Some Notes on Algebraic Methods for Geometry Theorem Proving[1]

DONGMING WANG
Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria and Institute of Systems Science, Academia Sinica, Beijing, China

Abstract. A new geometry theorem prover which provides the first complete implementation of Wu's method and includes several Gröbner bases based methods is reported. This prover has been used to prove a number of non-trivial geometry theorems including several *large* ones with less space and time cost than using the existing provers. The author presents a new technique by introducing the notion of *normal ascending set.* This technique yields in some sense *simpler* non-degenerate conditions for Wu's method and allows one to prove geometry theorems using characteristic sets but Gröbner bases type reduction. Parallel variants of Wu's method are discussed; an implementation of the parallelized version of his algorithm utilizing workstation networks has also been included in our prover. Timing statistics for a set of typical examples is given.

Key words. Geometry theorem prover, Wu's method, Gröbner bases method, normalization, parallelization.

1. Introduction

Inspired by the pioneering work of Wu [15–17] algebraic methods for mechanical geometry theorem proving have been extensively studied [2, 5, 6, 14]. Research on various improvements of Wu's method and the application of Gröbner bases method has made possible to prove hundreds of non-trivial geometric theorems within reasonable computing time. Despite this success, many research problems on this subject however remain still. For example, there is even no complete implementation of Wu's general method due to the lack of polynomial factorization over algebraic number fields. Also, problems concerning *simpler* non-degenerate conditions, the inherent relationships between Wu's method and Gröbner bases method and the parallel variants of related algorithms have not, or not completely, been considered. In this paper we report our recent work on these problems. In concrete term, we have made a complete implementation of Wu's method in MAPLE system which uses a newly discovered factorization method over algebraic number fields in combination with that of Hu and Wang [4]. On this basis a new geometry theorem prover implementing Wu's method and several Gröbner bases-based methods has become available. We present a new technique by introducing the notion of *normal ascending set.* This technique yields in some sense simpler non-degenerate conditions for Wu's method and allows one to prove geometric theorems using characteristic sets but Gröbner bases type reduction. We shall also be concerned with the parallelization of Wu's method which takes into account a powerful splitting technique of Wu [18]. Various experimental results, in particular, on large examples, are given. Some unclear phenomena observed during our experiments are also pointed out.

[1]This work is supported by the Austrian Ministry of Science under ESPRIT Basic Research Action 3125 (MEDLAR).

The reader of this paper is assumed to be familiar with notions including class, leading variable, initial, rank, reduced, (weak, quasi-) ascending set, pseudo-remainder, basic set, characteristic set and the reducibility of an ascending set in the theory and method of characteristic sets and some basic notions on Gröbner bases. For all details, the reader is referred to [1, 7, 8, 12, 16, 17]. For any polynomial G, polynomial set PS and (quasi-, weak) ascending set AS, we shall denote the class of G by $\mathrm{cls}(G)$, the leading variable of G by $\mathrm{lvar}(G)$, the degree of G in x by $\deg_x(G)$, the initial of G by $\mathrm{ini}(G)$, the pseudo-remainder of G with respect to AS by $\mathrm{prem}(G, \mathit{AS})$, the set of common zeros of polynomials in PS by $\mathrm{Zero}(\mathit{PS})$, and $\mathrm{Zero}(\mathit{PS}) \setminus \mathrm{Zero}(G)$ by $\mathrm{Zero}(\mathit{PS}/G)$.

All experimental results presented in this paper are made in MAPLE 4.3 running on Apollo DN3500 workstations. The timings are recorded in CPU seconds and include those for garbage collection.

2. Wu's Method and Its Implementation

2.1. FORMULATION OF MECHANIZATION PROBLEM

Following Wu [17] the mechanization problem of geometry theorem proving consists mainly of three steps: The first is the algebraization of geometry, namely, reducing the problem of geometry theorem proving to purely algebraic problem by introducing number system and coordinate system, starting from the axiom system. Then the hypothesis and conclusion of a theorem can be expressed as sets of algebraic relations. The second step is to well order those algebraic relations which correspond to the hypothesis of the theorem and decide whether the algebraic relations corresponding to the conclusion can be deduced from the well-ordered algebraic relations according to a certain procedure. This is called the *mechanization* of geometry, while the procedure for determining the inference relation is called a *mechanization method*. The final step is to implement the method on computer in order to achieve the proof of each theorem.

Consider now those geometric theorems whose hypothesis and conclusion are given by algebraic equations only, viz,

$$\mathit{HYP} = \{h_1(x_1, \ldots, x_n), \ldots, h_s(x_1, \ldots, x_n)\} = 0$$

and

$$\mathit{CON} = \{g_1(x_1, \ldots, x_n), \ldots, g_t(x_1, \ldots, x_n)\} = 0,$$

where h's and c's are all polynomials in the variables $x_1, \ldots, x_n$ with integer coefficients while $x_1, \ldots, x_n$ are some geometric entities like coordinates of points and areas of triangles. As observed by Wu, theorems of this *equation-type* contain the major part of theorems in elementary geometries. As also pointed out precisely by Wu, most geometric theorems are true only *generically*, i.e., under certain non-degenerate conditions such as a triangle does not degenerate to a line. Therefore, the problem of proving theorems of equation-type consists in first finding some non-degenerate conditions $\Delta = \{d_1, \ldots, d_e\} \neq 0$ and then determining whether the conclusion CON is a formal consequence of the hypothesis HYP under the conditions $\Delta \neq 0$, i.e., determining whether for all polynomial $G \in \mathit{CON}$, any zero of HYP which is such that $\Delta \neq 0$ is also a zero of G in some extension of the associated geometry field.

Note in particular that the conditions $\Delta \neq 0$ cannot be arbitrary. They should indeed represent the non-degenerate cases of the theorem. A precise theoretical analysis of these conditions requires the aid of algebraic geometry, in particular, the notion of dimension and irreducible decomposition of algebraic varieties. See [17] for a detailed elaboration.

2.2. WU'S DECISION ALGORITHM

For the second step of the mechanization problem Wu introduced a powerful decision method which forms the essential part of his theory. We have presented in [13] an algorithmic version of his method which consists essentially of the following steps:

Wu_Theorem_Prover(Input: HYP, CON)

1. Compute a characteristic set CS of HYP. If it is contradictory, then the hypothesis of the theorem is self-contradictory and the procedure terminates. Otherwise, let $I_1, \ldots, I_r$ be the initials of the polynomials in CS and proceed to the next step.

