

Does It Always Do That? Practical Automatic Lightweight Nondeterminism and Flaky Test Detection and Debugging for Python Libraries

Alex Groce

*School of Informatics, Computing & Cyber Systems
Northern Arizona University*

Josie Holmes

*School of Informatics, Computing & Cyber Systems
Northern Arizona University*

Abstract

A critically important, but surprisingly neglected, aspect of system reliability is system predictability. Many software systems are implemented using mechanisms (unsafe languages, concurrency, caching, stochastic algorithms, environmental dependencies) that can introduce behavioral nondeterminism. Users of software systems, especially other software using library calls in a single-threaded context, often expect that systems will behave deterministically in the sense that they have predictable results from the same series of operations. Equally importantly, even when it does not impact correctness or user experience, nondeterminism must be understood and managed for effective (especially automated) testing. Nondeterministic behavior that is not either controlled or accounted for can result in flaky tests as well as causing problems for test reduction, differential testing, and automated regression test generation. We show that lightweight techniques, requiring little effort on the part of developers, can be used to extend an existing testing system to allow detection of nondeterminism. We introduce a set of lightweight nondeterminism properties, inspired by real faults, and tailor these notions to the practical, automatic, checking of Python library code. In particular, we

Email addresses: `agroce@gmail.com` (Alex Groce), `https://agroce.github.io` (Alex Groce), `josie.holmes@nau.edu` (Josie Holmes)