SAV project proposal
Templates in Function Postconditions

Ravichandhran Kandhadai Madhavan

April 12, 2013

1 Problem Definition

The goal of the project is to (a) enable the use of (quantifier-free) linear templates with unknown coefficients in the postconditions and (b) to develop an inference engine for finding an instantiation of the linear template that is inductive. In this project, the focus is only on Leon programs that can be expressed as linear transition systems. In such programs every primitive expression is a linear combination of program variables and function invocations. However, the programs may have *if-then-else* constructs and *let* constructs. Also, every expression in the program is of type *Int*. The templates in the postconditions can use the parameters, the result variable and also user-defined functions. Formally, a template is an expression of the form $a_0 + a_1x_1 + \cdots + a_{k-1}x_{k-1} + a_kf_k(\bar{a}_k) + \cdots + a_nf_n(\bar{a}_n)$ *op* 0 where *op* $\in \{\leq, =, \neq\}$, each x_i is a program variable, each f_i is a user-defined function symbol, each a_i is a constant or a unknown coefficient (referred to as a *template variable*) and each $\bar{a}_i$ is a set of template variables.

2 The Approach

[2, 3] proposes an approach based on constraint solving for inferring inductive invariants that are instantiations of a predefined linear template. This approach can be adapted to construct the inference engine required by the project. Consider a function (belonging to the above restricted language) whose postcondition uses templates. Assume that the postcondition is a single atomic predicate. The verification condition computed by Leon for the function would be of the form: $\varphi : \bigwedge_i \varphi_i$, where, $\varphi_i : \forall \overline{x}.\phi[\overline{x}, \overline{a}] \Rightarrow p[\overline{x}, \overline{a}]$, $\overline{x}$ is a set of program variables, $\overline{a}$ is a set of template variables, ϕ is a conjunction of linear atomic predicates defined over the program and template variables and p is a single atomic predicate. The goal is to find an assignment to $\overline{a}$ such that φ holds. As described in [2], by *Farka's Lemma*, the values of $\overline{a}$ that satisfy φ can be obtained by solving a system of non-linear real valued inequalities generated from φ. To solve the inequalities, the plan is to use the Z3 SMT solver integrated into Leon.

Allowing conjunctions in the postcondition is straight-forward as the generated verification condition can be translated to the form given by φ. However, supporting disjunctions in the postcondition is challenging as the verification condition generated by Leon cannot be directly reduced to the form given by

φ. In such cases, the plan is to construct a formulae of the required form that is stronger than the verification condition. A solution for the stronger formulae is also a solution for the original verification condition.

In addition to implementing the above approach, I plan to extend it to handle templates with user-defined functions using the ideas presented in [1]. I also plan to combine this approach with the function unrolling mechanism of Leon to discover invariants that requires reasoning about multiple procedures.

3 Completeness of the approach

There are two sources of incompleteness: (a) Farka's Lemma is incomplete for integer arithmetic. There exists instantiations of templates that are inductive but cannot be found by the proposed approach. (b) Handling of disjunctions in the postconditions also introduces incompleteness. As a part of the project I propose to study and characterize formulas (of the form given by φ) for which Farka's Lemma may be incomplete.

4 Deliverables

An enhanced Leon verifier that can support linear templates in postconditions. Ideally, it should be able to handle all linear transition systems when expressed as Leon programs.

References

[1] Beyer, D., Henzinger, T. A., Majumdar, R., and Rybalchenko, A. Invariant synthesis for combined theories. In *Proceedings of the 8th international conference on Verification, model checking, and abstract interpretation* (Berlin, Heidelberg, 2007), VMCAI'07, Springer-Verlag, pp. 378–394.

[2] Coln, M., Sriram, S., and Henny, S. Linear invariant generation using non-linear constraint solving. In *Computer Aided Verification* (2003).

[3] Sriram, S., Henny, S., and Zohar, M. Constraint-based linear-relations analysis. In *Symposium on Static Analysis* (2004).