, s)$, where $t \equiv s$ (it can easily be shown that this does not affect refutational completeness).

These techniques allow to significantly reduce the number of clauses conveying exactly the same information.

Moreover, non-equational theories such as parallelism and collinearity are also treated by a special mechanism, based on the use of constrained clauses.

More precisely, a new function symbol "*dir*" (*dir* standing for "direction") of arity 1 is introduced. Its intended meaning is that $dir(d)$ is the equivalence class of the line $d$ w.r.t. the "*parallel*" relation. Therefore, checking whether two lines $(A, B)$ and $(C, D)$ are parallel will be done very easily by checking whether $dir((A, B)) = dir((C, D))$. Checking whether the points $A, B, C$ are collinear is done by checking (for example) that $dir((A, C)) = dir((A, B))$. This avoids adding axioms for the transitivity of the parallelism relation and allows to deal with this theory in a more effective way.

The following clauses are introduced into the clause sets:

$$\begin{array}{l}
[parallel((A,B),(C,D)) = true \mid dir((A,B)) \not= dir((C,D))] \\
[dir((A,C)) = dir(A,B)) \vee dir((A,B)) \not= dir((B,C)) \mid true] \\
[collinear(A,B,C) = true \mid dir((A,B)) \not= dir((A,C))] \\
[dir((A,C)) \not= dir(A,B)) \vee (A,B) = (A,C) \mid true]
\end{array}$$

Another special mechanism, based on the use of ordering constraints, has also been added in order to handle clauses that are symmetric w.r.t. a permutation of the variables. For example, consider the clause:

$$C : perpendicular(x, y) \wedge perpendicular(x, z) \to parallel(y, z)$$

and the permutation $\sigma = \{y \to z, z \to y\}$.

It is easy to see that $\sigma$ is a symmetry for clause $C$, since $C\sigma$ is syntactically equivalent to $C$ *modulo* the commutativity of $\|$ and $\vee$. Thus, given the two clauses $perpendicular(a, b)$ and $perpendicular(a, c)$, two distinct clauses will be generated by applying the $E$-hyper-resolution rule: $parallel(b, c)$ and $parallel(c, b)$.

Obviously, these two clauses are *equivalent* modulo commutativity of $\|$, hence one of these two clauses will be actually *deleted* by the subsumption algorithm. It would be more efficient to *prevent* the generation of such clauses, instead of discarding them afterwards. This is done by introducing special kind of *constraints* into the clauses. These constraints express ordering conditions on the variables, in order to "destroy" the symmetry, thus preventing the generation of equivalent clauses. Here, it suffices to add the constraint $x < y$, ensuring that $x$ must be lower than $y$ according to some (arbitrarily chosen) ordering $<$ in the initial clause. This prevent the generation of $parallel(b, c)$ (if $b > c$) or $parallel(c, b)$ (if $b < c$).

Unification problems are solved modulo these ordering constraints, i.e. the system check that the obtained solutions satisfy the constraints. As in Constraint Logic Programming, the constraints are solved as soon as the value of the variables are known. This allows to decrease strongly the cost of the unification algorithm (since several solutions may be discarded) and the number of deleted clauses without too much additional computation cost.

The addition of such ordering constraints may be done automatically. Indeed, it suffices to add for each clause $C$ and for each pair of variable $(x, y)$ such that $C\{x \leftrightarrow y\} \equiv C$ the constraints $x < y$ (or $y < x$).

As it is well known, adding such special mechanisms devoted to the handling of the considered theories is essential for improving the efficiency of provers. This is of course also the case for $\text{HOARD}_{\text{ATINF}}$. Without them, no proof would be obtained in reasonable time for most of the considered problems.

### 2.5 Axiomatization

Our goal is to produce human-oriented proofs, and, in the framework of this work, students oriented-proofs. Therefore, complete, but *complex* axiomatizations such as Tarski's or Hilbert's, would be useless.

The axiomatization will mainly depend on the *level of knowledge* of students. It will contain *all the geometric axioms* that are supposed to be known at a certain stage. In order to build our system such geometric axioms have been gathered from school textbooks (of the adequate level) and translated into the considered fragment of first-order logic.

Moreover, several axioms have had to be added in order to formalize "default" (or "implicit") knowledge freely used in textbooks, mainly because it is either trivial or common sense. Such axioms includes for example basic properties of geometric objects (i.e. $(A, B) = (B, A)$) or basic definitions (i.e. the definition of the midpoint of a segment) or trivial reasoning steps about geometric object (i.e. if $I \in (A, B)$ and $B \in (A, C)$ then $I \in (A, C)$). This implicit knowledge has been added either as built-in features or formalized as set of first-order clauses.

This axiomatization is obviously not complete w.r.t. Euclidean plane geometry, but completeness is not really relevant for our purpose (but soundness is, of course, an important issue).

The axiomatization that we used for all the examples in this paper contains 141 clauses.

### 2.6 An Example

*Remark 8.* It is worth noticing that despite numerous attempts, using different strategies, no proof of the simple theorems stated in Examples 2, 3 and 4 has been obtained with the well-known generic theorem prover OTTER (of course we have used the same axiomatization). OTTER ran out of memory after some hours of computation having generated thousands of clauses.

*Remark 9.* All the examples treated by $\text{HOARD}_{\text{ATINF}}$ given in this paper run on a Pentium 200 with 64 Mb of memory.

*Example 2 (Problem proposed to 15 years old French high school students).* We consider the following problem (see Figure 2): Let $A, B, C$ be a triangle such that $|AC| = |BC|$. Let $I$ be the middle of $[B, C]$, $J$ be the middle of $[A, C]$ and $K$ the middle of $[A, B]$. Let $E$ and $F$ the middle of $[A, K]$ and $[K, B]$ respectively. Prove that $(E, J, I, F)$ is a rectangle.

$\text{HOARD}_{\text{ATINF}}$ computes a proof automatically in about 20 seconds 217 clauses are generated and 2432 are generated and discarded by forward subsumption.

The proof (in pseudo-French) is given in Annex A. Note that several of the considered steps are simple applications of basic definitions that are trivial for

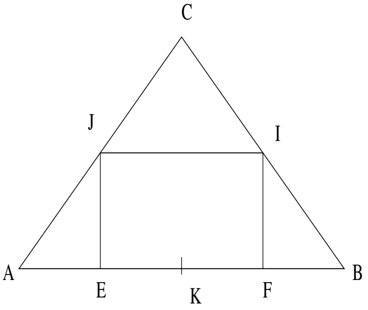


**Fig. 2.** A figure for Example 2

human beeings, and thus omitted in human's proofs (for example $E$ is the midpoints of $(A, K)$ and $(K, E)$ implies that $(A, K) \parallel (K, E)$). Of course, these steps could have been removed by the system as well, but we prefer to make them explicitly for the sake of completeness.

A more careful analysis of the proof also shows that several subparts are highly similar. They correspond to applications of the same sequence of geometric theorems on different points. In order to reduce the size of the proof — thus making it more readable — other techniques are necessary to detect such similarities. We have already developed these techniques, see Section 3.3 (Remark 13) for more details.

*Example 3.* We consider a parallelogram $(A, B, C, D)$ such that $d(A, C) = d(B, D)$. Let $E, F, G, H$ the midpoints of $[A, B]$, $[B, C]$, $[C, D]$, $[D, A]$ respectively. Prove that $(E, F, G, H)$ is a square.

The proof of this theorem generated by HOARD$_{\text{ATINF}}$ is given in Annex B.

The next example (taken from [8]) is interesting because its proof needs the introduction of new auxiliary points. It allow to compare features of HOARD$_{\text{ATINF}}$ and GEX [8] on a running example.

*Example 4 (taken from [8], page 240, originally given in [15]).*

We consider a trapezoid $(A, B, C, D)$ with $(A, B) \parallel (C, D)$. $M, N$ are the midpoints of $[A, C]$ and $[B, D]$ respectively. $E$ is the intersection of $(M, N)$ and $(C, B)$. Prove that $E$ is the midpoint of $[C, B]$.

In order to construct the proof, the midpoint of $[A, D]$ must be constructed. In our formalism, this entails the use of a clause corresponding to a constructive axiom, of the form:

$$\neg point(A) \vee \neg point(B) \vee midpoint(m(A, B), [A, B])$$

This axiom comes from the skolemization of the following formula:

$$\neg point(A) \vee \neg point(B) \vee (\exists M) midpoint(M, [A, B])$$

$m$ is the skolem function introduced from $(\exists M)$.

This technique avoids explicit generation of auxiliary points (as in GEX [8]). See Section 5 for more details about this problem.

However, it is not reasonable to keep such skolem terms in the proof presented to the user, because they will darken the proof. Therefore, HOARD$_{\text{ATINF}}$ eliminates them after the proof has been generated and automatically replaces the introduction of these terms by the explicit construction of new points (using anti-skolemization techniques). The proof it gives is the following (the translation into pseudo-french is too verbose, we do not give it).

Note that non-degeneracy conditions are *proven* here and not assumed from a given figure (see Section 5). This explain the difference in length between the HOARD$_{\text{ATINF}}$'s and GEX's proofs. As previously mentioned for "implicit" axioms, the corresponding parts of the proof need not to be explicitly displayed to the user, which makes the proof shorter and more readable.

```
there_exists(p7,midpoint(p7,segment(D,A)))
$
midpoint(N,segment(D,B))
$
parallel(segment(p7,N),segment(A,B))
midpoint(p7,segment(D,A))
midpoint(N,segment(D,B))
midpoints_theorem
$
non(D = A)
non(collinear(D,A,B))
$
non(p7 = D)
midpoint(p7,segment(D,A))
non(D = A)
$
collinear(N,D,B)
midpoint(N,segment(D,B))
$
collinear(p7,D,A)
midpoint(p7,segment(D,A))
$
non(p7 = N)
non(p7 = D)
collinear(N,D,B)
collinear(p7,D,A)
non(collinear(D,A,B))
$
parallel(segment(p7,N),segment(D,C))
parallel(segment(p7,N),segment(A,B))
parallel(segment(D,C),segment(A,B))
transitivity_parallelism
$
parallel(segment(p7,M),segment(D,C))
```

```
midpoint(p7,segment(D,A))
midpoint(M,segment(A,C))
midpoint_theorem
$
parallel(segment(p7,N),segment(p7,M))
parallel(segment(p7,N),segment(D,C))
parallel(segment(p7,M),segment(D,C))
transitivity_parallelisme
$
collinear(p7,N,M)
parallel(segment(p7,N),segment(p7,M))
$
droite(p7,N) = droite(N,M)
non(N = M)
non(p7 = N)
collinear(p7,N,M)
$
non(D = B)
non(collinear(D,A,B))
$
non(N = B)
midpoint(N,segment(D,B))
non(D = B)
$
non(collinear(D,C,B))
$
non(N = E)
non(N = B)
collinear(C,B,E)
collinear(N,D,B)
non(collinear(D,C,B))
$
droite(N,M) = droite(N,E)
non(N = E)
non(N = M)
collinear(N,M,E)
$
non(A = C)
non(collinear(A,C,B))
$
non(M = C)
midpoint(M,segment(A,C))
non(A = C)
$
collinear(M,A,C)
midpoint(M,segment(A,C))
$
non(M = E)
non(M = C)
collinear(C,B,E)
collinear(M,A,C)
non(collinear(A,C,B))
$
droite(N,M) = droite(M,E)
non(M = E)
non(N = M)
collinear(N,M,E)
$
droite(N,E) = droite(M,E)
droite(N,M) = droite(M,E)
droite(N,M) = droite(N,E)
$
droite(p7,N) = droite(M,E)
droite(p7,N) = droite(N,M)
droite(N,M) = droite(M,E)
$
parallel(segment(M,E),segment(A,B))
parallel(segment(p7,N),segment(A,B))
droite(p7,N) = droite(M,E)
$
midpoint(E,segment(C,B))
collinear(C,B,E)
non(collinear(A,C,B))
parallel(segment(M,E),segment(A,B))
midpoint(M,segment(A,C))
midpoint_theorem
$
```

## 3 Some Additional Features

In this section, we describe some other original features of HOARD$_{\text{ATINF}}$ and we give examples of applications of these features. Some of the ideas behind these capabilities have been proposed in the field of automated deduction since many years ago and some techniques have been presented (Bledsoe, Bundy, . . . ), but the techniques proposed here are original in a large extent and, as far as we know, combined for the first time.

