IC/91/6

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

SEARCHING DEPENDENCY BETWEEN ALGEBRAIC EQUATIONS: AN ALGORITHM APPLIED TO AUTOMATED REASONING

Yang Lu

and

Zhang Jingzhong

INTERNATIONAL ATOMIC ENERGY AGENCY

UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

IC/91/6

International Atomic Energy Agency

and

United Nations Educational Scientific and Cultural Organization

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

SEARCHING DEPENDENCY BETWEEN ALGEBRAIC EQUATIONS: AN ALGORITHM APPLIED TO AUTOMATED REASONING *

Yang Lu **

International Centre for Theoretical Physics, Trieste, Italy

and

Zhang Jingzhong

Institute of Mathematical Sciences, Academia Sinica,
610015 Chengdu, Sichuan, People's Republic of China.

MIRAMARE – TRIESTE

January 1991

* Submitted for publication.

** Permanent address: Institute of Mathematical Sciences, Academia Sinica, 610015 Chengdu, Sichuan, People's Republic of China.

ABSTRACT

An efficient computer algorithm is given to decide how many branches of the solution to a system of algebraic also solve another equation. As one of the applications, this can be used in practice to verify a conjecture with hypotheses and conclusion expressed by algebraic equations, despite the variety reducible or irreducible.

1 Introduction

Given a system of algebraic equations

$$\begin{cases} F_1(u_1,u_2,\cdots,u_n,x_1,x_2,\cdots,x_s)=0, \\ F_2(u_1,u_2,\cdots,u_n,x_1,x_2,\cdots,x_s)=0, \\ \cdots\cdots \\ F_s(u_1,u_2,\cdots,u_n,x_1,x_2,\cdots,x_s)=0, \end{cases} \tag{1}$$

and another one

$$G(u_1,u_2,\cdots,u_n,x_1,x_2,\cdots,x_s)=0 \tag{2}$$

where u_i and x_j we regard as parameters and indeterminates for $i=1,\cdots,n$ and $j=1,\cdots,s$, respectively, the problem which this paper concerns is whether the solution to $\{F_1=0,\cdots,F_s=0\}$ also solves $G=0$ or not? In other words, regarding $\{F_1=0,\cdots,F_s=0\}$ as the hypothesis of a theorem to be proved, and $G=0$ the desired conclusion, the task is to verify the theorem is ture or not.

It is easy to see that an efficient algorithm for such a problem is of significance both in theory and in practice.

In general, the solution to $\{F_1 = 0, \cdots, F_s = 0\}$ depends upon n parameters as following,

$$\begin{cases} x_1 = x_1(u_1, u_2, \cdots, u_n) \\ x_2 = x_2(u_1, u_2, \cdots, u_n) \\ \cdots\cdots \\ x_s = x_s(u_1, u_2, \cdots, u_n) \end{cases} \tag{3}$$

which consists of a number of branches. It is possible in some cases that $G = 0$ holds for several branches of the solution but doesn't hold for branches else.

The algorithms have been given in the case that variety $\{F_1, \cdots, F_s\}$ is irreducible.[1–3][6–9] Otherwise, one used to do decomposition to transform a reducible variety into irreucible sub-varieties then examine all of them one by one.[1] As we know, there was no algorithm for reducible case without doing decomposition until recently the authors (with Hou Xiaorong)[10] gave a decision procedure by which we can find directly how many branches of the solution to $\{F_1 = 0, \cdots, F_s = 0\}$ also solve $G = 0$, despite the variety reducible or irreducible. In this paper, it is demonstrated how the theoretical result is implemented on computer and how efficient the algorithm is.

2 Algorithm

To deal with this problem, we suppose $\{F_1 = 0, \cdots, F_s = 0\}$ has been transformed into the "triangular form":

$$\begin{cases} f_1(u_1, u_2, \cdots, u_n, x_1) = 0, \\ f_2(u_1, u_2, \cdots, u_n, x_1, x_2) = 0, \\ \cdots\cdots \\ f_s(u_1, u_2, \cdots, u_n, x_1, x_2, \cdots, x_s) = 0, \end{cases} \tag{4}$$

that is, for every equation $f_j = 0$, only one new indeterminate x_j is introduced. We call $\{f_1, \cdots, f_s\}$ an "ascending chain". This assumption known as "well-ordering principle" works without loss of generality because an algorithm was given some years ago [4–6][1] to transform a system of polynomials into an ascending chain which possesses the same set of zeros in any extended field.

