Ritt-Wu's Decomposition Algorithm*

Shang-Ching Chou and Xiao-Shan Gao†

Department of Computer Sciences
The University of Texas at Austin, Austin, Texas 78712 USA

Abstract An improved Ritt-Wu's decomposition (of an algebraic set into the union of irreducible varieties) algorithm is given. The algorithm has been used to prove geometric theorems that Wu's original method addresses. Unlike Wu's original approach, nondegenerate conditions are given explicitly at the beginning, not generated during the proof process. A program based on this improved version of the algorithm proved more than 500 theorems, including Morley's trisector theorem.

Keywords Wu's method, mechanical theorem proving, prover, elementary geometry, degenerate conditions, Ritt-Wu's principle, algebraic variety, ideal, ascending chain, the dimension theorem, Morley's trisector theorem.

1 Introduction

In 1977 Wu Wen-Tsün introduced an algebraic method which could be used to prove theorems not involving betweenness in Euclidean geometry [16]. Since then, hundreds of non-trivial theorems have been proved based on this method. Further work [17] showed that the algebraic tools and algorithms of the method were already begun in the work of J. F. Ritt [14]. The key to the method is Ritt-Wu's principle and Ritt-Wu's Decomposition Algorithm [17], [18]. If one implements the algorithms literally according to the description of the work by Ritt and Wu, the size of polynomials produced in the process will become larger and larger. In practice, some modifications of Ritt-Wu's original algorithms are used. Especially, Wu himself uses the notion of an ascending chain in loose sense [17] to reduce the size of polynomials produced. However, ascending chains in Wu's loose sense still cannot prevent the size growth of polynomials in many cases.

This paper presents an improved version of Ritt–Wu's decomposition algorithm using a new algorithm, W–*prem*, for computing pseudo remainders (Sections 2–4), and gives an efficient method for proving geometry theorems according to Formulation F2 (see below) based on our improved algorithms (Sections 5–8). Besides the improved algorithms, we also prove many related theorems of

*The work reported here was supported in part by the NSF Grant CCR-8702108.
†On leave from Institute of Systems Science, Academia Sinica, Beijing.

theoretical as well as practical interest; we will like to draw particular attention to Theorem (4.4).

In Section 5–8, we will address the same kinds of geometric statements as Wu's original method addresses. A geometric statement is valid only under certain nondegenerate conditions. There are two approaches (formulations) to dealing with nondegenerate conditions:

Formulation (Approach) F1. Introducing parameters and the notion of "generally true" for a geometry statement. The present techniques can prove a statement to be generally true, at the same time giving nondegenerate conditions (usually in algebraic form) automatically.

Formulation (Approach) F2. Giving nondegenerate conditions in geometric form manually (or mechanically) at the beginning as a part of the hypotheses. Then the prover only needs to answer whether the conclusion follows the hypotheses *without adding* any other conditions.

Our prover [3] mainly uses Formulation F1. This paper addresses F2 using Wu's method. Our improved algorithm/program has proved more than 500 theorems according to Formulation F2. Among the work related to Formulation F2, we mention the work of D. Kapur [9] and H. P. Ko [13]. We will discuss their work in Section 8.

2 Preliminary Definitions and Algorithms

Let K be a computable field such as $\mathbf{Q}$, the field of rational numbers, and $y = y_1, y_2, \ldots, y_m$ be indeterminates. Unless stated otherwise, all polynomials mentioned in this section are in $A = K[y_1, \ldots, y_m] = K[y]$.

Let $f \in K[y]$. The class of f, denoted by $class(f)$, is the largest i such that y_i occurs in f. If $f \in K$, then $class(f) = 0$. Let $c = class(f) > 0$. We call y_c, denoted by $lv(f)$, the *leading variable* of f. Considering f as a polynomial in y_c, we can write f as

$$a_n y_c^n + a_{n-1} y_c^{n-1} + \cdots + a_0$$

where $a_n, ..., a_0$ are in $K[y_1, \ldots, y_{c-1}]$, $n > 0$, and $a_n \neq 0$. We call a_n the *initial* or *leading coefficient* of f and n the *leading degree* of f, denoting them as $lc(f)$ and $ld(f)$, respectively.

Now we present the pseudo division algorithm, a basic step for most algorithms. Let f and g be in $K[y]$ and v be one of the $y_1, ..., y_m$. Suppose that $deg(f, v) > 0$. Considering f and g as polynomials in v, we can write g and f as $g = a_n v^n + \cdots + a_0$, $f = b_k v^k + \cdots + b_0$. First set $r = g$. Then repeat the following process while $m = deg(r, v) \geq k$: $r := b_k r - c_m v^{m-k} f$, where c_m is the leading coefficient of r in the variable v. It is easy to see that m strictly decreases after each iteration. Thus the process terminates. At the end, we have the *pseudo remainder* $prem(g, f, v) = r = r_0$ and the following formula

$$b_k^s g = qf + r_0, \quad \text{where } s \leq n-k+1 \text{ and } deg(r_0, v) < deg(f, v).$$

Let $c = class(f) > 0$. A polynomial g is *reduced with respect to* polynomial f if $deg(g, y_c) < deg(f, y_c)$. Note that $prem(g, f, y_c)$ is reduced with respect to f; we denote $prem(g, f, y_c)$ simply by $prem(g, f)$.

Definition (2.1). Let $C = f_1, f_2, \ldots, f_r$ be a sequence of polynomials in $K[y]$. We call it a *quasi ascending chain* or a *triangular form* if either $r = 1$ and $f_1 \neq 0$, or $r > 1$ and $0 < class(f_1) < class(f_2) < \cdots < class(f_r)$.

Let $f_1, \ldots, f_r$ be a quasi ascending chain with $class(f_1) > 0$. We define $prem(g; f_1, \ldots, f_r)$ inductively to be $prem((prem(g, f_r); f_1, ..., f_{r-1})$. Let it be R. Then we have the following important *Remainder Formula*:

$$I_1^{s_1} \cdots I_r^{s_r} g = Q_1 f_1 + \cdots + Q_r f_r + R \tag{2.1.1}$$

where the I_i are the initials of the f_i, $s_1, \ldots, s_r$ are non-negative integers, and $Q_1, \ldots, Q_r$ are polynomials. Furthermore, $deg(R, x_i) < deg(f_i, x_i)$, for $i = 1, \ldots, r$, where $x_i = lv(f_i)$.