### 3.1 Proof Verification

HOARD$_{\text{ATINF}}$ can be used to check the alleged “proofs” built by the students, in order to detect incorrect (in particular incomplete) “proofs”. However, the system does not merely check the soundness of the proof is the standard sense. Indeed, it is well known that human beings do not in general write “complete” proofs: several steps are usually considered implicit, often because they are trivial, and also because making them explicit would hide the main ideas and reduce proof readability.

Proof check is performed as follows:

- In the tutorial system CABRI-EUCLIDE (see Introduction), quite naturally, each step of the proof must be specified by a partial *conclusion* $C$, a set of *hypothesis* $\mathcal{H}$ and a geometric *theorem* (or definition) $T$.
- In order to check that the step is correct, $\text{HOARD}_{\text{ATINF}}$ tries to compute a proof of $C$ given $\mathcal{H}$, using both the set of axioms corresponding to $T$ *and* the "default" ("implicit") axioms i.e. the theory that are "well known" in the state of the student's knowledge (see Section 2.5).
- If a proof is found, then the proof step is validated. Otherwise, further analysis may be performed in order to find an *explanation* of why the proof is not correct. For example:
  - Checking whether other theorems can be used for deriving the desired conclusion.
  - Try to identify missing hypotheses that could be used to prove $C$.
  - Construct a partial counter-example if the step is incorrect.

*Example 5.* We consider the following proof step (taken from a student's "proof" built using CABRI-Euclide [21]).

"$(A, B, C, D)$ is a parallelogram, $I$ is the midpoint of $[B, D]$. Hence $I$ is the midpoint of $[A, C]$ (since the diagonals of a parallelogram intersect in their middles)".

Here the assertion "$I$ is the midpoint of $[A, C]$" cannot be deduced from the hypothesis. The proof step is not correct (sound) since the conclusion is not a logical consequence of the premises.

The system tries to computes (i.e. to *abduce*) a set of *additional* hypothesis allowing to prove the desired result. This is done by backward chaining, starting from the conclusion and trying to compute a sufficient set of hypothesis (w.r.t. a given set of axioms). Here it detects that the hypothesis $\neg collinear(\{A, B, C, D\})$ (i.e. a nondegeneracy condition) is sufficient to complete the proof. Depending on the context, the system could accept the proof step as it is, point out the missing hypothesis to the user, check whether the missing hypothesis is valid or not, etc.

*Remark 10.* $\neg collinear(\{A, B, C, D\})$ is an example of hypothesis which is often omitted in human proofs.

### 3.2 Model Building

When the validity (respectively contradiction) of a formula cannot be proved, it is very useful to get a counter-example (respectively model) of it, in order to provide a *convincing evidence* of the non-validity (non-contradiction) of the formula.

In this section, we show how to extract models from clause sets that are saturated (i.e. no new clauses can be derived) under $E$-hyper-resolution. This technique can be seen as a generalization of the method proposed in [23,22] for a particular class of clauses.

```
**Procedure** *ModelBuilding*
**INPUT** A set of clauses $S$
**OUTPUT** A set of ground equations $E$ such that $E \models S$
**Begin**
  **Let** $E := \emptyset$
  **Let** $S_E = \{C \in S \mid C \equiv \bigvee_{i=1}^{n} t_i = s_i\}$
  **While** $S_E \not\models \emptyset$
  **Begin**
    **Let** $C := \min(S_E)$ % w.r.t. the multiset extension of the order $<$
    **Let** $L := \max(C)$ % w.r.t. the order $<$ on literals
    $E := E \cup \{L\}$
    $S_E = \{C \in S \mid C \equiv \bigvee_{i=1}^{n} t_i = s_i, \forall i \in [1..n] t_i \not\models_E s_i\}$
  **End**
**End**
```

**Theorem 2.** *Let $S$ be a set of clauses saturated under E-hyper-resolution and ordered paramodulation together with standard deletion rule: (conditional) demodulation and subsumption. Then ModelBuilding$(S)$ terminates. Moreover, ModelBuilding$(S)$ is a model of $S$.*

The models constructed by the *ModelBuilding* procedure are Herbrand models (i.e. defined on the set of ground terms). Obviously, it is more useful — in the particular context of geometry — to construct *diagrams* as models. Hence, we have added a special mechanism to transform automatically Herbrand models into geometric diagrams. This entails numerical solving of sets of equations and inequations, since we have to find the value of the coordinates such that all the assertions in the Herbrand model holds (i.e. we must find a solution of a system of equations and inequations). To this purpose, we use MAPLE[2], together with specific heuristics devoted to numerical solving of inequations. The pre-generation of Herbrand model (or at least of a partial Herbrand model) is useful (and often crucial) to guide the numerical solver.

*Example 6.* Assume we want to build a counter-example for the wrong "proof" of Example 5. We have to find a model satisfying "$(A, B, C, D)$ is a parallelogram", "$I$ is the midpoint of $[B, D]$" but not "$I$ is the midpoint of $[A, C]$".

During the generation of the Herbrand model, we deduce in particular the assertion: "*collinear*$(\{A, B, C, D\})$" which is a logical consequence of the set of hypothesis. Then, all these assertions are translated into sets of equations and inequations. We obtain:

1 $X_I = (X_B + X_D)/2$ i.e. $I$ is the midpoint of $[B, D]$
2 $Y_I = (Y_B + Y_D)/2$
3 $X_A - X_B = X_D - X_C$ i.e. $A, B, C, D$ is a parallelogram
4 $Y_A - Y_B = Y_D - Y_C$
5 $X_I \not\models (X_A + X_C)/2$ i.e. $I$ is not the midpoint of $[A, C]$

[2] HOARD$_{\text{ATINF}}$ has not yet integrated a symbolic computation system, MAPLE is used here in an ad hoc manner.

6 $Y_I \neq (Y_A + Y_C)/2$
7 $(X_A - X_B) \times (Y_A - Y_C) = (X_A - X_C) \times (Y_A - Y_B)$ i.e. *collinear*$(\{A, B, C\})$
8 $(X_A - X_B) \times (Y_A - Y_D) = (X_A - X_D) \times (Y_A - Y_B)$ i.e. *collinear*$(\{A, B, D\})$

We use MAPLE to solve the system $\{1, 2, 3, 4, 7, 8\}$. We obtain a parameterized solution of the system, that can be instantiated by arbitrary values in order to find the coordinates corresponding to a diagram which is the graphical realization of the obtained Herbrand model. Notice that this would not have been possible if the assertion "*collinear*$(\{A, B, C, D\})$" had not been added into the set of hypothesis. Indeed, $1, 2, 3, 4$ do not carry enough information to find a solution satisfying $5, 6$.

### 3.3 Proof Generalization and Analogical Reasoning

The analysis of the concepts of analogy and reasoning by analogy has a very long tradition in western Philosophy and Mathematics. Analogy seems to be one of the most important (informal) techniques used by humans, particularly when dealing with mathematical problems (see e.g. [25]). Consequently, application of analogy to the field of automated deduction seems quite natural. However work in this direction is surprisingly scarce albeit the ignorance of analogy by theorem provers was early identified as a major drawback for their performances [2] and reasoning by analogy is considered as a challenge in automated deduction in the Wos' well-known list [35]. Of course, a good use of analogy is still a very important issue in automated deduction.

In the present work, we focus our interest on the *intrinsic* value of analogical reasoning for a better understanding of proofs, rather than in its uses to improve provers performances.

The main goals are to be able to verify (possibly proposing some changes) the analogies proposed by students and to suggest proofs (or segment of proofs) analogous to those proposed by them. Analogy detection is especially important for the presentation and structuring of proofs: detecting analogies between parts of a proof is a way of introducing lemmata, thus making proofs shorter and more readable. Using analogy a priori can be considered as a form of *planning*.

The analogy reasoning performed by $\text{HOARD}_{\text{ATINF}}$ is mainly based on the method developed in [11] for detecting and using analogies in resolution proofs. In this section, we briefly recall the basis of this approach, and we explain how it has been adapted to geometric reasoning.

The method relies on generalization techniques. It can be divided into two steps:

1. **Generalization step**. It occurs just after proving a (new) formula in a theory (i.e. a theorem). The formula is transformed into a *more general formula schema*. Predicate and function symbols are replaced by higher-order variables, and specific generalization rules are applied, in order to transform the considered formula into a more general one, while preserving the proof. The idea is to try to find a formula which is *more general* than the original one but that admit a "similar" (i.e. with the same structure)

proof. In some sense, the generalization algorithm infers, from a given proof, information useful for a larger class of problems.

2. **Matching step**. Then the system tries to compare the problem at hand to the previously generalized formulae in order to detect potential similarities. If such similarities are only partial, then the system tries to infer *lemmata* that have to be proven in order to "complete" the analogy. The calculus for solving such matching problems has been presented in [13,12]. It is based on higher-order unification techniques.

We refer to [11,13,12] for a detailed presentation of this approach but the description we give here should enable the reader to understand how it works. The main modifications that have been introduced into the algorithms in order to deal more easily with geometry are the following:

- First, it is clear that some of the function and predicate symbols are part of the language, hence *should not be generalized* at all (i.e. should not be replaced by higher-order variables). For example, it would not make sense, from a geometric point of view, to detect analogies by replacing the predicate "*parallel*" by the predicate "*perpendicular*" in a formula. Therefore, the application of the generalization rules (step 1) (see also [11]) must be carefully controlled. Only function symbols without any intended meaning should be generalized. Though this restricted generalization reduces the number of potential similarities, it also has the big advantage to strengthen the constraints on the matching problem (step 2), thus making the matching process much more easy.
- Second, the matching algorithm should be performed modulo the particular theories of geometry predicates (commutativity, circularity and parallelism) as explained in Section 2.4.

As a consequence of these changes in the algorithms, the next example (as well as the others in our experimentations) only uses a very weak version of the method presented in [11,13,12].

*Remark 11.* Of course the full version of the algorithm could be useful, for example for detecting similarities between proofs in different geometries or in different axiomatizations of the same geometry. Indeed, abstracting the predicate and function symbols of the considered axiomatization could be useful for studying properties of these axiomatization, as well as properties of the considered geometric problems.

*Example 7.* We consider the following source problem $(P)$. "Let $A, B, C, D$ be a quadrilateral. Let $I, J, K, L$ be the midpoint of $[A, B], [B, C], [C, D], [D, A]$ respectively. Prove that $(I, J, K, L)$ is a parallelogram."

This problem is easy and well known. The proof built by HOARD$_{\text{ATINF}}$ is the following:

```
midpoint(I,segment(A,B))
$
midpoint(J,segment(B,C))
$
midpoint(K,segment(C,D))
$
midpoint(L,segment(A,D))
$
parallel(segment(I,J),segment(A,C))
midpoint(I,segment(A,B))
midpoint(J,segment(B,C))
midpoints_theorem
$
parallel(segment(L,K),segment(A,C))
midpoint(L,segment(A,D))
midpoint(K,segment(C,D))
midpoints_theorem
$
parallel(segment(I,J),segment(L,K))
parallel(segment(I,J),segment(A,C))
parallel(segment(L,K),segment(A,C))
transitivity_parallelism
$
parallel(segment(L,I),segment(B,D))
midpoint(I,segment(A,B))
midpoint(L,segment(A,D))
midpoints_theorem
$
parallel(segment(J,K),segment(B,D))
midpoint(J,segment(B,C))
midpoint(K,segment(C,D))
midpoints_theorem
$
parallel(segment(I,L),segment(J,K))
parallel(segment(I,L),segment(B,D))
parallel(segment(J,K),segment(B,D))
transitivity_parallelism
$
parallelogramme(I,J,K,L)
parallel(segment(I,J),segment(L,K))
parallel(segment(L,I),segment(J,K))
parallelogram
$
```

Then, we consider a new problem, noted $(P')$. "Let $A, B, C$ be a triangle. Let $P$ be a point. We construct the points $P_1, P_2, P_3$ as the images of $P, P1, P2$ respectively, by the symmetry w.r.t. the points $A, B$, and $C$ respectively. Let $I$ be the midpoint of $[P, P3]$. Prove that $(A, B, C, I)$ is a parallelogramm."