Definition 1. By $\text{lcoeff}(f_j, x_j)$ or l_j denote the leading coefficient of f_j with respect to x_j, for $j = 1, \cdots, s$. An ascending chain $\{f_1, \cdots, f_s\}$ is called a *proper ascending chain* if any branch of the solution to $\{f_1 = 0, \cdots, f_s = 0\}$ solves none of $\{l_1 = 0, \cdots, l_s = 0\}$.

Definition 2. By $\text{resultant}(p, q, x)$ denote the resultant of any two polynomials, p and q, with respect to indeterminate x, which is a determinant in terms of the coefficients of $p(x)$ and $q(x)$ to verify if the two polynomials have a common solution. Given an ascending chain $\{f_1, \cdots, f_s\}$ and another polynomial

G, putting

$$r_{s-1} := \text{resultant}(G, f_s, x_s),$$
$$r_{s-2} := \text{resultant}(r_{s-1}, f_{s-1}, x_{s-1}),$$
$$r_{s-3} := \text{resultant}(r_{s-2}, f_{s-2}, x_{s-2}),$$
$$\cdots\cdots$$
$$r_1 := \text{resultant}(r_2, f_2, x_2),$$
$$r_0 := \text{resultant}(r_1, f_1, x_1),$$

we call r_0 *the resultant of ascending chain* $\{f_1, \cdots, f_s\}$ *with respect to polynomial* G and denote it simply by

$$\text{res}(f_1, \cdots, f_s, G).$$

Referring to MAPLE, by degree(p, T) and ldegree(p, T) denote the highest and lowest degrees of a polynomial p with respect to variable T, respectively. We have the following criterion:

Theorem 1. Given a proper ascending chain $\{f_1, \cdots, f_s\}$, a polynomial g and a new variable T, put

$$p(T) := \text{res}(f_1, \cdots, f_s, g + T).$$

All the branches of solution to $\{f_1 = 0, \cdots, f_s = 0\}$ also solve $g = 0$ if and only if

$$\text{degree}(p(T), T) = \text{ldegree}(p(T), T). \tag{5}$$

The next one works for general cases:

Theorem 2. Given a proper ascending chain $\{f_1, \cdots, f_s\}$, a polynomial g and a new variable T, put

$$k := \text{ldegree}(\text{res}(f_1, \cdots, f_s, g + T), T). \tag{6}$$

Then, there are exactly k branches of the solution to $\{f_1 = 0, \cdots, f_s = 0\}$ also solve $g = 0$.

Applying Theorem 1 to automated theorem proving, we have

Corollary 1. Assume $\{f_1, \cdots, f_s\}$ is a proper ascending chain. Let $\{f_1 = 0, \cdots, f_s = 0\}$ and g be the hypothesis and conclusion of a conjecture to be verified, respectively. The conjecture is true in general if and only if

$$\text{degree}(p(T), T) = \text{ldegree}(p(T), T).$$

where $p(T)$ means $\text{res}(f_1, \cdots, f_s, g + T)$.

Corollary 2. Assume $\{f_1, \cdots, f_s\}$ is a proper ascending chain. Let $\{f_1 = 0, \cdots, f_s = 0\}$ and g be the hypothesis and conclusion of a conjecture to be verified, respectively. The conjecture is true for exact k of the branches of the algebraic variety which expresses the hypothesis, if and only if

$$k = \text{ldegree}(\text{res}(f_1, \cdots, f_s, g + T), T). \tag{7}$$

Corollary 3. Given a proper ascending chain $\{f_1, \cdots, f_s\}$ and a polynomial g, none of the branches of solution to $\{f_1 = 0, \cdots, f_s = 0\}$ solves $g = 0$ if and only if

$$\text{res}(f_1, \cdots, f_s, g) \neq 0. \tag{8}$$

The last one gives following algorithm to check an ascending chain to be proper or not:

Corollary 4. An ascending chain $\{f_1, \cdots, f_s\}$ is proper if and only if $l_1 \neq 0$ and for $j = 2, 3, \cdots, s$,

$$\text{res}(f_1, \cdots, f_{j-1}, l_j) \neq 0, \tag{9}$$

where l_j denote the leading coefficient of f_j with respect to x_j.

