http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (7): 1431-1439, 2010
ISSN 1812-5638

Application of Wu's Method to Proving Total Correctness of Recursive Program

Yong Cao
Institute of Computer Science and Engineering,
University of Electronic Science and Technology, China

Abstract: Software testing can find errors, but can not find them absent. Because program verification can provide this guarantee, it has arguably been one of the most fertile application areas of formal methods and becomes more and more important in software development. Proving the correctness of a program entails showing that if the program is initiated from a machine state satisfying a certain precondition then the state on termination satisfies some desired post-condition. The emphasis was put on formal verification in program proving previous. If we can algebraize program proving, the correctness proving can be mechanized and automated easily and the efficiency of proving can be also improved greatly. Wu's method has been shown to improve efficiencies in mechanical geometry theorem proving, where it can replace the well-known Grobner basis method. It seems to have promise to bring the powerful mathematical machinery to settle the program total correctness problem. This study applies Wu's characteristic set method and mathematical induction to proving the total correctness of recursive program. Our method is based on the algebraic theory of polynomial set, which have wider application domain. We implement this method with the computer algebra tools MMP. The application of the method is demonstrated on a few examples. Compared with proving tool Isabelle, our method is more efficient through experiments and greatly improves the efficiency of proving especially in polynomial program. In this way, an algebraic approach based on mathematics mechanization is introduced to software verification. A new idea for program correctness proving is proposed in this study.

Key words: Wu's characteristic set methods, Grobner basis, termination, partial correctness, triangularize, mathematical induction

INTRODUCTION

Testing is becoming one of the key-aspects of software development methodologies. Usually, methods for program correctness proving are testing. They can find errors but cannot cover all bugs. Formal verification approaches, e.g., formal program correctness proving, can provide this guarantee (Mao and Wu, 2005). As software development with reuse is seen as one of the most important factors, formal verification becomes a very attractive approach to the perfect development of software systems for the reusable component should be completely correct. Since Floyd-Hoare-Dijkstras inductive assertion method using pre/post-conditions was proposed in the late sixties for verifying properties of imperative programs, the success however has been limited. Recently, due to the advance of computer algebra, several methods based on symbolic computation can be applied successfully to program correctness proving.

Wu's algorithm is an efficient method for solving systems of polynomial equations, which mainly calculates a characteristic set of a finite set of polynomials. It is especially relevant in the framework of algorithmic geometry, where it can replace the well-known Grobner basis method. Also, it seems to have promise to bring the powerful mathematical machinery to settle the program total correctness problem. In fact many programs or program fragments (They should include assignment statements which can be transformed into transition equations. Each assignment state is like x = e where e is an expression. It can also be transformed into the transition equation like x - e = 0.) can be represented as algebraic transition systems (Sankaranarayanan *et al*., 2004) and transitions between program states can be represented as polynomial equations. Once program can be algebraized, the correctness proving can be mechanized and automated easily and the efficiency of proving can be also improved greatly.

Total correctness of programs means two things: termination and partial correctness with respect to a specification. This study introduces an approach of polynomial algebra to prove program correctness, shows Wu's method can be applied to program correctness proving and proposes a method based on mathematics mechanization to verify correctness of recursive program. Wu's method is an algorithm of representing the zeros of a set of polynomials with the characteristic sets and we

can use Wu's method to prove recursive program correctness by calculating the characteristic set. Our method can be more efficient and implemented on computer easily and mechanically and presents a new idea for proving the total correctness of program. It is a polynomial algebraic method and polynomial algebraic method is different from classical logical method because the latter emphasizes on logical reasoning and the former emphasizes on computation.

Some work has been proposed to recursive program correctness proving. Krauss (2006) developed an automated tool for defining partial recursive functions in Higher-Order Logic and providing appropriate reasoning tools for them. He implemented his tool as a definitional specification mechanism for Isabelle/HOL (http://isabelle.in.tum.de/). Popov and Jebelean (2009) describe an innovative method for proving the total correctness of tail recursive programs having a specific structure.

Wu's method: Wu's method is an efficient method in symbolic computation, which finds zeros of a set of polynomials by looking for the characteristic sets. Roughly speaking, one would like to obtain a triangular base B of S, i.e., a sequence of polynomials $(p_1, \ldots, p_m)$ such that every polynomial in S is a linear combination of polynomials in B. In this section, we give a brief review of the necessary results in the theory of Wu's method and (Wu, 2001) for a detailed exposition.