(i) A quasi ascending chain is called an *ascending chain in Ritt's sense* if f_j are reduced with respect to f_i for $i < j$.

(ii) A quasi ascending chain is called an *ascending chain in Wu's sense* (abb. wu-asc chain) if the initials I_j of the f_j are reduced with respect to f_i for $i < j$.

Extensive computational experience suggests that for nontrivial problems, ascending chains in Ritt's sense as well as in Wu's sense are very expensive to compute both from space and time considerations. As we discuss later in the paper, on Morley's theorem, huge intermediate polynomials are generated for computing these ascending chains. In this paper, we introduce ascending chains in weak sense which are not that expensive to compute.

(iii) A quasi ascending chain is called an *ascending chain in weak sense* (abb. w-asc chain) if $prem(I_i; f_1, ..., f_r) \neq 0$, for $i = 1, ..., r$.

Obviously, an ascending chain in Ritt's sense is an ascending chain in Wu's sense; an ascending chain in Wu's sense is an ascending chain in weak sense. The key to our improved version of the algorithm is to use ascending chains in weak sense. Whenever we talk about an ascending chain, it can be any one of the above three.

We define a partial order $<$ in $K[y]$: $f < g$ if $class(f) < class(g)$ or $class(f) = class(g) > 0$ and $ld(f) < ld(g)$. If neither $f < g$ nor $g < f$, then we say f and g are of the same rank. Obviously, every nonempty polynomial set S has a minimal element, i.e., the one which is not higher than any other element in S.

Definition (2.2). Let $C = f_1, ..., f_r$ and $C_1 = g_1, ..., g_m$ be two ascending chains. We define $C < C_1$ if there is an s such that $s \leq \min(r, m)$ and f_i and g_i are of the same rank for $i < s$ and that $f_s < g_s$, or $m < r$ and f_i and g_i are of the same rank for $i \leq m$.

Proposition (2.3). The partial order $<$ among the set of all ascending chains is well-founded, i.e, there are no infinite, strictly decreasing sequences of

ascending chains.

Proof. See Lemma 1 of [17]. .QED.

Definition (2.4). Let S be a nonempty polynomial set. A minimal w-asc chain in the set of all w-asc chains formed from polynomials in S is called a *w-basic set* of S.

Unless stated otherwise, whenever we talk about a finite polynomial set S, we assume S does not contain zero. By (2.3), every nonempty polynomial set S has a w-basic set.

Theorem (2.5). Let S be a finite, non-empty polynomial set. There is an algorithm to construct a w-basic set of S.

Proof. Let f_1 be a minimal polynomial of S (in the order $<$). If $class(f_1) = 0$, then it forms a w-basic set of S. Now suppose $class(f_1) > 0$. Let S_1 be the set of all polynomials in S, whose classes are higher than $class(f_1)$ and whose initials I are such that $prem(I; f_1) \neq 0$. If S_1 is empty, then f_1 forms a w-basic set of S. Now suppose S_1 is nonempty. Continuing this way, at step k, we have a w-asc chain $C = f_1,...,f_k$ in S. Let S_k be the set of all polynomials in S, whose classes are higher than $class(f_k)$ and whose initials I are such that $prem(I; f_1, ..., f_k) \neq 0$. If S_k is empty, then $f_1, ..., f_k$ is a w-basic set of S. Otherwise, chose a minimal element f_{k+1} in S_k. $f_1, ..., f_k, f_{k+1}$ form a w-asc chain again. Eventually, we arrive at a w-basic set of S in no more than m steps. .QED.

In the original presentation of Ritt-Wu's principle (cf. [14], [17]) the key operation $prem(f; ASC)$ (where ASC is an ascending chain) is repeatedly used. Since the main purpose of triangulation is to reduce the class or the leading degree of f, we need only to take fewer pseudo remainders than $prem(f; ASC)$ takes. This can reduce the size growth of polynomials produced. The following W–*prem* is one of our key steps to control the size growth of polynomials.

Algorithm W–*prem* (2.6). Given a polynomial g and an ascending chain $ASC = f_1, ..., f_r$ with non-constant f_1. We define W–$prem(g; ASC)$ to be:

Case 1. $prem(g; f_1, ..., f_r)$ if $prem(lc(g); f_1, ..., f_r) = 0$.
Case 2. g if $class(f_r) < class(g)$.
Case 3. W–$prem(prem(g, f_r); f_1, ..., f_{r-1})$ if $class(f_r) = class(g)$.
Case 4. W–$prem(g; f_1, ..., f_{r-1})$ if $class(f_r) > class(g)$.

The remainder formula is still valid for W–*prem*, except $deg(R, x_i) < deg(f_i, x_i)$ (where $x_i = lv(f_i)$) is not necessarily true.

Proposition (2.7). For an ascending chain $ASC = f_1, ..., f_r$ with $class(f_1) > 0$ and a polynomial g, if W–$prem(g; ASC) = 0$, then $prem(g; ASC) = 0$.

Proof. The proposition is easy to prove using induction on r. .QED.

We introduce a new notation extremely useful for the rest of the paper:

$$PD(ASC) = \{g \mid prem(g; ASC) = 0\}.$$

The following proposition insures the termination of the triangulation procedure

of Ritt-Wu's principle, when using W–*prem*.

Proposition (2.8). Let $B = f_1, \ldots, f_r$ be a w-basic set of polynomial set S with $0 < class(f_1)$, and h be a polynomial. Suppose $g = \text{W–}prem(h; f_1, \ldots, f_r)$ is not zero. Then the set $S_1 = S \cup \{g\}$ has a w-basic set lower than B.

Proof. See [4]. .QED.