3 Proof

The key is to prove Theorem 2 which other ones are inferred from.

Lemma 1. Let $f(x)$ and $g(x)$ be polynomials with degrees μ and ν, roots $x^{(1)}, x^{(2)}, \cdots, x^{(\mu)}$ and $\underline{x}^{(1)}, \underline{x}^{(2)}, \cdots, \underline{x}^{(\nu)}$, and leading coefficients l and $\underline{l}$, respectively. We have

$$\text{resultant}(f, g, x) = l^\nu \underline{l}^\mu \prod_{i=1}^{\mu} \prod_{j=1}^{\nu} (x^{(i)} - \underline{x}^{(j)}). \tag{10}$$

This well-known fact can be found in algebra texbooks.

Lemma 2. Given notations as above, then

$$\text{resultant}(f, g, x) = l^\nu \prod_{i=1}^{\mu} g(x^{(i)}). \tag{11}$$

It is easy to prove by using Lemma 1.

Now we need more notations. Put $\mu_j := \text{degree}(f_j, x_j)$ for $j = 1, 2, \cdots, s$. Since the system of equations (4) is in triangular form, theoretically speaking, all indeterminates $x_1, x_2, \cdots, x_s$ can be solved one by one, with several branches each. Thus, we can denote the full solution to (4) by

$$\begin{cases} x_1 = x_1^{(i_1)}(u_1, \cdots, u_n) \\ x_2 = x_2^{(i_1\, i_2)}(u_1, \cdots, u_n) \\ x_3 = x_3^{(i_1\, i_2\, i_3)}(u_1, \cdots, u_n) \\ \cdots\cdots \\ x_s = x_s^{(i_1\, i_2\, \cdots\, i_s)}(u_1, \cdots, u_n) \end{cases} \tag{12}$$

where i_j ranges from 1 to μ_j, for $j = 1, 2, \cdots, s$, if the ascending chain is proper.

For simplicity, we put

$$\vec{u} := (u_1, \cdots, u_n),$$

$$\vec{x} := (x_1, \cdots, x_s),$$

and

$$\vec{x}^{(i_1\, i_2\, \cdots\, i_s)} := (x_1^{(i_1)}(\vec{u}), x_2^{(i_1\, i_2)}(\vec{u}), \cdots, x_s^{(i_1\, i_2\, \cdots\, i_s)}(\vec{u})). \tag{13}$$

Lemma 3. Given notations as above, put $\nu_s := \text{degree}(G, x_s)$,

$$\nu_j := \text{degree}(\text{res}(f_{j+1}, f_{j+2}, \cdots, f_s, G), x_j) \tag{14}$$

for $j = 1, 2, \cdots, s-1$, and

$$L := l_1(\vec{u})^{\nu_1} \prod_{j=2}^{s} \prod_{i_1=1}^{\mu_1} \prod_{i_2=1}^{\mu_2} \cdots \prod_{i_{j-1}=1}^{\mu_{j-1}} (l_j(\vec{u}, \vec{x}^{(i_1\, i_2\, \cdots\, i_s)})^{\nu_j}. \tag{15}$$

where every $l_j(\vec{u}, \vec{x}^{(i_1\, \cdots\, i_s)})$ is an algebraic function of $\vec{u}$ obtained by substituting a branch of the solution, $\vec{x}^{(i_1\, \cdots\, i_s)}$, for the $\vec{x}$ in $l_j(\vec{u}, \vec{x})$, the leading coefficient of f_j with respect to x_j. Then, we have

$$\text{res}(f_1, \cdots, f_s, G) = L \prod_{i_1=1}^{\mu_1} \prod_{i_2=1}^{\mu_2} \cdots \prod_{i_s=1}^{\mu_s} G(\vec{u}, \vec{x}^{(i_1\, i_2\, \cdots\, i_s)}). \tag{16}$$

It is not difficulty to prove Lamma 3 by using Lemma 2 successively.

Now let us go to prove Theorem 2. Replacing G in (16) with $g+T$, the right hand side becomes a polynomial $p(T)$ with roots

$$-g(\vec{u}, \vec{x}^{(i_1\, i_2\, \cdots\, i_s)})$$

where i_j ranges from 1 to μ_j for $j = 1, 2, \cdots, s$. The lowest degree of a polynomial is nothing but the number of the vanishing roots, so that among the $\mu_1 \mu_2 \cdots \mu_s$ roots, $-g(\vec{u}, \vec{x}^{(i_1\, \cdots\, i_s)})$, exactly k of them are vanishing. This completes the proof.