We first give some notational conventions, which are also directly useful for our purpose of proving program correctness. Let, $K[x_1, \ldots, x_n]$ be the ring of polynomials in n variables over a field K. For the sake of simplicity, K is supposed to be an algebraically closed field. Let $\deg_{x_j}(f)$ be the maximum degree of the variable x_j in the polynomials $f \in K[x_1, \ldots, x_n]$. Let $G = < g_1, \ldots, g_r >$ be a finite sequence of polynomials $g_1, \ldots, g_r \in K[x_1, \ldots, x_n]$. In the following, all the polynomials mentioned are in $K[x_1, \ldots, x_n]$. A fixed ordering on the set of variables is considered. Without loss of generality, we assume that the given variables ordering is:

$$x_1 \prec x_2 \prec \cdots \prec x_{n-1} \prec x_n \tag{1}$$

Definition 1: Given a polynomial:

$$f = I_d(x_1,\ldots,x_{j-1})x_j^d + I_{d-1}(x_1,\ldots,x_{j-1})x_j^{d-1} + \ldots + I_0(x_1,\ldots,x_{j-1}) \tag{2}$$

Its initial polynomial, In(f) is defined to be the polynomial $I_d(x_1,\ldots,x_{j-1})$. We use In(P) in the following to denote the set of the initial polynomials in P:

$$In(P) = \{In(p_i) : p_i \in P\} \tag{3}$$

Definition 2: A variable x_j is said to be in f if some monomial in f with nonzero coefficient contains a positive power of x_j. If x_j is in f and no variable x_i with $x_i > x_j$ is in f, then:

$$\text{class}(f) = j \text{ and } \text{cdeg}(f) = \deg_{x_j}(f) \tag{4}$$

If no variables exist in f, then class(f) = 0 and cdeg(f) = 0.

Definition 3: We say that f is of lower rank than g, if either:

$$\begin{aligned} &1.\ \text{class}(f) < \text{class}(g), \text{ or} \\ &2.\ \text{class}(f) = \text{class}(g) \text{ and } \text{cdeg}(f) < \text{cdeg}(g). \end{aligned} \tag{5}$$

The basic idea of Wu's algorithm is to enlarge the initial set S with new polynomials obtained by a certain pseudo-division procedure. Let p be a polynomial in $K[x_1, \ldots, x_n]$. The partial degree of p with respect to the variable x_i is denoted $\deg_{x_j}(p)$. The head variable of p is the variable x_c, occurring in p with maximal c and, in this case, c is called the class of p; the initial polynomial In(p) of p is the coefficient of $x_c^{\deg_{x_c}(p)}$ in p. For instance, the polynomial $x_1^2+x_2+x_3^4+x_2^2x_3^4$ has class 3 and initial polynomial $1+x_2$.

Theorem 1: For any two polynomials f and g with class(f) = j., there exist a non-negative number α and polynomials q and r, such that:

$$In(f)^{\alpha} g = qf + r \tag{6}$$

where,

$$\deg_{x_j}(r) < \deg_{x_j}(f) \quad \text{and} \quad \alpha \le \deg_{x_j}(g) - \deg_{x_j}(f) + 1$$

If r = g, then g is said to be reduced with respect to f and r is denoted by prem(g, f, x_j) as usual (prem is abbreviation of pseudo-remainder).

Definition 4: G is said to be an ascending set, if one of the following two conditions holds:

- r = 1 and g_1 is not equal to zero
- r > 1 and $0 < \text{class}(g_1) < \text{class}(g_2) < \ldots < \text{class}(g_r) \le n$ and each g_i is reduced with respect to each of the preceding polynomials $g_j (1 \le j < i)$

Considering an ascending set G, a polynomial f and using pseudo-division successively:

$$\begin{aligned} r_t &= f \\ r_{t-1} &= \text{prem}(r_t, g_t, x_t), \\ r_{t-2} &= \text{prem}(r_{t-1}, g_{t-1}, x_{t-1}), \\ &\vdots \\ r_0 &= \text{prem}(r_1, g_1, x_1) \end{aligned} \tag{7}$$

We obtain:

$$\text{In}(g_t)^{\alpha_t} \cdots \text{In}(g_1)^{\alpha_1} p = \sum_{i=1}^{t} h_i g_i + r_0$$

where,

$$\text{In}(g_i)^{\alpha_i} r_i = h_i g_i + r_{i-1}$$

The remainder r_0 is usually denoted by prem(f, G).

Definition 5: The ascending set *G* is a characteristic set of P if and only if:

$$(\forall f \in P)[\text{prem}(f, G) = 0]. \tag{8}$$

From the above definition, we can present an algorithm on how to compute a characteristic set of a set of polynomials. We call an algorithm to compute a characteristic set of a set of polynomials Wu's method. The algorithm can be modified in various ways for better performance. However, the common algorithm in Wu's method is doing pseudo-division successively, with respect to an ascending set. It adds the remainders to the set of polynomials successively until all the remainders are 0.

Definition 6: We use Zero(S) to denote the zero set of S:

$$\text{Zero}(S) = \left\{ \begin{array}{l} \langle a_1, a_2, \ldots, a_n \rangle \in K^n : \\ (\forall p \in S)\left[p(a_1, a_2, \ldots, a_n) = 0\right] \end{array} \right\} \tag{9}$$

Theorem 2: Let, B be a characteristic set of S. Then:

$$\begin{aligned} &\text{Zero}(S) = (\text{Zero}(B) \setminus \text{Zero}(\text{In}(B))) \\ &\cup \bigcup_{i=1}^{n} (\text{Zero}(S \cup \text{In}(g_i))) \end{aligned} \tag{10}$$

Theorem 3: Given each $P_i \cup B$, we calculate the corresponding characteristic set by using Wu's method. Let, $\{B_i, 1 \le i \le \mu\}$ be all the characteristic sets obtained that include B and have:

$$\text{Zero}(P) = \bigcup_{i=1}^{u} (\text{Zero}(B_i) \setminus \text{Zero}(\text{In}(B_i))) \tag{11}$$

