

**CHENNAI – PONDICHERRY**

**Automated Behavioral Testing of Refactoring Engines**

**Abstract**

Refactoring is a transformation that preserves the external behavior of a program and improves its internal quality. Usually, compilation errors and behavioral changes are avoided by preconditions determined for each refactoring transformation. However, to formally define these preconditions and transfer them to program checks is a rather complex task. In practice, refactoring engine developers commonly implement refactorings in an ad hoc manner since no guidelines are available for evaluating the correctness of refactoring implementations. As a result, even mainstream refactoring engines contain critical bugs. We present a technique to test Java refactoring engines. It automates test input generation by using a Java program generator that exhaustively generates programs for a given scope of Java declarations. The refactoring under test is applied to each generated program. The technique uses SafeRefactor, a tool for detecting behavioral changes, as an oracle to evaluate the correctness of these transformations. Finally, the technique classifies the failing transformations by the kind of behavioral change or compilation error introduced by them. We have evaluated this technique by testing 29 refactorings in Eclipse JDT, NetBeans, and the JastAdd Refactoring Tools. We analyzed 153,444 transformations, and identified 57 bugs related to compilation errors, and 63 bugs related to behavioral changes.