Master's Thesis

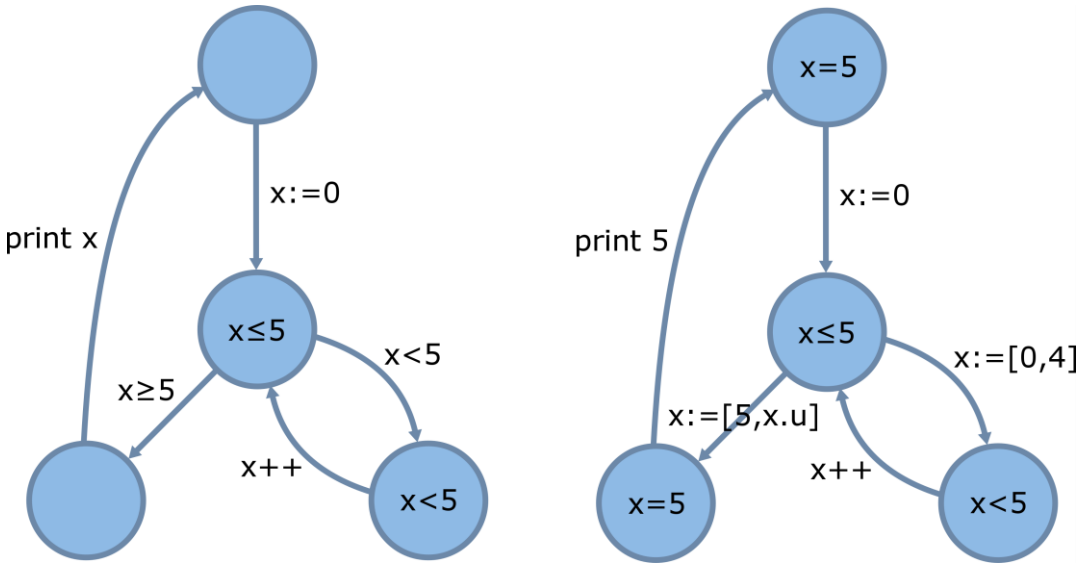
Incremental Policy Iteration for Static Analysis of PLC Programs

Problem Statement

An important program analysis is the static analysis, which determines, among other information, value sets of variables at program locations without explicitly running the program. Because classical static analysis oftentimes takes a lot of steps, a technique called *policy iteration*[2] was developed, which generates a set of simpler programs with similar behaviour to the original program and uses analysis results from these programs to calculate the result for the actual program.

Task

ARCADE, a program analysis tool for industrial control software developed at i11, already contains a rudimentary implementation for policy iteration on intervals. However, the basic heuristic doesn't always find the optimal policy for nested loops, since sometimes, the policy for the outer loop may depend on the results of the inner loop. The goal of this thesis is to develop an incremental approach to policy iteration, i.e., deferring the choice of policies until some partial result of the static analysis has been found.



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 .