ABSTRACT 2011-02-18

A Foundation for Flow-Based Program Matching Using Temporal Logic and Model Checking

Julia Lawall, University of Copenhagen

Reasoning about program control-flow paths is an important functionality of a number of recent program matching languages and associated searching and transformation tools. Temporal logic provides a well-defined means of expressing properties of control-flow paths in programs, and indeed an extension of the temporal logic CTL has been applied to the problem of specifying and verifying the transformations commonly performed by optimizing compilers. Nevertheless, in developing the Coccinelle program transformation tool for performing Linux collateral evolutions in systems code, we have found that existing variants of CTL do not adequately support rules that transform subterms other than the ones matching an entire formula. Being able to transform any of the subterms of a matched term seems essential in the domain targeted by Coccinelle.

In this work, we propose an extension to CTL named CTL-VW (CTL with variables and witnesses) that is a suitable basis for the semantics and implementation of the Coccinelle's program matching language. Our extension to CTL includes existential quantification over program fragments, which allows metavariables in the program matching language to range over different values within different control-flow paths, and a notion of witnesses that record such existential bindings for use in the subsequent program transformation process. We for use in the context of Coccinelle. We then assess the performance of the approach in practice, using a transformation rule that fixes several reference count bugs in Linux code.

This is joint work with Julien Brunel, Damien Doligez, Rene Rydhof Hansen, and Gilles Muller, and was published in POPL 2009