*Pynguin: Automated Unit Test Generation for Python*

Iaguta Alen-Mihael Ion Bogdan Guceanu Marian-George

Pynguin is a test generator written in Python, a general purpose language. Below, we’ll look at a couple of features that make pynguin a well-designed tool, with a specific goal in mind, but also what makes it unique.

*Generation*

Pynguin is an automation tool for a task that invokes automation as a base principle. What makes it so great it’s that it automatically generates the Python test cases, but it does so in PyTest style. This is necessary as it assures uniformity across the tests.

*Modularity and extensibility*

This tool is designed to be modular. This openness allow the users to define some in-case modules, thus making it extensible. Key features regarding this topic include the ability to implement algorithms and also type-inference techniques, so that they get the most coverage possible.

*Installation and features*

Pynguin is easily installable through the python package manager, pip. This makes it at-hand for any user that may want to test their code. It also supports various test input generation algorithms and coverage criteria, helping improve test coverage levels significantly. This is clearly stated below.

*Usage and results*

Pynguin has been successful in achieving up to 68.0% branch coverage on 118 Python modules. That number may not necessarily sound impressive, but this is what the tool was able to achieve with an almost bare-bones setup. It is a command-line tool with an API for controlling the test generation process efficiently.