Thesis Proposal

School of Computer Science
Institute for Software Research
Software Engineering

Automatic Program Transformation for Program Repair and Improving Analysis

Rijnard van Tonder

Wednesday, May 9, 10am
Gates & Hillman Centers 6501

Software bugs are not going away. Millions of dollars and thousands of developer-hours are spent finding bugs, debugging the root cause, writing a patch, and reviewing fixes. However, truly fixing bugs remains a predominantly manual and expensive process. Automatic Program Repair (APR) promises to cut the costs of fixing bugs manually. However, APR is hard: the root concern is that automatic program transformation can lead to intractable and undesirable behavior. Current state of the art in APR research predominantly rely on existing tests to identify bugs, validate fixes, or both. However, tests are susceptible to patch overfitting and preclude fixing certain classes of bugs (e.g., leaks) and previously unknown bugs. In contrast, static analysis techniques and related logic-based reasoning use semantic abstractions to efficiently detect wider bug classes without tests.

My first insight is that existing semantic abstractions in high quality analyses can provide precise correctness validation for automatic fixes. Second, these semantic abstractions present new ways of driving unassisted search and application of program fragments to fix real programs, including previously unknown bugs. In particular, I will demonstrate application of separation logic for fixing heap-related defects, and extend logic-based reasoning to additional classes of bugs for fixing real bugs in real programs. Preliminary results from applying semantic-driven program transformation for APR reveal that the analysis itself can be improved. For example, typical analysis behavior conservatively aborts when an error is detected, whereas fixing the error allows analysis to continue. My insight is that program transformation can augment both static and dynamic analysis behavior to improve analysis results. I will develop automated program transformation techniques to demonstrate analysis improvement (e.g., to find more bugs, fix more bugs, and reduce false positives). The thesis is that semantic-driven search and application of program transformations, using logic-based validation enable efficient, scalable, and unassisted automated program repair of real world-programs and can improve the effectiveness of existing program analyses.

Committee: Dr. Claire Le Goues (chair), Dr. Christian Kästner, Dr. Jan Hoffmann, Dr. Manuel Fahndrich (Facebook)