17-665 Program Analysis
Project
Evelyn Kuo

Project Checkpoint

So far I have set up a Github repository for my code and have tried out the various tools (pynguin, deal, tstl, klara, crosshair, auger) mentioned in the project proposal on a simple function called triangle which categorizes the triangle given its three side lengths. The general structure is here and each tool has its associated files for testing grouped together.

```
project/
|   README.md
└──triangle/
|   |   triangle.py: base version of file that you want to generate test cases for
|   |   triangle_*.py: modified versions to work with different tools
|   └──tests/
|       └── (tool)/: subfolders corresponding to each tool
|       |       test_triangle.py: generated or created test files from the tool it belongs to
```

So far, penguin, deal, klara seem the most straightforward and close to what I expected for automatic test generation, so I plan on continuing to use them. Auger requires a bit more time to see if it fits well, though it did manage to produce something that looks like it might work for what I have in mind. Lastly, TSTL and crosshair don't seem to be in line with what tools I was looking for, so I don't plan on continuing to use them for the project. Namely, TSTL seems to require too much setup and Crosshair is more of a real-time checker which makes it harder for it to be compared to the other tools. I still have yet to decide on what shared group of code I plan to test the tools on as I do not know of any standardized benchmarks for automated testing and don't want to run into the issue of using benchmarks which may have been pre-selected to perform better on a certain tool. I think this is the only major blocker as of now. For metrics for comparing the different tools, I plan to have one metric measure the number of characters needed to be changed for a file to be prepped for a certain tool. This metric is to be a proxy for ease of use of the tool. Another metric I plan to use is the number of detected bugs to measure how good the tool is at catching bugs, though I need to look more into the documentation to see if I can standardize the number of tests generated. Lastly, I also plan to qualitatively take note of how good the tool is at pointing out a failing test case which is related to ease of use.

Link to Github Repository: https://github.com/enkokoro/program_analysis_project