

Max Moulds

CS 362 - Random Testing - Quiz 2 - testme

The function testme asks for a char and a group of characters and then compares them to a set of predefined characters and the the word “reset” and if the right sequence of chars is given to the function from the random char function inputChar then the state level is elevated, eventually to 9 when the program waits for the reset word to occur in another function inputString which returns a randomly generated string.

If one was to truly test every random possibility there would be  $96!/(96-9)!$  permutations for the inputChar function alone and another  $26!/(26-6)!$  permutations for the inputString function. These numbers are derived after initially limiting possibilities in inputChar to just 96 ASCII chars and in inputString to 26 alphabet characters in lowercase, so it is by no measure fully exhaustive of all possible inputs from the two functions.

For inputChar, further limiting of the test cases was required. A set containing 9 characters was used. A single character was chosen at random from that array of 9 and this significantly limited the number of permutations required to complete the testme function. The 9 characters were directly taken from the testme function and this means that no matter what random character is returned from inputChar it will satisfy one level of the if statements used to change the state variable.

For inputString, further limitation of possible permutations was not needed after reducing the inputChar permutations. The number of required iterations was reduced to about half the max size of a signed integer. This allowed runtimes to range from 1 min to 5 mins and some runs were able to print the error message in under 5 million iterations while others were around 25 million. This was significantly lower than I expected.

Here is an excerpt of the output from a test.

```
....  
Iteration 27444958: c = (, s = dmbec, state = 9  
Iteration 27444959: c = }, s = bbdgb, state = 9  
Iteration 27444960: c = }, s = wexsb, state = 9  
Iteration 27444961: c = {, s = reset, state = 9  
error
```

Additionally, the development process tested my understanding of combinatorial math, which is not as strong as it should be and I must admit some of my figures calculated using combinatorics may be inaccurate. Initially, I tried permuting all possibilities but decided that approach was untenable and abandoned the approach in favor of a reduced set of permutations that when run would complete in a reasonable amount of time on my system. I also used the tC variable to limit the iterations on long running tests during development.