2. Form the remainder set RS of all polynomials $G \in CON$ with respect to CS. If $R \equiv 0$ for all $R \in RS$, then the theorem is true under the subsidiary conditions $\Delta = \{I_1, \ldots, I_r\} \neq 0$ and go to Step 4. Otherwise, proceed to the next step.

3. Verify the irreducibility of CS. If CS is irreducible, then the theorem is not true under the subsidiary conditions $\Delta \neq 0$ and the procedure terminates. Otherwise, decompose HYP into irreducible components and go to Step 1 by considering all those components which have the same dimension as that of HYP.

4. Check whether the subsidiary conditions $\Delta \neq 0$ are indeed non-degenerate conditions and consider further the degenerate cases as particular theorems by adjoining the condition $D = 0$ for each $D \in \Delta$ to the hypothesis as it is interesting.

As the subsidiary conditions represent in general just the non-degenerate conditions of the theorem, we did not include fully the last step in our description and implementation of the algorithm. The returned conditions are subject to further analysis for making clear whether they are indeed the non-degenerate conditions after the theorem has been proved. This is usually necessary only if one is interested in these particular cases.

2.3. SOME STRATEGIES AND TECHNIQUES

Although the above algorithm is designed already for practical use, several strategies and techniques may be taken into account for the sake of efficiency. Let us explain them as follows.

The basic operation underlying the algorithm Wu_Theorem_Prover is the pseudo-division of two polynomials F and G with respect to some variable x. While dividing G by F, one gets a remainder formula of the form

$$I^s \cdot G = Q \cdot F + R,$$

where I is the leading coefficient of F in x. The integer s is determined to be the smallest possible so that the formula holds true according to Wu. This is important for reducing the degree and number of terms of the remainder R. Instead of I^s we now use $I_1^{s_1} \cdots I_e^{s_e}$, where $I_1, \ldots, I_e$ are all the distinct irreducible factors of I, and choose the smallest $s_1, \ldots, s_e$ for the corresponding remainder formula to hold true. For this modification the determination of R requires the computation of GCD (greatest common divisor) and takes thus more time. As the modification often removes some redundant factors, further computation will benefit, in particular, if the occurring polynomials become large.

In practice one need not compute exactly the characteristic set at the first step. Instead, a certain triangular form (quasi-characteristic set) is often used. Namely, the so-called *medial set* computed by either the algorithm triset or the algorithm CharacteristicSetN with quasi-basic set described in [12] is usually sufficient (the term characteristic set used hereafter may also be such triangular form). The exact characteristic set need be computed only if the remainder of some conclusion polynomial with respect to the triangular form is non-zero. In this case, one has to perform further decomposition in the third step.

A powerful technique was often used by Wu (see [18] for example) for proving difficult theorems on small computers. The idea is based on an observation illustrated as follows. Let $CS = \{C_1, \ldots, C_r\}$ be the characteristic set of the hypothesis polynomial set *HYP*. As to confirm the theorem, the remainder R_i of the conclusion polynomial g_i with respect to *CS* is expected to be 0. Suppose that the successive formation of the remainder has been done up to some j so that we have

$$R_i = \text{prem}(g_i, \{C_{j+1}, \ldots, C_r\}).$$

If R_i contains a variable x whose order precedes the leading variable of C_j, then all remainders of the coefficients of R_i in x with respect to $CS' = \{C_1, \ldots, C_j\}$ are *usually* 0. Therefore, the formation of R_i with respect to CS' can be split to that of all such coefficients with respect to CS'. This splitting technique has been described in [13] as an algorithm, named *Sprem*, for computing the pseudo-remainder. Its use not only improves the algorithm Wu_Theorem_Prover using serial computation for large problems but also allow us to introduce parallel features as will be presented in Section 4. We note that, to make the splitting possible we should take the parameters under consideration, namely, order the parameters with higher rank as long as the characteristic set is still ascending. In this way the splitting happens for almost all examples. In our implementation this arrangement of variables is made automatically. However, the use of the splitting technique is recommended only for large problems. Extensive use of this technique for small problems may slow down the computation.

During both the computation of characteristic set and the successive formation of pseudo-remainders, some redundant factors appear often. It is advantageous to remove these factors at a certain stage by examining the initials of the polynomials occurring in the computation and the contents of the polynomials (with respect to leading variable) in the characteristic set.

2.4. POLYNOMIAL FACTORIZATION OVER ALGEBRAIC NUMBER FIELDS

The third step of Wu's decision algorithm requires polynomial factorization over successive algebraic extensions of the basic field. This is generally considered as a difficult problem. Except for Chou's implementation [2] which can factorize polynomials of degree 2, all other implementations of Wu's method do not include this step. The general factorization algorithms are too complicated and not available in current computer algebra systems. MAPLE can only factorize univariate polynomials with rational coefficients over the algebraic number fields obtained by adjoining roots of univariate minimal polynomials with rational coefficients to the rational field $\mathbf{Q}$. This is somewhat far from our requirement in which both polynomials to be factorized and the minimal polynomials are multivariate with coefficients in some algebraic extension field of $\mathbf{Q}$.

In [4] we presented a factorization method which is quit suitable for our purpose. The author has implemented this method in MAPLE. It is efficient for factorizing polynomials of degree 2 and 3, while degrees of the minimal polynomials are also not too high. However, the method is still too slow for polynomials of higher degree. Recently, we have found another method which can be used for factorizing polynomials of rather high degree and reduces dramatically the difficulty of the factorization problem. Combining these two methods we are now able to factorize most polynomials encountered in our test. See [13] for some examples on irreducible decomposition which requires polynomial factorization of higher degree over algebraic number fields. The full detail of our new method and further experimental results will be reported elsewhere.

To our present attention of geometry theorem proving, the factorization of polynomials over algebraic number fields, as a matter of fact, is unnecessary in general. Some researchers have considered the need of factorization as a defect of Wu's method and tried to avoid it. This is, however, unreasonable. When factorization is indeed required, the geometric configuration corresponding to the hypothesis of the theorem is composed always of several components. In this case, the theorem is often true for some components but not true for the others. A method without factorization can only say that the theorem is not true, but no other information. Nevertheless, in practical case, to determine for which component the theorem does not hold is even more important. If, on the other hand, we turn our attention to prove theorems including all degenerate cases, or to determine the simplest non-degenerate conditions, or to disprove theorems, further decomposition of the hypothesis polynomial set must be considered. Then polynomial factorization is absolutely necessary. In addition, an efficient factorization method provides also means for solving many other geometric problems beyond theorem proving.