4 Examples

The resultant computation is supported by current softwares of computer algebra such as MAPLE, MACSYMA, REDUCE and MATHEMATICA. Somebody thought of it can do nothing because the computation would be too big, but in practice the complexity is acceptable for quite a number of non-trivial examples. All the ascending chain appeared in examples of this section have been checked to be proper by Corollary 4.

Example 1. A difficult geometry theorem was conjectured by Thébault in 1938 and proved by Taylor in 1983 with a proof of 26 pages! Then it was verified[1] on a SYMBOLICS 3600 computer by using *successive pseudo division* to all the irreducible sub-varieties obtained from decomposition to a reducible variety which expresses hypothesis of the theorem; the program took 44 hours CPU time, while more than 500 theorems else took only a few seconds each. The wonderful theorem is stated as follows:

Given a triangle ABC with incenter w and circumcircle Γ, take any point D on edge BC. Let w_1 be the center of the circle which contacts DB, DA and Γ, and w_2 the center of the circle which contacts DC, DA and Γ. To prove w_1, w_2 and w are collinear.

We took one algebraic interpretation somewhat diference from that taken in [1] for the Thébault–Taylor theorem and ran a program in MAPLE(version 4.3) on a SUN386*i* station with CPU time 1041.58 seconds, and then on a CONVEX C210 with CPU time 268.35 seconds only. The answer the screen showed was: *The conjecture is true for 2 of the 8 branches.* Our program follows below:

```
tt:=proc()
f1:=4*x1^2*u1^2*u2^4+6*u1^2*u2^4-2*u2^2-2*u2^6-2*u2^2*u1^4-2*u2^6
  *u1^4+u1^2+u1^2*u2^8+4*u2^4*u3-4*u2^4*u1^4*u3-4*u2^4*u3^2*u1^2;
f2:=(4*u2^4-8*u2^4*u3*u1^2-4*u1^4*u2^4+8*x1*u1^2*u2^4)*x2^2+(-4*
  u2^2*u1^4+8*u2^4+4*u1^2-4*u2^6*u1^4-4*u2^6+8*u1^4*u2^4+4*u1^2*
  u2^8+8*u1^2*u2^4-8*u1^2*u2^2-4*u2^2-8*u2^6*u1^2)*x2+(-2*u1^2+4*
  u1^2*u2^4-2*u1^2*u2^8)*x1-1-8*u2^4*u3+8*u2^2*u3*u1^2+u1^4+2*u2
  ^4+4*u2^6*u1^4*u3+4*u1^4*u3*u2^2-12*u2^4*u3*u1^2-2*u1^4*u2^4-2*
  u2^8*u3*u1^2-8*u2^4*u1^4*u3-2*u3*u1^2+u1^4*u2^8+4*u3*u2^2-u2^8+
  8*u2^6*u3*u1^2+4*u2^6*u3;
f3:=(4*u2^4-4*u1^4*u2^4-8*u2^4*u3*u1^2-8*x1*u1^2*u2^4)*x3^2+(8*u1
  ^4*u2^4+4*u1^2-4*u2^2*u1^4-8*u1^2*u2^2-4*u2^6*u1^4+8*u1^2*u2^4-
  8*u2^6*u1^2+4*u1^2*u2^8-4*u2^2+8*u2^4-4*u2^6)*x3+(2*u1^2-4*u1^2
  *u2^4+2*u1^2*u2^8)*x1-1+8*u2^2*u3*u1^2+8*u2^6*u3*u1^2+u1^4+2*u2
  ^4-u2^8+4*u3*u2^2-12*u2^4*u3*u1^2-2*u1^4*u2^4-8*u2^4*u3-8*u2^4*
  u1^4*u3-2*u3*u1^2+u1^4*u2^8+4*u1^4*u3*u2^2+4*u2^6*u3+4*u2^6*u1
```

```
  ^4*u3-2*u2^8*u3*u1^2;
f4:=x4*u1*u2+1-u1^2*u2^2;
f5:=x5*u1*u2-u1^2+u2^2;
f6:=(2*u1^2-2*u1^2*u2^4+2*x4*u1^2*u2^2+2*x5*u1^2*u2^2)*x6+(-u1^2-
   u1^2*u2^4)*x5+(-u1^2-u1^2*u2^4)*x4-u1^4+u2^4-1+u1^4*u2^4;
f7:=(-u2^2+u1^2*u2^4-u2^2*u1^4+u1^2)*x7+(u2^2-u2^2*u1^4-u1^2+u1^2
  *u2^4-2*x5*u1^2*u2^2)*x6+(u1^2*u2^4+u1^2)*x5-u2^4+u1^4;
g:=(4*x2*u2^4+4*x3*u2^4)*x7+(-4*u2^2-4*u2^6+8*u2^4)*x6-4*u2^2-4*
  u2^6+1+6*u2^4-4*x2*x3*u2^4+u2^8;
r6:=resultant(g+T,f7,x7);
r5:=resultant(r6,f6,x6);
r4:=resultant(r5,f5,x5);
r3:=resultant(r4,f4,x4);
r2:=resultant(r3,f3,x3);
r1:=resultant(r2,f2,x2);
r0:=resultant(r1,f1,x1);
lprint('The conjecture is ture for',ldegree(r0,T),'of the',degree
  (r0,T),'branches')
end;
```