The interesting point here is that from a human point of view $(P')$ is strongly similar (analogous) to $(P)$. We use the system HOARD$_{\text{ATINF}}$ in order to detect the analogy and reconstruct the proof of $(P')$ from the one of $(P)$.

In a first step the proof of $(P)$ is generalized. Then the system tries to match the new problem $(P')$ with $(P)$. In this case the matching does not succeed, since $(P')$ is not a mere instance of $(P)$. But a *partial matching* is possible. This matching can be transformed into a total one, modulo a set of *lemmata* corresponding to hypotheses of $(P')$ that can not be matched to those of $(P)$.

We obtain the following list of hypotheses:

```
midpoint(A,segment[X1,X2]).
midpoint(B,segment[X2,X3]).
midpoint(C,segment[X3,X4]).
midpoint(I,segment[X4,X1]).
```

It should be noted that $X1, X2, \ldots, X4$ are *variables.* In order to reconstruct the proof, we have to find values of $X1, \ldots, X4$ such that the properties above hold.

The last hypothesis can be matched with `midpoint(I,segment(P3,P)`, which leads to:

```
midpoint(A,segment[P,X2]).
midpoint(B,segment[X2,X3]).
midpoint(C,segment[X3,P3]).
```

The remaining hypotheses cannot be matched. However, they can easily be proven from the original one, i.e. we have:

```
midpoint(A,segment[P,P1]).
midpoint(B,segment[P1,P2]).
midpoint(C,segment[P2,P3]).
```

where the variables $X2, X3$ have been instanciated. Thus, we obtain the following proof:

```
[partial] % the analogy is only partial
parallelogramm(A,B,C,I)
parallel(segment(A,B),segment(I,C))
parallel(segment(I,A),segment(B,C))
$
parallel(segment(A,B),segment(I,C))
parallel(segment(A,B),segment(P,P2))
parallel(segment(I,C),segment(P,P2))
$
parallel(segment(A,B),segment(P,P2))
midpoint(A,segment(P,P1))
midpoint(B,segment(P1,P2))
$
parallel(segment(I,C),segment(P,P2))
midpoint(I,segment(P,P3))
midpoint(C,segment(P2,P3))
```

```
$
midpoint(I,segment(P,P3))
$
parallel(segment(I,A),segment(B,C))
parallel(segment(A,I),segment(P1,P3))
parallel(segment(B,C),segment(P1,P3))
$
parallel(segment(A,I),segment(P1,P3))
midpoint(A,segment(P,P1))
midpoint(I,segment(P,P3))
$
parallel(segment(B,C),segment(P1,P3))
midpoint(B,segment(P1,P2))
midpoint(C,segment(P2,P3))
$
midpoint(A,segment(P,P1))
symetry(P,P1,A)
$
midpoint(B,segment(P1,P2))
symetry(P1,P2,B)
$
midpoint(C,segment(P2,P3))
symetry(P2,P3,C)
$
```

*Remark 12.* All these steps — generalization of the original problem, matching, instanciation of the variables, proof of the remaining lemmata and reconstruction of the whole proof — are fully automated. Computation time is rather short (less than 1s). The interested reader may refer to [11,13,12] for a detailed presentation of the algorithms.

*Remark 13.* Analogy reasoning is also very useful for *structuring proofs*. For example in the proof of the Theorem given in Annex A, due to the obvious symmetry in the problem, the proof of $(E, J) \parallel (C, K)$ and $(F, I) \parallel (C, K)$ are analogous. Using our proof generalisation and matching algorithms in a systematic way on these subproofs could allow to detect such similarities and avoid redundancies in proofs. Appropriate lemmata (obtained from the generalized subproof) could be automatically introduced in the proof. Obviously, a proof presented using lemmata is much more easy to understand and to interact with.

## 4 Reasoning from Diagrams: An Example

Using models and/or counter-examples for guiding proof search is a very natural idea that have been considered since the very beginning of Automated Deduction [15]. It is especially useful in geometry where using diagrams to guide proofs (or refutations) is a common attitude among humans.

A model can be used to check the validity of a given assertion in a particular context. Hence, it is often used to discard goals that *cannot* be proven, because they are false in at least one model. This is especially useful for *backward reasoning*, since it allows to prune the search space by deleting numerous subgoals that cannot be true in all models of the hypotheses, therefore cannot be a logical consequence of them (see for example [34]).

Models (interpretations) can be also useful in *forward reasoning.* When looking for contradiction of set of formulas, if the set can be partitioned in formulas satisfied by the interpretation and formulas falsified in it, it is useless to deduce exclusively from formulas satisfied in the interpretation. This is the principle of the two most popular resolution strategies: support and semantic resolution.

The set of support strategy [37] was born almost simultaneously with resolution. The set of clauses is partitioned into a satisfiable one and a non-satisfiable one (a simple theorem ensures the existence of such interpretations for unsatisfiable set of clauses). Specification of the interpretation is not required. The more successful approach for using semantic information in the context of resolution theorem proving is surely *semantic resolution*, introduced by Slagle in [33,6]. Semantic resolution corresponds roughly, as Slagle pointed out, to the heuristic used in the **G**eometric **T**heorem-proving **M**achine [15] to prune the proof search in plane geometry. It can be seen as an extension of the set of support strategy. The principle of semantic resolution is again to discard some clauses that *cannot* possibly lead to a refutation. This is done — roughly speaking — by preventing the application of the resolution rule between two clauses that are true in the considered interpretation.

Therefore, in all semantic-based approaches, models play the role of a *filter*, and are used to cut some of the branches in the proof search. As such, this was proved to be a powerful and elegant way of guiding proof search and strongly improved the performances of theorem provers (see for example [34] for some striking examples). Macro inference rules such as positive or negative hyperresolution may be seen as particular cases of semantic resolution.

In [30] models are used in two different ways: to prune subgoals that are false in the model (as in [15]) and to *suggest inferences.* This second use is closer to what we do in our approach. Reiter also uses the state of the proof to suggest changes in the model (e.g. a *construction* in geometry).

An interesting recent work that deserves to be mentioned is [24]: "on Horn clause OSHL (the prover written by Plaisted and Zhu) behaves a lot like the geometry theorem proving prover of Gelernter et al. [15] which used geometric diagrams and backward chaining with Horn clause-like inference rules to guide its search."

Here we propose to go one step further: instead of using models to *discard* information (or to suggest inferences as in [30]), we propose to use them *a-priori* for *suggesting* possible lemmata, in order to *structure* or to *plan* proofs.

Given a set of hypothesis $\mathcal{H}$, a goal $G$ (the formula to prove) and a particular diagram $D$ (which is a model of $\mathcal{H}$), we try to find one (or several) logical relations $L$ that are satisfied in $D$. Then these relations can be seen as lemmata, thus suggesting the following (generic) *proof plan*:

- Prove that $\mathcal{H}$ models $L$.
- Prove that $\mathcal{H} \cup \{L\}$ implies $G$.

This technique may be seen as a way of adding controlled version of the (analytic or non-analytic) "cut" rule into the resolution calculus. It is well known that clever application of the cut rule frequently amounts to significantly prune the proof search and to decrease the size of the obtained proof (see for example [19]). On the other hand, identifying "interesting" lemmata for application of this rule is very difficult, since unrestricted application would be obviously impossible in practice, due to the huge number of potential candidates. Finding "good" lemmata often requires advanced knowledge about the considered domain, and is at the very heart of the mathematical activity.

We consider that the use of diagrams may be a good way to control the application of the cut rule, in a realistic and useful way. The following is an example of application of this technique.

*Example 8 (Exercise proposed to 16 years old French high school students).* We consider the following problem.

*"Let $(A, B, C)$ be a triangle. Let $M$ be the midpoint of $[A, B]$, $G_1$ be the midpoint of $[M, B]$. Let $G_3, G_2$ be two points on $(A, C)$ such that $\boldsymbol{CG_2} = \boldsymbol{G_2G_3} = \boldsymbol{G_3A}$. Let $G$ be a point such that $\boldsymbol{CG} = 2/3 \times \boldsymbol{CG_1}$ and let $I$ be the midpoint of $[C, M]$. Prove that $G$ belongs to $(B, G_2)$."*

It should be clear that the function of models here is different from that mentioned in Section 3.2. In Section 3.2. model building must be been as a way to exhibit counterexamples of the initial problem (in refutational approaches formula are *negated*) after a failed search. Here, models will be built *before* the search starts and will be used to *guide* it. Therefore in this case, the pre-generation of the Herbrand model is not performed, The system starts by generating automatically a diagram for this problem (i.e. a model of the hypotheses) using the technique presented in Section 3.2, i.e. the generation of algebraic conditions corresponding to a logical axiomatization.

Then, it tries to generate lemmata by detecting assertions that are true on the figure. This is done as follows.

- First, choose a (finite) set of functional symbols and predicate symbols. Here only the elementary predicate symbols `parallel` and `perpendicular` and the constructor "$x, y \to (x, y)$" (the line between two points $x$ and $y$) are chosen.
- Then, enumerate all possible atomic assertions (up to a maximal depth) that can be constructed on the chosen signature. Assertions that are already known or that are equivalent to the considered goal (e.g. $(B, G) \parallel (B, G_2)$) should be eliminated immediately.
- Check (using numerical computation) which assertions are true or false in the model.

Here it generates (among others) the following properties:

$$\begin{gathered} \mathit{parallel}((B, G_2), (M, G_3)) \\ \mathit{parallel}((G_2, I), (M, G_3)) \\ \mathit{parallel}((B, G), (B, I)) \end{gathered}$$

*Remark 14.* The construction of the diagram and the generation of the lemmata can be done automatically in almost negligible computation time on this example (HOARD$_{\text{ATINF}}$ used less than 1s).

The generation of these lemmata suggests the following proof plan:

1. Try to prove the property $G \in (B, G_2)$ (i.e. theorem's conclusion) using *all available lemmata.*
2. Then, try to prove from $\mathcal{H}$ the lemmata that *were actually used* in the proof generated during Step 1.
3. Finally reconstruct the proof of $G \in (B, G_2)$ using the proof built in steps 1 and 2.

The first step is actually very easy. It can be done in 4.7 s. 95 clauses are generated. The second step is more difficult. The system runs about 50s, and generates 1286 clauses (126 are kept, others are deleted by backward or forward subsumption). The third step is trivial and can actually be neglected (less than 1s).

A rough comparison can be done: proving the initial assertion without using the lemmata takes 80s and entails the generation of 2386 clauses.

## 5 Related Work

As already mentioned in Section 1 there are a few logic-based running systems for geometry theorem proving. Two important and representative among them (an "old" one and a very recent one) are GEOM [10] and Geometry Expert (GEX) [8].

GEOM aims at a better better understanding of using Prolog in geometry theorem proving. Many of the points investigated in [10] are the same as those investigated in [8] and in the present work.

The interesting work [10] has several limitations with respect to ours, for example, GEOM allows to express hypotheses and goals *only* from a fixed list (lines, parallel segments, . . . ). This makes the system rigid, the user cannot specify, e.g. circles. As for diagrams, GEOM uses them both as a filter (as in [15]) and also as a guide (i.e. as an example) to propose conclusions. This second use is only suggested through a very simple example and is not systematic as in HOARD$_{\text{ATINF}}$. Of course, our work benefits of the advances of automated deduction these last years.

Our work shares many features with that of Chou, Gao and Zhang [8] but has several important differences.