2.5. IMPLEMENTATION OF A NEW PROVER

The author has implemented the algorithm Wu_Theorem_Prover in the MAPLE system together with the strategies discussed in Subsection 2.3. The MAPLE function factor is used for factorization of polynomials over rationals and our newly discovered method in combination with that presented in [4] has been implemented for factorizing polynomials over algebraic number fields. This provides thus the first complete implementation of Wu's method. Our prover also includes three Gröbner bases based decision algorithms given in [5, 6] and the method using normalized characteristic sets with both pseudo-division and Gröbner bases type reduction described in the next section. A parallelized version of Wu's decision algorithm (see Section 4) up to 4 processors has also been implemented on the basis of the MIMD model. It can be run on several distributed workstations connected by a local network. The data communication is realized through the network and simple use of the MAPLE functions save and read for writing and reading data to/from disc files in an internal form. Twenty predicates for translating geometric statements into algebraic equations have been included. More will be added soon. The implemented provers has been considered as part of a general geometry problem solver (GEOPER) under development. It is now running with MAPLE 4.1 on MicroVAX II under VMS operating system and with MAPLE 4.3 on Apollo DN3500/DN4000/DN10000 under UNIX operating system. See Table 1 in Subsections 3.5 for timing statistics on some typical examples. More details about the implementation issue will be published elsewhere.

3. Normalization of Ascending Sets and Its Applications

3.1. NORMALIZATION OF ASCENDING SETS

Consider a non-contradictory (weak) ascending set AS. Let us make a change of variables so that it can be written in the form

$$AS \qquad \begin{array}{l} A_1(u_1,\ldots,u_d,y_1), \\ A_2(u_1,\ldots,u_d,y_1,y_2), \\ \qquad \cdots\cdots \\ A_r(u_1,\ldots,u_d,y_1,y_2,\ldots,y_r). \end{array}$$

Denote the initial of A_i by I_i which is usually in $\boldsymbol{K}[u_1,\ldots,u_d,y_1,\ldots,y_{i-1}]$ for each i. If, in particular, I_i is in $\boldsymbol{K}[u_1,\ldots,u_d]$ for all i, we shall call AS a *normal (weak) ascending set*. Conventionally, any contradictory ascending set is considered as a normal ascending set.

THEOREM 1. *Given any non-contradictory (weak, quasi-) ascending set AS as above, the algorithm* Normal *below, if does not fail, computes a normal weak ascending set AS^* and a polynomial F such that*

$$\mathrm{Zero}(AS/F) \subset \mathrm{Zero}(AS^*), \quad \mathrm{Zero}(AS^*/J^*) \subset \mathrm{Zero}(AS/J), \tag{3.1}$$

in which J and J^* are respectively the products of initials of the polynomials in AS and AS^*.

```
Normal(Input: AS; Output: [AS*, F])
  {A_1, ..., A_r} := AS; F := 1
  if r = 1 then AS* := {A_1}
  else
    for i from r − 1 step −1 to 1 do
      I_r = ini(A_r)
      if cls(I_r) = cls(A_i) then
        y_i := lvar(A_i)
        compute H = gcd(A_i, I_r, y_i) and Q such that H = P A_i + Q I_r
        if cls(H) = cls(A_i) then
          D := Remove(A_i/H, H, y_i)
          F := F H
          if cls(D) = cls(A_i) then A_i := D
          else AS* := Fail; stop
        y_r := lvar(A_r); m := deg_{y_r}(A_r); B := A_r − I_r y_r^m
        A_r := prem(H y_r^m + Q B, A_i)
    [AS_0, F_0] := Normal({A_1, ..., A_{r−1}})
    F := F F_0
    if AS_0 = Fail then AS* := Fail
    else AS* := AS_0 ∪ {A_r}
```

where the subalgorithm Remove *is written as*

```
Remove(Input: A, B, x; Output: D)
  G := gcd(A, B, x)
  if deg_x(G) = 0 then D := A
  else D := Remove(A/G, G, x)
```

and $\gcd(A,B,x)$ denotes the greatest common divisor of A and B with respect to x.

Proof. As in the algorithm, we denote the number of polynomials in AS by r. Then $r = 1$ is a trivial case. For $r > 1$, let us set $AS_r = AS$ first.

Consider the repetition statement for i from $r-1$ to 1, decreasing 1 in each step. Let $A_{i+1\ j}$ be the jth element of AS_{i+1} and $I_{i+1\ j} = \mathrm{ini}(A_{i+1\ j})$ with, in particular, $I_{i+1\ r} \in \boldsymbol{K}[u_1, \ldots, u_d, y_1, \ldots, y_i]$. If $I_{i+1\ r}$ has its class less than that of $A_{i+1\ i}$, we set $AS_i = AS_{i+1}$ and proceed with the iteration for the next i. Suppose, otherwise, that $\mathrm{cls}(I_{i+1\ r}) = \mathrm{cls}(A_{i+1\ i})$. Then we have two cases:

Case 1. $A_{i+1\ i}$ and $I_{i+1\ r}$ are relatively prime with respect to the leading variable y_i of $A_{i+1\ i}$, i.e., the greatest common divisor H of $A_{i+1\ i}$ and $I_{i+1\ r}$ with respect to y_i is in $\mathbf{K}[u_1, \ldots, u_d, y_1, \ldots, y_{i-1}]$. Then there are polynomials $P, Q \in \boldsymbol{K}[u_1, \ldots, u_d, y_1, \ldots, y_i]$ such that

$$(3.2) \qquad PA_{i+1\ i} + QI_{i+1\ r} = H \in \boldsymbol{K}[u_1, \ldots, u_d, y_1, \ldots, y_{i-1}].$$

Writing $A_{i+1\ r} = I_{i+1\ r} y_r^m + B$ and multiplying both sides of (3.2) by y_r^m, one gets

$$(3.3) \qquad QA_{i+1\ r} = Hy_r^m + QB - PA_{i+1\ i}\ y_r^m.$$

Set $A_{ir} = \mathrm{prem}(Hy_r^m + QB, A_{i+1\ i})$ and $A_{ij} = A_{i+1\ j}$ for $j = 1, \ldots, r-1$. Clearly, A_{ir} and $A_{i+1\ r}$ have the same class $d+r$, so the classes of $A_{i1}, \ldots, A_{ir}$ are steadily increasing, i.e., $AS_i = \{A_{i1}, \ldots, A_{ir}\}$ is at least a quasi-ascending set. We want to show that

$$\mathrm{Zero}(AS_{i+1}/F_i) \subset \mathrm{Zero}(AS_i), \quad \mathrm{Zero}(AS_i/J_i) \subset \mathrm{Zero}(AS_{i+1}/J_{i+1}),$$

with $F_i = 1$ and J_k the product of initials of the polynomials in AS_k for $k = i, i+1$.