3 A Modification of Ritt-Wu's Principle

Now let us fix an extension E of the base field K. We denote $Zero(S)$ the common zeros of polynomials in S, i.e., the set

$$\{(a_1, ..., a_m) \in E^m \mid h(a_1, ..., a_m) = 0, \text{ for all } h \in S\}.$$

Let G be another polynomial set. We denote $Zero(S/G)$ to be $Zero(S) - \bigcup_{g \in G} Zero(g)$. The following improvement of Ritt–Wu's Principle is used in our prover.

Theorem (3.1). (Ritt-Wu's Principle). Let $S = \{h_1, ..., h_n\}$ be a finite nonempty polynomial set. There is an algorithm to construct a w-asc chain ASC, called a *characteristic set* of S and denoted by Char–Set(S), such that either

(3.2). ASC consists of a non-zero constant in $K \cap Ideal(S)$; in this case $Zero(S)$ is empty, or

(3.3). $ASC = f_1, \ldots, f_r$ with $class(f_1) > 0$ and such that $f_i \in Ideal(S)$ and $\text{W–}prem(h_j; f_1, \cdots, f_r) = 0$for all $i = 1, \ldots, r$ and $j = 1, \ldots, n$; in this case we have:

$$(3.3.1)\quad Zero(S) = Zero(ASC/\{lc(f_1), \cdots, lc(f_r)\}) \bigcup \cup_{i=1}^{r} Zero(S \cup \{lc(f_i)\}),$$

$$(3.3.2)\quad Zero(S) = Zero(PD(ASC)) \bigcup \cup_{i=1}^{r} Zero(S \cup \{lc(f_i)\}).$$

Proof. By (2.5), we can construct a w-basic set B_1 of $S_1 = S$. If B_1 consists of only one nonzero constant, then we have (3.2). Otherwise, we can expand S_1 to S_2 by adding nonzero $\text{W–}prem(g; B_1)$ for all elements g in S_1. If $S_2 = S_1$, then we have (3.3). Otherwise, we can construct a w-basic set B_2 of S_2. By (2.8), $B_1 > B_2$. If B_2 does not consist of one nonzero constant, then we can expand S_2 to S_3 using the same procedure. Thus we have a strictly increasing sequence of polynomial sets:

$$S_1 \subset S_2 \subset \cdots,$$

with a strictly decreasing sequence of w-asc chains

$$B_1 > B_2 > \cdots .$$

By (2.3), this decreasing sequence can be only finite. Thus, there is an integer $k \geq 1$ such that either B_k consists of a nonzero constant or $S_k = S_{k+1}$; then we have either (3.2) or (3.3), respectively. Formulas (3.3.1) and (3.3.2) are direct consequences of the Remainder Formula (2.1.1) and (2.7). .QED.

Example (3.4). Let $S = \{h_1, ..., h_7\}$ in Example (6.2) (the Morley configuration).

$S_1 = S$; $B_1 = h_2, h_3, h_5, h_7$.
$S_2 = S \cup \{f_1, f_4, f_6\}$; $B_2 = f_1, h_2, h_3, f_4, h_5, f_6, h_7$, where
$f_1 = \text{W–}prem(h_1; B_1) = prem(h_1, h_2)$,
$f_4 = \text{W–}prem(h_4; B_1) = prem(h_4, h_5)$,
$f_6 = \text{W–}prem(h_6; B_1) = prem(h_6, h_7)$.

The w-asc chain $ASC = B_2$ is a characteristic set of S. The numbers of terms of the polynomials in B_2 (abb. the *ntps*) are 14, 4, 2, 28, 4, 59, 13. It took about 15 sec to complete the computation on a Symbolics 3600. If we use Wu-asc chain, it took about 30 minutes to reach the 5th iteration B_5. The *ntps* of B_5 are 14, 14, 2, 301, 23, 1977, 47. The program stuck in a computation (for more than 2 hours) of a $prem(p, q)$, where the numbers of terms in p and q are 2388 and 1977.

Remark. Algorithm (3.1) is a "bottom-up" triangular procedure. It has been observed that "top-down" triangular procedures ([1], [12], [10]) are much faster. However, top-down versions are not complete, i.e., triangular forms obtained by such kinds of procedures generally do not satisfy (3.3). It is a good strategy to incorporate a top-down triangular procedure into Algorithm (3.1). This can reduce the number of iterations in (3.1) and polynomial sizes in many cases. It was first done in [12]. For example, we can produce a triangular form TR_i for S_i by a top down procedure, then set S_i to be $S_i \cup TR_i$ and take a w-basic set B_i of S_i, and so on. However, this approach should be combined with w-asc chain and W–prem, otherwise the size control generally cannot be insured. Example (3.4) is such an example, for which this approach does not help much without using W–prem.

4 A Modification of Ritt-Wu's Decomposition Algorithm

Theorem (4.1). Ritt-Wu's Zero Decomposition Algorithm (the Refined Form). Let S and G be two non-empty polynomial sets. There is an algorithm either to detect the emptiness of $Zero(S/G)$ or to decompose $Zero(S/G)$ in the following form:

$$Zero(S/G) = \bigcup_{1 \leq i \leq k} Zero(ASC_i/I_i \cup G), \tag{4.1.1}$$

$$Zero(S/G) = \bigcup_{1 \leq i \leq k} Zero(PD(ASC_i)/G) \tag{4.1.2}$$

where each ASC_i is an *irreducible* w-asc chain,[1] the I_i is the initial set of the ASC_i, and $prem(g; ASC_i) \neq 0$ for all $g \in G$ and $i = 1, ..., k$.

Proof. Let $ASCs$ be a set of w-asc chains, initialized to be empty at the beginning.

Step 1. According to (3.1) we can construct a w-asc chain having the property of either (3.2) or (3.3). In the case of (3.2), $Zero(S/G)$ is empty. In the case of (3.3), we have a w-asc chain ASC and a polynomial set S' (i.e., S_k in the proof of (3.1)) having ASC as one of its w-basic sets. $Zero(S) = Zero(S')$.

