Upper approximation method for polynomial invariants

O. Maksymets

We present a solution for polynomial invariant generation problem for programs. We adopt iteration upper approximation method that was successfully applied on free algebras for polynomial algebras. Set of invariant is interpreted as an ideal over polynomial ring. Relationship, intersection problems solution are proposed. Intersection of Gröbner basis are used to solve intersection problem. Inverse obligatory is applied to solve relationship problem.

Introduction

After verification of programs based on Floyd-Hoare-Dijkstra's inductive approval inductive approval, using pre/postconditions and loop invariants [1] in the seventies (Wegbreit, 1974, 1975; German and Wegbreit, 1975; Katz and Manna, 1976; Cousot and Cousot, 1976; Suzuki and Ishihata, 1977; Dershowitz and Manna, 1978) there was silent period in this domain. Recently significant progress in development of automated provers, SAT solvers and models checkers had place. All mentioned tools use assertions as input data. Therefore, during last years problem of finding assertion for programs became actual again.

We interpret program as U-Y schema on algebra of polynoms. Iterative algorithms applied for free algebras and vector space [5] was adopted in this paper for polynomial space.

An *invariant* of a program at a location is an assertion that is true of any program state reaching the location. Proposed approach generates basis of invariants for each program state taking in consideration assertions that were in initial state.

This work was inspired by related work done in generating invariants for polynomial space using Gröbner basis (Müller-Olm and Seidl, 2004b, Sankaranarayanan et al., 2004, Rodriguez-Carbonell and Kapur, 2007). We argue some opportunity to discover more invariants using iterational method, that looks promising on smaller problems.

Preliminaries

Let A be U-Y program over memory[3] with set of variables $R = \{r_1, \ldots, r_m\}$ that defined on algebra of data (D, Ω). $K(\Omega, Eq)$ is an algebra class that includes algebra (D, Ω) [2]. We consider (D, Ω) is algebra of polynomials $\Re[r_1, \ldots, r_m]$ and $T(\Omega, R)$ is algebra of terms on R from class $K(\Omega, Eq)$.

Definition 1. *(Algebraic Assertions) An algebraic assertion ψ is an assertion of the form $\bigwedge_i p_i(r_1, \ldots, r_m) = 0$ where each $p_i \in \Re[r_1, \ldots, r_m]$. The degree of an assertion is the maximum among the degrees of the polynomials that make up the assertion.*

Definition 2. *(Ideals) A set $I \subseteq \Re[r_1, ..., r_n]$ is an ideal, if and only if*

1. $0 \in I$.
2. *If $p_1, p_2 \in I$ then $p_1 + p_2 \in I$.*
3. *If $p_1 \in I$ and $p_2 \in \Re[r_1, ..., r_n]$ then $p_1 \cdot p_2 \in I$ [4].*

An ideal generated by a set of polynomials P, denoted by $((P))$ is the smallest ideal containing P. Equivalently,

$$((P)) = \{g_1 p_1 + \ldots + g_m p_m | g_1, \ldots, g_m \in R[r_1, \ldots, r_n], p_1, \ldots, p_m \in P\}$$

An ideal I is said to be finitely generated if there is a finite set P such that $I = ((P))$. A famous theorem due to Hilbert states that all ideals in $\Re[r_1, \ldots, r_n]$ are finitely generated. As a result, algebraic assertions can be seen as the generators of an ideal and vice-versa. Any ideal defines a variety, which is the set of the common zeros of all the polynomials it contains.

Definition 3. *(Ideals intersection) A set K is an intersection of ideals $I = \{f_1, \ldots, f_l\}$ and $J = \{g_1, \ldots, g_m\}$ if*

$$K = \{s(r_1, \ldots, r_n) | s(r_1, \ldots, r_n) = \sum_{i=1}^{l} pi \cdot f_i = \sum_{j=1}^{m} qj \cdot g_j,$$
$$p_1, \ldots, p_l, q_1, \ldots, q_m \in \Re[r_1, \ldots, r_n]\} \quad (1)$$

Theorem 1 (Ideal intersection)**.** *Let I and J be ideals in $R[r_1, \ldots, r_2]$.*

$$I \cap J = (t \cdot I + (1 - t) \cdot J) \cap \Re[r_1, \ldots, r_2] \quad (2)$$

where t is a new variable [4].

Proof. Note that $tI + (1 - t) J$ is an ideal in $\Re[x_1, \ldots, x_n, t]$. To establish the desired equality, we use strategy of proving containment in both directions.

Suppose $f \in I \cap J$. Since $f \in I$, we have $t \cdot f \in tI$. Similarly, $f \in J$ implies $(1 - t) \cdot f \in (1 - t) J$. Thus, $f = t \cdot f + (1 - t) \cdot f \in tI + (1 - t) J$. Since $I, J \subset \Re[x_1, \ldots, x_n]$.