Both systems use a logical approach with forward chaining. Geometry Expert is implemented in C, and HOARD$_{\text{ATINF}}$ is implemented in Prolog.

We compare both approaches on the basis of their design philosophy and the potential capabilities of their underlying methods.

A comparison from a experimental point of view is part of future work.

- **Axiomatization.** The Geometric axiomatization (i.e. the set of geometric rules) used by the Geometric Expert is *fixed* (it is chosen using *only* geometric criteria).
  On the contrary HOARD$_{\text{ATINF}}$ can be used with *different* axiomatizations. This is needed because the prover has to be used by students with different *levels of knowledge.*
  As explained in Section 2.5, this axiomatization must correspond to the set of theorems, definitions and axioms which are available to the students *in the considered environment.* Hence, it cannot be chosen by the prover, but *must be given to the system* (together with the considered geometric configuration and conclusion). The proof *must* be given using the appropriate axiomatization. Indeed, a correct proof may be useless, if it uses definitions, properties or constructions that are not yet known by the student.
  More generally, to be *convincing*, a proof must use theorems, theories, ..., known by the reader or explainable from the reader knowledge.
  The axiomatization in [8] consists in a set of function-free Horn clauses (possibly containing existential quantifiers). Our system can handle *non-Horn clauses* which could allow to encode *case reasoning* into the system (using for example formulas of the form $parallel((A, B), (C, D)) \vee nonparallel((A, B), (C, D)))$. Moreover existential quantifiers are replaced by function symbols (using skolemization).
  Very likely this flexibility makes the system usable for domains other than geometry with very few modifications (only some particular theories from the considered domain could have to be included into the unification algorithm).
- **Treatment of equality.** Geometry Expert does not use the equality predicate. Equality intervenes as associated to particular predicates (e.g. *eqangles*) and the equal objects are described in extension in the database. In contrast, HOARD$_{\text{ATINF}}$ incorporates an equality axiomatization in an adequate form, adapted to the inference process, in particular for handling some geometric theories such as parallelism (see Section 2.4). Specific inference rules are normally used for reasoning with equality: paramodulation (which corresponds to instantiation and replacement of equals by equals) and E-unification (i.e. unification modulo a set of equations).
  Dealing explicitely with equality is necessary in teaching the notion of proof. It must be handled *similarly as human beings do it.* Encoding it at the object level into geometric axiomatizations would not be really helpful.
  Efficient treatment of the equality predicate in Automated Deduction is well known as a difficult problem. Our new calculus ($E$-hyper-resolution) provides useful features for handling it in the context of the considered class of problems: $E$-hyper-resolution prevents the generation of non-ground clauses and strongly reduces the search space.

- **Handling theories.** As HOARD$_{\text{ATINF}}$, Geometry Expert uses special mechanisms to handle basic geometric properties (commutativity, collinearity, transitivity, ...).

  The main idea in [8] is to reflect these properties in the structured database. We put them in the inference rule (more precisely in the unification algorithm).

  Geometry Expert also uses a special mechanism for generating new auxiliary points during proof search. Explicit generation of auxiliary points is avoided into our system, since this process is encoded into the considered axiomatization, using axioms with *constructors*, for example:

  $$(AB) \parallel (CD) \vee intersection((AB), (CD)) \in (AB)$$

  Using this axiom, the point corresponding to the intersection of $(AB)$ and $(CD)$ is added into the set of clauses as soon as $(AB)$ and $(CD)$ are proven to be non-parallel. This point is denoted by $intersection((AB), (CD))$.

  The inference rules in [8] does not use function symbols, hence the use of existential quantifiers and explicit skolemization is needed.

  It is worth noticing that the heuristic used by Geometry Expert to guide the application of the generation of new auxiliary point rule can be *simulated* by the strategy described in Appendix B, which consists in first saturating the set of clauses without considering non-nested clauses, and then adding the set of non-nested clauses into the set of saturated clauses. Since non-nested clauses (in our terminology) corresponds exactly to existential axioms in [8], this strategy is essentially equivalent to the one used by the Geometry Expert for generating auxiliary points.

  However, since we want to preserve refutational completeness, we do not — in contrast to [8] — impose further restrictions on the use of auxiliary points. Chou, Gao and Zhang assume that all the auxiliary points are constructed using only *non-auxiliary* points. In our terminology, this could correspond to limiting the depth of terms to 2 (i.e. all terms occurring in the clause set must be constructed from constant symbols). Such additional restrictions could of course be easily added into our system (at the cost of loosing refutational completeness).
- **Strategy.** Whereas Geometry Expert uses a depth-first strategy, the system HOARD$_{\text{ATINF}}$ uses a *breath-first* one for non-nested clauses. This strategy is necessary, since non nested clauses may lead to non-termination. On the other hand, Geometry Expert does not use a complete search strategy: it uses a depth-first strategy, together with some restrictions on the use of the non-nested geometric rules (i.e. the rules used to add new auxiliary points) in order to insure termination of the process (i.e. to limit the number of points that can be considered). Since all the other rules correspond to nested clauses (since they do not contain any function symbol) this condition is sufficient to ensure the termination of the saturation process in [8] (see Appendix B).
- **Computing geometric figures.** As in [8], HOARD$_{\text{ATINF}}$ allows to compute automatically geometric figures for a given geometric configuration.

However, the method in [8] is restricted to very particular geometric configuration (called *linear* by the authors) whereas our method can be used on any configuration, provided that the corresponding set of algebraic equations is solvable. Even if it is not solvable, the generation of additional logical consequences of the clause set may help to find a solution.
In particular, it is easy to see that geometric configurations that are linear (in the sense of [8]) corresponds to a set of algebric equations that is *already* in solved form. Therefore, the generation of the model is straightforward in this case (no use of any formal system such as Maple, and no generation of further properties from the geometric configuration is needed).
- **Using diagrams.** In [8], diagrams are used only to handle negative conditions (notice that this could result in logically incorrect "proofs"!). We have investigated in the present work the use of diagrams to *guide* the search for a proof, by suggesting *proof plans.* To the best of our knowledge, no other work proposes a systematic use of diagrams to get proof plans.

## 6 Conclusion and Future Work

The reported research has hopefully shown that a general approach to a qualitative improving of theorem proving is realistic. Obviously, to be usable in practice it must be combined with techniques specialized to a particular domain.

In this work, computer assisted learning of the notion of proof through geometry has been chosen and the corresponding specialized techniques have been introduced.

It should be emphasized that our present goal and the one for the near future *are not* to compete with the excellent (algebraic-based) existing geometric theorem provers (benefiting from significant programming efforts for many years) but to *set theoretical bases* in order to incorporate powerful features of human reasoning using general techniques. This goal seems to us *necessary* in some uses of automated theorem proving, particularly in assisted learning. The experiments with our prototype for assisted learning have confirmed this idea. A lot of work remains to be done. We mention the main directions of present and future research:

- **Combining HOARD$_{\text{ATINF}}$ with other existing systems**. We shall deepen our comparison with Geometry Expert [8] and evaluate to which extent our approaches can be combined to obtain a more powerful system.
We shall also study the possibility of incorporating some features of HOARD$_{\text{ATINF}}$ to the well-known system Cinderella [31].
- **Incorporating symbolic computation systems**. Geometric problems often require various form of symbolic computation (e.g. dealing with relation on lengths or angles). These kinds of reasoning cannot be efficiently encoded in first-order logic. In the present work we have used MAPLE in an ad hoc manner. In the future the use of symbolic computation tools (MAPLE, MATHEMATICA, . . . ) should be systematic.

- **Experimenting with abduction**. Abduction, i.e. identifying hypothesis allowing to prove a given formula (or to explain a given fact) is an important form of the human inference activity. Abduction is one more application of our general approach (see [7,3,12]). Recently, in [29] an algebraic-based method has been proposed for discovering additional hypotheses in order to make true a given statement. We will compare possibilities of both approaches.
- **Exploiting full capabilities of our approach to analogy.** To handle analogy in this work only a very weak version of the algorithm implementing our approach [11,13,12] was necessary. Using the full capabilities of it will allow to investigate much deeper geometric properties (see Remark 11).
- **Using diagrams in logical frameworks**. One of the authors have worked on geometry modeling in Coq ([26,27]) and appreciated the utility of diagrams and the use of analogy (for example in generating lemmata or in proof planning) when interactively proven in Coq. Combining ideas and techniques in this paper with logical frameworks is an important spin-off of our work, presently under investigation.

## References

1. N. Balacheff. Apprendre la preuve. In J. Sallantin and J.-J. Szczeniarz (eds.), *Le concept de preuve à la lumière de l'intelligence artificielle*, pages 197–236. PUF, Paris, 1999.
2. W. W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9:1–35, 1977.
3. C. Bourely, G. Défourneaux, and N. Peltier. Building proofs or counterexamples by analogy in a resolution framework. In *Proceedings of JELIA 96*, LNAI 1126, pages 34–49. Springer, 1996.
4. R. Caferra and M. Herment. A generic graphic framework for combining inference tools and editing proofs and formulae. *Journal of Symbolic Computation*, 19(2):217–243, 1995.
5. R. Caferra, M. Herment, and N. Zabel. User-oriented theorem proving with the ATINF graphic proof editor. In *Fundamentals of Artificial Intelligence Research*, LNCS 535, pages 2–10. Springer, 1991.
6. R. Caferra and N. Peltier. Extending semantic resolution via automated model building: Applications. In *Proceeding of IJCAI'95*, pages 328–334. Morgan Kaufman, 1995.
7. R. Caferra and N. Peltier. Disinference rules, model building and abduction. *Logic at Work*. Essays dedicated to the memory of Helena Rasiowa (Part 5: Logic in Computer Science, Chap. 20). Physica-Verlag, 1998.
8. S.-C. Chou, X.-S. Gao, and J.-Z. Zhang. A deductive database approach to automated geometry theorem proving and discovering. *Journal of Automated Reasoning*, 25(3):219–246, 2000.
9. S.-C. Chou. *Mechanical Geometry Theorem Proving.* Mathematics and its Applications. D. Reidel, 1988.
10. H. Coelho and L. Moniz Pereira. Automated reasoning in geometry theorem proving with Prolog. *Journal of Automated Reasoning*, 2(4):329–390, 1986.

11. G. Défourneaux, C. Bourely, and N. Peltier. Semantic generalizations for proving and disproving conjectures by analogy. *Journal of Automated Reasoning*, 20(1–2):27–45, 1998.
12. G. Défourneaux and N. Peltier. Analogy and abduction in automated reasoning. In M. E. Pollack (ed.), *Proceedings of IJCAI'97*, pages 216–225. Morgan Kaufmann, 1997.
13. G. Défourneaux and N. Peltier. Partial matching for analogy discovery in proofs and counter-examples. In W. McCune (ed.), *Proceedings of CADE-14*, LNAI 1249, pages 431–445. Springer, 1997.
14. C. Fermüller and A. Leitsch. Decision procedures and model building in equational clause logic. *Journal of the IGPL*, 6(1):17–41, 1998.
15. H. Gelernter, J. Hansen, and D. Loveland. Empirical explorations of the geometry theorem-proving machine. In J. Siekmann and G. Wrightson (eds.), *Automation of Reasoning*, vol. 1, pages 140–150. Springer, 1983. Originally published in 1960.
16. H. Hong, D. Wang, and F. Winkler. *Algebraic Approaches to Geometric Reasoning.* Special issue of the Annals of Mathematics and Artificial Intelligence 13 (1-2). Baltzer, Amsterdam, 1995.
17. J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, 1991.
18. D. Kapur and J. E. Mundy. *Geometric Reasoning.* MIT Press, 1989.
19. A. Leitsch. *The Resolution Calculus.* Texts in Theoretical Computer Science. Springer, 1997.
20. V. Luengo. A semi-empirical agent for learning mathematical proof. In S. P. Lajoie and M. Vivet (eds.), *Artificial Intelligence and Education*, pages 475–482. IOS Press, Amsterdam, 1999.
21. V. Luengo. Cabri-Euclide: Un micromonde de preuve intégrant la réfutation. Thèse de doctorat, I.N.P.G., France, Septembre 1997.
22. N. Peltier. Combining resolution and enumeration for finite model building. In P. Baumgartner and H. Zhang (eds.), *FTP'00* (*Third International Workshop on First-Order Theorem Proving*), St-Andrews, Scotland, pages 170–181. Technical Report, Universität Koblenz-Landau, July 2000.
23. N. Peltier. On the decidability of the PVD class with equality. Technical report, LEIBNIZ Laboratory, 2000. To appear in the Logic Journal of the IGPL.
24. D. A. Plaisted and Y. Zhu. Ordered semantic hyperlinking. *Journal of Automated Reasoning*, 25(3):167–217, 2000.
25. G. Polya. *How to Solve It: A New Aspect of Mathematical Method* (second edition). Princeton University Press, 1973.
26. F. Puitg and J.-F. Dufourd. Formal specifications and theorem proving breakthroughs in geometric modelling. In *Theorem Proving in Higher-Order Logics*, LNCS 1479, pages 401–422. Springer, 1998.
27. F. Puitg and J.-F. Dufourd. Formalizing mathematics in higher-order logic: A case study in geometric modelling. *Theoretical Computer Science*, 234:1–57, 2000.
28. A. Quaife. Automated development of Tarski's geometry. *Journal of Automated Reasoning*, 5:97–118, 1989.
29. T. Recio and M. Vélez. Automatic discovery of theorems in elementary geometry. *Journal of Automated Reasoning*, 23(1):63–82, 1999.
30. R. Reiter. A semantically guided deductive system for automatic theorem proving. *IEEE Transactions on Computers*, C-25(4):328–334, 1976.
31. J. Richter-Gebert and U. Kortenkamp. *The Interactive Geometry Software Cinderella.* Springer, 2000.

