

---

# An approach and benchmark to detect behavioral changes of commits in continuous integration

Benjamin Danglot\*<sup>1</sup>

<sup>1</sup>Davidson – Davidson College – France

## Résumé

When a developer pushes a change to an application's codebase, a good practice is to have a test case specifying this behavioral change. Thanks to continuous integration (CI), the test is run on subsequent commits to check that they do not introduce a regression for that behavior. In this paper, we propose an approach that detects behavioral changes in commits. As input, it takes a program, its test suite, and a commit. Its output is a set of test methods that capture the behavioral difference between the pre-commit and post-commit versions of the program. We call our approach DCI (Detecting behavioral changes in CI). It works by generating variations of the existing test cases through (i) assertion amplification and (ii) a search-based exploration of the input space. We evaluate our approach on a curated set of 60 commits from 6 open source Java projects. To our knowledge, this is the first ever curated dataset of real-world behavioral changes. Our evaluation shows that DCI is able to generate test methods that detect behavioral changes. Our approach is fully automated and can be integrated into current development processes. The main limitations are that it targets unit tests and works on a relatively small fraction of commits. More specifically, DCI works on commits that have a unit test that already executes the modified code. In practice, from our benchmark projects, we found 15.29% of commits to meet the conditions required by DCI.

---

\*Intervenant