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 (2): 349-353, 2010
ISSN 1812-5638

Finding Loop Invariants Based on Wu's Characteristic Set Method

Yong Cao and Qing-Xin Zhu
Institute of Computer Science and Engineering,
University of Electronic Science and Technology, China

Abstract: Loop invariants are important parts in program verification and proof. Correspondingly, techniques for automatically checking and finding invariants have been studied for many years. In present study, an approach using Wu's characteristic set method for automatically finding polynomial invariants of imperative programs is presented. Present method is based on the algebraic theory of polynomial set over polynomial rings, 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 other polynomial algebraic approaches, our method is more efficient through experiments.

Key words: Wu's method, loop invariants, characteristic set, computer algebra

INTRODUCTION

There has recently been a surge of interest in research on automatically finding loop invariants of imperative programs. Program verification based on Floyd-Hoare's inductive assertion method, using preconditions, post-conditions and loop invariants, was considered a major research problem in the seventies, leading to the development of many program verification systems. Karr first showed an arithmetic algorithm for finding invariant linear equalities at any program point of a procedure in literature (Rodriguez-Carbonell and Kapur, 2004a, b). Recently, Muller-Olm and Seidl (2004) proposed an interprocedural method for computing invariant polynomial of bounded degree in program. Due to the advance of computer algebra and there appears to be a revival of research activities relating to mechanically finding loop invariants, especially using abstract interpretations, quantifier elimination and the associated widening operators (Muller-Olm and Seidl, 2002, 2004; Rodriguez-Carbonell and Kapur, 2004a, b).

Wu's method is an algorithm of representing the zeros of a set of polynomials with the characteristic sets, which is invented by Chinese famous mathematician Wu Wen-Tsun. As an important method and tool, Wu's method performs mechanical geometry theorem proving better than other ways in symbolic computation and has aroused attention to symbolic computation once more. It mainly calculates a characteristic set of a finite set of polynomials and Wu Wen-Tsun successfully applied it to geometry theorem proving. Wu's work inspired the study of more efficient characteristic set algorithms and he becomes a leader in mathematics mechanization in China (Mao and Wu, 2005).

An invariant of a loop is hypothesized as a parameterized formula. Parameters in the invariant are discovered by generating constraints on the parameters by ensuring that the formula is indeed preserved by the execution path corresponding to every basic cycle of the loop. Constraints on parameters appearing in a parameterized formula hypothesized as an invariant can be successively generated by considering every basic cycle through the program location (typically the beginning of the loop) to which the invariant is associated.

The other methods for finding loop invariants are generally based on Grobner bases which are the theory of ideals over polynomial rings. Before we use the methods based on Grobner bases, we must prove that the methods are used over ideals (Sankaranarayanan *et al*., 2004). It is not necessarily for us to use Wu's characteristic set method because it is algebraic theory of polynomial set over polynomial rings. Wu's characteristic set method can be used to find loop invariants and we implement the approach with the computer algebra tools MMP (http://www.mmrc.iss.ac.cn/mmp). Compared with other approaches, our method is more efficient through experiments.

WU'S CHARACTERISTIC SET METHOD

Most approaches to geometric reasoning are formulated in terms of synthetic geometry which is a logical system expressed using properties of points, lines,

Corresponding Author: Yong Cao, No. 2, South Baihua Street, Chengdu City, P.O. Box 610041, Sichuan Province, Peoples Republic of China Tel: 86-28-87048656, 13882252841

angles, etc., considered as primitive elements. In contrast, Wu's method is algebraic. Hypotheses (or given conditions) and conclusions (conjectures) in a geometry problem are represented as polynomial equations. Geometry theorem can be proved using a fairly simple and elegant algebraic method such as Wu's method. A geometry statement of the form a finite set of hypotheses (or given conditions) implying a conclusion is considered. Hypotheses (or given conditions) are polynomial equations expressing geometric relations and the conclusion is also a polynomial equation stating a geometric relation to be derived; a subset of variables corresponding to coordinates of points that can be arbitrarily chosen in a geometry statement are viewed as parameters. Unlike the methods of Tarski, Seidenberg, Monk, Collins and Ben-Or *et al*., variables in Wu's method range over an algebraically closed field.

Definition 1: Given a polynomial:

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

Its initial polynomial, In(f) is defined to be the polynomial $I_d(x_1,...,x_{j-1})$.