Step 2. Check whether the w-asc chain $ASC = f_1, ..., f_r$ is reducible. If it is, then there is an integer $k > 0$ such that $f_1, ..., f_{k-1}$ is irreducible, but $f_1, ..., f_k$ is reducible. By (9.4) in [4], we can find two polynomials g and h with $class(f_k) = class(g) = class(h)$ and $gh \in Ideal(f_1, ..., f_k)$. We have a decomposition: $Zero(S') = Zero(S' \cup \{g\}) \cup Zero(S' \cup \{h\})$. Obviously, $S' \cup \{g\}$ and $S' \cup \{h\}$ have w-basic sets strictly lower than that of S'. We can take each of $S' \cup \{g\}$ and $S' \cup \{h\}$ as a new S, and go to step 1.

Step 3. Let $ASC = f_1, ..., f_r$ and $I = \{lc(f_1), ..., lc(f_r)\}$. By (3.3.2) we have: (4.1.3)

$$Zero(S/G) = Zero(S'/G) = Zero(ASC/I \cup G) \bigcup \cup_{p \in I} \{Zero(S' \cup \{p\}/G)\}.$$

Step 4. If $prem(g; ASC) = 0$ for some $g \in G$, then $Zero(ASC/I \cup G)$ is empty. Otherwise, we add this w-asc chain ASC to $ASCs$.

Step 5. For each p in I, let $p' = prem(p; ASC)$; since W–$prem(p; ASC) \neq 0$, $p' \neq 0$ by (2.7). For each $Zero(S' \cup \{p\}/G) = Zero(S' \cup \{p, p'\}/G)$ in (4.1.3), take $S' \cup \{p, p'\}$ as a new S, then go to step 1. Repeat this process recursively. Since $S' \cup \{p, p'\}$ has a w-basic set *strictly* lower than that of S' by (2.8), this recursive process will finally terminate. For otherwise, we would have a strictly decreasing sequence of w-asc chains, contradicting (2.3). The termination of each branch happens when I consists of constant polynomials. Upon termination, we have two cases:

(i) $ASCs$ is empty. This means that S does not have common zeros.

(ii) $ASCs = \{ASC_1, ..., ASC_k\}$ $(1 \leq k)$, then we have the decomposition (4.1.1). Since $Zero(ASC_i/I_i) \subset Zero(PD(ASC_i)) \subset Zero(S)$, (4.1.2) follows from (4.1.1). .QED.

Theorem (4.2). Ritt-Wu's Zero Decomposition Algorithm (the Coarse Form). The same statement as in (4.1), except we do not require that each w-asc ASC_i be irreducible.

Proof. The only thing needed to change in Algorithm (4.1) is to drop step 2 in the proof of (4.1). .QED.

The reader who is not interested in the details of the further improvements of the above algorithms can skip the rest of this section.

Remark 1. Since multivariate factorization is available in many algebraic sys-

[1] For the definition and properties of irreducible ascending chains see [3].

tems, we suggest to keep step 2 and check the reducibility of $prem(f_k; f_1, ..., f_{k-1})$. Using factorization of multivariate polynomials over the integers is actually a *good* strategy to reduce polynomial sizes. If we want to obtain the refined form, we can put off factorization over extension fields to the last step when we have the coarse form. This turns out to be very effective to obtain the refined form. Among the 8 examples in [6], there are very few involving factorization over extensions after we get the coarse form, and only one w-asc chain is actually reducible over extension fields.

Remark 2. The branches produced in the recursive step 5 can be as many as thousands and most of them are redundant. If G is not empty, we have the following two techniques (1) and (2) to control branching.

(1) If some $d \in G$ is reduced to zero by S' (the set produced in Step 1) using some other reductions (e.g., the reduction used in the Gröbner basis method), then $Zero(S/G)$ is empty. This is one of the most effective techniques to control branching in the algorithm, especially such situations happen at early stages.

(2) Let $ASC = f_1, ..., f_r$. If $prem(d, ASC) = 0$ for some $d \in G$, $Zero(PD(ASC)/G)$ is empty. More important, we do not have to add initials of those f_i which are not used in computing $prem(d, ASC)$ to S'.

(3) We still can use the above techniques (1) and (2) even G is empty. We only need to write (3.3.2) (or (3.3.1)) in a different form:
(3.3.2′)
$Zero(S) = Zero(PD(ASC))\bigcup \cup_{1\le i\le r} Zero(S \cup \{lc(f_i)\}/\{lc(f_1), ..., lc(f_{i-1})\})$.

In this way, the final decomposition would be slightly different:

$$Zero(S/G) = \bigcup_{1\le i\le k} Zero(PD(ASC_i)/G \cup D_i). \tag{4.1.2′}$$

Since $S \subset PD(ASC_i)$ for $i = 1, ..., k$, $Zero(PD(ASC_i)) \subset Zero(S)$. Thus we actually can get rid of D_i in (4.1.2′), and have the same decomposition (4.1.2) or (4.1.1).

Techniques (1)–(3) can reduce branches by a magnitude of one or two orders for essentially large problems. In Example (3.4), it took 5860 sec to decompose $Zero(S)$ without using (3); there were 2346 characteristic sets produced in the process and 76 asc chains in (4.1.2) (i.e., $k = 76$). It took only 944 sec using (3). There were 106 characteristic sets produced in the process and 45 asc chains in (4.1.2).

Theorem (4.3). Let E be an algebraically closed extension of the base filed K and $G = \{1\}$. Then (4.1.2) becomes

$$Zero(S) = \bigcup_{1\le i\le k} Zero(PD(ASC_i)) \tag{4.3.1}$$

which is a decomposition of algebraic set $Zero(S)$ into the union of the irreducible varieties $Zero(PD(ASC_i))$. Here each $PD(ASC_i)$ is a prime ideal. Or

alternatively,

$$Radical(S) = \bigcap_{1\le i\le k} PD(ASC_i). \tag{4.3.2}$$

The decompositions in (4.1)–(4.3) are generally redundant, i.e., some $Zero((PD(ASC_i))$ may be contained in others. The following theorem, the proof of which is rather long, removes some redundancy without any cost.