In a word, the essence of Wu's method is that we can obtain:

$$\text{In}(g_t)^{\alpha_t} \cdots \text{In}(g_1)^{\alpha_1} p = \sum_{i=1}^{t} h_i g_i + r_0$$

and Wu's characteristic set $\{g_1, g_2, \ldots, g_t, r_0\}$ through pseudo-division if:

$$\text{In}(g_t)^{\alpha_t} \neq 0, \bullet\bullet\bullet\bullet\bullet, \text{In}(g_1)^{\alpha_1} \neq 0$$

$g_i = 0$ and $r_0 = 0$ then:

$$p = \text{In}(g_t)^{\alpha_t} \cdots \text{In}(g_1)^{\alpha_1}$$

$f = hg = 0$ and Zero(f) = Zero(p)/Zero(h), where:

$$h = \text{In}(g_t)^{\alpha_t} \cdots \text{In}(g_1)^{\alpha_1}$$

In this way, we can obtain a characteristic set of a set of polynomials.

APPLICATION OF WU'S METHOD TO VERIFYING CORRECTNESS OF RECURSIVE PROGRAM

In geometric proving we algebraize the proposition (conclusion or conjectures) which need to be proved and the given conditions (or hypotheses) firstly. Suppose that the algebraized conclusion is p and the algebraized given conditions (or hypotheses) are $S = \{f_1 = 0, f_2 = 0, \ldots, f_s = 0\}$. Then we set p as goal and set $S = \{f_1 = 0, f_2 = 0, \ldots, f_s = 0\}$ as constraint conditions. From the hypothesis (constraint condition) polynomials we compute a set of polynomials in a triangular form called Wu's characteristic set $B = (g_1 = 0, \ldots, g_t = 0)$ which every polynomial in S is a linear combination of polynomials in B, namely $f_i \in S$ and:

$$\text{In}(g_t)^{\eta_t} \ldots \text{In}(g_1)^{\eta_1} f_i = \sum_{j=1}^{t} h_j g_j$$

Next we check whether the conclusion polynomial p pseudo-divides to 0 using the polynomials in the characteristic set, namely we use every element of the characteristic set pseudo-divide the conclusion p to obtain pseudo-remainder r.

Then, we know there should be:

$$\text{In}(g_t)^{\alpha_t} \cdots \text{In}(g_1)^{\alpha_1} p = \sum_{i=1}^{t} h_i g_i + r_0$$

If $r_0 = 0$ and

$$In(g_t)^{\alpha_t} \neq 0,\ldots., In(g_1)^{\alpha_1} \neq 0$$

then we can draw a conclusion p = 0. Characteristic sets are computed using pseudo-division by introducing a total ordering on dependent variables. The inequations:

$$In(g_t)^{\alpha_t} \neq 0,\ldots., In(g_1)^{\alpha_1} \neq 0$$

have been called nondegenerate conditions or subsidiary conditions in the literature (Mao and Wu, 2005).

Hypothesize that r_0 is terminate polynomial assertion when the program terminates. If $r_0 = 0$, namely r_0 satisfies termination condition and state transition equations or constraint conditions $f_1 = 0, \ldots, f_s = 0$, then the program will terminate and we will deduce that the inductive polynomial assertion p = 0 is true.

According the above discussion, we can apply Wu's method to verifying correctness of recursive program and formalize our method as follow.

Suppose that recursive program is P. Extract termination condition, state transition equations (constraint conditions) and goal from recursive program and triangularize state transition equations.

Termination condition:

$$t(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0 \tag{12}$$

Goal or conclusion:

$$p(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0 \tag{13}$$

State transition equations (constraint conditions):

$$\begin{aligned} &f_1(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0\\ &f_2(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0\\ &\cdots\cdots\\ &f_s(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0 \end{aligned} \tag{14}$$

where $u_1,u_2,\ldots,u_r$ are independent parameters and $x1,\ldots,x_s$ are dependent parameters.

We triangularize state transition equations (constraint conditions) and obtain:

$$\begin{aligned} &f_1^*(u_1,u_2,\cdots,u_r,x_1)=0\\ &f_2^*(u_1,u_2,\cdots,u_r,x_1,x_2)=0\\ &\cdots\cdots\\ &f_s^*(u_1,u_2,\cdots,u_r,x_1,\cdots,x_s)=0 \end{aligned} \tag{15}$$

Assume that the given variables' ordering (We must give variables' ordering in advance. Wu's method is an elimination method and the variables' ordering can point out the elimination sequence of variables. For example when we eliminate variable x_i the given ordering guarantees that variable whose order greater than x_i will not recur in computing. Otherwise, the computation which finds the characteristic set does not terminate.), find loop invariants of recursive program and relation formulae among states and prove that the sequence of states forms a descending chain and there exists states which satisfy termination condition and lay a foundation for induction. If the corresponding relation formulae among certain states have been reduced with state transition equations but the states do not satisfy termination condition, then the program will not terminate.

Assume the induction hypothesis and apply Wu's method and mathematical induction to proving partial correctness of the recursive program. If the termination and partial correctness of the program are proved, then the program is totally correct.