32. M. Rusinowitch. Démonstration automatique par des techniques de réécriture. Thèse d'état, Université Nancy 1, France, 1987. Also available as textbook, Inter Editions, Paris, 1989.
33. J. R. Slagle. Automatic theorem proving with renamable and semantic resolution. *Journal of the ACM*, 14(4):687–697, 1967.
34. J. Slaney. SCOTT: A model-guided theorem prover. In *Proceedings IJCAI-93*, vol. 1, pages 109–114. Morgan Kaufmann, 1993.
35. L. Wos. *Automated Reasoning: 33 Basic Research Problems*. Prentice Hall, 1988.
36. L. Wos, R. Overbeek, E. Lush, and J. Boyle. *Automated Reasoning: Introduction and Applications* (second edition). McGraw-Hill, 1992.
37. L. Wos, G. Robinson, and D. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *Journal of the ACM*, 12:536–541, 1965.

**Acknowledgements**

We thank an anonymous referee for a lot of detailed and constructive remarks that helped us to improve the presentation of this paper.

## A An (Automatically Generated) Proof for Example 2

*Remark 15.* The justifications provided by the system (e.g. "propriete 2, chap. 8", etc.) refer directly to the geometric theorems and definitions given in the considered textbook (see Section 2.5).

```
- e est le milieu de [ak] (hypothese)
- (ak) et (ke) sont paralleles, car e est le milieu de [ak] (par definition)
- f est le milieu de [bk] (hypothese)
- (bk) et (kf) sont paralleles, car f est le milieu de [bk] (par definition)
- k est le milieu de [ab] (hypothese)
- (ab) et (bk) sont paralleles, car k est le milieu de [ab] (par definition)
- (ab) et (kf) sont paralleles car (bk) et (kf) sont paralleles,
et (ab) et (bk) sont paralleles (propriete 1, rappels)
- (ab) et (ak) sont paralleles car: k est le milieu de [ab] (par definition)
- (ab) et (ke) sont paralleles car (ak) et (ke) sont paralleles
et (ab) et (ak) sont paralleles (propriete 1, rappels)
- (ke) et (kf) sont paralleles car (ab) et (kf) sont paralleles
et (ab) et (ke) sont paralleles (propriete 1, rappels)
- (ke) et (ef) sont paralleles car (ke) et (kf) sont paralleles (par definition)
- (ak) et (ef) sont paralleles car (ak) et (ke) sont paralleles
et (ke) et (ef) sont paralleles (propriete 1, rappels)
- j est le milieu de [ac] (hypothese)
- i est le milieu de [bc] (hypothese)
- (ab) et (ij) sont paralleles car j est le milieu de [ac]
et abc est un triangle et i est le milieu de [bc] (propriete 1, chap. 8)
- (ak) et (ij) sont paralleles car: (ab) et (ak) sont paralleles
et (ab) et (ij) sont paralleles (propriete 1, rappels)
- (ij) et (ef) sont paralleles car: (ak) et (ef) sont paralleles
et (ak) et (ij) sont paralleles (propriete 1, rappels)
```

```
- (ck) et (je) sont paralleles car e est le milieu de [ak]
et ack est un triangle et j est le milieu de [ac] (propriete 1, chap. 8)
- (ck) et (if) sont paralleles car f est le milieu de [bk]
et bck est un triangle et i est le milieu de [bc] (propriete 1, chap. 8)
- (if) et (je) sont paralleles car: (ck) et (je) sont paralleles
et (ck) et (if) sont paralleles (propriete 1, rappels)
- fije est un parallelogramme car (ij) et (ef) sont paralleles
et (if) et (je) sont paralleles (definition 14, rappels)
- abc est isocele en c (hypothese)
- (ck) est la mediane issue de c du triangle cab car k est le milieu de [ab]
(definition 10, rappels)
- (ck) est la hauteur issue de c du triangle cab car abc est isocele en c
et abc est un triangle et (ck) est la mediane issue de c du triangle cab
(propriete 5, chap. 10)
- (ab) et (ck) sont perpendiculaires car (ck) est la hauteur issue de c du
triangle cab (definition 10, rappels)
- (ck) et (ij) sont perpendiculaires car (ab) et (ij) sont paralleles et
(ab) et (ck) sont perpendiculaires (propriete 3, rappels)
- (ij) et (if) sont perpendiculaires car (ck) et (if) sont paralleles
et (ck) et (ij) sont perpendiculaires (propriete 3, rappels)
- fije est un rectangle car: fije est un parallelogramme et (ij) et (if) sont
perpendiculaires (propriete 30, rappels)
```

## B An (Automatically Generated) Proof for Example 3

```
- h est le milieu de [ad] (hypothese)
- g est le milieu de [cd] (hypothese)
- (ac) et (gh) sont paralleles car h est le milieu de [ad], acd est un triangle
et g est le milieu de [cd] (propriete 1, chap. 8)
- f est le milieu de [bc] (hypothese)
- e est le milieu de [ab] (hypothese)
- (ac) et (ef) sont paralleles car f est le milieu de [bc], abc est un triangle
et e est le milieu de [ab] (propriete 1, chap. 8)
- (gh) et (ef) sont paralleles car (ac) et (gh) sont paralleles
et (ac) et (ef) sont paralleles (propriete 1, rappels)
- (bd) et (gf) sont paralleles car g est le milieu de [cd], bcd est un triangle
et f est le milieu de [bc] (propriete 1, chap. 8)
- (bd) et (he) sont paralleles car h est le milieu de [ad], abd est un triangle
et e est le milieu de [ab] (propriete 1, chap. 8)
- (gf) et (he) sont paralleles car (bd) et (gf) sont paralleles
et (bd) et (he) sont paralleles (propriete 1, rappels)
- fghe est un parallelogramme car (gh) et (ef) sont paralleles, ghef est un
quadrilataire et (gf) et (he) sont paralleles (definition 14, rappels)
- (ac) et (bd) sont perpendiculaires (hypothese)
- (ac) et (he) sont perpendiculaires car (ac) et (bd) sont perpendiculaires
et (bd) et (he) sont paralleles (propriete 3, rappels)
- (he) et (ef) sont perpendiculaires car: (ac) et (ef) sont paralleles, (ac)
et (he) sont perpendiculaires (propriete 3, rappels)
- fghe est un rectangle car fghe est un parallelogramme et (he) et (ef) sont
```

```
perpendiculaires (propriete 30, rappels)
- longueur(segment(a,c)) = longueur(segment(b,d)) (hypothese)
- longueur(segment(g,h)) = longueur(segment(a,c))*0.5 car h est le milieu de
[ad], acd est un triangle et g est le milieu de [cd] (propriete 1, chap. 8)
- longueur(segment(g,f)) = longueur(segment(b,d))*0.5 car g est le milieu de
[cd], bcd est un triangle et f est le milieu de [bc] (propriete 1, chap. 8)
- longueur(segment(g,h)) = longueur(segment(g,f)) car
longueur(segment(a,c)) = longueur(segment(b,d)),
longueur(segment(g,h)) = longueur(segment(a,c))*0.5
et longueur(segment(g,f)) = longueur(segment(b,d))*0.5 (par definition)
- longueur(segment(e,f)) = longueur(segment(a,c))*0.5 car f est le milieu de [bc]
abc est un triangle et e est le milieu de [ab] (propriete 1, chap. 8)
- longueur(segment(g,f)) = longueur(segment(e,f)) car
longueur(segment(a,c)) = longueur(segment(b,d)),
longueur(segment(g,f)) = longueur(segment(b,d))*0.5
et longueur(segment(e,f)) = longueur(segment(a,c))*0.5 (par definition)
- longueur(segment(h,e)) = longueur(segment(b,d))*0.5 car h est le milieu de [ad]
abd est un triangle et   e est le milieu de [ab] (propriete 1, chap. 8)
- longueur(segment(g,f)) = longueur(segment(h,e)) car
longueur(segment(g,f)) = longueur(segment(b,d))*0.5
et longueur(segment(h,e)) = longueur(segment(b,d))*0.5 (par definition)
- fghe est un losange car longueur(segment(g,h)) = longueur(segment(g,f)),
longueur(segment(g,f)) = longueur(segment(e,f)), ghef est un quadrilataire
et longueur(segment(g,f)) = longueur(segment(h,e)) (definition 15, rappels)
- fghe est un carre car fghe est un losange et fghe est un rectangle
(propriete 32, rappels)
```

# Higher-Order Intuitionistic Formalization and Proofs in Hilbert's Elementary Geometry

Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck

Laboratoire des Sciences de l'Image, de l'Informatique
et de la Télédétection (UMR CNRS 7005)
Université Louis-Pasteur de Strasbourg
Pôle API, boulevard S. Brant
67400 Illkirch, France
{dehlinge,dufourd,schreck}@lsiit.u-strasbg.fr

**Abstract.** We propose the basis of a higher-order logical framework to axiomatize and build proofs in Hilbert's elementary geometry in which intuitionistic aspects are emphasized. More precisely, we use the Calculus of inductive constructions and the system Coq to specify geometric concepts and to study and interactively handle proofs for the first two groups of Hilbert's axiomatics. It is the first step to a formalization well adapted to the definition of primitive operations that are used in many different geometric algorithms.

## 1 Introduction

We study a formalization of geometry well adapted to computer-related aspects of geometry, including computational geometry, geometric modelling, geometric construction and robotics. In this paper, we propose the basis of a higher-order logical framework to axiomatize and build proofs in Hilbert's elementary geometry in which intuitionistic aspects are emphasized.

The interest of an axiomatization of geometry for computer science has been well demonstrated by several authors, for instance Toussaint [16] and Knuth [14], in order to better understand geometric universes and well define primitive operations that are used in many different algorithms. Our research group has reached the same conclusion for topology-based geometric modelling [22]. Moreover, in spite of the impressive results of the mechanical geometry theorem proving based on polynomial algebra [23,3], the crucial issue of round-off errors often makes logical reasoning in pure geometry preferred to computing in analytic geometry when possible. This is the case for knowledge-based systems that symbolically solve geometric constraints, for instance in CAD [6].