Theorem (4.4). Let $n = length(S)$ be the number of polynomials in S. Suppose that the emptiness of $Zero(S)$ is not detected in algorithm (4.1) or (4.3) and the set unions in (4.1.1) and (4.1.2) (either in the refined form or in the coarse form) are arranged in such a way that $length(ASC_i) \le n$ for $i \le l$, and $length(ASC_i) > n$ for $i > l$ for some integer $0 \le l \le k$, then $0 < l$, and we have the decomposition

$$Zero(S/G) = \bigcup_{1\le i\le l} Zero(PD(ASC_i)/G). \tag{4.4.1}$$

Proof. If it is the refined form (4.1), then the theorem is a consequence of the Affine Dimension Theorem [8]. In general case, see [4]. .QED.

Remark. Notice also that the formula:

$$Zero(S/G) = \bigcup_{1\le i\le l} Zero(ASC_i/I_i \cup G)$$

is generally not true even for the refined form. This is the key advantage to use $Zero(PD(ASC_i))$ instead of $Zero(ASC_i/I_i)$.

Theorem (4.5). There is an algorithm to remove the redundancy in the decomposition (4.3.1) *completely*, thus providing an *irredundant* decomposition of an algebraic set.

Proof. Let $ASC = f_1, ..., f_r$ be an irreducible asc chain. Let GB_z be a Gröbner basis of $\{f_1, ..., f_r,\ z \cdot lc(f_1) \cdots lc(f_r) - 1\}$ in a compatible ordering with $y_1^{i_1} \cdots y_m^{i_m} < z$. Then $GB_z \cap K[y]$ is a Gröbner basis of the prime ideal $PD(ASC)$ (for the proof of this statement, see p.85 of [3]). Once we have a Gröbner basis of each $PD(ASC_i)$ in (4.4.1), we can get the irredundant decomposition easily. See [6] for examples using this theorem. .QED.

5 A Method for Formulation F2

Let E be the field associated with a given geometry. Suppose the hypothesis of a geometry statement can be algebraically expressed by a set of polynomial equations $\{h_1(y_1, ..., y_m) = 0, ..., h_n(y_1, ..., y_m) = 0\}$ together with a set of polynomial inequations $\{s_1(y_1, ..., y_m) \neq 0, ..., s_q(y_1, ..., y_m) \neq 0\}$ expressing the non-degenerate conditions and the conclusion by a polynomial equation

$g(y_1, ..., y_m) = 0$. Then the equivalent algebraic form of the geometry statement is

(5.1) $\forall y_1 \cdots y_m \in E[(h_1 = 0 \wedge \cdots \wedge h_n = 0 \wedge s_1 \neq 0 \wedge \cdots \wedge s_q \neq 0) \Rightarrow g = 0]$.

Let $S = \{h_1, ..., h_n\}$ and $G = \{s_1, ..., s_q\}$, then the above formula is equivalent to

(5.2) $$Zero(S/G) \subset Zero(g).$$

Method (5.3). This method is to confirm (5.1), or in the case when E is algebraically closed, to decide whether (5.1) is valid.

Using Algorithms (4.1) or (4.2), and Theorem (4.4) to decompose $Zero(S/G)$ into

$$Zero(S/G) = \bigcup_{1 \leq i \leq l} Zero(PD(ASC_i)/G).$$

Each $Zero(PD(ASC_i)/G)$ is called a *component* of $Zero(S/G)$.

Case 1. $prem(g; ASC_i) = 0$ for all $i = 1, ..., l$. Then (5.2), hence formula (5.1) is valid by Theorem (5.4) below.

Case 2. $prem(g; ASC_i) \neq 0$ for some i. If E is algebraically closed and each ascending chain ASC_i is irreducible, then formula (5.1) is not valid by Theorem (5.5) below. .QED.

In case 2 and when formula (5.1) is disproved, we don't have any information about the reason why (5.1) is false: it is false because the geometry statement is *generally false* or because some nondegenerate conditions are missing. This is why the authors are in favor of Formulation F1 in Section 1 to introduce the notion "*generally (generically) true*", which is inherent to a given geometry statement regardless of how much nondegenerate conditions are added.

Theorem (5.4). Let the notations be the same as in Section 4 and g be any polynomial. Suppose we have decomposition (4.4.1) (in the coarse or refined form). If $prem(g; ASC_i) = 0$ for all $i = 1, ..., l$, then $Z(S/G) \subset Zero(g)$.

Proof. Since $prem(g; ASC_i) = 0$, $g \in PD(ASC_i)$. Hence $Zero(PD(ASC_i)) \subset Zero(g)$ for all i. By (4.4.1), $Zero(S/G) \subset Zero(g)$. .QED.

Theorem (5.5). Let the notations be the same as in Section 4 and g be any polynomial. Suppose we have decomposition (4.4.1) in the refined form (i.e., all ASC_i are irreducible) and all zeros are taken from an *algebraically closed* extension E of K. Then

(i) Each $Zero(PD(ASC_i)/G)$ is non-empty.
(ii) $Zero(S/G) \subset Zero(g)$ if and *only if* $prem(g; ASC_i) = 0$ for all $i = 1, ..., l$.

Proof. See Theorems (4.8) and (9.3) in [4]. .QED.

6 Examples

Example (6.1) (Pascal's Theorem). Let A, B, C, D, F and E be six points on a circle (O). Let $P = AB \cap DF$, $Q = BC \cap FE$ and $S = CD \cap EA$. Show that P, Q and S are collinear (Figure 1).