According to Cao and Zhu (2010) we obtain the loop invariant $L(x_1, x_2, \ldots, x_s)$ of recursive program P. Suppose x_s is a output variable and $x_1, x_2, \ldots, x_{s-1}$ are input variables and by loop invariant $L(x_1, x_2, \ldots, x_s)$ we compute the relation formula:

$$x_s = H(x_1,x_2,\cdots,x_{s-1}) \tag{16}$$

which is between input variables and output variable in each recursion. Assume there exist n recursions in P and when n = 1 we compute whether $x_s = H(x_1, x_2, \ldots, x_{s-1})$ satisfies termination condition namely we compute program state $x_1^{(1)},x_2^{(1)},\cdots,x_s^{(1)}$ satisfy exit conditions and find out whether $x_s = H(x_1, x_2, \ldots, x_{s-1})$ satisfies termination condition through substituting $x_1^{(1)},x_2^{(1)},\cdots,x_s^{(1)}$ for $x_1, x_2, \ldots, x_s$. We assume the induction hypothesis:

$$x_s^{(n-1)} = Hx_1^{(n-1)},x_2^{(n-1)},\cdots,x_{s-1}^{(n-1)}$$

is true and the hypothesis need to be proved is:

$$x_s^{(n)} - Hx_1^{(n)},x_2^{(n)},\cdots,x_{s-1}^{(n)} = 0$$

We use I to be divided by $f^*_1, f^*_2,\ldots,f^*_s$ to obtain:

$$I=\sum_{i=1}^{s} h_i f_i^* + R_n \tag{17}$$

According to the induction hypothesis:

$$x_s^{(n-1)} = Hx_1^{(n-1)}, x_2^{(n-1)}, \cdots, x_{s-1}^{(n-1)}$$

and constraint conditions f^*_1, f^*_2,...,f^*_s we find out whether R_n is 0. If R_n =0 then I=0 and we know that the program is partially correct. Because termination of the program has already been proved:

$$\text{When, } n = x_s^{(1)} = H\left(x_1^{(1)}, x_2^{(1)}, \cdots, x_{s-1}^{(1)}\right)$$

total correctness of the recursive program is proved.

We demonstrate how to apply Wu's method to recursive program correctness proving by using Euclid's algorithm (gcd: greatest common divisor) example. The Example 1 (Hu *et al*., 2003) will be described as follow:

$$\begin{aligned} GCD(x,y) == \text{if } & x = 0 \text{ then } y \\ & \text{else if } y < x \text{ then } GCD(y,x) \\ & \text{else if } GCD(x, y-x) \end{aligned} \tag{18}$$

At first, we compute an invariant of recursion according to (Cao and Zhu, 2010).

$$z = rx^* + sy^* \tag{19}$$

r, s ∈ Z and x^* and y^* are initial values of x and y.

The iterative calls of recursive program produce a sequence of arguments or states $x_n = x^*$, $y_n = y^*$, x_{n-1}, y_{n-1}, ... Termination condition, state transition equations (constraint conditions) and goal will be described as algebraic formulae as follow:

We give termination condition according to exit condition of recursion and output value at exit.

$$z = GCD(0, y_i) = y_i, \quad \text{if} \quad x_i = 0, \quad i = 1, 2, \cdots, n \tag{20}$$

State transition equations (constraint conditions):

$$\begin{cases} f_1 = x_{i-1} - y_i \\ \qquad \text{if} \quad y_i < x_i \\ f_2 = y_{i-1} - x_i \\ i = 1, 2, \cdots, n-1, n \end{cases} \tag{21}$$

$$\begin{cases} f_3 = x_{i-1} - x_i \\ \qquad \text{Otherwise} \\ f_4 = y_{i-1} - y_i + x_i \\ (x_i > 0 \wedge y_i \geq 0) \vee (x_i \geq 0 \wedge y_i > 0) \\ i = 1, 2, \cdots, n-1, n \end{cases} \tag{22}$$

Goal:

$$\begin{aligned} & z = GCD(x_n, y_n) = y_i \quad \text{and} \quad y_i \quad \text{is} \quad \text{in} \\ & \text{the} \quad \text{above} \quad \text{termination} \quad \text{condition.} \end{aligned} \tag{23}$$

We assume that the given variables' ordering is

$$x_n \succ y_n \succ x_{n-1} \succ y_{n-1} \succ \cdots \succ x_1 \succ y_1 \tag{24}$$

In order to use pseudo-division correctly, we should triangularize state transition equations. Equation 21 and 22 will be transformed into:

$$\begin{cases} f_1^* = x_i - y_{i-1} \\ \qquad \text{if} \quad y_i < x_i \\ f_2^* = y_i - x_{i-1} \\ i = 1, 2, \cdots, n-1, n \end{cases} \tag{25}$$

$$\begin{cases} f_3^* = x_i - x_{i-1} \\ \qquad \text{Otherwise} \\ f_4^* = y_i - x_{i-1} - y_{i-1} \\ (x_i > 0 \wedge y_i \geq 0) \vee (x_i \geq 0 \wedge y_i > 0) \\ i = 1, 2, \cdots, n-1, n \end{cases} \tag{26}$$

