TSTL: Progress Report

Neil Parmar

There were no particular bugs encountered for the System Under Test, however, TSTL correctly verified certain assert conditions which would explicitly fail. TSTL generated a failure report when the stack was empty and the count condition was not equal to the initialization value. It generated a result stating that, this counter failed on an empty stack where it tried to match with the count not being zero. However, this assert condition, when it fails it derives an output such as the "failcount += 1".

The original implementation of TSTL for stacks had certain but limited complications with very minimal initialization and usage for automated testing. However, the current version of "stacktstl.tstl" file has concrete operations which determine specific scenarios, for example when certain assert conditions fail. The code though would not be considered thoroughly in accordance with automated testing. However, intention is to implement the particular scenarios with more automated testing techniques which correctly identify the false negatives. Moreover, the TSTL codes implementation has been considerably focused towards prioritizing the assert conditions over other related stack conditions which is still under progress.

Currently I am performing certain minimal operations upon TSTL while trying to prioritize the asserts conditions over the normal stack operations. I am particularly working upon the scenario where the assert conditions fail based upon the stack priority. One scenario of looking at it is to understand the set of assert operations, which particularly try and jeopardize the normal stack operations which have not been carried-out. Another scenario would be to identify the assert operations while the stack is empty, and also a scenario would be to identify the assert operations while the relevant stack operations have been performed and lastly the condition where the stack operations are stalled due to the assert condition being generated based upon a particular scenario. This is a general and vague picture of my proposal and there would be a certain amount of limitations for all these particular condition.

The SUT, is just a code which falls under the category of "well done". However, the particular code generates certain peculiar behavior while computing the stack size. This is still under progress, which is generally quite strange, as a returning a size is not under a very complex operation and would be certain result of certain syntactical mistake. Moreover, one specific problem which I have encountered while working with tstl is that, it considerably does not save or update itself periodically. I am particularly not sure about its internal working but this could be best explained with an example. So, while I try and test the assert condition for the empty stack, I worked on scenarios which could make the test fail as well as succeed. However, even

after saving the successful version of it, it still generates the error of failcount. This result may be due to system error or due to something weird happening with tstl.

The code coverage for the stack is about 15.15% with 11 branch conditions and 15 statements. However, I have not performed many operations on the code coverage and particularly not sure about it’s working.