The obvious non-degenerate conditions in this problem seem to be "the three pairs of lines, AB and DF, BC and FE, and CD and EA, have normal intersections". Let $B = (u_1, 0)$, $A = (0, 0)$, $C = (u_2, u_3)$, $O = (x_2, x_1)$, $D = (x_3, u_4)$, $F = (x_4, u_5)$, $E = (x_5, u_6)$, $P = (x_6, 0)$, $Q = (x_8, x_7)$, and $S = (x_{10}, x_9)$. Then the problem can be specified as follows:

$h_1 = 2u_2x_2 + 2u_3x_1 - u_3^2 - u_2^2 = 0$	$OA \equiv OC$.
$h_2 = 2u_1x_2 - u_1^2 = 0$	$OA \equiv OB$.
$h_3 = x_3^2 - 2x_2x_3 - 2u_4x_1 + u_4^2 = 0$	$OA \equiv OD$.
$h_4 = x_4^2 - 2x_2x_4 - 2u_5x_1 + u_5^2 = 0$	$OA \equiv OF$.
$h_5 = x_5^2 - 2x_2x_5 - 2u_6x_1 + u_6^2 = 0$	$OA \equiv OE$.
$h_6 = (u_5 - u_4)x_6 + u_4x_4 - u_5x_3 = 0$	P, D and F are collinear.
$h_7 = (u_6 - u_5)x_8 - (x_5 - x_4)x_7 + u_5x_5 - u_6x_4 = 0$	Q, F and E are collinear.
$h_8 = u_3x_8 - (u_2 - u_1)x_7 - u_1u_3 = 0$	Q, B and C are collinear.
$h_9 = u_6x_{10} - x_5x_9 = 0$	S, E and A are collinear.
$h_{10} = (u_4 - u_3)x_{10} - (x_3 - u_2)x_9 + u_3x_3 - u_2u_4 = 0$	S, C and D are collinear.
$s_1 = (u_4 - u_3)x_5 - u_6x_3 + u_2u_6 \neq 0$	Lines AE and CD have a normal intersection.
$s_2 = u_3x_5 - u_3x_4 - (u_2 - u_1)u_6 + (u_2 - u_1)u_5 \neq 0$	Lines BC and EF have a normal intersection.
$s_3 = u_1u_5 - u_1u_4 \neq 0$	Lines AB and DF have a normal intersection.
$g = x_7x_{10} - (x_8 - x_6)x_9 - x_6x_7 = 0$	Conclusion: S, Q and P are collinear.

$Zero(S/G) = Zero(PD(ASC_1)/G)$ (in 6.9s [2]), where $ASC_1 = \text{Char–Set}(S)$. Since $prem(g;\, ASC_1) = 0$ (in 0.4s), the theorem has been confirmed.

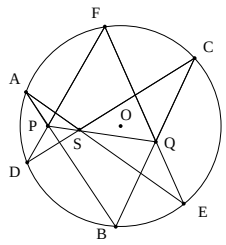

Figure 1: Pascal's Theorem

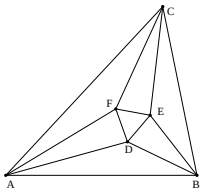

Figure 2: Morley's Theorem

Example (6.2) (Morley's Trisector Theorem.) The points of intersection D, E and F of the adjacent trisectors of the angles of any triangle ABC are the vertices of an equilateral triangle (Figure 2).

Let $B = (y_1, 0)$, $A = (0, 0)$, $D = (y_2, y_3)$, $C = (y_5, y_4)$, $F = (y_8, y_7)$, and $E = (y_{10}, y_9)$. Then the problem can be specified as follows:

[2] Meaning that it took 6.9 seconds to complete the computation on a Symbolics 3600.

$h_1 = (y_3^3 + (-3y_2^2 + 6y_1y_2 - 3y_1^2)y_3)y_5 + ((-3y_2 + 3y_1)y_3^2 + y_2^3 - 3y_1y_2^2 + 3y_1^2y_2 - y_1^3)y_4$
$-y_1y_3^3 + (3y_1y_2^2 - 6y_1^2y_2 + 3y_1^3)y_3 = 0$ $\tan(\angle CBA) - \tan(3\angle DBA) = 0.$

$h_2 = (y_3^3 - 3y_2^2y_3)y_5 + (-3y_2y_3^2 + y_2^3)y_4 = 0$ $\tan(\angle CAB) - \tan(3\angle DAB) = 0.$

$h_3 = y_6^2 - 3 = 0$ $\tan(\pm\pi/3) = \pm\sqrt{3}.$

$h_4 = (((y_3^2 + y_2^2 - y_1y_2)y_5 - y_1y_3y_4)y_6 + y_1y_3y_5 + (y_3^2 + y_2^2 - y_1y_2)y_4)y_8 + ((y_1y_3y_5 + (y_3^2 + y_2^2 - y_1y_2)y_4)y_6 - (y_3^2 + y_2^2 - y_1y_2)y_5 + y_1y_3y_4)y_7 - ((y_3^2 + y_2^2 - y_1y_2)y_5^2 + (y_3^2 + y_2^2 - y_1y_2)y_4^2)y_6 - y_1y_3y_5^2 - y_1y_3y_4^2 = 0$

$\tan(\angle BAD + \angle DBA + \angle ACF) = \pm\sqrt{3}.$

$h_5 = (y_1y_3y_5 - y_1y_2y_4)y_8 + (y_1y_2y_5 + y_1y_3y_4)y_7 = 0$ $\tan(DAB) = \tan(CAF).$

$h_6 = ((2y_4y_5 - y_1y_4)y_8 + (-y_5^2 + y_1y_5 + y_4^2)y_7 - y_4y_5^2 - y_4^3)y_{10} + ((-y_5^2 + y_1y_5 + y_4^2)y_8 + (-2y_4y_5 + y_1y_4)y_7 + y_5^3 - y_1y_5^2 + y_4^2y_5 - y_1y_4^2)y_9 + (-y_4y_5^2 - y_4^3)y_8 + (y_5^3 - y_1y_5^2 + y_4^2y_5 - y_1y_4^2)y_7 + y_1y_4y_5^2 + y_1y_4^3 = 0$

$\tan(ACF) = \tan(ECB).$

$h_7 = (y_1y_3y_5 + (-y_1y_2 + y_1^2)y_4 - y_1^2y_3)y_{10} + ((y_1y_2 - y_1^2)y_5 + y_1y_3y_4 - y_1^2y_2 + y_1^3)y_9 - y_1^2y_3y_5 + (y_1^2y_2 - y_1^3)y_4 + y_1^3y_3 = 0$ $\tan(ABD) = \tan(EBC).$

$s_1 = y_1y_4 \neq 0$ A, B, C are not collinear.