We start to use pseudo-division to calculate Wu's characteristic set. There exists $F = r_n x_n + s_n y_n - z = 0$, where $x_n = x^*$ and $y_n = y^*$. If $y_n < x_n$, we use F to be divided by f^*_1 to obtain:

$$\begin{aligned} & F = r_n f_1^* + R_n' \\ & R_n' = s_n y_n + r_n y_{n-1} - z = 0 \end{aligned} \tag{27}$$

We use R'_n to be divided by f^*_2 to obtain:

$$\begin{aligned} & R_n' = s_n f_2^* + R_n \\ & R_n = s_n x_{n-1} + r_n y_{n-1} - z = 0 \end{aligned} \tag{28}$$

Let $r_{n-1} = s_n$ and $s_{n-1} = r_n$, $R_n = r_{n-1}x_{n-1} + s_{n-1}y_{n-1} - z = 0$. If $y_n \geq x_n$, we use F to be divided by f^*_3 to obtain:

$$\begin{aligned} & F = r_n f_3^* + R_n' \\ & R_n' = s_n y_{n-1} + r_n x_{n-1} - z = 0 \end{aligned} \tag{29}$$

We use R'_n to be divided by f^*_4 to obtain:

$$\begin{aligned} & R_n' = r_n f_4^* + R_n \\ & R_n = (r_n + s_n) x_{n-1} + s_n y_{n-1} - z = 0 \end{aligned} \tag{30}$$

Let $r_{n-1} = r_{n-1}+s_n$ and $s_{n-1} = s_n$, $R_n = r_{n-1}x_{n-1} + s_{n-1}y_{n-1} - z = 0$.

We apply pseudo-division to R_n (we substitute n-1 for subscript n and substitute n-2 for subscript n-1 in f^*_1, f^*_2, f^*_3 and f^*_4 here) and repeat to apply pseudo-division to calculating pseudo-remainder. There exists:

$$\begin{aligned} &R_{n-1} = r_{n-2}x_{n-2} + s_{n-2}y_{n-2} - z = 0 \\ &R_{n-2} = r_{n-3}x_{n-3} + s_{n-3}y_{n-3} - z = 0 \\ &\vdots \end{aligned} \tag{31}$$

In this way, we obtain a Wu's characteristic set $\{f^*_1, f^*_2, \ldots, R_2\}$. If $x_n, x_{n-1}, x_{n-2}, \ldots$ form a descending chain (we will prove it later) and there should exist $x_m = 0$ which satisfies termination condition and

$$R_{m+1} = r_m x_m + s_m y_m - z = 0 \tag{32}$$

when $x_m = 0$, the program will terminate and $z = s_m y_m$. If $s_m = 1$ and $m = 1$, $R_2 = y_1 - z = 0$ and $z = y_1 = GCD(0, y_1)$, namely z is output variable value.

Suppose that the relation formula between input variable values x_n and y_n and output variable value z or the relation formula among program states is:

$$\begin{aligned} &z = r_n x_n + s_n y_n = r_i x_i + s_i y_i = y_1. \\ &i = 1, 2, \ldots, n-1 \end{aligned} \tag{33}$$

According the above discussion, we can verify correctness of recursive program GCD by applying Wu's method and mathematical induction.

Proof: First we prove termination of recursive program and lay a foundation for induction.

If $x_1 = 0$ and $y_1 > 0$, then $x_1 = 0$ satisfies termination condition and $y_1 - z = 0$ satisfies the relation between input variables x_n and y_n and output value z. Here, the program will terminate and output variable value $z = GCD(0, y_1) = y_1$.

If $y_2 = 0$ and $x_2 > 0$, then because $y_2 < x_2$, we use $F = x_2 - z$ to be divided by $f^*_1 = x_2 - y_1 = 0$ and $f^*_2 = y_2 - x_1 = 0$ to obtain $R_2 = y_1 - z = 0$ and $x_1 = 0$ which satisfies termination condition. Therefore we obtain Wu's characteristic set $\{f^*_1, f^*_2, R_2\}$ and $F = f^*_1 + 0 \bullet f^*_2 + R_2$. Similarly the program will terminate and output variable value $z = y_1$.

If $x_i > 0$ and $y_i > 0$, then we calculate by using state transition equations. If $y_i < x_i$, then $x_{i-1} = y_i < x_i$. The chain $x_n, x_{n-1}, \ldots, x_i, \ldots, x_1$ is descending at this moment.

If $y_i \geq x_i$ then there exists $y_i = kx_i + r$, where $k = \lfloor y_i/x_i \rfloor$ and $0 \leq r < xi$, namely $r = MOD(y_i, x_i) < x_i$. We obtain:

$$\begin{cases} x_{i-k} = \cdots = x_{i-2} = x_{i-1} = x_i \\ y_{i-k} = y_i - kx_i = r \end{cases} \tag{34}$$

Apparently $y_{i-k} < x_{i-k}$, $x_{i-(k+1)} = y_{i-k} < x_{i-k}$. Therefore the chain $x_n, x_{n-1}, \ldots, x_i, \ldots, x_1$ is always descending. Because $x_i \in N(i = n, n-1, \ldots)$, there must exist $x_1 = 0$ and the program will terminate.