Since A_{ir} can be written as a linear combination of $A_{i+1\ i}$ and $A_{i+1\ r}$ with polynomial coefficients, the first relation holds obviously. Before showing the second, we see that the initial I_{ir} of A_{ir} is the product of H and some factors of the initial of $A_{i+1\ i}$. Hence, for any $z \in \mathrm{Zero}(AS_i/J_i)$ we have $A_{i+1\ j} = A_{ij} = 0, I_{i+1\ j} = I_{ij} \neq 0$ for $i = 1, \ldots, r-1$ and $I_{ir} \neq 0, H \neq 0$ at z. From (3.3) and the determination of A_{ir} one can see that $QA_{i+1\ r} = 0$ at z. By (3.2) we have $QI_{i+1\ r} \neq 0$ at z. It follows that $A_{i+1\ r} = 0, I_{i+1\ r} \neq 0$ at z. Therefore, $z \in \mathrm{Zero}(AS_{i+1}/J_{i+1})$ and we have shown the second zero relation.

Case 2. $A_{i+1\ i}$ and $I_{i+1\ r}$ are not relatively prime with respect to the leading variable y_i of $A_{i+1\ i}$. In this case, they have a common divisor of class $d+i$. Let us simply remove all possible factors of H, the GCD of $A_{i+1\ i}$ and $I_{i+1\ r}$ with respect to y_i, from $A_{i+1\ i}$ as done by the subalgorithm Remove, denote the obtained polynomial by D, and set $F_i = H$. If D has its class less than that of $A_{i+1\ i}$, then the algorithm terminates and *Fail* is returned. Otherwise, let $AS' = \{A_{i+1\ 1}, \ldots, A_{i+1\ i-1}, D, A_{i+1\ i+1}, \ldots, A_{i+1\ r}\}$, in which the classes of polynomials are steadily increasing. Denote as usual the product of initials of the polynomials in AS' by J'. Then, we have $\mathrm{Zero}(AS_{i+1}/F_i) \subset \mathrm{Zero}(AS')$ and $\mathrm{Zero}(AS_{i+1}/J_{i+1}) = \mathrm{Zero}(AS'/J')$. As D and $I_{i+1\ r}$ are now relatively prime with respect to y_i, the problem is reduced, by considering AS' as AS_{i+1}, to Case 1. Therefore, we can determine an AS_i such that

$$\mathrm{Zero}(AS_{i+1}/F_i) \subset \mathrm{Zero}(AS') \subset \mathrm{Zero}(AS_i),$$

$$\mathrm{Zero}(AS_i/J_i) \subset \mathrm{Zero}(AS'/J') = \mathrm{Zero}(AS_{i+1}/J_{i+1}).$$

In this way the repetition statement, if does not fail, produces a sequence of quasi-ascending sets $AS = AS_r, \ldots, AS_1$ satisfying

$$\mathrm{Zero}(AS_r/F_{r-1}) \subset \mathrm{Zero}(AS_{r-1}), \ldots, \mathrm{Zero}(AS_2/F_1) \subset \mathrm{Zero}(AS_1),$$

$$\mathrm{Zero}(AS_1/J_1) \subset \cdots \subset \mathrm{Zero}(AS_{r-1}/J_{r-1}) \subset \mathrm{Zero}(AS_r/J_r).$$

Setting $\bar{F} = F_{r-1} \cdots F_1$, we have therefore

$$\text{Zero}(AS/\bar{F}) = \text{Zero}(AS_r/\bar{F}) \subset \text{Zero}(AS_1),$$

$$\text{Zero}(AS_1/J_1) \subset \text{Zero}(AS_r/J_r) = \text{Zero}(AS/J).$$

Let $AS_1 = \{A_{11}, \ldots, A_{1r}\}$ and $AS_1' = \{A_{11}, \ldots, A_{1\ r-1}\}$. Notice first that A_{1r} has its initial in $\boldsymbol{K}[u_1, \ldots, u_d]$. Since AS_1' contains $r-1$ polynomials, we can compute, if not fail, a normal weak-ascending set AS_0 and a polynomial F_0 by induction such that

$$\text{Zero}(AS_1'/F_0) \subset \text{Zero}(AS_0), \quad \text{Zero}(AS_0/J_0) \subset \text{Zero}(AS_1'/J_1'),$$

where J_0 and J_1' are respectively the products of initials of the polynomials in AS_0 and AS_1'.

Now, let $AS^* = AS_0 \cup \{A_{1r}\}$ and $F = \bar{F}F_0$. Then the zero relations in (3.1) hold. Certainly, all initials of the polynomials in AS^* are in $\mathbf{K}[u_1, \ldots, u_d]$. Therefore, these initials are all reduced with respect to those polynomials in AS^*. Namely, AS^* is a weak ascending set and the theorem is proved.

We note that it is easy to get a normal ascending set from a normal weak ascending set AS, in replacing A_i by the remainder A_i^* of A_i with respect to $\{A_1, A_2^*, \ldots, A_{i-1}^*\}$ for $i = 2, \ldots, r$; for the replacement does not change the class of each A_i. If we denote the obtained normal ascending set by AS^*, then the zero relations in (3.1), with $F = 1$, hold as well.

One can prove that the algorithm Normal always succeeds with $F = 1$ if AS is irreducible. Otherwise, it can either return an additional non-constant polynomial F or fail. This is actually not a week point of the algorithm, but provides on the contrary even more important information. In the whole theory of characteristic sets, the initials of polynomials in an ascending set play a special role. While considering the zero set of an ascending set, we always exclude those zeros which make the vanishing of some initials. From the proof, one sees that if a non-constant F is returned, it says that some polynomials and the initials of the others in the ascending set have common divisors. This happens, typically, in the case such as $AS = \{x_1(x_1 - 1), x_1x_2 + 1\}$. Then the removal of factors like x_1 is just what we expect to do. For situation where the algorithm fails, we consider a typical example $AS = \{x_1^2 - 1, x_1x_2^2 + x_2^2, x_2x_3 + 2\}$. With respect to the ordering $x_1 \prec x_2 \prec x_3$, AS is an ascending set with classes $1, 2, 3$ and $J = (x_1 + 1)x_2$, the product of initials of the three polynomials. When the set $\text{Zero}(AS/J)$ is considered, the factor x_2^2 of the second polynomial becomes apparently redundant. While applied to this example, the algorithm Normal fails due to the fact that AS, after the removal of this factor, is no longer an ascending set, but its failure just tells us the possibility of removing such a factor and provides in addition a factorization of the polynomial $x_1x_2^2 + x_2^2$. Ascending sets like AS appear however abnormal and are not what we usually would like to have. The factorization of polynomials such as $x_1x_2^2 + x_2^2$ in this example allows us to decompose the considered ascending set into simpler ones. In other words, the failed cases can be further considered. For a more complete result, we have

THEOREM 2. *Given any polynomial set PS, one can compute a list of normal ascending sets CS_i such that*

$$\text{Zero}(PS) = \bigcup_i \text{Zero}(CS_i/J_i),$$

where J_i is the product of initials of the polynomials in CS_i for each i.