$s_2 = y_1^2 \neq 0$ Line AB is non-isotropic.

$s_3 = (y_5 - y_1)^2 + y_4^2 \neq 0$ Line BC is non-isotropic.

$s_4 = y_5^2 + y_4^2 \neq 0$ Line AC is non-isotropic.

$g = (y_6y_8 - y_7 - y_2y_6 + y_3)y_{10} + (y_8 + y_6y_7 - y_3y_6 - y_2)y_9 + (-y_2y_6 - y_3)y_8 + (-y_3y_6 + y_2)y_7 + (y_3^2 + y_2^2)y_6 = 0$

Conclusion: $\tan(\angle EDF) = \pm\sqrt{3}.$

$Zero(S/G) = Zero(PD(ASC_1)/G)$ (in 756.7s), where $ASC_1 = \text{Char–Set}(S)$ (i.e., B_2 in Example (3.4)). Since $prem(g; ASC_1) = 0$ (in 6.4s), the theorem has been confirmed. There are 18 triangles DEF thus formed. The proof by our prover applies to all 18 equilateral triangles. The idea of the above specification of the equation part comes from Wu's work [17]. The specification of the non-degenerate conditions is due to us. An isotropic line is a line perpendicular to itself. It does not exist in Euclidean geometry. Thus we have proved Morley's theorem under the only non-degenerate condition that A, B, and C are not collinear. For more examples See [4].

7 Experimental Results

We have implemented Method (5.3) in our prover [2]. More than 500 theorems have been proved in this way. In particular, we have experimented with the same set of the 512 theorems in [3] (using the same coordinates and equations). For 413 of the 512 theorems, the prover can generate non-degenerate conditions *all* in geometric form by the method in [2]. For most of those 413 theorems, we use such machine generated non-degenerate conditions in geometric form as the inputs to our new method. We paid particular attention to a few problems among these 413 theorems, specifying non-degenerate conditions manually. For example, we proved Feuerbach's theorem under the only non-degenerate that "the vertices of the triangle are not collinear."

For the remaining 91 theorems, some non-degenerate conditions in polynomial inequations were generated by our previous method. First we simply deleted these algebraic inequations, using the rest machine generated non-degenerate

conditions in geometric form as inputs. About half of these 91 theorems were confirmed this way. We have to pay more attention to the remaining half, adding more non-degenerate conditions in geometric form manually.

In this way, we have proved 493 of the 512 theorems. We had trouble with the remaining 19 theorems within the time or space limit of the computer. Among the 493 theorems proved, 471 were proved within 300 seconds; 12 within one hour.

The following table gives time statistics of 30 theorems for various methods. Wu(F1) is the method we mainly use for Formulation F1. The sign * means that some non-degenerate conditions are still in algebraic form. The sign ? means that it took more than 4 hours without results. Wu(F2) is the method in this paper; 0.4 + 0.3 means that it took 0.4 sec to decompose and 0.3 sec to check $prem(g; ASC_i) = 0$. GB(F1) is our Gröbner basis method for Formulation F1 [7]. Strictly speaking, non-degenerate conditions produced by this method is in algebraic form. However, by a theorem due to us, if GB(F1) confirms a theorem in a class of constructive geometry statements, then it is valid under geometric non-degenerate conditions produced by the method in [5]. GB(F2) is our Gröbner basis method for Formulation F2 [7], [3].

Ex. No.	Theorem	Sources	Wu(F1)	Wu(F2)	GB(F1)	GB(F2)
Ex[1]	Parallelogram	Section 2 [7]	0.1	0.23 + 0.02	0.45	0.71
Ex[2]	Theorem of Centroid	Ex1, [7]	0.35	0.33 + 0.02	0.83	0.38
Ex[3]	Theorem on Altitudes	Ex133, [3]	0.83	0.5 + 0.1	2.5	75.7
Ex[4]	Ceva's Theorem	Ex334, [3]	0.81	0.4 + 0.3	1.1	48.3
Ex[5]	Simson's Theorem	Ex2, [7]	1.2	0.6 + 0.1	1.6	144.1
Ex[6]	Brahmagupta	Section 4 [7]	1.1	5.1 + 0.1	2.4	1.48
Ex[7]	Pappus' Theorem	Ex6, [7]	1.3	2.5 + 0.2	6.4	44.6
Ex[8]	Pappus Dual	Ex17, [3]	1.7	0.8 + 0.3	15.8	293.9
Ex[9]	Butterfly Theorem	Ex5, [7]	1.5	6.1 + 0.1	24.5	?
Ex[10]	Simson Dual	Ex21, [3]	4.8	20.9 + 1.2	?	?
Ex[11]	Four Simson Lines	Ex311, [3]	3.2	4.9 + 1.3	?	?
Ex[12]	Ogilvy's Theorem	Ex65, [3]	238.5	20.6 + 195.6	?	?
Ex[13]	Secant Theorem	Ex390, [3]	3.3	0.7 + 2.1	11,975.2	2950.4
Ex[14]	Pascal's Theorem	This paper	3.9	8.9 + 0.3	1401.5	?
Ex[15]	Brianchon's Theorem	Ex19, [3]	9.6	5.0 + 5.3	?	?
Ex[16]	Pappus Point	Ex7, [7]	5.6	8.8 + 1.9	54.6	?
Ex[17]	Isosceles Midpoint	Ex8, [7]	2.6	3.7 + 0.1	1.5	3.3
Ex[18]	Gauss' Theorem	Ex9, [7]	1.5	0.15 + 0.05	1.3	0.35
Ex[19]	Gauss Point	Ex10, [7]	3.7	5.7 + 0.6	18.8	95.1
Ex[20]	Gauss Conic	Ex17, [3]	1511.3	101.3 + 1502.1	5,225.4	?
Ex[21]	Feuerbach's Theorem	Ex204,[3]	2.8	27.3 + 1.0	28.2	197.9
Ex[22]	Miquel Point	Ex308, [3]	40.2	4.3 + 38.1	90.3	?
Ex[23]	Miquel Circle	Ex309, [3]	412.3	1.9 + 555.2	137.7	?
Ex[24]	Steiner's Theorem	Ex13, [3]	10.2	8.6 + 4.2	?	?
Ex[25]	Kirman's Theorem	Ex14, [3]	8.2	6.7 + 6.0	?	?
Ex[26]	Pascal Conic 1	Ex9, [3]	11.8	2.8 + 11.2	71.6	?
Ex[27]	Pascal Conic 4	Ex12, [3]	210.3	8.6 + 184.2	?	?
Ex[28]	Morley's Theorem	This paper	20.2 *	756.7 + 6.4	308.1 *	?
Ex[29]	V. Pratt's Theorem	Ex80, [3]	275.3	?	?	?
Ex[30]	Coxeter's Theorem	Ex40, [3]	218.2 *	?	?	?