Theorem 1: For any two polynomials f and g, there exist a non-negative number α and polynomials q and r, such that:

$$In(f)^{\alpha}g = qf+r \quad (2)$$

which, is called f pseudo-divides g and r is called pseudo-remainder of f pseudo-dividing g.

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

$$In(g_t)^{\gamma_t} \ldots In(g_1)^{\gamma_1} f_i = \sum_{j=1}^{t} h_j g_j$$

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 (prem is abbreviation of pseudo-remainder).

$$\begin{aligned} r_t &= prem(p, g_1)\\ r_{t-1} &= prem(r_t, g_2)\\ &\vdots\\ r_1 &= prem(r_2, g_t) \end{aligned} \quad (3)$$

and we know there should be:

$$In(g_t)^{\alpha_t} \cdots In(g_1)^{\alpha_1} p = \sum_{i=1}^{t} h_i g_i + r_1$$

If $r_1 = 0$ and $In(g_t)^{\alpha t} \neq 0$,...., $In(g_1)^{\alpha 1} \neq 0$, 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 $In(g_t)^{\alpha t} \neq 0$,...., $In(g_1)^{\alpha 1} \neq 0$, have been called nondegenerate conditions or subsidiary conditions in the literature (Mao and Wu, 2005).

Definition 2: We use Zero(S) to denote the zero set of S:

$$Zero(S) = \left\{ \begin{array}{l} \langle a_1, a_2, \ldots, a_n \rangle \in K^n : \\ (\forall p \in S)\left[p(a_1, a_2, \ldots, a_n) = 0 \right] \end{array} \right\} \quad (4)$$

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

$$\begin{aligned} &Zero(S) = \left(Zero(B) \setminus Zero(In(B))\right)\\ &\cup \bigcup_{i=1}^{n} \left(Zero(S \cup In(g_i))\right) \end{aligned} \quad (5)$$

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

$$Zero(P) = \bigcup_{i=1}^{u} (Zero(B_i) \setminus Zero(In(B_i))) \quad (6)$$

FINDING LOOP INVARIANTS BASED ON WU'S CHARACTERISTIC SET METHOD

An invariant at any program point is a formula relating program variable such that whenever control passes through the point, the formula is true. A path in a program is assumed to be a sequence of assignments, possibly interspersed with Boolean tests (which is assumed to evaluate to true) arising due to conditional statements and loops. For a given loop L, let $p_1, \ldots, p_{m,\ldots}$ be all possible basic cycles, consisting of Boolean tests and assignment statements, each of which starts at the beginning of the body of the loop and ends at the same point. For a formula $I(\bar{x})$ to serve as a loop invariant, it should be the case that the Hoare triple $\{I(\bar{x})\}\ p_i\ \{I(\bar{x})\}$ where p_i stands for the code fragment

corresponding to the path p_i and $\bar{x}$ is the tuple of all of the variables in the program.

Loops are supposed to have form as follow:

$$\begin{aligned}&\text{While } E(\bar{x}) \text{ do}\\&\text{path 1: } \langle p_{11}, \ldots, p_{1k_1}\rangle\\&\text{path 2: } \langle p_{21}, \ldots, p_{2k_2}\rangle\\&\vdots\\&\text{path m: } \langle p_{m1}, \ldots, p_{mk_m}\rangle\\&\vdots\\&\text{End}\end{aligned} \tag{7}$$

where, $p_{ij} = 0$ is a assignment statement ($x_{ij} = e_{ij}(\bar{x}) \Leftrightarrow x_{ij} - e_{ij}(\bar{x}) = 0$, where e_{ij} is a expression) which is polynomial generally and $\bar{x} = (x_1 x_2 \ldots x_n)$ denotes the tuple of program variables and *E* are Boolean expressions.

Suppose that polynomials p and q. Let r is pseudo-remainder of p pseudo-dividing q. $\exists y r(y) = 0 \rightarrow p(y) = 0 \wedge q(y) = 0$. When $p = 0$ and $q = 0$ we can also obtain $r = 0$. Therefore we have $Zero(r) = Zero(q \wedge q) = Zero(p) \cap Zero(q)$ under nondegenerate conditions. Similarly if r is pseudo-remainder of polynomial p pseudo-dividing polynomial set Q ($q_1, q_2, \ldots, q_m$) then there should be:

$$Zero(r) = Zero(p \wedge \wedge_{i=1}^{m} q_i) = Zero(p) \cap Zero(\cap_{i=1}^{m} q_i)$$

under nondegenerate conditions.

