CS569 (Spring 2016) --- Project

Competitive Milestone 2

Wen-Yin Wang

05/18/2016

Previous implement

In previous project assignment, competitive milestone 1, I have been implemented my project in original random testing including get data setting in command line. I assert that the inputs are working correctly, and no need to revising in that part. However, my algorithm only executes one time due to the loop, and it affects the lower coverage. And I am wondering if there have any other ways can keep repeating the test until TIMEOUT and other methods having higher coverage. So, I started my revising implement.

Revising implement

Before I started my revising implement, I added two functions, collectCoverage and printCoverage, in previous extension submit time. The function collectCoverage is used to collect and count for branches and statements after each test; and the function printCoverage is used to print total coverage report. First of all, I want my program can keep doing testing until TIMEOUT. So, I added a while loop if the elapsed time is smaller than TIMEOUT. Within the while loop, I placed my previous double loops that contain my entire testing algorithm. In this step, unfortunately, I cannot found out any differences in coverage. So, I was trying to find the reason. In the second steps, I removed a restart function in SUT module within the first loop. Since the given random seed is same in the entire program, the results from random function are same. So, I removed the restart function within the depth loop. And the restart function only be called when the random act are not passed the safely checking. Until know, I already found out my coverage is higher than previous implement. Since I used randomEnabled function in SUT module to do my random testing. So, I did not change anything about random function and random seed. But I am thinking about using randomEnableds to replace randomEnable. I did not implement it at this time, but I will try in next time,

Future works

As I mentioned before, I will add randomEnableds in my randomAct function. The number n I used for randomEnableds will be a random number. I hope it works well and I can get a higher coverage. Besides, I will revise the displaying for clear and pretty output. Also, I will tend to implement a failure handler to deal with exceptions. I will aim to develop my program in higher coverage and lower failure rate.

Pseudocode

```
collectCoverage{
        for s <- all current branches:
                if s is a new branch:
                        branchCount[s] = 1
                else:
                        branchCount[s] += 1

         for s <- all current statements:
                if s is a new statement:
                         statementCount[s] = 1
                else:
                        statementCount[s] += 1
}

printCoverage{
        sort branchCount[…]
        sort statementCount[..]
        print all branches and statements
}

randomAct{
        act = sut.randomEnabled(config.seed)
        ok = sut.safely(act)
        if not ok:
                if config.faults:
                        write file
                sut.failure()
                sut.restart()
        return ok
}

main{
        make config from command line

        if config.fault :
                open file

        sut = sut.sut()
        sut.restart()
        start = time.time()
        elapsed = time.time() – start
        while elapsed < config.timeout do:
                sut.restart()
                for i = 1 to config.depth:
                        for j = 1 to config.width:
```

```
                                randomAct()
                                elapsed = time.time() - start
                                if config.running = 1:
                                        if got a new branch:
                                                print new branch
                                        if got a new statement:
                                                print new statement
                                if elapsed > config.timeout:
                                        break
                        collectCoverage()
                printCoverage()

                if config.fault:
                        close file

                if config.coverage:
                        sut.internalReport

                print results
}
```