Assume the induction hypothesis that the relation formula between input variables x_{n-1} and y_{n-1} and output value z namely $r_{n-1}x_{n-1} + s_{n-1}y_{n-1} - z = r_{n-1}x_{n-1} + s_{n-1}y_{n-1} - y_1 = 0$ is true, namely output variable value $z = GCD(x_{n-1}, y_{n-1}) = r_{n-1}x_{n-1} + s_{n-1}y_{n-1} = y_1$. We need to prove that there exists $F = r_n x_n + s_n y_n - y_1 = 0$, namely $z = GCD(x_n, y_n) = r_n x_n + s_n y_n = y_1$.

$$F = r_n x_n + s_n y_n - y_1 \tag{35}$$

If $y_n < x_n$, we use F to be divided by f^*_1 and f^*_2 to obtain:

$$F = r_n f^*_1 + s_n f^*_2 + (s_n x_{n-1} + r_n y_{n-1} - y_1) \tag{36}$$

Let, $R = s_n x_{n-1} + r_n y_{n-1} - y_1 = 0$, we obtain Wu's characteristic set $\{f^*_1, f^*_2, R\}$ and

$$P = In(f^*_1)^{\alpha_1} In(f^*_2)^{\alpha_2} F = r_n f^*_1 + s_n f^*_2 + R$$

where, $In(f^*_1) = 1$ and $In(f^*_2) = 1$. When $s_n = r_{n-1}$, $r_n = s_{n-1}$, $P = F = 0$. Obviously output variable value $z = GCD(x_n, y_n) = r_n x_n + s_n y_n = y_1$.

If $y_n \geq x_n$, we use F to be divided by f^*_3 and f^*_4 to obtain:

$$F = r_n f^*_3 + s_n f^*_4 + ((r_n + s_n)x_{n-1} + s_n y_{n-1} - y_1) \tag{37}$$

Let, $R = (r_n + s_n)x_{n-1} + s_n y_{n-1} - y_1$, we obtain Wu's characteristic set $\{f^*_3, f^*_4, R\}$ and

$$P = In(f^*_3)^{\alpha_3} In(f^*_4)^{\alpha_4} F = r_n f^*_3 + s_n f^*_4 + R$$

where, $In(f^*_3) = 1$ and $In(f^*_4) = 1$. When $r_n = r_{n-1} - s_{n-1}$, $s_n = s_{n-1}$, $P = F = 0$. Here we can prove that output variable value $z = GCD(x_n, y_n) = r_n x_n + s_n y_n = y_1$.

Therefore, there exists $r_n \in Z$ and $s_n \in Z$ and the relation formula between input variables x_n and y_n and output variable value z namely $F = r_n x_n + s_n y_n - z = r_n x_n + s_n y_n - y_1 = 0$ and output value $z = GCD(x_n, y_n) = GCD(x^*, y^*) = y_1$. The program is partially correct. Because termination has been proved, we can obtain the program is totally correct.

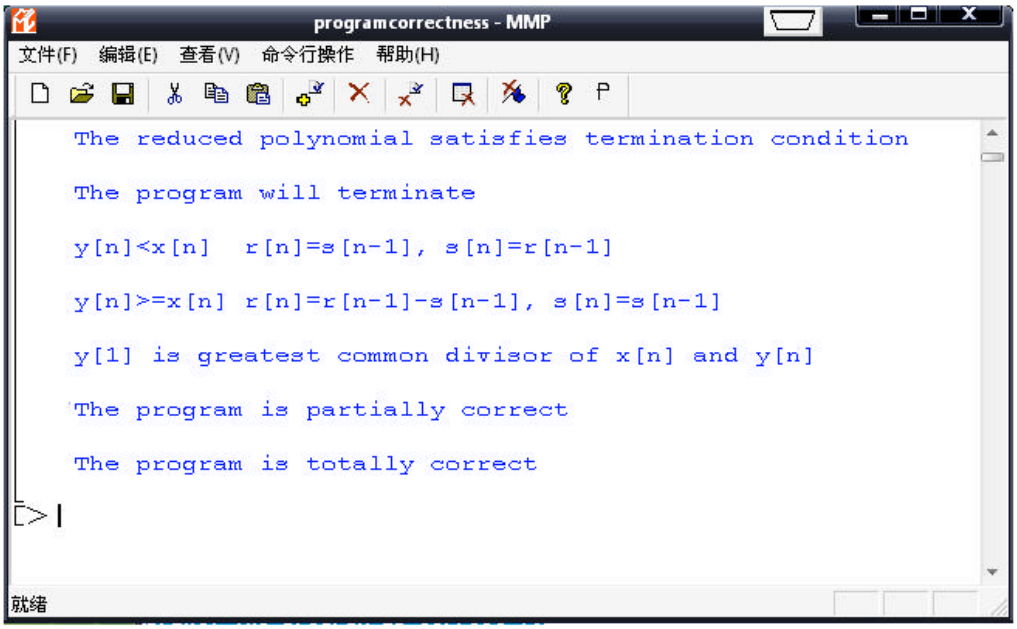

Fig. 1: The result of proving the program GCD

We conducted our project in our lab and completed the necessary experiments from Sep 2009 to Nov 2009. We make program to implement our reasoning to prove the total correctness of the example on automated reasoning platform MMP (http://www.mmrc.iss.ac.cn/mmp). We give variables' ordering and extract state transition equations, terminate condition and goal and transform them into algebraic formulae and triangularize state transition equations. The formulae are the input of our program and the program can automatically generate invariants and compute to decide whether the recursive program is partially correct and will terminate. The result is shown in Fig. 1.

