RefDistiller: A Refactoring Aware Code Review Tool for Inspecting Manual Refactoring Edits

Authors: Everton L. G. Alves, Myoungkyu, SongMiryung Kim

http://dx.doi.org/10.1145/2635868.2661674

Manual refactoring edits are error prone, as refactoring requires developers to coordinate related transformations and understand the complex inter-relationship between affected types, methods, and variables. Recent studies show that developers often mix refactoring with other semantic changes and developers do most refactoring manually. RefDistiller is a refactoring-aware code review tool that can help developers detect potential behavioral changes in manual refactoring edits. It first detects the types and locations of refactoring edits by comparing two program versions. Based on the reconstructed refactoring information, it then detects potential anomalies in refactoring edits using two techniques: a template-based checker for detecting missing edits and a refactoring separator for detecting extra edits that may change a program's behavior.

RefDistiller helps developers by detecting deviations from pure refactoring edits, ensuring confidence in the correctness of manual refactoring. It employs two techniques: RefChecker for missing edits and RefSeparator for extra edits. By analyzing original and manually refactored versions using RefFinder, RefDistiller identifies potential edits' types and locations. RefChecker verifies required code modifications and reference bindings preservation, reporting any missing edits. RefSeparator generates a pure refactoring version and compares it with the manual one using ChangeDistiller's syntactic differencing, reporting any extra edits' locations. The closest work to this tool is GhostFactor which automatically checks correctness of manually performed refactoring by checking required conditions. The idea of GhostFactor is similar to RefChecker's templates; however, it handles three refactoring types. Currently, RefChecker reports nine types of warnings for six refactoring types. In contrast to RefSeparator, GhostFactor cannot detect extra edits from manual refactorings.

The tool was validated by evaluating its effectiveness on a data set with one hundred manual refactoring bugs. These bugs are described by the developers of the tool to be "hard to detect because they do not produce any compilation errors nor are caught by the pre- and post-condition checking of many existing refactoring engines". So, these bugs are logic / runtime bugs, as their related code is syntactically correct, but does not behave as expected due to flaws in the logic or algorithms implemented withing the code. It is estimated that RefDistiller can identify 97% of the erroneous edits, of which 24% are not detected by extensive, automatically generated test suites.