**Pynguin: Automated Unit Test Generation for Python**

Team members: Iliescu Andrei, Iancu Aurelian, Ovidiu Gligor

Pynguin presents an approach to automatically generating unit tests for Python code. The paper's motivation stems from the challenges developers face in ensuring code quality and correctness through effective unit testing, particularly in dynamically typed languages like Python or JavaScript where traditional static analysis tools may be less effective. To address this, Pynguin leverages evolutionary algorithms to systematically explore the space of possible unit tests, aiming to generate tests that maximize code coverage.

The aim of Pynguin is to decrease the need for writing unit tests manually by automating the process. The tool uses a combination of dynamic program analysis and search-based techniques to iteratively generate test cases that exercise different execution paths within the target code. Pynguin introduces several innovative strategies for guiding the search process, such as code instrumentation, test prioritization, and assertion generation, which contribute to the effectiveness and efficiency of test generation.

Validation of Pynguin's effectiveness was conducted through empirical studies involving real-world Python projects. The tool was evaluated against manually written test suites, comparing code coverage metrics and fault-detection capabilities. Additionally, Pynguin was subjected to benchmarking against other state-of-the-art test generation tools to assess its performance and scalability.

https://doi.org/10.1145/3510454.3516829