SMT-Based Array Invariant Generation*

Daniel Larraz, Enric Rodríguez-Carbonell, and Albert Rubio

Universitat Politècnica de Catalunya, Barcelona, Spain

Discovering loop invariants is an essential task for verifying the correctness of programs or computer systems in general. In this talk we present a technique for generating universally quantified loop invariants over array variables.

Namely, programs are assumed to consist of unnested loops and contain linear expressions in assignments, `if` and `while` conditions, as well as in array accesses. Now, let $\overline{a} = (A_1, \ldots, A_m)$ be the array variables of a program. Given a positive integer $k > 0$, our method generates invariants of the form

$$\forall \alpha \ : \ 0 \leq \alpha \leq \mathcal{C}(\overline{v}) - 1 \ : \ \Sigma_{i=1}^{m} \Sigma_{j=1}^{k} a_{ij} A_i[d_{ij}\alpha + \mathcal{E}_{ij}(\overline{v})] \ + \mathcal{B}(\overline{v}) + b_\alpha \alpha \ \leq \ 0$$

where $\mathcal{C}$, $\mathcal{E}_{ij}$ and $\mathcal{B}$ are linear polynomials with integer coefficients over the scalar variables of the program $\overline{v} = (v_1, \ldots, v_n)$ and $a_{ij}, d_{ij}, b_\alpha \in \mathbb{Z}$, for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, k\}$. This family of properties is quite general and allows us to handle a wide variety of programs for which we can automatically generate non-trivial invariants.

Unlike previous approaches based on abstract interpretation or first-order theorem proving, our method builds upon the so-called *constraint-based* invariant generation approach. This method produces linear invariants, i.e., invariants expressed as linear inequalities over scalar variables, by transforming the problem of the existence of an inductive invariant for a loop into a satisfiability problem in propositional logic over non-linear arithmetic, thanks to Farkas' Lemma. Despite the potential of the method, its application has been limited so far due to the lack of good solvers for the obtained non-linear constraints.

However, recently significant progress has been made in SMT modulo the theory of non-linear arithmetic. In particular, the Barcelogic SMT solver has shown to be very effective on finding solutions in the presence of non-linear integer arithmetic. It can also combine integers and reals, which is very useful when handling the constraints generated by the constraint-based invariant generation approach.

Our techniques have been successfully implemented in the CppInv tool. By using the Barcelogic SMT solver as a back-end, it automatically generates inductive loop invariants (both linear scalar invariants as well as array invariants) for programs written in a subset of the C++ language. We believe that the combination of our tool with some static analysis to infer the set of potentially interesting invariants for proving some given property would be very useful in the automation of the verification process.

* This work has been partially supported by the Spanish MEC/MICINN under grant TIN 2010-68093-C02-01