Invariant polynomial $I(\bar{x}) = 0$ should satisfies $E(\bar{x}) = 0$ and $p_{ij} = 0$ ($i = 1, 2, \ldots, m; j = 1, 2, \ldots, k_m$), namely:

$$Zero\left(I(\bar{x})\right) \subset \left(Zero\left(E(\bar{x})\right) \cap \bigcap_{i=1}^{m}\bigcap_{j=1}^{k_m} Zero(P_{ij})\right) \tag{8}$$

Therefore we have:

$$I(\bar{x}) \wedge \left(E(\bar{x}) \wedge \wedge_{i=1}^{m} \wedge_{j=1}^{k_m} P_{ij}\right) \rightarrow I(\bar{x}) \tag{9}$$

Suppose that the invariant assertion $I(\bar{x})$ is a parameterized polynomial equation over program variables or states:

$$I(x_1, x_{2,\ldots,} x_n) = \sum_{i_1, i_2, \ldots, i_n} a_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = 0 \tag{10}$$

For a given loop L, suppose that $p_1, \ldots, p_m$ are all possible closed paths, consisting of Boolean tests and assignment statements, each of which starts at the beginning of the body of the loop and ends at the same point. Suppose a closed path p_i which includes a sequence of assignment statements $p_{i1}, p_{i2}, \ldots p_{ik_i}$. We compute characteristic set B $\{p'_{i1}, p'_{i2}, \ldots p'_{ik_i}\}$ and $I(\bar{x})$ pseudo-divide the characteristic set B to obtain pseudo-remainder $I'(\bar{x})$. $s = I(\bar{x}) - I'(\bar{x}) = 0$ where s is a polynomial formula. Because s equals zero for all the possible values of $x_1, x_2, \ldots, x_n$ (Sankaranarayanan *et al*., 2004) and its all coefficients are identically zero ($I(\bar{x})$ is a invariant). The coefficients of s are linear combination of coefficients of invariant assertion $I(\bar{x})$. In the closed path we can determine the partial or total coefficients of invariant assertion $I(\bar{x})$. Let the invariant assertion of which partial coefficients are ascertained be $I_d(\bar{x})$ and obviously $I_d(\bar{x})$ are derived from $I(\bar{x})$. Because the same parameterized assertion can be used for multiple closed paths, $I_d(\bar{x})$ is invariant assertion of another closed path. Repeat the above way and we can compute other coefficients of invariant assertion $I_d(\bar{x})$. Similarly, the rest coefficients of $I(\bar{x})$ may be deduced by analogy. Finally we compute constant coefficient of invariant by substituting initial values of variables. In this way, we can obtain the invariant assertion $I(\bar{x})$ for every closed path.

Algorithm 1:

Input: Algebraized goal $I(\bar{x})$ with parameters and constraint conditions of all closed paths $p_{i1}, p_{i2}, \ldots pik_i$. which have been represented as equations.

Output: Invariant assertion of loop

Begin

for i=1 to m{
Compute characteristic set B = $\{p'_{i1}, p'_{i2}, \ldots p'ik_i.\}$ of constraint conditions $\{p_{i1}, p_{i2}, \ldots pik_i.\}$ of the ith closed path;
$I(\bar{x})$ pseudo-divide B to obtain $I'(\bar{x})$;
$s = I(\bar{x}) - I'(\bar{x})$;
Because all coefficients of s are identically zero, we compute part parameters of $I(\bar{x})$;
}
Substitute initial values of variables to compute constant coefficient of invariant;
return $I(\bar{x})$; /*Its coefficients are determined/

End.

In order to demonstrate our method, we give an example which is shown as follow.

Example 1:

```
function product (X, Y: integer) return z: integer
    var x, y, z: integer
    ⟨x, y, z⟩ = ⟨X, Y, 0⟩;
    while y<>0 do
    ⟨x, y, z⟩ = ⟨2x, y div 2, z⟩;
    [] ⟨x, y, z⟩ = ⟨2x, (y-1) div 2, x+z⟩;
    end while
    end
```

We transform the assignment statements of each closed path in Example 1 into polynomial equations.

$$\begin{array}{ll} \tau_1 : p_{11} = x - 2u & \tau_2 : p_{21} = x - 2u \\ \quad p_{12} = y - (1/2)v & \quad p_{22} = y - (1/2)(v-1) \\ & \quad p_{23} = z - (u + w) \end{array} \tag{11}$$

Suppose that I(x, y, z) is a polynomial equation in x, y, z in which the degree of every variable is at most 1.

