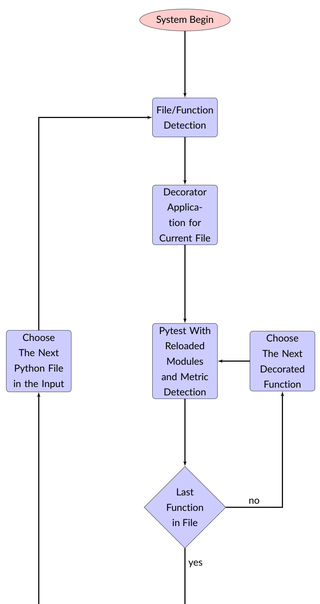
Automatic Detection of Pseudo-tested Methods using Python and Pytest

Nicholas Tocci
Gregory Kapfhammer

Introduction

Software systems are very large and complex. Due to this, modern Python programs are difficult to test due to the lack of type safety. Another concern is the possible misleading nature of statement coverage since it doesn't factor in branches and iteration, there is no information on the data state, and the quality of the oracle. These factors can cause a psuedo-tested methods to exist in Python programs. A psuedo-tested method is method that is tested, but which passes regardless of the output of a function. Function-Fiasco helps determine how much of the code is adequately tested and provides a metric that is more representative of the actual conditions of the system.

Implementation



Function-Fiasco is an automatic tool that detects pseudo-tested methods in Python progams.

Scan the QR Code to visit our GitHub page

Preliminary Results

	System Name	State Cov	Function Cov	NUMM	NUMTM	Fiascoed	Pseudo	NUMTTM	UC	Change
1	Hashids-Python	0.97	0.94	16	15	10	8	7	0.44	0.50
2	Bleach	0.48	0.41	368	152	8	2	150	0.41	0.00
3	Pycco	0.77	0.86	22	19	6	5	14	0.64	0.22
4	Howdoi	0.78	0.95	20	19	2	0	19	0.95	0.00
5	Flashtext	0.81	0.33	42	14	7	4	10	0.24	0.09
6	Honcho	0.85	0.69	58	40	7	5	35	0.60	0.09
7	Maya	0.90	0.50	88	44	13	3	41	0.47	0.03
8	Gator	0.99	0.86	92	79	54	30	49	0.53	0.33
9	Hatch	1.00	0.56	134	75	14	6	69	0.51	0.05
10	Nikola	0.67	0.44	732	319	16	9	310	0.42	0.02

Function-Fiasco can successfully detect pseudo-tested methods in Python based systems.

Future Work

Function-Fiasco has many features that will be implemented which include:

- Further type fuzzing capability
- Parameterized test observation
- Further system evaluation

Conclusion

Pseudo-tested methods are an issue that exist in Python based systems. Function-Fiasco has the capability to detect such methods that may lead to unexpected issues. Function-Fiasco can aid in the implementation of Python systems.

Get Involved

If you would like to get involved, please feel free to enter bugs into the issue tracker on our GithHub page, or submit a pull request to aid in the implementation.

Acknowledgements

Made in cooperation with Cory Wiard.

ALLEGHENY COLLEGE
COMPUTER SCIENCE