Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

Masterarbeit

Finding Policies for Advanced Control Flow Conditionals

Problem Statement

Functional safety is an important goal of modern software development. In addition to specialized development and management techniques (V-Model, Scrum, etc.), there are crucial tools such as automated testing and verification to ensure the necessary standards of quality. With this goal in mind, i11 - Embedded Software is developing the code analysis tool ARCADE (Aachen Rigourous Code Analysis & Debugging Environment) which supplies different techniques for program analysis.

An important subset of analyses is the static analysis, which determines, among other information, value sets of variables at program locations without explicitly running the program. Viewed from a formal angle, these analyses search for a fixed point of the transition function of the program (that is, the function that calculates exactly one step of program execution). This fixed point is usually calculated via a simple fixed point iteration, with optional techniques such as widening to ensure termination at the cost of accuracy.

Because fixed point iteration oftentimes takes a lot of steps, these optional techniques are important in order to yield usable results. For this reason, a technique called *policy iteration*[1] was developed. This technique has the goal of finding these fixed points more quickly than simple fixed point iteration. The basic idea of policy iteration is to represent the transition function f as infimum of a set G of simpler functions (which we call policies), which allows computations of fixed points of f as infimum of fixed points in G. To decide which policy to analyze first, we use a heuristic that, based on the structure of the conditional, attempts to select a policy which will yield a good fixed points after only a few steps of calculation.

Task

ARCADE already contains a rudimentary implementation for policy iteration on intervals. However, the policies are only found for simple control flow conditionals (i.e. $x < 5$), but cannot be found for more complex conditionals (i.e. $x*y < z+x$). Goal of this thesis is to extend the policy iteration in order for it to be able to handle more classes of conditionals on the domain of intervals.

A first approach to this may be linearization, i.e. replacing the conditionals with linear inequalities that overapproximate them. More involved approaches may call for a different way of generating policies.

Qualifications

Knowledge of formal methods, especially in the context of static analysis, is greatly appreciated. Since the policy iteration has been implemented in the C++-Version of ARCADE, applicants should be well-versed in programming C++.

Advisor

Marcus Völker, M. Sc. RWTH
voelker@embedded.rwth-aachen.de

[1] Costan, A.; Gaubert, S.; Goubault, E.; Martel, M. & Putot, S. (2005), A Policy Iteration Algorithm for Computing Fixed Points in Static Analysis of Programs., in Kousha Etessami & Sriram K. Rajamani, ed., 'CAV' , Springer, pp. 462-475 .