To establish containment in the opposite direction, suppose $f \in (tI + (1 - t) J) \cap \Re[r_1, \ldots, r_n]$. Then $f(r) = g(r, t) + h(r, t)$, where $g(r, t) \in tI$ and $h(r, t) \in (1 - t) J$. First set $t = 0$. Since every element of tI is a multiple of t, we have $g(r, 0) = 0$. Thus, $f(r) = h(r, 0)$ and hence, $f(r) \in J$. On the other hand, set $t = 1$ in the relation $f(r) = g(r, t) + h(r, t)$. Since every element of $(1 - t) J$ is a multiple of $1 - t$, we have $h(r, 1) = 0$. Thus, $f(r) = g(r, 1)$ and, hence, $f(r) \in I$. Since f belongs to both I and J, we have $f \in I \cap J$. Thus, $I \cap J \supset (t \cdot I + (1 - t) \cdot J) \cap \Re[r_1, \ldots, r_2]$and this completes the proof. ■

$A = \{a_0, a_1, \ldots, a_*\}$ is a nodes set of U-Y schema. N_{a_i} is basis of assertions that we have in node a_i on current step of method. $N_{a_0}, N_{a_1}, \ldots, N_{a_*}$ is a set of assertion basis for nodes of U-Y schema. We consider set of conditions U with elements of structure $u = (p(r_1, \ldots, r_n) = 0)$, where $p(r_1, \ldots, r_n) \in \Re[r_1, \ldots, r_n]$. Set of assignments Y has elements structure $r_i := p(r_1, \ldots, r_n)$, where $p(r_1, \ldots, r_n) \in \Re[r_1, \ldots, r_n]$.

Algorithm of UAM

Let provide listing of upper approximation method (UAM) from [2]

Input: N_0 is start conditions and U-Y scheme A.

Output: N is set of invariants.

$N_{a_0} := N_0$
ToVisit.push(a_0)
Visited := {}
while ToVisit $\neq \emptyset$ **do**
  c := ToVisit.pop()
  Visited := Visited + c
  for all (c, y, a') **do**
    if Not a' in Visited **then**
      $N'_a := ef(N_c, y)$
      ToVisit.push(a'))
    end if
  end for
end while
ToVisit := $A/\{a_0\}$
while ToVisit $\neq \emptyset$ **do**
  $c := takefrom$ ToVisit
  if $N_c \neq \emptyset$ **then**

$N := N_c$
for all (a', y, c) **do**
$N := N \cdot ef(N_c, y)$
end for
if then$(N \neq N_c)$
$N_c := N$
ToVisit := ToVisit $+\{a|$ for every $(c, y, a)\}$
end if
end if
end while

Therefore to apply algorithm for polymonial algebra relationship, intersection and stabilization problems should be solved.

Relationship Problem. Given the algebraic basis of assertions set M and the operator $y \in Y$. Construct the algebraic basis assertions set $ef(M, y)$ that implies after assignment operator. We consider particular case of invertible assignments to solve relationship problem. In this case equality that assignment presents $r_i' = p(r_1, \ldots, r_n)$ can be transform as $r_i = p(r_1, \ldots, r_i', \ldots, r_n)$, where r_i' is new value of variable. Effect function that execute assignment of schema a is simple replacement old variable with new polynom.

Intersection Problem. Given the algebraic basis of assertions sets I and J. Construct the algebraic basis assertions set $I \cap J$. Accordingly to Theorem 1 intersection construction can be held using (2).

Stabilization Problem. Show that the construction process of basis assertions sets associated to the program states stabilizes. Investigation of this problem is out of scope of this paper.

Conclusion

In this paper we present theoretical basis for application of UAM on program with polynomial algebra. Ideal interpretation for program invariants was chosen. Operations defined on Gröbner basis satisfy all requirements stated in [2] to apply UAM, but additional proofs required.

Future work will refer to method application and deep investigation of stabilization problem.

References

[1] T. Hoare. The Verifying Compiler: A Grand Challenge for Computing Research. Journal of the ACM, No. 50(1), P. 63–69, 2003

[2] A. B. Godlevskii, Y. V. Kapitonova, S. L. Krivoi, A. A. Letichevskii Iterative Methods of Program Analysis, Cybernetics and Systems Analysis Vol. 25, No. 2, 1989, 139–152.

[3] A. A. Letichevsky. On finding invariant relations of programs. In Algorithms in Modern Mathematics and Computer Science (Urgench, 1979), number 122 in LNCS, pages 304-314, 1981.

[4] B. Buchberger, F. Winkler Gröbner Bases and Applications, Cambridge University Press, 1998

[5] O. M. Maksymets, Check of Invariants generated by Iterative Algorithm for programs on Absolutely Free Algebra using Mathematical Induction, Problems of Programming 2012, Vol. 2-3, 228-333

Authors

Oleksandr Mykolaiovych Maksymets — the 3rd year post-graduate student, faculty of cybernetics, Taras Shevchenko national university of Kiev, Kiev, Ukraine; E-mail: *maksymets@gmail.com*