SIf the program does not terminate, we can prove that it dose not satisfy termination condition. The example (Example 2) is shown as follow:

$$\begin{aligned} GCD1(x,y) == & \text{if } x = 0 \text{ then } y \\ & \text{else if } y < x \text{ then } GCD1(x-y, x) \\ & \text{else } GCD1(x, y-x) \end{aligned} \tag{38}$$

First, we extract termination condition, state transition equations and goal from the program and triangularize constraint conditions.

Termination condition:

$$z = GCD1(0, y_i) = y_i, \text{ if } x_i = 0, \ i=1,2,\cdots,n \tag{39}$$

State transition equations (constraint conditions):

$$\begin{cases} f_1^* = x_i - y_i - x_{i-1} \\ \qquad\qquad \text{if} \quad y_i < x_i \\ f_2^* = y_i - y_{i-1} \\ \\ i = 1,2,\cdots,n-1,n \end{cases} \tag{40}$$

$$\begin{cases} f_3^* = x_i - x_{i-1} \\ \qquad\qquad \text{Otherwise} \\ f_4^* = y_i - x_{i-1} + y_{i-1} \\ \\ (x_i > 0 \wedge y_i \geq 0) \vee (x_i \geq 0 \wedge y_i > 0) \\ i = 1,2,\cdots,n-1,n \end{cases} \tag{41}$$

Goal:

$$z = GCD(x_n, y_n) = y_i \text{ and } y_i \text{ is in the above termination condition.} \tag{42}$$

But the Example 2 may not terminate. When the relation formula among program states is reduced with respect to state transition equations, the corresponding states do not satisfy termination condition.

Proof: It can be easily seen that $x_n, x_{n-1}, \ldots, x_i, \ldots, x_1$ form a descending chain and there may exist $x_m = 0$ if $y_m > 0$. For

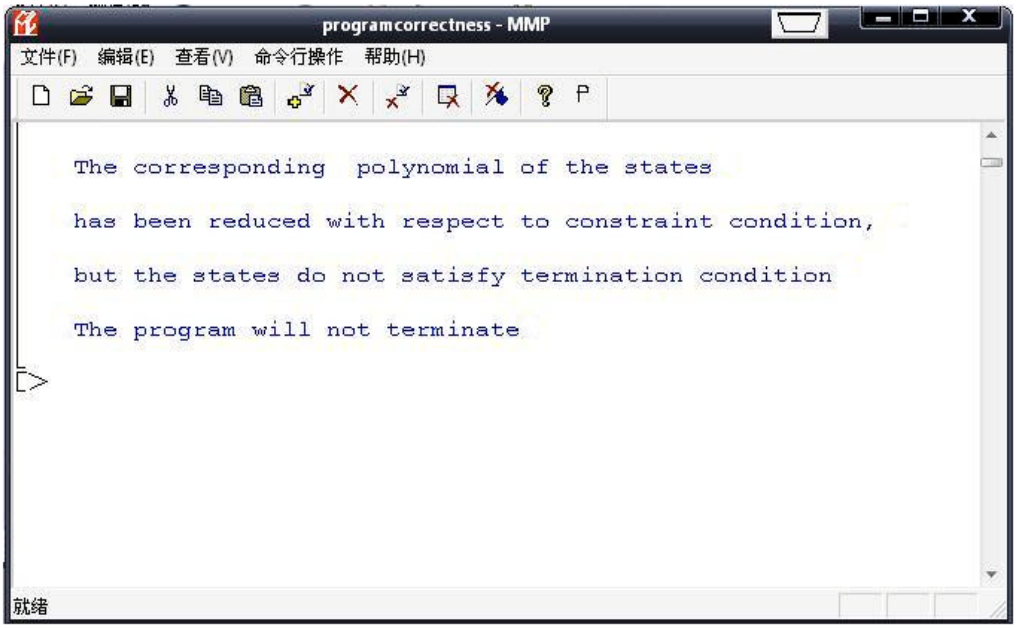

Fig. 2: The result of proving the program GCD1

the sake of simplicity let m = 1. According to relation formula among the states and state transition equations, if $x_1 = 0$ and $y_1 > 0$, then $y_1 - z = 0$ and $x_1 = 0$ which satisfies termination condition. We obtain the program will terminate and output value $z = y_1$ at this moment.

