Oregon state university
Cs 562
Zhen Tang

Part 3: Bug/progress report (20%)

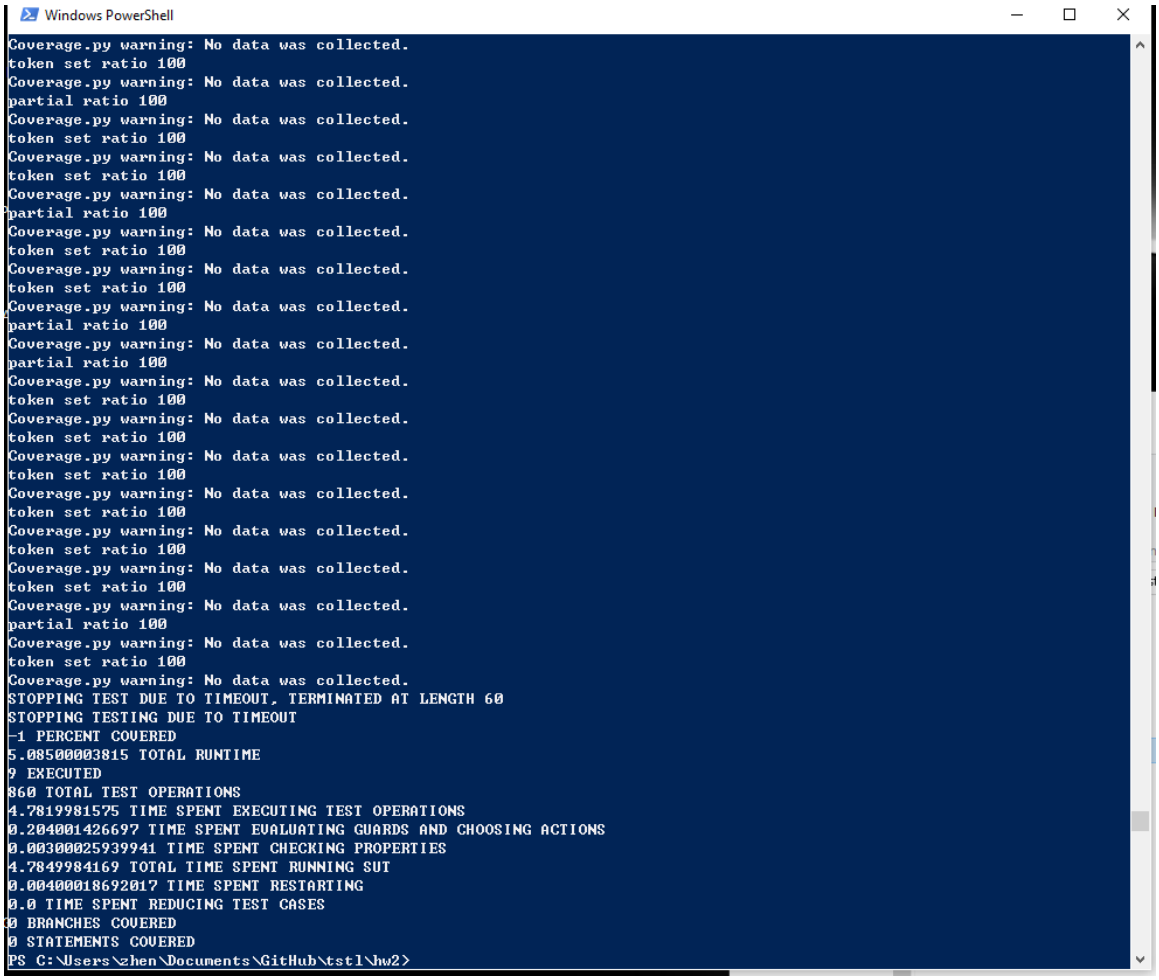
As I described in the midterm report. The system I am testing is a string matching library which provides functions to do string matching with different requirements. After testing this library in many different cases, I would like to say that most functions in this library I have tested works as expected. I haven't found any bugs yet.

Since the library is used for string matching and it is actually not easy to check the correctness of the ratio. My test are mainly concerned about the stability of the functions. In the first several weeks, I tried some ways to build some strings based on the same substring and then apply the partial ratio function to compare those strings with the substring. However, the way I construct those testing strings seems too straightforward at the beginning that they just keep coping and pasting themselves to building longer strings. That guaranteed the output of the function would always be 100 which is easy for me to have assertion after each steps. I think it would be interesting if I could find some different ways to construct strings to cover more cases. After finishing the testing about partial ratio function, I'm now working on construct appropriate test strings for testing token sort ratio function and token set ratio function. The basic construct methods for those test strings are also simple. Similarly to the testing for ratio function, we would constructs those testing strings from a basic word set in order to guarantee the output, like what I did for ratio function. Until now, this system passed all of tests I created. I also have some effort on testing "process" function but there are not enough decent results.

Based on the work I have already done and the number of functions in the library, I think I would be able to test all the functions which are explained in its document. In the last days of the term, I would try to do more testing work on three aspects. First of all, optimizing string construct methods to cover more cases for testing ratio functions. On the other hand, it is important to get to know the length limit of strings that can be accepted by those functions. Although for most cases users would only use the functions for some works with simple input string, like keyword matching, knowing the limitation of the function, which were not explained in its document, is still needed. Moreover, checking the correctness of the ratio output is also an interesting issue, which is worth trying if have enough time.

As for code coverage, I haven't found a proper way to test it. As it shown in the picture below. It keep printing "Coverage.py warning: No data was collected." And shows that -1 percent covered, which is obviously impossible.

Oregon state university
Cs 562
Zhen Tang



As a result, it is hard to say whether the SUT is a good one or not since my test only cover limited cases of the functions in the library. If based on the result I already get, I would like to say the library is not a bad designed one. At least it successfully achieves its basic designed features.

My test already covered all the functions in the library. But I still need to have more efforts to cover more cases.