Example 2. The famous Feuerbach Theorem: A circlr which contacts 3 lines has to contact the nine-point-circle of the triangle formed by these lines. For this we ran a program in MAPLE(version 4.3) on a SUN386i station with CPU time 6.63 seconds, and then on a CONVEX C210 with 1.72 seconds. The answer the screen showed was: *The conjecture is true for 4 of the 4 branches,* that means the theorem true in general. In fact, for any triangle, one inscribed circle and three escribed circles, each of the four contacts the nine-point-circle. The program follows below:

```
fb:=proc()
f1:=-u1^3+4*u1^2*x1+4*u2^2*x1^2*u1-4*u2*x1^2*u1-4*u1*x1^2-8*u2^2*
  x1^3+8*u2*x1^3+4*x1^2*u1^3-8*x1^3*u1^2+4*u1*x1^4;
f2:=-2*u2*x1-2*x1*x2+2*x1+2*u1*x2-u1;
f3:=4*u1*x3-u1^2-u2+u2^2;
f4:=4*x4-1-2*u2;
f5:=u1^4-2*u1^2*u2+2*u1^2*u2^2+u2^2-2*u2^3+u2^4+u1^2-16*u1^2*x5;
g:=normal(((x1-x3)^2+(x2-x4)^2)^2+x5^2+x1^4-2*((x1-x3)^2+(x2-x4)
  ^2)*(x5+x1^2)-2*x5*x1^2);
r4:=resultant(g+T,f5,x5);
r3:=resultant(r4,f4,x4);
r2:=resultant(r3,f3,x3);
```

```
r1:=resultant(r2,f2,x2);
r0:=resultant(r1,f1,x1);
lprint('The conjecture is true for',ldegree(r0,T),'of the',degree
  (r0,T),'branches')
end;
```

Example 3. On three sides of a triangle ABC, three equilateral triangles BCA_1, CAB_1 and ABC_1 are drawn. To prove lines AA_1, BB_1 and CC_1 are concurrent. To this theorem we use a modified algorithm by replacing g in resultant computation with the *final pseudo remainder* obtained from Wu's algorithm, successive pseudo division. The answer the screen showed is *The conjecture is true for 2 of the 8 braches.* In fact, there are 8 instances as some of the equilateral triangles may be inward to triangle ABC and some else may be outward. The conjecture is true only for two instances, that is, three equilateral triangles all inward or all outward. The program in MAPLE follows below, with CPU time (including that for well-ordering and successive division) 26.37 seconds and 5.47 seconds on a SUN386*i* station and a CONVEX C210, respectively.

```
et:=proc()
F1:=2*x1-1;
F2:=4*x2^2-3;
F3:=x3^2+x4^2-u1^2-u2^2;
F4:=x3^2+x4^2-(x3-u1)^2-(x4-u2)^2;
F5:=(x5-1)^2+x6^2-(u1-1)^2-u2^2;
F6:=(x5-1)^2+x6^2-(x5-u1)^2-(x6-u2)^2;
f1:=F1;
f2:=F2;
f3:=resultant(F3,F4,x4);
f4:=normal(F4);
f5:=resultant(F5,F6,x6);
f6:=normal(F6);
A:=array(1..3,1..3,[[x2-u2,u1-x1,(x2-u2)*u1-(x1-u1)*u2],[x4,1-x3,
  x4],[x6,-x5,0]]);
with(linalg,det):
g:=normal(det(A));
p5:=prem(g,f6,x6);
p4:=prem(p5,f5,x5);
p3:=prem(p4,f4,x4);
p2:=prem(p3,f3,x3);
p1:=prem(p2,f2,x2);
```

```
p0:=prem(p1,f1,x1);
if p0=0 then print('The conjecture is true in general') else
r5:=resultant(p0+T,f6,x6);
r4:=resultant(r5,f5,x5);
r3:=resultant(r4,f4,x4);
r2:=resultant(r3,f3,x3);
r1:=resultant(r2,f2,x2);
r0:=resultant(r1,f1,x1);
lprint('The conjecture is true for',ldegree(r0,T),'of the',degree
  (r0,T),'branches');
fi
end;
```