Similarly $y_n, y_{n-1}, \ldots, y_i, \ldots, y_1$ form a descending chain and there may exist $y_m = 0$ if $x_m > 0$. Let m = 1 and if $y_1 = 0$ and $x_1 > 0$, then $F = x_1 - z = 0$ is reduced with respect to $f^*_1 = x_1 - y_1 - x_0 = 0$ and $f^*_2 = y_1 - y_0 = 0$ and $y_1 = 0$ does not satisfy termination condition. Because $x_1 > y_1$, we should apply f^*_1, f^*_2 to pseudo-division to obtain R_1. Wu's characteristic set $\{f^*_1, f^*_2 R_1\}$ is obtained, but after computation we know $R_1 = x_1 - z = F$ and F is reduced with respect to f^*_1, and f^*_2. Because $x_1 > y_1$ again, f^*_1 and f^*_2 will be applied to pseudo-division to obtain R_1 repeatedly and the process will be infinite. The program will not terminate when $y_1 = 0$.

The result of proving program GCD1 on a PC with automated reasoning platform MMP is as shown in Fig. 2.

Definition 7: The states of program are reduced if the corresponding relation formula among the program states is reduced with respect to state transition equations.

From above discussion we know that if the reduced states of program do not satisfy termination condition, then the program will not terminate.

Suppose that the program will terminate, there must exist states satisfy termination condition and the reduced states will be transformed into termination states. Let, f be the corresponding relation formula among the reduced states. According to definition 7 and theorem 1 we use f to be divided by state transition equations to obtain pseudo-remainder R = f. The reduced states do not change in pseudo-remainder R and will not be transformed into termination states forever (there exits a infinite sequence of reduced states). This yields a contradiction. Therefore the program will not terminate.

EXPERIMENTS

The theorem prover Isabelle (http://isabelle.in.tum.de/) is a generic system for implementing logical formalisms and Isabelle/HOL is the specialization of Isabelle for HOL (high order logic). As mentioned ealier, our method has been implemented on MMP in some examples. Similarly we use Isabelle to prove the correctness of the examples. Compared with the other traditional approaches such as Isabelle, our method is proved to be more efficient through experiments especially in polynomial program. Its performance can be seen in Table 1. The experimental environment was an Intel Pentium 4 with 1.8 GHz of processor and 512 MB of memory.

We use Isabelle2008 to prove the examples on windows platform through simulator Cygwin which simulates a Unix-like environment. We generate some

Table 1: The comparison of execution time of experiment results between Isabelle and MMP (in sec)

Examples	Description	Source	Isabelle	MMP
GCD	Greatest common divisor	Hu *et al.* (2003)	1.1	0.31
Fibonacci	$an = a_{n-1}+a_{n-2}$	Hu *et al.* (2003)	0.9	0.27
Gauss	Σn	Hu *et al.* (2003)	0.6	0.19
Factorial	n!	Hu *et al.* (2003)	0.6	0.12

definitions, funs and lemmas etc for proving the examples. Factorial: 1 primrec and 5 lemmas; Fibonacci: 1 definition and 12 lemmas; Gauss: 1 primrec and 7 lemmas; GCD: 3 definitions, 1 fun and 20 lemmas.

CONCLUSIONS

There exists quite a lot formal verification approaches to prove program correctness. If we can make program correctness proving algebraized, the efficiency of proving will be improved greatly and it will be mechanized and automated easily. In this study, we transform program into algebraic transition system and apply Wu's method to recursive program correctness proving. The new approach gives new theoretical insights into program correctness proving. Compared classical formal method, our method is more efficient in many programs especially in polynomial program; compared (Popov and Jebelean (2009) our method more widely applicable because the method by Popov and Jebelean (2009) can only be applicable to Tail Recursive Programs which have a specific structure and dose not include multi-ply conditional statement etc. Mathematics mechanization is a powerful and promising methodology in program correctness proving. There exists diversiform programs and their proving methods should be diversiform. We need combine diversified program verification techniques to settle complicated verification problems.

ACKNOWLEDGMENTS

The results described in this study was supported by a grant from the National Science Foundation of China (NSFC) (60671033) and the Doctoral Foundation of the Ministry of Education of China (20060614 05).

REFERENCES

Cao, Y. and Q.X. Zhu, 2010. Finding loop invariants based on Wu's characteristic set method. Inform. Technol. J., 9: 349-353.

Hu, Z.G., J. Wu and Z.H. Deng, 2003. Methodology of Program Design. National Defence Industrial Press, Beijing.

Krauss, A., 2006. Partial Recursive Functions in Higher-Order Logic. In: Automated Reasoning, Furbach, U. and N. Shankar (Eds.). LNCS., 4130, Springer-Verlag, Berlin, Heidelberg, ISBN-13: 978-3-540-37187-8, pp: 589-603.

Mao, W. and J.Z. Wu, 2005. Application of wu's method to symbolic model checking. Proceedings of the International Symposium on Symbolic and Algebraic Computation, July 24-27, Beijing, China, pp: 237-244.

Popov, N. and T. Jebelean, 2009. Using computer algebra techniques for the specification, verification and synthesis of recursive programs. Math. Comput. Simulat., 79: 2302-2309.

Sankaranarayanan, S., H.B. Sipma and Z. Manna, 2004. Nonlinear loop invariant generation using Grobner bases. Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Jan. 14-16, ACM, Venice, Italy, pp: 318-329.

Wu, W.T., 2001. Mathematics Machenization. Kluwer Sience Press, Beijing.