In computer science, the need to actually build and manipulate geometric objects favors an intuitionistic approach, proscribing the *excluded middle* axiom or equivalent ones [11]. This point of view has been defended by Heyting [12] and developped by von Plato [17,18,19] in a new approach of affine concepts. But instead of following a not much frequented way to address geometry (see for instance [23,1] for various propositions related with this topic), we opted to

J. Richter-Gebert and D. Wang (Eds.): ADG 2000, LNAI 2061, pp. 306–323, 2001.

attack a commonly accepted geometry, hence our choice of Hilbert's axiomatics. Thus, following the progression of Hilbert's *Grundlagen der Geometrie* [13], we have tried to express the beginning of this axiomatics and the proofs of fundamental theorems in the chosen logical framework. When it was not natural for us, we have adapted Hilbert's formulation, especially by distinguishing intuitionistic and classical aspects.

More precisely, our framework is the Calculus of inductive contructions (CIC) which is based on type theory and offers a high abstract generic level of specification and proof. It is implemented in system Coq developped at the INRIA and already used by von Plato for geometry and our research group for discrete topology [20,21,22].

The paper is structured as follows. Section 2 shortly presents the calculus of inductive constructions and Hilbert's geometry with its five axiom groups. Section 3 gives some basic formal definitions of geometric types and relations in Coq. Section 4 specifies the axioms of the first Hilbert's group. Sections 5 and 6 each present a theorem of the first group. Section 7 does the same for the axioms and two theorems of the second group. Then, Section 8 develops the proof of a larger theorem which requires additional data types, and Section 9 gives some conclusions. Rather than giving a complete view of the results, we will focus on the novelties that come up at each step.

## 2 Basic Notions

### 2.1 Calculus of Inductive Constructions

Type theories are logical frameworks the objects of which are typed lambda-terms. They can be distinguished by the power of their typing system and their underlying logic (see for instance [10]). Examples are Church's higher order logic, Martin-Löf intuitionistic type theory and the Calculus of inductive constructions (CIC) [5]. A typed lambda-term formalism is related with its logical semantics by the Curry-Howard isomorphism, which states that:

- any proof $p$ is represented by a lambda-term $P$
- any proposition $s$ is represented by a type $S$
- $p$ is a proof of $s$ iff $P$ is of type $S$

The typing system of CIC is based on Girard's polymorphic lambda-calculus, a very powerful typing system that allows the representation of proofs as well as types by lambda-terms, the underlying logic being intuitionistic. Another important feature of CIC is the possibility to inductively define types that are used to define usual logical constructions with connectors and quantifiers as well as user application-oriented data types and predicates. Fundamentally, the generic inference rules, or type-checking rules, of CIC allow introduction or elimination of the universal quantifier, of the lambda-abstraction or of inductive constructs. All usual quantifiers and connectors manipulation rules can be considered to be derived from them.

To develop our specifications and proofs we have worked with system Coq, a CIC-based conversational proof assistant. The user communicates with the system through a high-level language called Gallina that allows the definition of axioms, parameters, types, functions and predicates. Coq also supports interactive proof constructions in which reasoning steps are specified by the user through commands called tactics, which in fact implement the CIC inference rules. This point will be illustrated in the subsequent geometric proofs.

### 2.2 Hilbert's Axiomatics

In the late XIXth century, geometry is still based on Euclid's theory, which itself is based on intuition. The creation of non-euclidian geometries leads a few mathematicians to go back to the roots of geometry and find alternate or better ways to found it. In 1899, David Hilbert publishes *Grundlagen der Geometrie* [13] in which he proposes a new axiomatics for elementary geometry. Hilbert pays special attention to issues such as axiom classification, independence and minimality. He is also the first one to abstractly link arithmetics and geometry by creating a linear and area calculus system. Hilbert classifies his axioms in five groups:

I. Incidence axioms
II. Order axioms
III. Congruence axioms
IV. Parallel axiom
V. Continuity axioms

We have only studied group I and a significant part of group II so far. Each group features a number of important theorems in addition to the axioms, the proof of many of which is not given. Hilbert's style is informal, therefore many formalization problems arise. We will focus on the methodology we used to overcome these problems. Besides, Hilbert works in a classical logical framework, whereas we try to be constructive. Therefore, we are especially concerned about the use of excluded middle axiom or its equivalents.

## 3 Definitions

Group I of Hilbert's axioms deals with incidence. They express relations between points, lines and planes, these are abstract beings at the very beginning. Thus, in Gallina, they may be seen as belonging to one of three types designated by parameters declared by:

```
Parameters point, line, plane: Set.
```

In Gallina, $t : T$ reads "$t$ is of type $T$", and $Set$, which has nothing to do with set theory, is used to refer to "the type of concrete types", i.e. the type of the type of the objects that are actually built and studied. This command adds `point`, line and plane as three new constants of type Set in the environment. It can be understood as the axiom "there exist three objects of type Set named point, line and plane". Hilbert then considers two abstract binary incidence relations, one between points and lines, the other between points and planes. They are also declared through parameters:

```
Parameter inc-pt-ln: point -> line -> Prop.
Parameter inc-pt-pl: point -> plane -> Prop.
```

In these functional notations, Prop is the built-in type of logical propositions. Group II of axioms deals with a ternary betweenness relation between points. As before, it is declared through a parameter:

```
Parameter between: point -> point -> point -> Prop.
```

Thus, (between A B C) stands for "$B$ is between $A$ and $B$". Finally, it is convenient to define some other relations built from those parameters, using the Gallina definition mechanism. For instance, alignment of three points is defined by:

```
Definition aligned3: point -> point -> point -> Prop:=
        [A,B,C: point]
              (EX l: line
                   | (inc-pt-ln A l) /\ (inc-pt-ln B l)
                                     /\ (inc-pt-ln C l)).
```

This definition abbreviates the term [A,B,C: point](...) into the symbol aligned3 of type point → ... → Prop. The square-bracketed expression [A,B,C: point] represents the lambda-abstraction of variables A,B and C, and the expression (EX l: line | ...) represents the existential quantification over variable l. Thus, the whole term intuititively means "given three points A,B et C, there exists a line l such that A,B et C are incident to l". Likewise, we represent incidence between a line and a plane with the following definition:

```
Definition inc-ln-pl: line -> plane -> Prop:=
          [a: line ;a': plane]
                (A: point)
                      (inc-pt-ln A a) -> (inc-pt-pl A a').
```

where the notation (A: point)(...) represents the universal quantification of variable A. The whole definition means "given a line a and a plane a', for any point A, if A is incident to a, then it is incident to a'". Many other definitions are built in a similar way. Note that, adapting Hilbert's notation, we name points, lines and planes with capital, minuscules and quoted minuscules, respectively.

## 4 Axioms of the First Group

We can then use these variables and definitions to give a naive Gallina translation of the first group of Hilbert's axioms. The first of them is written by Hilbert:

> **Axiom (I,1):** "For any given two points $A$ and $B$, there is a line $l$ to which both $A$ and $B$ are incident."

According to this formulation, $A$ and $B$ may be identical. However, when he refers to several beings of the same kind, Hilbert always assumes that they are all distinct. This will prove important to use axiom (I,1) constructively. Thus, when translating these axioms into Gallina, we systematically mention that all variables of the same kind are distinct. For instance, a Gallina version of axiom (I,1) in which $A$ and $B$ are supposed to be distinct is the following:

```
Axiom AI-1:
      (A,B: point)
            ~A=B -> (EX a: line |
                    (inc-pt-ln A a) /\ (inc-pt-ln B a)).
```

This term can be understood as: "for any two points `A` and `B`, if `A` and `B` are not equal, then there exists a line `a`, such that `A` is incident to `a` and `B` is incident to `a`". In a similar way, Hilbert writes the second axiom:

> **Axiom (I,2):** "There is no more than one line to which two given points $A$ and $B$ are incident."

Note that $A$ and $B$ are here obviously meant to be distinct, as this statement would otherwise be false in "natural" geometry, which Hilbert's axioms are supposed to formalize. This axiom is much harder to express formally. Indeed, Hilbert's formulation actually means "the number of lines to which two given points $A$ and $B$ are incident is no more than one". Should we try to translate this statement directly, we would have in some way to formally describe the counting of lines satisfying a proposition. This is rather cumbersome as it would probably require the introduction of concepts such as naturals, infinite sets and cardinals, especially if performed in a constructive framework. But the main problem is that we have no way to make sure that our method of counting lines will yield the same results as Hilbert's, as he did not give any properties of this process. The best we can do is saying that line counting is a rather easy and clear concept, and that Hilbert probably meant the same thing as we do. However, we choose a formal presentation of this axiom which is syntactically far from a direct translation, but semantically acceptable:

```
Axiom AI-2:
      (A,B: point)(a,b: line)
            ~A=B -> (inc-pt-ln A a) -> (inc-pt-ln B a)
            -> (inc-pt-ln A b) -> (inc-pt-ln B b)
            -> a=b.
```

which can be read as: "For any two given points A and B and any two given lines a and b, if A and B are distinct, if A and B are incident to a, and if A and B are incident to b, then a and b are identical". Although other formulations are equivalent in classical logic, ours is rather powerful in intuitionistic logic, mainly because it yields a positive result [11], which had already been used by [18]. Similar considerations that we cannot develop any further can be done on the six remaining axioms of the first group.

## 5 Theorems of the First Group

This section discusses the proof of Theorem I, which is given by Hilbert in his work. For modularity reasons, we have split it into four parts, numbered Ia to Id. Thus, Theorem Ia informally writes:

**Theorem Ia:** "Two lines have at most one common point."

which can be translated in Gallina into:

```
Theorem Ia:
     (a,b: line)(A,B: point)
          ~a=b -> (inc-pt-ln A a) ->(inc-pt-ln B a)
          -> (inc-pt-ln A b) -> (inc-pt-ln B b)
          -> A=B.
```

where counting is handled in a way similar to axiom (I,2). Thus Theorem Ia can read: "all points common to two distinct lines are equal". To illustrate the basic mechanisms of the Coq system, we detail the proof of this theorem. In fact, the proof requires the addition of an axiom of *excluded middle*, or at least of *decidability of point equality*. But we can start by constructively proving a weaker version of this theorem:

```
Theorem Ia-weak:
      (a,b: line)(A,B: point)
           ~a=b -> (inc-pt-ln A a) -> (inc-pt-ln B a)
           -> (inc-pt-ln A b) -> (inc-pt-ln B b)
           ~~A=B.
```

which can read: "any points common to two distinct lines cannot be proved different".

*Proof.* – Tactic `Unfold 1 not` unfolds the first occurence of symbol `~`, thus expanding the subterm `~~A=B` into `~A=B` → `False`.
- Tactic `Intros` introduces `(a,b: line)`, `(A,B: point)`, `~a=b`, `(inc-pt-ln A a)`, `(inc-pt-ln B a)`, `(inc-pt-ln A b)`, `(inc-pt-ln B b)`, `A=B` as hypotheses, the only conclusion being `False`.
- Tactic `Absurd a=b` allows us to prove any goal, in particular `False`, provided we prove `~a=b` and `a=b`.

- Tactic `Assumption` proves `~a=b` which is among the hypotheses.
- Tactic `Apply AI_2 with A:=A B:=B` applies axiom `AI_2` with convenient effective parameters, and leads to five subgoals, which are all among hypotheses.
- Finally, tactic `Assumption` used five times removes all of them.

□

Then, the complete proof of Theorem `Ia` is easily obtained from the Theorem `Ia-weak` and the excluded middle axiom:

```
Axiom EM: (p: Prop) p \/ ~p.
```

or, what is equivalent here, the axiom of decidability of point equality:

```
Axiom DEC_EQ_PT (A,B: point) A=B \/ ~A=B.
```

The use of these axioms is natural in classical logic, but is prohibited in intuitionistic logic. In fact, the lack of ways to infer a point equality with our axioms suggests that Theorem `Ia` cannot be proved constructively. Thus if we assume `DEC_EQ_PT`, a proof of Theorem `Ia` is:

*Proof.*
- As above, Tactic `Intros` introduces all premises into the local context.
- Tactic `(Elim DEC_EQ_PT A B)` generates two subgoals:
  - `A=B`→`A=B`: removed by Tactic `Trivial`.
  - `~A=B`→`A=B`: Tactic `Intro` introduces `~A=B` into the local context.
  - Tactic `Absurd ~A=B` replaces the current goal by two subgoals `~A=B` and `~~A=B`.
    - ∗ Tactic `Assumption` removes the first subgoal.
    - ∗ Tactic `Apply Ia-weak with a:=a b:=b` applies Theorem `Ia-weak`, and generates one subgoal for each premise of `Ia-weak`.
    - ∗ Finally, Tactic `Assumption` used five times removes the remaining subgoals.

