

Milestone 2 Reporter

Xiang Li

932-514-926

Algorithm:

The newCover.py has two separate phases. The phase one uses random test to test the states and their active actions. Meanwhile, the newCover.py can computer the coverage of each branches. So using these accounted data, phase two can randomly test more braches and actions, which can improve coverage rate.

According to the code newCover.py, my algorithm inherits its structure. I also use the two phases structure to build the code. In newCover.py, professor Alex computer the mean number of all actions' cover time. And use this number to find the actions' cover time that below it. In my code, I also divided a test into two phases. Each phase take half of the running time. To compute a benchmark, I was trying to sort the list of action's cover times. And then find the median number of this list. If the actions' whose cover time is below the median number, then grip the state and restore the state in to a new list. In the phase two, randomly run the states that in the new list again to enhance the coverage ratio.

Revised Code:

For the milestone 2, the most things I did is changing the command line format and making input arguments more flexible. For the milestone 1, my code had to input all the arguments one time, and if the tester misses one or two arguments, the code would crash. But in this time, I imported the package argparse, this can help your code receive the argument more flexible and dynamic.

More over, at the first time I receive timeout when professor ran my code. I also revised the time budget and let it would not crashed again.

Future work:

I will try my best to apply the algorithm, which is mentioned in my project proposal: FAR. It's like to compute the new generating sequences by some vectors. I will read the paper carefully to figure out the method to generate these vectors distances. Than according to these distances, I will generate the new sequence. I think this algorithm can improve the branch coverage from random testing. If this method fail, I will use other method to improve random tester, such as dividing Time into more pieces. Also, I will follow the instruction of final project to revise the project.

And I also have a mind that, my coverage at this time is not very high, I want to add BFS into my code to improve coverage. This breadth fast search algorithm might spend more

time processing it because the time complexity is $O(b^d)$. b is branching factor of graph, and d is the distance between start node and next node. Although, the BFS is not efficient, it is still a good way to get more coverage. In my code I have two phases, the first part is random test. I can add BFS into the second part or create a third part. So in the last rest of the class, my focus will be the coverage problem.