$$I(x,y,z) = a_1xyz + a_2xy + a_3xz + a_4yz + a_5x + a_6y + a_7z + a_8 = 0 \tag{12}$$

At first we compute characteristic set B of $\{p_{11} = x\text{-}2u, p_{12} = y\text{-}(1/2)v\}$. Then we compute the pseudo-remainder r of I(x,y,z) pseudo-dividing the characteristic set B through MMP.

```
>css:=[[a1*x*y*z + a2*x*y + a3*x*z + a4*y*z+ a5*x +
a6*y + a7*z +a8], [x-2u],[y-(1/2)v]]:
>ord:=[x, y, z, u, v]:
>charser(css, ord);
```

where, variables' ordering is given by ord and charser is a function in MMP.

After computation we obtain:

$$I'(x,y,z) = a_1xyz + a_2xy + 2a_3xz + \frac{1}{2}a_4yz + 2a_5x + \frac{1}{2}a_6y + a_7z + a_8 = 0 \tag{13}$$

Successively we compare the coefficients of I(x,y,z) and I'(x,y,z) to obtain:

$$s = a_3xz - 2a_4yz + a_5x - \frac{1}{2}a_6y = 0 \tag{14}$$

According to literature (Sankaranarayanan *et al*., 2004) a polynomial s is zero for all the possible values of $x_1; x_2,\ldots, x_n$ if its all coefficients are identically zero. This implies that the coefficient of every term in the polynomial is 0, namely $a_3 = a_4 = a_5 = a_6 = 0$.

$$I(x,y,z) = a_1xyz + a_2xy + a_7z + a_8 \tag{15}$$

In succession we compute characteristic set B of $\{p_{21} = x\text{-}2u, p_{22} = y\text{-}(1/2)(v\text{-}1), p_{23} = z\text{-}(u+w)\}$. We compute the pseudo-remainder r of I(x,y,z,) pseudo-dividing the characteristic set B through MMP.

```
>css:=[[a1*x*y*z + a2*x*y  + a7*z  +  a8],[x-2u],[y-(1/2)
(v-1)], [z-(u+w)]]:
>ord:=[x, y, z, u, v, w]:
>charser(css, ord);
```

We obtain:

$$I'(x, y, z) = 2a_1x^2y + 2a_1xyz + a_2xy + (a_7 - a_2)x + a_7z + a_8 = 0 \tag{16}$$

We compare the coefficients of I(x,y,z) and I'(x,y,z) to obtain $a_1 = 0$ and $a_7\text{-}a_2 = 0$. Now we have $I(x,y,z) = a_2xy+a_7Z+a_8 = 0$. When loop is entered, the initial values x* and y* of the variables will be substituted in I(x, y, z) to obtain $I(x^*, y^*, 0) = a_2x^*y^* + a_8 = 0$ which gives $a_8 = -a_2x^*y^*$.

In this way the loop invariant is found:

$$I(x, y, z) = a_2xy + a_2z - a_2x^*y^* = 0 \Rightarrow xy + z - x^*y^* = 0 \tag{17}$$

In the above computation we hypothesize that the bounded degree is 1. When we hypothesize that the bounded degree is greater than 1, we can still obtain loop invariant $xy + z - x^*y^* = 0$.

EXPERIMENTS

Impressive results achieved using Wu's method have once again sparked a great deal of interest in automatic geometry theorem proving. Kapur and Wan (1990) in particular has reported success in using Wu's method for all of these problems and his program took much less time. Based on the experiments of Kapur and Wan (1990), it appears that the method using Wu's characteristic set computation is more efficient than the method using Grobner basis algorithm.

Let Wu's characteristic set $B = \{g_1, g_2,\ldots,g_t) \subseteq K(x_1, x_2,\ldots,x_n\}$ and $\deg(g_i) \leq d, \leq i \leq t$. By literature (Gallo and Mishra, 1994) the complexity of computing Wu's characteristic set is $O(s^{O(n)}(d+1)^{O(n^2)})$. Then the Wu's algorithm for finding loop invariants takes $O(s^{O(n^2)}(d+1)^{O(n^4)} + mn)$, where m is the number of closed paths. Therefore, the complexities are all $O(s^{n^{O(1)}}(d+1)^{n^{O(1)}})$ which is called a singly exponential complexity w.r.t. n. If we adopt Grobner base algorithm the complexity is:

$$O(2(\frac{d^2}{2} + d)^{2^{n-1}} + mn)$$

(Lakshman, 1990), namely:

$$O(2(\frac{d^2}{2} + d)^{2^{O(n)}})$$

which is double exponential w.r.t n. The method using Wu's algorithm is superior to Grobner base obviously. The experiments also validate the case.