We omit the proof of this theorem. Since for application to geometry theorem proving the abnormal cases happen rather seldom, the algorithm Normal is sufficient for our present purpose. If the algorithm returns some removed factors, we treat them generally as possible non-degenerate conditions.

The algorithm Normal has a top-down scheme. It can be modified to have a bottom-up scheme as well. In its implementation, we also remove some redundant factors such as those which are the initials of some polynomials and meantime the factors of other polynomials for efficiency as well as for brevity. The output weak ascending set AS^* of the algorithm Normal will be called the *normalized (weak) ascending set* of AS, while the procedure for getting AS^* from AS will be called the *normalization* of AS.

3.2. SIMPLIFICATION OF ASCENDING SETS

In this and the following subsections we present some applications of the normalization of ascending sets. The first of such applications is simplification. In many instances the normalized ascending set has a much simpler form than that of the original one. Take for example the following ascending set

$$\begin{aligned} AS = \{2x^2 - 1,& \\ &22540009426896317848166025972903447015109 2xy^2 \\ &- 132036318262485267264375273692698717304108yx \\ &+ 193362711280036780235451438281814822 28562x \\ &- 159381935137720544849472622882685111499038y^2 \\ &+ 933637760064779112172849860363357528469 84y \\ &- 136728084376176387925486745949931970974 65\}, \end{aligned}$$

where the big integer coefficients are relatively prime. To our great surprise, the normalization of this ascending set leads to a much simpler one

$$AS^* = \{2x^2 - 1, -4yx + 4x + 2y^2 - 4y + 3\},$$

which has the same set of zeros as AS does.

This simplification is very critical for our new factorization method which we shall explain in our further work. However, normalization cannot always produce a simpler ascending set as in this example. It is not very clear to the author in which occasion a simpler form can be obtained.

3.3. SIMPLER NON-DEGENERATE CONDITIONS

The second application of normalization is for geometry theorem proving in which the non-degenerate conditions are desired to be simpler in some sense. One criterion for this simplicity can be that the polynomial inequations involve less variables, in particular, only the parameters. This is considered to be reasonable because the theorem should be generally true for arbitrary parameters. While those parameters are constrained by some polynomial equalities, the theorem can thus be considered as degenerate cases.

For Hilbert's pure intersection point theorems as well as their extension by the author [9, 11], non-degenerate conditions can always be found as inequations involving only parameters as variables. For Wu's general method and its various modifications it is not possible to get such conditions. Now, we use instead the normalized characteristic set in Wu's decision algorithm. If the remainders of all conclusion polynomials are zero, then the theorem is proved under the conditions that the product of initials of the polynomials in the normalized characteristic set is non-zero. This is true simply because the conclusion polynomials multiplied by some factors of these initials can be written as linear combinations of the hypothesis polynomials. Therefore, the non-degenerate conditions as polynomial inequations now involve only parameters and can be considered simpler in this sense. Also, in many cases the conditions obtained in this way consist of a smaller number of inequations.

EXAMPLE 1 (Olympiad, 1977 [6]). The hypothesis of this theorem can be algebraized as a set of 7 polynomial equations. The quasi-characteristic set of the corresponding polynomial set consists of 7 polynomials shown below:

$$\begin{aligned}
c_1 &= y_1^2 + 2y_1u - 2u^2,\\
c_2 &= (y_1^2 + 2y_1u + 2u^2)y_2 - 2u^2y_1 - y_1^2u,\\
c_3 &= -uy_3 + y_1y_2 - y_1u + uy_2,\\
c_4 &= 2y_4 - y_1,\\
c_5 &= 2y_5 - y_1,\\
c_6 &= -(y_1^2 + 2y_1u + 2u^2)y_6 + 2y_1u^2 + y_1^2u,\\
c_7 &= -uy_7 - y_1y_6 + y_1u - uy_6.
\end{aligned}$$

The theorem can be proved by computing the remainders of the two conclusion polynomials, providing the non-degenerate conditions

$$u(y_1^2 + 2y_1u + 2u^2) \neq 0.$$

Now, normalization of the above quasi-characteristic set produces another quasi-ascending set of the form

$$\{c_1, 2y_2 - u, c_3, c_4, c_5, 2y_6 - u, c_7\},$$

which yields a simpler non-degenerate condition

$$u \neq 0.$$

EXAMPLE 2 (Secant Theorem). We introduce the coordinates of points as $A(-u_1, 0)$, $B(u_1, 0)$, $C(u_2, u_3), O(0, y_1), D(u_4, y_2)$ and $T(y_3, 0)$. Then the hypothesis and conclusion of this theorem can be expressed as

$$\begin{aligned}
h_1 &= 2u_3y_1 - u_2^2 + u_1^2 - u_3^2, && O \text{ is the circumcenter of } \Delta ABC\\
h_2 &= -y_2^2 + 2y_1y_2 + u_1^2 - u_4^2, && D \text{ is on the circle } O\\
h_3 &= (-y_2 + u_3)y_3 + y_2u_2 - u_4u_3, && T \text{ is on the line } CD
\end{aligned}$$

and

$$c = (y_3 + u_1)^2(y_3 - u_1)^2 - [(y_3 - u_2)^2 + u_3^2][(y_3 - u_4)^2 + y_2^2],$$
$$|TA|^2|TB|^2 = |TC|^2|TD|^2$$

where the hypothesis polynomial set $HYP = \{h_1, h_2, h_3\}$ is already an weak ascending set. Whether to use this set or its strong sense, the theorem may be proved under the subsidiary conditions

$$u_3 \neq 0, \quad y_2 \neq u_3.$$

In the second condition the dependent y_2 is however involved. If the weak ascending set HYP is normalized, then we get a weak ascending set

$$CS^* = \{h_1, h_2, -(u_4 + u_2)y_3 + u_2u_4 + y_2u_3 + u_1^2\}$$

with a non-trivial factor $u_4 - u_2$ removed. Thus, the theorem is proved under the subsidiary conditions

$$u_3 \neq 0, \quad u_4 \neq -u_2, \quad u_4 \neq u_2.$$

These conditions correspond to the same geometric meaning as that of $u_3 \neq 0, y_2 \neq u_3$, but all inequations involve only parameters.