□

The second part of Hilbert's Theorem I expresses:

> **Theorem Ib:** "Two distinct planes either have no common point or have a common line."

Its naive Gallina translation is:

```
Theorem Ib:
      (a',b': plane)
            ~a'=b'
            -> ~(EX A: point | (inc-pt-pl A a')
                                         /\ (inc-pt-pl A b'))
               \/ (EX a: line | (inc-ln-pl a a')
                                         /\ (inc-ln-pl a b')).
```

As previously, we can prove this theorem using the excluded middle or an equivalent. Here again, we conjecture that it cannot be proved constructively. This time, the reason is that it is a *classification theorem*, and that nothing in our axioms allows reasoning by cases. This anomaly is fixed by adding the excluded middle to the axioms. As in Theorem `Ia`, we can constructively prove for this theorem a weaker version, with which the excluded middle directly yields the full theorem. The third part of Hilbert's Theorem I writes:

> **Theorem Ic:** "Two distinct planes with a common line have no common point outside this line."

There are several ways to formalize this theorem, for instance:

```
Theorem Ic-neg:
      (a',b': plane)(a: line)(A: point)
            ~a'=b'
            -> (inc-pt-pl A a') -> (inc-pt-pl A b')
            -> (inc-ln-pl a a') -> (inc-ln-pl a b')
            -> ~~(inc-pt-ln A a).
```

which expresses the fact that the planes `a'` and `b'` in play have no common point outside the common line `a`, or:

```
Theorem Ic-pos:
      (a',b': plane)(a: line)(A: point)
            ~a'=b'
            -> (inc-pt-pl A a') -> (inc-pt-pl A b')
            -> (inc-ln-pl a a') -> (inc-ln-pl a b')
            -> (inc-pt-ln A a).
```

which expresses the fact that all points common to planes `a'` and `b'` are incident to the common line `a`.

We have proved `Ic-neg` constructively, and `Ic-pos` classically with the excluded middle. Once again, we conjecture that `Ic-pos` cannot be proved constructively. Finally, the fourth part of Theorem I writes:

> **Theorem Id:** "A plane and a line that is not incident to this plane have at most one common point."

No new kind of problem arises in the treatment of this theorem.

## 6 Theorem II

In Group I of Hilbert's axiomatics, Theorem II deals with existence and unicity of the plane incident to a line and a point, or to two lines:

> **Theorem II:** "There is a single plane incident to two given distinct convergent lines, or to a given line and point apart from this line."

Like Theorem I, Theorem II is split into four parts, or “subtheorems”. The last two parts, which address unicity of the considered planes, can be formalized easily and proved constructively. The first two parts, which address existence of the planes, are slightly harder to formalize, and their proofs use the decidability of point equality.

Here, decidability occurs in a new way. Indeed, decidability was previously used at the very beginning of the proof to produce two cases, one being proved constructively, and the other being proved by refutation. Thus Theorems `Ia` to `Id` can be seen as direct classical corollaries of constructive lemmas. In the first two parts of Theorem II, *decidability applies to points built in the middle of the proofs*, and is used to decide which of two constructions must be chosen. Here constructive parts and classical decision parts are interwoven in a more complex way than in Theorem I, thus non-constructivity must be handled in a different manner. A method would be to prove a lemma for each constructive part of the complete classical proof. The problem is that those lemmas are not very meaningful by themselves. Our method is to weaken the two existence-related subtheorems by adding an hypothesis, which will be used during the proof to make the decision that was previously made by the use of decidability in the classical proof. This extra hypothesis, called `can-build-other-point-on-line` is a much weaker and more specialized version of point equality decidability:

```
Definition can-build-other-point-on-line:=
      (A: point)(a: line)
            (inc-pt-ln A a)
            -> (EX B: point | ~A=B /\ (inc-pt-ln B a)).
```

which states that, given a line and a point on this line, one can build another point on this line. Using this definition we can express and constructively prove the following theorem:

```
Theorem IIexll-weak:
      can-build-other-point-on-line ->
            (a,b: line)(A: point)
            ~a=b
            -> (inc-pt-ln A a) -> (inc-pt-ln A b)
            -> (EX a’: plane |
                     (inc-ln-pl a a’) /\ (inc-ln-pl b a’)).
```

which states that, assuming `can-build-other-point-on-line`, one can build a plane incident to two distinct given lines that have a common point. Once again, we have managed to express the full theorem as a classical corollary of this constructive theorem, as `can-build-other-point-on-line` is a consequence of point equality decidability. However, this weak constructive version is actually much weaker, as `can-build-other-point-on-line` must be assumed whenever we want to use it. This extra hypothesis can also be understood as a property we wish we had in our axiomatics, and could be used as a hint as to how to modify Hilbert's classical axiomatic in order to build a constructive one. The other existence subtheorem is handled in the same way.

All theorems of the first group have been formally proved in classical logic, and almost none were proved in constructive logic. This strongly suggests that Hilbert's axioms are not naturally well adapted for constructive reasoning, and need to be tweaked in order to fill our needs.

# 7 Axioms and Theorems of the Second Group

## 7.1 Axioms of the Second Group

The four axioms of Group II of Hilbert's geometry describe relation `between` (see Sect. 3). They are straightforwardly translated to Gallina, the first one being split into two parts numbered **a** and **b**:

> **Axiom (II,1) part a:** "If $B$ is between $A$ and $C$, then $A$, $B$, $C$ belong to a line."

translated in Gallina into:

```
Axiom AII-1a:
      (A,B,C: point)
            ~A=B -> ~A=C -> ~B=C -> (between A B C)
            -> (aligned3 A B C).
```

Note that as usual we only consider distinct points.

> **Axiom (II,1) part b:** "If $B$ is between $A$ and $C$, then $B$ is also between $C$ and $A$."

translated into:

```
Axiom AII-1b:
      (A,B,C: point)
            ~A=B -> ~A=C -> ~B=C -> (between A B C)
            -> (between C B A).
```

> **Axiom (II,2):** "Given two points $A$ and $C$, there exists at least one point $B$ belonging to line $AC$ such that $C$ is between $A$ and $B$."

Thanks to Axiom `AII-1a`, explicitely stating that $A$, $B$ and $C$ are aligned is redundant:

```
Axiom AII-2:
      (A,B: point)
      ~A=B -> (EX C: point |
                 ~A=C /\ ~B=C /\ (between A B C)).
```

Another option was to give the name "line $XY$" to the line built between $X$ and $Y$ using axiom `AI-1`.

> **Axiom (II,3):** "Among three points on a line, there is no more than one that is between the other two."

translated into:

```
Axiom AII-3:
      (A,B,C: point)
            ~A=B -> ~A=C -> ~B=C -> (between A B C)
            -> ~(between B A C).
```

Propositions `~(between C A B)`, `~(between A C B)` and `~(between B C A)` can then be deduced using `AI-1` and `AII-1b`.

> **Axiom (II,4) (or axiom of Pasch):** "Let $A$, $B$ and $C$ be three unaligned points and $a$ a line of plane $ABC$ that meets none of $A$, $B$ and $C$; if line $a$ meets any point of segment $AB$, then it meets either a point of segment $BC$ or a point of segment $AC$."

To make things shorter, we first introduce a predicate which states that a given line meets a point of a given segment represented by its ends:

```
Definition line-meets-seg:=
      [a: line][A,B: point]
            (EX C: point | ~A=C /\ ~B=C
                 /\ (inc-pt-ln C a) /\ (between A C B)).
```

Now we can write Axiom (II,4):

```
Axiom AII-4:
      (A,B,C: point; a: line)
           ~(aligned3 A B C)
           -> ~(inc-pt-ln A a)
           -> ~(inc-pt-ln B a) -> ~(inc-pt-ln C a)
           -> (line-meets-seg a A B)
           -> (line-meets-seg a A C) \/ (line-meets-seg a B C).
```

This axiom is central in Hilbert's axiomatic, as it spawns the most interesting and meaningful consequences. In our axiomatic, it is also the first axiom that allows to infer a *disjunction*.

### 7.2 Theorems of the Second Group

Hilbert proposes a number of theorems and gives nonconstructive proofs for them. As in Theorem `IIexll-weak`, we obtain constructive analog theorems by adding weaker versions of excluded middle to the premises of these theorems.

**Theorem III:** "Given two points $A$ and $C$, there is at least one point $D$ on the line $AC$ between $A$ and $C$."

In Gallina, this classical theorem writes:

```
Theorem TIII:
      (A,C: point)
           ~A=C
           -> (EX D: point | ~A=D /\ ~C=D
                       /\ (between A D C)).
```

The only purely classical step in Hilbert's proof is the construction of a point `E` not aligned with `A` and `C`, which cannot be done constructively without an extra assumption. Such an assumption is that, given two distinct points, one is able to build a third point such that the three points are not aligned:

```
Definition can-build-unaligned-point: Prop:=
     (A,B: point) ~A=B -> (EX C: point | ~(aligned3 A B C)).
```

Thus, adding this definition to the premises of Theorem III makes it constructively provable. This hypothesis is natural enough to consider that we do not go too far away from the spirit of Hilbert's axiomatics. Theorem IV is handled in a similar way:

**Theorem IV:** "Of three aligned points $A$, $B$ and $C$, there is one that is between the two others."

This classical theorem writes:

```
Theorem TIV:
     (A,B,C: point)
      ~A=B -> ~A=C -> ~B=C -> (aligned3 A B C)
      -> (between B A C) \/ (between A C B) \/ (between A B C).
```

It is easily proved in classical logic, and requires the addition of two hypotheses to be proved in constructive logic. The first one is that all intersection points of distinct lines are equal, the second one is that the relation between is decidable. They are respectively named `eq-line-inter` and `dec-between`. Finally, Theorem V writes:

**Theorem V:** "Given four points of a line, they can be designated by $A$, $B$, $C$ and $D$ so that $B$ is between $A$ and $C$ and between $A$ and $D$, and that $C$ is between $A$ and $D$ and between $B$ and $D$."

This theorem has not been formally proved as it is a special case of Theorem VI, which is proved further. However, Hilbert's proof uses two interesting lemmas which can be seen as transitivity properties of between:

> **Lemma V-1:** "If $B$ belongs to segment $AC$ and $C$ belongs to segment $BD$, then $B$ and $C$ belong to segment $AD$."

> **Lemma V-2:** "If $B$ belongs to segment $AC$ and $C$ belongs to segment $AD$, then $C$ belongs to segment $BD$ and $B$ belongs to segment $AD$."

Both of the lemmas can be proved classically, but also constructively when adding the previously defined hypotheses `can-build-not-aligned-point` and `eq-line-inter`. They will be crucial in the proof of Theorem VI, as they provide a rather powerful manipulation tool for relation `between` that is much more intuitive than Pasch's axiom.

## 8 Theorem VI

Theorem VI is an example of complex result that requires the introduction of new proof techniques. It writes:

> **Theorem VI:** "Given a finite number of points of a line; it is possible to designate them by $A$, $B$, $C$, $D$, $E$, $\ldots$, $K$ so that $B$ is between $A$ and every other point, that $C$ is between $A$, $B$ on the one hand and $D$, $E$, $\ldots$, $K$ on the other hand, that $D$ is between $A$, $B$, $C$ on the one hand and $D$, $E$, $\ldots$, $K$ on the other hand, and so on. Only the opposite denomination $K$, $\ldots$, $E$, $D$, $C$, $B$, $A$ shares the same properties."

