

# Improving Usability of Flaky Test Detection Tools

Hao Yuan (Frankie)      Mentor: Wing Lam  
Email: {haoyuan4, winglam2}@Illinois.edu



## Flaky Tests

Tests that pass or fail non-deterministically (different results for the same code)

### Types of Flaky Test

- Order Dependent (OD): Tests that are flaky because of only the order tests run
- Non-Order Dependent (NOD): All other types of flaky tests besides OD

### Issues

- Wasting machine testing time
- Misleading developers (especially in regression tests)

Google had around 63 thousand flaky tests that run over the course of a week  
<https://testing.googleblog.com/2017/04/where-do-our-flaky-tests-come-from.html>

## iDFlakies

A framework for detecting and partially classifying flaky tests

**Requirement:** a fully passing original order of tests to ensure the project was properly setup

1. Shuffles and runs tests in different test orderings
2. Analyzes and classifies flaky tests from the test result

## My Goal

Improve usability of iDFlakies by giving users the option to relax iDFlakies' requirement

## My Work

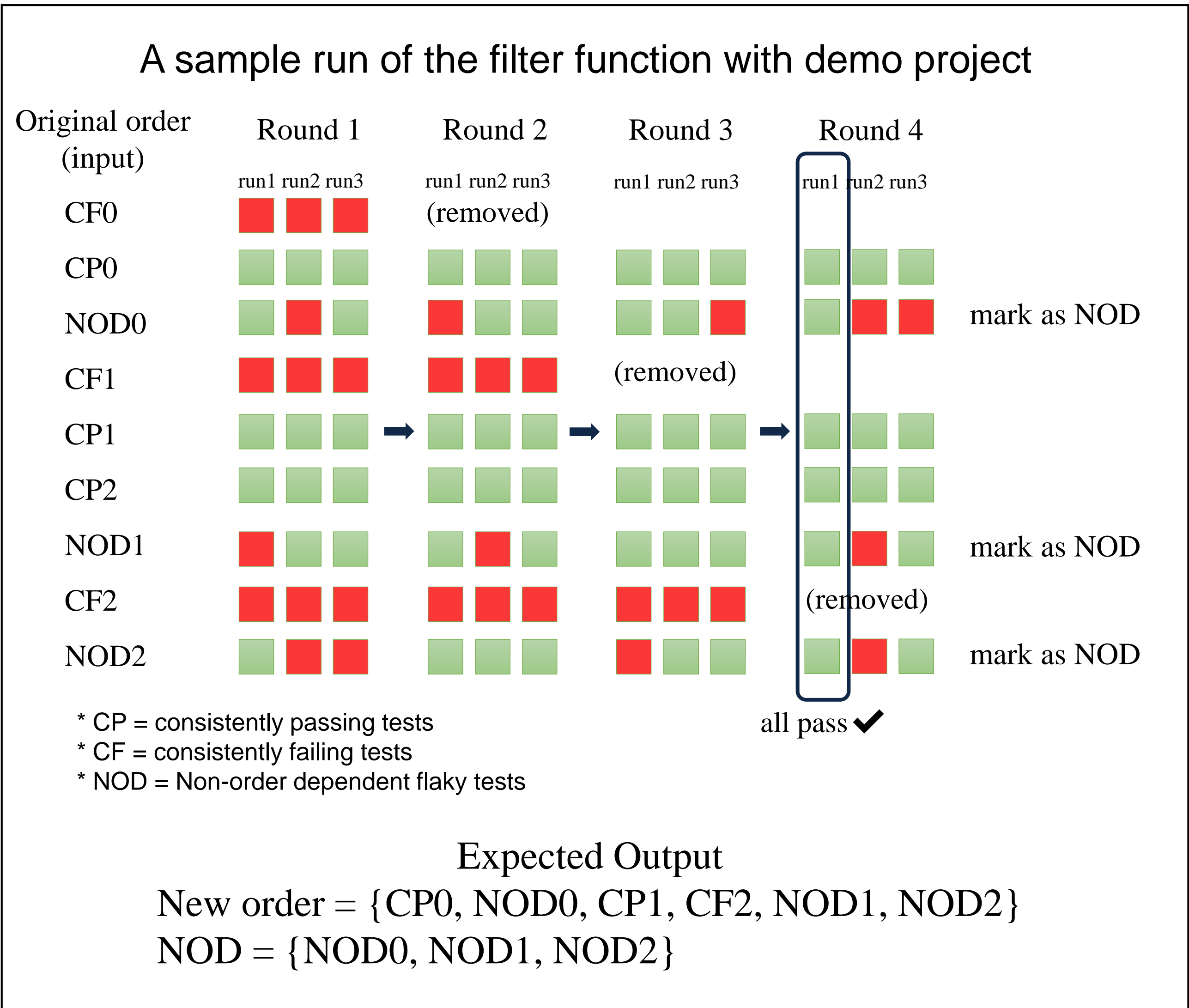
### Removing Failing Tests

A filter that analyzes test results of each round and removes consistently failing tests to obtain fully passing original order.  
The filter is also able to identify NOD tests.

- Input: an array list of original order
- Options: #runs per round, remove failing tests one per round or all at once
- Output: a new passing original order without consistently failing tests & a list of confirmed NOD tests

### Demo Project

- Contains 3 consistently passing tests, 3 consistently failing tests, and 3 NOD tests that fail one-third times
- All consistently failing tests should be removed and NOD tests should be recorded by the new filter.
- GitHub Repository: <https://github.com/frankyhao/mavendemoproject/>



Framework & dataset: <https://sites.google.com/view/flakytestdataset>