The geometric meaning of these conditions is quite clear: $u_3 \neq 0$ says that the triangle ABC (respectively, the circle O) does not degenerate to a line; the condition $u_4 \neq -u_2$ means that the line CD is not parallel to line AB which ensures the existence of the intersection point, while $u_4 \neq u_2$ says that the point D does not coincide with the point C. However, in the last degenerate case the theorem is still true. It can be proved by taking the condition $u_4 = u_2$ into account to form the special (tangent) theorem.

3.4. A COMBINATION METHOD FOR THEOREM PROVING

As the third application the normalization of ascending sets provides the possibility of using characteristic sets but Gröbner bases type reduction for geometry theorem proving. This is done by computing first a normalized characteristic set of the hypothesis polynomial set and then the normal forms of the conclusion polynomials modulo the characteristic set. Namely, we may design a structurally simple algorithm as follows

```
N_Theorem_Prover(Input: HYP, CON; Output: SC, Yes or NC)
    CS := characteristic set of HYP
    if CS is contradictory then stop [SC]
    [CS*, F] := Normal(CS)
    if CS* = Fail then stop [NC]
    RS := φ
    for each G ∈ CON while RS = φ do
        R := normal form of G modulo CS* in Q(u)[y]
        if R ≠ 0 then RS := RS ∪ {R}
    if R = φ then [Yes] else [NC]
```

where *SC*, *Yes*, and *NC* mean respectively that the hypothesis of the theorem is self-contradictory, the theorem is proved to be true under some subsidiary conditions including $F \neq 0$, and the theorem is not confirmed.

The termination and correctness of the above algorithm are obvious. It is quite surprising that many geometric theorems can be confirmed by this algorithm. The inherence of this fact is still unclear to the author. While the characteristic set is irreducible, its normalization, in many instances, but not all, is the same as the Gröbner basis in $\mathbf{Q}(u)[y]$ with respect to the purely lexicographical ordering. Even if not, the theorem may be proved as well.

Our motivation of using the algorithm N_Theorem_Prover came from the feeling that the formation of successive pseudo-remainders and the computation of Gröbner bases require respectively more time than that of the formation of normal forms and the computation of characteristic sets. Our experiments (below) have shown that it is not so. However, the algorithm still remains significant and can be probably further improved. Its current implementation yields a result as good as the fully Gröbner bases-based algorithm. Similar to the latter, we are able to determine the non-degenerate conditions in our algorithm by collecting the denominators appearing in the representation of the conclusion polynomials in terms of polynomials in the characteristic set. This process is certainly time-consuming and has not been included in our implementation.

3.5. EXPERIMENTAL RESULTS

The following table shows the computing time (in seconds) of the above algorithm (N-GB) and our implementation of Wu's standard method (WU), Wu's method using normalized characteristic sets (N-WU) and one Gröbner bases-based method (GB), named RED in [6]. >7200 means that the computation was stopped after 7200 CPU seconds, while *** indicates that the computation was rejected by MAPLE for the reason "object too large."

Table 1. Timing Statistics for 4 Algorithms

Theorem	WU	N-WU	N-GB	GB
Ceva	2.867	3.583	7.133	20.916
Simson	3.583	2.683	8.167	25.934
Pappus	3.967	3.083	13.083	23.467
Secant	4.066	3.033	9.317	16.600
Ptolemy	4.700	3.367	25.866	28.967
Feuerbach	4.950	5.233	23.300	47.700
Butterfly	6.500	111.366	195.100	98.166
Pascal	11.417	732.367	1668.634	1236.616
EX401	30.050	***	>7200	>7200
Gauss Point	43.350	38.684	407.167	524.367
Morley	113.884	117.350	154.334	361.783
Steiner-Wu	161.467	68.833	685.616	459.266
EX453	200.683	>7200	>7200	382.316
Pascal-Conic	282.966	***	***	>7200
Brianchon	329.600	***	>7200	4078.400
EX144	396.417	3075.584	11801.067	>7200

In this table, EX144, EX401 and EX453 refer to the examples given in [2]. For some theorems shown above the conclusion contains more than one statement. The reported timings are those for proving all the statements. The Pascal theorem is proved for circle, while the Brianchon theorem is proved for ellipse. From this table, one sees that for geometry theorem proving Wu's standard decision algorithm (with our improvements at the implementation level) is still the fastest one. Our research effort has greatly reduced the computing time and space for large problems in comparison with the existing work (cf. [2]). Except for this method, all the other three have difficulty to solve the large problems. This is due to the normalization of ascending sets (which is rather slow and can produce very large polynomials for some problems) and the computational expenses on Gröbner bases and normal forms.

4. Parallelization of Geometry Theorem Proving

4.1. PARALLEL VARIANTS OF WU'S METHOD

The algorithm Wu_Theorem_Prover sketched in Section 2 consists of computing first the characteristic set CS and then the remainders of conclusion polynomials $G \in CON$ with respect to CS. In this algorithm parallel features may be introduced at several locations. First, we can use the parallelized algorithm to get the characteristic set CS for which a detailed discussion has been given in [13]. The second variant is to compute the remainders of all polynomials in CON with respect to CS in parallel instead of the sequential formation of RS. This is significant when CON contains more than one polynomial. We should note that among the theorems considered so far CON has often only one or two polynomials. This is due to the fact that those theorems are mainly taken from plane Euclidean geometry. While in three or higher dimensional geometry, more polynomials must be contained in CON.

The third parallel variant can be made by taking into account a powerful splitting technique of Wu. Namely, we split some intermediate remainder $\bar{G}$ which is got from the successive pseudo-division of the conclusion polynomial G by some polynomials in CS, if possible, into several polynomials $H_1, \ldots, H_h$ as illustrated in Section 2. If such splitting happens, we can then compute the further remainders of all H_j with respect to the remaining polynomials in CS synchronously.