In his book [13], Hilbert does not give any proof for this theorem. Except for the unicity part, we have proved it semi-constructively with the help of Coq in the style of Theorems III to V. The formulation of this theorem is much trickier than in the previous theorems for two reasons. The first reason is that Theorem VI applies to a *variable amount of points*, which requires quantification on point sets rather than on individual points. The second reason is that we need to formalize the notion of *point designation*. We choose to represent a point set by a linear list of points, the designation of a point in this set being assimilated to its rank in the corresponding list. The theorem then translates into:

> **Theorem VI (reformulated):** "Any list $l$ of distinct points of a line can be sorted into a list $l'$, such that each point in $l'$ is between all of its predecessors in $l'$ and all of its successors in $l'$."

In order to formally prove this theorem, we must axiomatize the notion of list and perform an induction on the list structure.

### 8.1 Axiomatization of Lists

We can immediately use Coq's built-in lists, the inductive polymorphic type of which is defined by:

```
Inductive list [A: Set]: Set:=
        nil: (list A)
      | cons: A -> (list A) -> (list A).
```

Thus `list` is an inductive type parametered by a set `A` and generated by the two constructors `nil` and `cons`. The standard Coq library provides some functions to manipulate or observe lists, such as `In` which tests whether an element belongs to a list, `length` that computes the length of a list, and `rev` that reverses a list. With these we define several auxiliary predicates: `(external-point A l)` which states that, on a line, a given point `A` is not between any two elements of the point list `l` (note that `A` may belong to `l`); `(distinct l)` which states that the points of a list `l` are all distinct; `(permutation l l')` which states that `l` is a permutation of `l'`; `(sorted l)` which states that `l` is sorted with respect to `between` as described above; `(aligned l)` which states that all points of `l` are aligned. Using all these notions Theorem VI formally writes:

```
Theorem TVI:
      dec-between ->
      can-build-unaligned-point -> eq-line-inter
      -> (l: (list point))
            (aligned l) -> (distinct l)
            -> (EX l': (list point) |
                     (permutation l l')
                     /\ (sorted l') /\ (distinct l')).
```

The proof of this theorem uses Lemma `EXISTS-EXTERNAL-POINT`, which finds in a point list a point that is external to this list. It is presented in the following section.

### 8.2 Proof of EXISTS-EXTERNAL-POINT

This lemma formally writes:

```
Lemma EXISTS-EXTERNAL-POINT:
      dec-between
      -> can-build-unaligned-point -> eq-line-inter ->
            (l: (list point))
                  (aligned l) -> ~l=(nil point)
                  -> (distinct l)
                  -> (EX A: point | (In A l)
                              /\ (external-point A l)).
```

*Proof.* The Coq system is asked to perform induction on the structure of `l` through Tactic `Induction`. There are two cases:

- `l=(nil point)`: a contradiction is inferred with hypothesis `l=(nil point)`.
- `l=(cons N l')`: a second induction is performed on `l'`, with anew two cases:
  - `l'=(nil point)`: `N` is easily proved to be a suitable external point to `l`.
  - `l'=(cons H1 l'')`: a third induction is performed on `l''`, with anew two cases:

  - ∗ `l''=(nil point)`: `N` is again easily proved to be a suitable external point to `l`.
  - ∗ `l''=(cons H2 l''')`: we are in the general case, which is handled as follows.

The general case is solved using as induction hypothesis the fact that `EXISTS-EXTERNAL-POINT` is satisfied for the list `l'=(cons H1 (cons H2 l'''))`. This hypothesis is used to build a suitable point `X` external to `l'` that also belongs to `l'`.

We then show that either `H1` or `H2` is distinct from `X` as all points in `l` are distinct; that point is named `H`. We then use the constructive version of Theorem `IV` (Sect. 7.2) to infer `(between N X H)` ∨ `(between X H N)` ∨ `(between X N H)`. We have to study the three cases:

- `(between N X H)`: we want to prove that `N` is a point external to `l` that belongs to `l`.
- `(between X H N)`: we want to prove that `X` is a point external to `l` that belongs to `l`.
- `(between X N H)`: we want to prove that `X` is a point external to `l` that belongs to `l`.

In order to take care of the first case, we must show that `(between N B C)` ∨ `(between N C B)` for any two points `B` and `C` in `l`. To do so, we compare each of `B` and `C` to each of `N`, `X` and `H`, and thus create $2^6$ subgoals, each corresponding to a combination of comparison results. The comparison is possible without assuming point equality decidability thanks to the assumption that all points in `l` are distinct. Among these subgoals, 57 lead to a contradiction and are solved automatically using tactic `Auto`, and the remaining 7 show strong similarities. Thus they are all solved with two local lemmas that are proved using Lemmas `V-1`, `V-2` and Axiom `AII-1b`, and that take advantage of those similarities. We then have to prove that `N` belongs to `l`, which we do by using the definition of `l`. The proofs for the second and third cases have the same structure as the one for the first case, except that `N` is replaced by `X` and that there is only one local lemma.

This ends the proof by induction on the list structure that `EXISTS-EXTERNAL-POINT` is satisfied for any list. □

Now we can prove Theorem VI itself.

### 8.3 Proof of Theorem VI

The proof is led by induction on the length of the entry point list `l`.

*Proof.* System Coq is asked to perform Tactic `Induction` on `(length l)`, i.e. on the structure of the naturals. There are two cases:

- `(length l)=O`: obviously `(nil point)` is a sorted permutation of l.

- `(length l)=(S n)`: a second induction is performed on `n`, with anew two cases:
  - `n=0`: list `l` has a single element, and we immediately prove that `l` is a sorted permutation of itself.
  - `n=(S n')`: a third induction is performed on `n'`, with anew two cases:
    - `n'=0`: list `l` has two elements, and we easily prove that `l` is a sorted permutation of itself.
    - `n'=(S n'')`: list `l` has three or more elements. We are in the general case, which is handled as follows.

In the general case, we apply `EXISTS-EXTERNAL-POINT` to extract from `l` a point `H` that is external to `l`. We then apply the induction hypothesis to the remaining list `l'` to produce `l''`, a sorted permutation of `l'`. The first and last element of `l''` are respectively denoted by `F` and `L`. As `H` is a point external to `l'` (and thus to `l''`), we have by definition `(between H F L)` ∨ `(between H L F)`. Hence two cases are to be studied:

- `(between H F L)`: we prove that `(cons H l'')` is a sorted permutation of `l`.
- `(between H L F)`: we prove that `(cons H (rev l''))` is a sorted permutation of `l`.

In both cases, most of the proof is performed using a common lemma that in turn applies Lemma `V-2`. □

In fact, the complete proof requires lots of tedious auxiliary operations on lists and proofs of their properties.

## 9 Conclusion

In this work we have studied the first two groups of Hilbert's axiomatics from a constructive point of view. In order to build a constructive specification, several variants for the same axioms and theorems have been proposed and compared. However, obtaining all of Hilbert results absolutely requires some purely classical features, such as decidability of some properties.

The specification itself was developed in the Gallina language of powerful system Coq. Generic definitions have been written, and numerous lemmas and theorems have been interactively formally proved using techniques such as induction. The proof scripts amounted to roughly 120 pages, about half of which for Theorem VI and its auxiliary types and lemmas.

This work only dealt with a very small part of Hilbert's book [13]. Even the study of the second group is far from being complete: polygonal lines, closed curves, notions of inside and outside still have to be introduced and treated. In the following groups, Hilbert introduces other fundamental notions such as angle and figure congruence. To our knowledge, these concepts have never been addressed from an intuitionistic point of view, and that would be a good challenge for the future.

As to the methodological aspects, the techniques we used to adjust classical specifications to the intuitionistic framework could probably be generalized and applied in fields other than geometry.

One may bring into question the advantages of axiomatic methods, like ours, with respect to polynomial algebraic methods [23,3]. When they are formalized and supported by proof assistants such as Coq, the former are as rigorous as the latter, as they do not let any exception case of the theorems or lemmas be eluded. Moreover, the former allow much easier understanding of the geometrical reasoning than the latter, which almost completely hide it. Also, axiomatic methods are not plagued by round-off issues. These are all good reasons to use symbolic approaches in CAD systems [6].

Conversely, general interactive proof assistants like Coq provide little help for proof discovery, whereas some geometry-dedicated algebraic systems are based on efficient mechanization processes [23]. In order to be competitive, automatic geometric provers based on axiomatic methods must probably be guided by semantic verification by comparison with a sketch [9]. This way is followed in some CAD systems [8]. Unfortunately, it is difficult to imagine systematic ways to reveal degenerate cases with these methods.

## References

1. Balbiani, P., Dugat, V., Fariñas del Cerro, L., Lopez, A.: Eléments de Géométrie Mécanique. Hermès (1994).
2. Barras, B. *et al.*: The Coq Proof Assistant Reference Manual (Version 6.3.1). INRIA (1999), `http://pauillac.inria.fr/coq/doc/main.html`
3. Chou, S.-C.: Mechanical Geometry Theorem Proving. D. Reidel (1988).
4. Coquand, T., Huet, G.: Constructions: A higher order proof system for mechanizing mathematics. EUROCAL ’85, Linz, LNCS **203**, Springer-Verlag (1985), 151–184.
5. Coquand, T., Paulin, C.: Inductively defined types. P. Martin-Löf and G. Mints, editors, COLOG-88, LNCS **417**, Springer-Verlag (1990), 50–66.
6. Dufourd, J.-F., Mathis, P., Schreck, P.: Geometric construction by assembling solved subfigures. Artificial Intelligence **99** (1998), 73–119.
7. Dufourd, J.-F., Puitg, F.: Functional specification and prototyping with combinatorial maps. Computational Geometry — Theory and Applications **16** (2000), 129–156.
8. Essert-Villard, C., Schreck, P., Dufourd, J.-F.: Sketch-based pruning of a solution space within a formal geometric constraint solver. Submitted (2000).
9. Gelernter, H.: Realization of a geometry theorem proving machine. Computers and Thought, Mac Graw Hill (1963), 134–163.
10. Girard, J.-Y., Lafont, Y., Taylor, P.: Proofs and Types. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (1989).
11. Heyting, A.: Intuitionism — An Introduction. North Holland (1956).
12. Heyting, A.: Axioms for intuitionistic plane affine geometry. Proceedings of an International Symposium on the Axiomatic Method with Special Reference to Geometry and Physics (1959), 160–173.
13. Hilbert, D.: Fondations de la Géométrie — Edition critique préparée par P. Rossier, CNRS, Dunod (1971).

14. Knuth, D.E.: Axioms and Hulls. LNCS **606**, Springer-Verlag (1992).
15. Paulin-Mohring, C.: Inductive Definition in the System Coq — Rules and Properties. Typed Lambda-Calculi and Applications, LNCS **664**, Springer-Verlag (1993).
16. Toussaint, G.: A new look at Euclid's second proposition. Technical Report No SOCS 90.21 (1990).
17. von Plato, J.: The axioms of constructive geometry. Annals of Pure and Applied Logic **76** (1995), 169–200.
18. von Plato, J.: Organization and development of a constructive axiomatization. LNCS **1158**, Springer-Verlag (1996), 288–296.
19. von Plato, J.: A constructive theory of ordered affine geometry. Indagationes Mathematicae N.S. **9**(4) (1998), 549–562.
20. Puitg, F., Dufourd J.-F.: Formal program development in geometric modelling. Current Trends in Applied Formal Methods, Boppard, LNCS **1641**, Springer-Verlag (1998), 62–76.
21. Puitg, F., Dufourd J.-F.: Formal specifications and theorem proving breakthroughs in geometric modelling. Theorem Proving in Higher Order Logics, Canberra, LNCS **1479**, Springer-Verlag (1998), 401–427.
22. Puitg, F., Dufourd J.-F.: Formalizing mathematics in higher logic: A case study in geometric modelling. Theoretical Computer Science **234** (M. Nivat, ed.), Elsevier Science (2000), 1–57.
23. Wu, W.-T.: Mechanical Theorem Proving in Geometries. Springer-Verlag (1994).

# Author Index