

Title: An Empirical Study on Static Analyzer Toolsets to Reduce False Positives, False Negatives in Python Type Checkers

Benedek Kaibas, Allegheny College, PA, 16335, USA

Dr. Gregory M. Kapfhammer, Allegheny College, PA, 16335, USA

Abstract—The rapid ascent of Python as a dominant programming language has necessitated a reliance on static type checkers to mitigate runtime exceptions in large-scale software systems. While established tools like Mypy and Pyright, alongside emerging high-performance alternatives such as Pyrefly and Zuban, are now integral to continuous integration pipelines, their reliability remains largely unverified against the language's rapidly evolving feature set. This article presents an empirical study addressing the critical trade-off between soundness and completeness in Python static analysis. We introduce Pytifex, a novel automated framework that bridges the gap between theoretical fuzzing and real-world software defects. Unlike traditional methods that rely on random code generation, Pytifex leverages a data-driven approach by mining closed GitHub issue reports specifically labeled as false positives and false negatives. It systematically refines and mutates these "seed" examples to generate high-level, adversarial test cases designed to expose regressions and soundness gaps. By subjecting both mature and nascent type checkers to this targeted differential testing, our research highlights significant discrepancies in error reporting and offers a robust methodology for reducing false diagnostics. This work provides tool maintainers with actionable insights to fortify their inference engines, ultimately enhancing the stability and correctness of the Python type-checking ecosystem.