Finally, if the remainder of some G with respect to the characteristic set CS is non-zero while CS is reducible, the proof of the theorem needs irreducible decomposition. Therefore, the parallel variant for this step can be made as well. This is similar to the parallelization of computing irreducible characteristic series discussed in detail in [13].

4.2. EXPERIMENTAL RESULTS

Based on these natural variants we have described in [13] a parallelized version of the algorithm Wu_Theorem_Prover. In additions to those given there we include other six test examples as Table 2 below to show the parallel computing time in comparison with that for sequential computation.

Table 2. Timing Statistics for Parallel Computing

Theorem	Serial	2 Processors	4 Processors
Olympiad	5.166	2.767	—
Problem 9	7.280	3.900	—
Problem 10	8.901	4.517	—
Tetrahedron	25.083	15.817	10.517
EX401	30.050	16.533	15.317
Gauss Point	43.350	28.117	21.017
Morley	113.884	57.433	—
Steiner-Wu	161.467	75.317	—
EX453	200.683	122.800	—
Pascal-Conic	282.966	231.417	159.767
Brianchon	329.600	178.600	181.334
Six-Point	>7200	>7200	—

Here problems 9 and 10 refer to those given in [13]. — indicates that we did not try that case with 4 processors; for only 2 processors are necessary in the present implementation. If we did, the other 2 processors must remain free.

4.3. ENCOUNTERED PROBLEMS

Parallel computation can reduce at average the computing time up to 42.3% with two processors and up to 49.4% with four processors. The first percentage is rather notable, whereas the second is not as ideal as we expected. The main reason for this is the unfull use of parallelism in our current implementation. At some stage we are unable to introduce parallel features to make all processors busy. In that case the computation is carried out still by one or two processors. Besides this respect, there are several encountered problems concerning the use of parallelism. We may discuss at least two as follows. In parallel computation, one has to manage the data communication. This costs of course a certain amount of time, especially, in the case of using many processors while the problem is smaller. In our implementation, data communication is realized through a local network and the reading and writing of files. Since MAPLE provides an efficient facility for reading and writing data from/to the disc files in its internal form. The data communication problem for four processors, if not too frequent, is not very serious.

The second problem concerns the practical computational complexity. Without a deep understanding of this issue, we are unable to optimally partition the jobs. For example, there are three jobs a, b and c for which the computation take 5, 90 and 100 seconds respectively and we need to distribute these jobs to two processors A and B. If we cannot estimate the computing time in advance and distribute just randomly the jobs, for instance, a to A and b, c to B, then about 200 seconds are required for the computation. If, on the other hand, we can predicate the amount of computing time for a, b and c to have an order $time(a) < time(b) < time(c)$ and distribute properly the jobs a, b to A and c to B, then the computing time required may be approximately 100 seconds, which speeds up the computation one time. This simple example demonstrates that the partition of jobs based on practical complexity analysis is of key importance in parallel processing and affects dramatically the efficiency. To avoid an improper partition, one way is to distribute only one job to each processor each time. The processor picks up another job while the current job has been done. This can bypass the complexity estimation, at least for the above example, but will cause expensive data communication in the case of many jobs. In our current implementation, the jobs are distributed to each processor only once and in an essentially random manner.

5. Further Remarks

In this section we give some remarks on a few large examples which are taken from various references and considered to be typical and representative. We shall use $maxt$, the maximal number of terms of polynomials occurring in the reductions, as an index to reflect the space complexity of a proof. This index can be used in particular for comparing different proofs when the comparison of computing time becomes impossible due to the use of different type of machines.

5.1. MORLEY'S THEOREM

This is one of the most difficult geometric theorems which was successfully proved by Wu [16]. Wu proved it with $maxt = 1960$ by using a tricky formulation of the problem together with his splitting technique. The theorem was proved by Gao [3] with $maxt \doteq 400$ using trigonometric functions. Chou [2] was able to prove the theorem with $maxt = 567$ using a formulation following Wu's trick. We have proved again the theorem with $maxt = 416$ using a similar formulation. What we can achieve now is that the $maxt$ can be reduced to 96 if we use the normalized characteristic set. However, we do not benefit with respect to computing time, but we do get simpler non-degenerate conditions. Note that all the above mentioned $maxt$s are only for one conclusion statement, while the timings shown in tables 1 and 2 are for the proof of two conclusion statements. The parallel computation with 2 processors on this example has ideally reduced the computing time to a half.

5.2. PASCAL-CONIC THEOREM

This theorem was discovered and proved by Wu in 1980 (see [18]). In Wu's proof which uses his splitting technique the $maxt$ is 1346. With the same formulation as Wu, we can prove the theorem with $maxt = 3432$ and CPU time $= 353.817$ seconds without using the splitting technique and with $maxt = 1346$ and CPU time $= 282.966$ seconds using the splitting technique. We are able to use more parallel processors for this example as the intermediate remainder can be split into 11 polynomials. The parallel computation has reduced the time up to 18.2% by two processors and 43.5% by four processors (see Table 2).

5.3. EX144, EX401 AND EX453

These are some of the examples in [2] which take much computing time and space. In particular, EX453 is the only theorem whose proof takes much more time using characteristic sets than using Gröbner bases. Chou proved this theorem with $maxt = 18290$. The author was able to prove it with $maxt = 1044$ by using Wu's splitting technique [9]. We can now prove the theorem with $maxt = 382$ by our new prover, using again the splitting technique of Wu. The parallel computation on this example can reduce the time up to 38.8% by two processors (see Table 2). Concerning the examples 144 and 401, the first was proved by Chou using two strategies with $maxt = 16537$ and $= 13017$, respectively, and the second was proved with $maxt = 22363$. However, we can prove these two theorems by our prover with $maxt = 2241$ and $maxt = 241$. We have also tried several other examples given in [2] which have their $maxt$s more than 10000. For these examples, our prover can always produce smaller $maxt$ (even using the same algebraic formulation).

5.4. SIX-POINT THEOREM

This theorem was discovered and proved by this author at the end of 1985 (see [10]). It ran about 14 CPU hours on an IMB4341 with $maxt = 10980$ using author's CPS prover designed for constructible theorems. In that prover the successive substitution is used instead of pseudo-division [11]. In the implementation the substitution was done simultaneously for several variables which can be explained as follows.

