A Semantic Framework for Program Debugging

Wei Li

State Key Laboratory of Software Development Environment
School of Computer Science and Engineering
Beihang University, China
liwei@nlsde.buaa.edu.cn

Abstract.

This work aims to build a semantic framework for automated debugging. A debugging process consists of tracing, locating, and fixing processes consecutively. The first two processes are accomplished by a tracing procedure and a locating procedure, respectively. The tracing procedure reproduces the execution of a failed test case with well-designed data structures and saves necessary information for locating bugs. The locating procedure will use the information obtained from the tracing procedure to locate ill-designed statements and to generate a fix-equation, the solution of which is a function that will be used to fix the bugs. A structural operational semantics is given to define the functions of the tracing and locating procedure. Both procedures are proved to terminate and produces one fix-equation. The main task of fixing process is to solve the fix-equation. It turns out that for a given failed test case, there exist three different types of solutions: 1. the bug is solvable, there exists a solution of the fix-equation, and the program can be repaired. 2. There exists a non-linear error in the program, the fix-equation generated at each round of the locating procedure is solvable, but a new bug will arise when the old bug is being fixed. 3. There exists a logical design error and the fix-equation is not solvable.