8 Related Work

Though people knew at the very beginning that Wu's method could prove theorems with Formulation F2, no attempt was ever made at that time, because people (including the first author) thought it tended to be much slower than the method based on Formulation F1. Wu, Chou, Gao and others proved hundreds of theorems based on Formulation F1.

The first author experimented with Formulation F2 during 1984–1985 using the Gröbner basic (GB) method. He was able to prove about 10 theorems and found it very slow [7]. The hardest one proved by him was perhaps Simson's theorem.

D. Kapur, on the other hand, was successful in proving more and harder theorems using the GB method based on a refutational approach according to Formulation F2 [9].

H. P. Ko [13] was the first to use Ritt-Wu's method to prove geometry theorems according to Formulation F2. Our work is in the same direction as Ko's work and has similarities and differences with her work [13]. Our method is faster than hers. Especially, our Theorem (4.4) makes the proof procedure faster and clearer. For example, in Example (6.1), Ko produced four components with ascending chains T_1, T_2, T_3, and T_4. According to Theorem (4.4), $Zero(T_i/I_i \cup G) \subset Zero(PD(T_1)/G)$ (for $i = 2, 3, 4$; here the I_i are the initial sets of the ascending chains T_i) and $Zero(S/G) = Zero(PD(T_1)/G)$. So we only need to check whether $prem(g; T_1) = 0$; or in terms of Formulation F1, to check whether g vanishes on the general components (generally true).

After this paper was submitted, we learned that D. Kapur and H. K. Wan also used the characteristic set method to prove geometry theorems according to Formulation F2 using a refutational approach [15], [11].

The authors wish to thank D. Kapur for his suggestion and comments.

References

[1] S.C. Chou, "Proving Elementary Geometry Theorems Using Wu's Algorithm",in *Automated Theorem Proving: After 25 years*, Ed. By W.W. Bledsoe and D. Loveland, AMS Contemporary Mathematics Series **29** (1984), 243-286.

[2] S.C. Chou, "Proving and Discovering Theorems in Elementary Geometries Using Wu's Method", PhD Thesis, Department of Mathematics, University of Texas, Austin (1985).

[3] S.C. Chou, *Mechanical Geometry Theorem Proving,* D. Reidel Publishing Company, Dordrecht, Netherlands, 1988.

[4] S.C. Chou and X. S. Gao, "Ritt-Wu's Decomposition Algorithm and Geometry Theorem Proving",Technical Report 89-09, Department of Computer Sciences, University of Texas at Austin, 1989.

[5] S.C. Chou and X.S. Gao, "A Class of Geometry Statements of Constructive Type and Geometry Theorem Proving", TR-89-37, Computer Sciences Department, The University of Texas at Austin, November, 1989.

[6] S.C. Chou and X.S. Gao, "Techniques for Ritt–Wu's Decomposition Algorithm", TR-90-2, Computer Sciences Department, The University of Texas at Austin, February, 1990.

[7] S.C. Chou and W.F. Schelter, "Proving Geometry Theorems with Rewrite Rules", *Journal of Automated Reasoning*, **2(4)** (1986), 253–273.

[8] R. Hartshorne, *Algebraic Geometry,* Springer-Verlag, 1978.

[9] D. Kapur, "Geometry Theorem Proving Using Hilbert's Nullstellensatz", in Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation, 202-208.

[10] D. Kapur and J. Mundy, "Wu's Method and its Application to Perspective Viewing", *Artificial Intelligence Journal*, V. **37** (1988), pp.15-36.

[11] D. Kapur and Hoi K. Wan, "Refutational Proofs of Geometry Theorems via Characteristic Sets", to appear in *ISSAC'90*.

[12] H.P. Ko and S.C. Chou, "Polynomial Triangulation for Pseudo Common Divisors", Technical Report, 85CRD242, General Electric Company, 1985.

[13] H.P. Ko, "Geometry Theorem Proving by Decomposition of Quasi-Algebraic Sets: An Application of the Ritt-Wu Principle", *Artificial Intelligence*, Vol. 37, pp95-122 (1988).

[14] J. F. Ritt, *Differential Algebra,* AMS Colloquium Publications, New York, 1950.

[15] Hoi Wan, "On proving Geometry Theorems via Characteristic Set Computation", MS Project Report, Department of Computer Science, RPI, Troy, Dec., 1987.

[16] Wu Wen-tsün, "On the Decision Problem and the Mechanization of Theorem Proving in Elementary Geometry", *Scientia Sinica* **21** (1978), 157-179.

[17] Wu Wen-tsün, "Basic Principles of Mechanical Theorem Proving in Geometries", *J. of Sys. Sci. and Math. Sci.* **4(3)**, 1984, 207-235, republished in *Journal of Automated Reasoning* **2(4)** (1986), 221-252.

[18] Wu Wen-tsün, "On Zeros of Algebraic Equations –An Application of Ritt's Principle", *Kexue Tongbao* **31(1)** (1986), 1-5.