Consider, for example, the intersection of two lines determined by two pairs of points which have been previously constructed. Let the coordinates of the intersection point be (x_i, x_{i+1}). Then x_i, x_{i+1} are algebraically constrained by two linear equations of the form

$$\begin{aligned} A_1 x_i + B_1 x_{i+1} + C_1 &= 0, \\ A_2 x_i + B_2 x_{i+1} + C_2 &= 0, \end{aligned}$$

where A_1, A_2, B_1, B_2, C_1 and C_2 are all polynomials in terms of the coordinates of the four constructed points. We solve the linear equations and then get $x_i = P_i/Q$ and $x_{i+1} = P_{i+1}/Q$ by assuming the subsidiary condition that the coefficient determinant $Q \neq 0$. Suppose in the reduction phase that we have a polynomial $G(\ldots, x_i, x_{i+1})$ and need to substitute x_i and x_{i+1} into G to get a rational expression. Not one by one, we substitute x_i and x_{i+1} simultaneously into G so that

$$G(\ldots, x_i, x_{i+1}) = G(\ldots, \frac{P_i}{Q}, \frac{P_{i+1}}{Q}) = \frac{G'}{Q^s}.$$

Since x_i and x_{i+1} have a common denominator Q, the reduction of fractions to the expression on the left-hand side avoids some redundant factors and thus reduces the computational complexity. This is also the unmentioned reason why we did not give the index sets every step in [10].

Without using this technique we have tried to prove the theorem by our new prover where the triangular form (quasi-characteristic set) is used. Up to the variable x_{14} we got a polynomial consisting of 18768 terms, even though we have removed some possible factors. For this reason, we have used the so-called *locally reduced triangular form* as an option, i.e., we allow to make polynomials reduced in light of stairs. Then we are able to prove the theorem with the same index set in every even step as in [10]. However, in every odd step the bigger polynomials with $maxt = 15612$ still occur due to the one-by-one reductions. That the number of terms of polynomials is controlled is owing to the removal of the redundant factors. It appears that the proof given in [10] still represents the simplest one so far.

Our Apollo DN3500 slows down totally if the main storage is exhausted. This happens normally when the occurring polynomials consists of more than 5000 terms. Due to this limit we were unable to repeat the proof of this theorem on Apollo DN3500 (the running time is more than six days and the CPU time is estimated to be about 10–12 hours). However, the theorem has been proved on our Apollo DN10000 with about two CPU hours. As the conclusion of this theorem contains two polynomials, the parallel approach should yield a better result. Since only one Apollo DN10000 is available in our institute, we are not able to experiment the parallel computation for this example.

5.5. CONCLUDING REMARK

We have noted some recent results on algebraic methods for geometry theorem proving. Particular attention is paid to reduce computing time and space and to obtain simpler non-degenerate conditions by using specific techniques like removing factors, remainder splitting, and normalization. Without attempt to prove hundred theorems, we have made experiments for the proof of a number of well-named theorems and the resolution of several large problems. These experimental results have demonstrated the accomplishments of our effort, in particular, for those large problems whose solutions by computer took too much space and time. The technique of normalization leads also to a new approach of combining characteristic sets and Gröbner bases for geometry theorem proving. We have investigated the parallelization of geometry theorem proving. Although the initial experiments do not speed up the computation as much as we expected, the gain of using parallelism appears clearly. This approach remains interesting and should be further studied. Our contribution also contains a new geometry theorem prover. This prover provides the first complete sequential implementation and a parallel implementation with up to four processors of Wu's method and includes several methods based on Gröbner bases computation.

References

1. Buchberger, B., 'Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory', in Multidimensional Systems Theory (N. K. Bose, ed.), D. Reidel Publishing Company, Dordrecht-Boston-Lancaster, 184–232 (1985).
2. Chou, S. C., *Mechanical Geometry Theorem Proving*, D. Reidel Publ. Comp., Dordrecht-Boston-Lancaster-Tokyo (1988).
3. Gao, X. S., 'Transcendental Functions and Mechanical Theorem Proving in Elementary Geometries', *MM Research Preprints*, Academia Sinica, China, No. 2, 37–47 (1987).
4. Hu, S. and Wang, D. M., 'Fast Factorization of Polynomials over Rational Number Field or its Extension Fields', *Kexue Tongbao* **31**, 150–156 (1986).
5. Kapur, D., 'Using Gröbner Bases to Reason about Geometry Problems', *J. Symbolic Computation* **2**, 399–408 (1986).
6. Kutzler, B. and Stifter, S., 'On the Application of Buchberger's Algorithm to Automated Geometry Theorem Proving', *J. Symbolic Computation* **2**, 389–397 (1986).

7. Ritt, J. F., *Differential Equations from the Algebraic Standpoint*, Amer. Math. Soc., New York (1932).

8. Ritt, J. F., *Differential Algebra*, Amer. Math. Soc., New York (1950).

9. Wang, D. M., 'Mechanical Approach for Polynomial Set and its Related Fields', Ph.D thesis, Academia Sinica (1987).

10. Wang, D. M., 'A New Theorem Discovered by Computer Prover', *J. Geometry* **36**, 173–182 (1989).

11. Wang, D. M., 'On Wu's Method for Proving Constructive Geometric Theorems', *Proc. IJCAI-89*, Vol. 1, pp. 419–424 (1989).

12. Wang, D. M., 'Characteristic Sets and Zero Structure of Polynomial Sets', Preprint, RISC-LINZ, Johannes Kepler University, Austria (1989).

13. Wang, D. M., 'On the Parallelization of Characteristic-Set-Based Algorithms', Preprint, RISC-LINZ, Johannes Kepler University, Austria (1990).

14. Wang, D. M. and Gao, X. S., 'Geometry Theorems Proved Mechanically Using Wu's Method — Part on Euclidean Geometry', *MM Research Preprints*, Academia Sinica, China, No. 2, 75–106 (1987).

15. Wu, W. T., 'On the Decision Problem and the Mechanization of Theorem-Proving in Elementary Geometry', *Scientia Sinica* **21**, 159–172 (1978); *Contemp. Math.* **29**, 213–234 (1984).

16. Wu, W. T., 'Basic Principles of Mechanical Theorem Proving in Elementary Geometries', *J. Sys. Sci. & Math. Scis.* **4**, 207–235 (1984); *J. Automated Reasoning* **2**, 221–252 (1986).

17. Wu, W. T., *Basic Principles of Mechanical Theorem Proving in Geometries* (Part on elementary geometries, in Chinese), Science Press, Beijing (1984).

18. Wu, W. T., 'Some Recent Advances in Mechanical Theorem Proving of Geometries', in Automated Theorem Proving: after 25 years, *Contemp. Math.* **29**, 235–242 (1984).