5 Conclusion: Algorithm in Large

Among the existing algorithms for searching dependency between algebraic equations, what is the essential difference of that suggested in this paper from others? In a word, it is a *global algorithm*. To see this more clearly, we consider a problem about transversality, a global property, between algebraic submanifolds. Let

$$\begin{cases} F_1(x_1, x_2, x_3) = 0 \\ F_2(x_1, x_2, x_3) = 0 \end{cases}$$

be an algebraic curve and

$$F_3(x_1, x_2, x_3) = 0$$

an algebraic surface. We want to know whether the curve transversally intersects the surface or not. Since the tangent vector of the curve is $\nabla F_1 \times \nabla F_2$ and the normal vector of the surface is ∇F_3, the transversal condition means

$$(\nabla F_1 \times \nabla F_2) \cdot \nabla F_3 \neq 0$$

for all the intersections.

Assume $\{F_1, F_2, F_3\}$ is transformed into a proper ascending chain, $\{f_1, f_2, f_3\}$, and put

$$g := (\nabla F_1 \times \nabla F_2) \cdot \nabla F_3.$$

Then, by Corollary 3, the curve intersects the surface transversally if and only if

$$\mathrm{res}(f_1, f_2, f_3, g) \neq 0.$$

Otherwise, with Theorem 2, the number of the tangent points is given by

$$\mathrm{ldegree}(\mathrm{res}(f_1, f_2, f_3, g + T), T).$$

ACKNOWLEDGMENTS

One of the authors (Y.L.) would like to thank Professor Abdus Salam, the International Atomic Energy Agency and UNESCO for hospitality at the International Centre for Theoretical Physics, Triéste.

References

[**1**] Chou, S. C., *Mechanical Geometry Theorem Proving,* D. Reidel publishing Company, (Amsterdam) 1988.

[**2**] Kapur, D., 'Using Gröbner bases to reason about geometry problems', *J. Symbolic Computation* **2**, 399-408 (1986).

[**3**] Kutzler, B. & Stifter, S., 'On the application of Buchberger's algorithm to automated geometry theorem proving', *J. Symbolic Computation* **2**, 389-397 (1986).

[**4**] Ritt, J. F., *Differential Equations from the Algebraic Standpoint,* Amer. Math. Soc., 1932.

[**5**] Ritt, J. F., *Differential Algebra,* Amer. Math. Soc., 1950.

[**6**] Wu Wen-tsün, 'On the decision problem and the mechanization of theorem proving in elementary geometry', *Scientia Sinica* **21**, 157-159, (1978).

[**7**] Yang Lu, 'A new method of automated theorem proving', in *The Mathematical Revolution Inspired by Computing,* J.H. Johnson and M.J. Loomes (eds), Oxford University Press 1991. pp.115-126

[**8**] Zhang Jingzhong & Yang Lu, 'Principles of parallel numerical method and single-instance method of mechanical theorem proving', *Math. in Practice and Theory (in Chinese),* 1, 34-43 (1989).

[**9**] Zhang Jingzhong, Yang Lu & Deng Mike, 'The parallel numerical method of mechanical theorem proving', *Theoretical Computer Science* **74**, 253-271 (1990).

[**10**] Zhang Jingzhong, Yang Lu & Hou Xiaorong, 'A criterion for dependency of algebraic equations and its application to automated theorem proving', (submitted).