GENERATING EXACT NONLINEAR RANKING FUNCTIONS BY SYMBOLIC-NUMERIC HYBRID METHOD*

SHEN Liyong · WU Min · YANG Zhengfeng · ZENG Zhenbing

DOI: 10.1007/s11424-013-1004-1
Received: 5 January 2011 / Revised: 30 January 2012
©The Editorial Office of JSSC & Springer-Verlag Berlin Heidelberg 2013

Abstract This paper presents a hybrid symbolic-numeric algorithm to compute ranking functions for establishing the termination of loop programs with polynomial guards and polynomial assignments. The authors first transform the problem into a parameterized polynomial optimization problem, and obtain a numerical ranking function using polynomial sum-of-squares relaxation via semidefinite programming (SDP). A rational vector recovery algorithm is deployed to recover a rational polynomial from the numerical ranking function, and some symbolic computation techniques are used to certify that this polynomial is an exact ranking function of the loop programs. At last, the authors demonstrate on some polynomial loop programs from the literature that our algorithm successfully yields nonlinear ranking functions with rational coefficients.

Key words Program verification, ranking function, semidefinite programming, symbolic-numeric hybrid method.

1 Introduction

For many applications, guaranteed termination of programs is necessary, especially in those situations where unexpected behavior can be catastrophic, such as spaceflight. Hence, termination determination of a program is an important and challenging research subject in the study of software verification. Although there exists already a rich body of literature on this subject, it is still an open task to design a termination checker for a realistic programming language, even when all the guards and assignments are given as piecewise linear functions. It is proved that the termination problem is undecidable in general in [1]. Some results are available to determine termination in simple cases. Tiwari[2] proved that termination of a class of single-path loops with linear guards and linear assignments is decidable and provided a decision procedure.
via constructive proofs. Based on this work, Braverman[3] showed that termination of a simple class of linear loops over the integer is decidable.

For loop programs with nonlinear guards and assignments, some other strategies exist for analyzing the termination. Using the finite difference trees, Bradley, et al.[4] presented a technique to prove termination of multiple path loops with polynomial guards and polynomial assignments. Babic, et al.[5] described an algorithm for proving termination of a class of loops with nonlinear assignments by divergence.

A recent trend on automatic proofs of termination for linear imperative loops has mostly focused on the synthesis of ranking functions. A ranking function for a loop maps the values of the loop variables to a well-founded domain, whose value monotonically decreases as the system moves forward. A ranking function is linear if it is a linear combination of the loop variables and a constant. Colón and Sipma[6] first presented an algorithm to generate linear ranking functions in a deductive manner, and then extended the algorithm to loops with multiple paths and the nested loops based on the manipulation of polyhedral cones in [7]. The authors gave a complete and efficient linear ranking function synthesis method for single loops that can be represented as a linear inequality system in [8]. It considers nondeterministic update of variable values to allow for abstraction. But this method does not work for multiple path loops. Based on building ranking function templates and checking satisfiability of template instantiations that are Presburger formulas, Bradley, et al.[9] can discover linear ranking functions for any linear loops over integer variables. The method is complete, which means that a rank function can be found if it exists, but neither efficient nor terminating on some loops. For loops with polynomial guards and polynomial assignments, Chen, et al.[10,11] reduced ranking function discovering to determining if semi-algebraic systems have real solutions. This method can yield all the polynomial ranking functions with the given degree bounds, however, it relies on cylindrical algebra decomposition (CAD), which is of high complexity.

Cousot[12] presented a numerical method of synthesizing nonlinear ranking functions for linear and quadratic loops by solving the constraint problem. The method is quite efficient, but it is incomplete due to the Lagrangian relaxation that it utilizes. Moreover, solving SDP systems and Lagrangian relaxation problem by fixed precision computation are subject to numerical errors, therefore the obtained numerical ranking function is not a certified ranking function, because it only satisfied the conditions of the ranking functions approximately, but not exactly.

In this paper, we propose a symbolic-numeric hybrid method on synthesizing nonlinear ranking functions for polynomial loops. We transform the problem of finding a polynomial ranking function of given degree into a polynomial optimization problem, then semidefinite programming (SDP) is applied to solve the optimization problem via polynomial sum-of-squares (SOS) relaxation technique. As known, SDP is the generalization of linear programming, which is based on interior-point method. Mostly important, SDP can be solved in polynomial-time. Therefore, numerical ranking functions of the loop programs, which approximately satisfies the ranking function constraints, will be achieved efficiently by solving the associated polynomial optimization problem via SDP. We next deploy two algorithms in symbolic computation to compute a ranking function which satisfies exactly the ranking function conditions: 1) rational vector recovery algorithm is used to compute the rational coefficients vector of the exact ranking function; 2) DISCOVERER[13] or RAGLIB[14] Maple package can be exploited to certify that one polynomial is a ranking function via determining if two semi-algebraic systems have no real solutions.

The rest of the paper is organized as follows. In Section 2, we recall some definitions of ranking functions and semi-definite programming, and then transform the problem of ranking function discovering into polynomial optimization solving. The methods on recovering and certifying exact ranking functions are proposed in Section 3. An algorithm and some examples