Table 1: The time comparison of finding loop invariants between the experimental results using Grobner bases (Pentium 4 with a 3.4 GHz processor and 2 GB of memory) in literature (Rodríguez-Carbonell and Kapur, 2007) and our results using Wu's method (Pentium 4 with a 1.8 GHz processor and 512MB of memory and obviously the computer performance applying Grobner bases is superior to mine). in: Seconds

Name	Function	Source	Grobner bases	Wu's method
Dijkstra	$\sqrt[2]{}$	(Dijkstra)	1.5	0.56
Divbin	Division	(Kaldewaij)	2.1	0.99
Freire1	$\sqrt[2]{}$	(Freire)	0.7	0.48
Freire2	$\sqrt[3]{}$	(Freire)	0.7	0.51
Cohencu	Cube	(Cohen)	0.7	0.33
Fermat	Factor	(Bressoud)	0.8	0.43
Wensley	Division	(Wegbreit)	1.1	0.47
Lcm	Lcm	(Dijkstra)	1.0	0.44
Petter3	Power sum	(Petter)	1.3	0.33
Petter4	Power sum	(Petter)	1.3	0.38
Petter5	Power sum	(Petter)	1.4	0.43

We have presented a sound method based on Wu's characteristic set method for finding polynomial invariants of imperative programs. The technique has been implemented using the algebraic geometry tool MMP. The implementation has successfully computed invariants for many non-trivial programs. Its performance is satisfactory as can be seen in Table 1. Compared with other approaches (Rodríguez-Carbonell and Kapur, 2007), our method is more efficient through these experiments.

CONCLUSION

An invariant of a program at a location can be seen as an assertion (we transform the assertion into polynomial equation in this study) that is true of any program state reaching the location. The importance of the automatic invariant finding problem for the verification and the analysis of programs are well-known. We have presented an approach for automatically finding invariants of loop programs using Wu's characteristic method and implement it on auto reasoning platform MMP. The technique has many advantages; first of all, our method has wider application domain. Secondly compared other methods by experiments we prove that our method is more efficient. Automatic proving and verification of program is a trend and algebraic method can be mechanized easily. For future research, we are interested in exploring the proposed research along such directions: generalize and extend our method to other program models and integrate this and other methods for mechanically program verification and proving etc.

ACKNOWLEDGMENTS

The study described in this research 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 (2006061405).

REFERENCES

Gallo, G. and B. Mishra, 1994. The complexity of resolvent resolved. Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, Jan. 23-25, Arlington, Virginia, pp: 280-289.

Kapur, D. and H.K. Wan, 1990. Refutational proofs of geometry theorems via characteristic set computation. Proceedings of the International Symposium on Symbolic and Algebraic Computation Aug. 20-24, Tokyo, Japan, pp: 277-284.

Lakshman, Y.N., 1990. On the complexity of computing a Grobner basis for the radical of a zero dimensional ideal. Proceedings of the 22th Annual ACM Symposium on Theory of Computing, May 14-16, Baltimore, Maryland, USA., pp: 555-563.

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, Jul. 2427, Beijing, China. pp: 237-244.

Muller-Olm, M. and H. Seidl, 2002. Polynomial constants are decidable. Proceedings of the 9th Static Analysis Symposium, Sept. 17-20, Springer-Verlag, Technical University of Madrid (Spain), pp: 4-19.

Muller-Olm, M. and H. Seidl, 2004. Computing interprocedurally valid relations in affine programs. Proceedings of the ACM SIGPLAN Principles of Programming Languages, Jan. 14-16, Venice, Italy, pp: 330-341.

Rodriguez-Carbonell, E. and D. Kapur, 2004a. An abstract interpretation approach for automatic generation of polynomial invariants. Proceedings of the 11th Static Analysis Symposium, Aug. 26-28, Springer-Verlag, Verona, Italy, pp: 2-18.

Rodriguez-Carbonell, E. and D. Kapur, 2004b. Automatic generation of polynomial loop invariants: Algebraic foundations. Proceedings of the International Symposium on Symbolic and Algebraic Computation, Jul. 4-7, Santander, Spain, pp: 266-273.

Rodríguez-Carbonell, E. and D. Kapur, 2007. Generating all polynomial invariants in simple loops. J. Symb. Comput., 42: 443-476.

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.