Test Report for CS 490 SpecExplorer Project

Author: Jun Yan Ma (James) (ma23@purdue.edu)

In the TFTP Visual Studio Solution, named TFTPproject, you can find the project TFTPproject.TestSuite that contains the generated test code.

I used the state checker pattern that provides actions that can be called in all model states for which the implementation is to be validated. These actions only sends information from generated test cases to the harness. The pattern works by performing the actual check in the test harness. If the test harness reaches a state that is wrong, the state checker method will be able to detect it. As such, this pattern is perfect for performing black-box testing of my buggy TFTP client.

The classes that contain the test code are located in **SendTestSuiteWithCheckers.cs** and **WriteSendTestSuiteWithCheckers.cs**, corresponding to testing the send sliced scenarios and write sliced scenarios respectively.

Each class contains in itself three tests, each test case corresponds to the different length input for **canGetExit()** and **canPutExit()** methods.

Upon running of the tests, every single test case (six of them) fails, with the error messages describing what state was expected, and what state the buggy TFTP client reached instead.

The error messages are:

SendTestSuiteWithCheckers* produced
Assert.AreEqual failed. Expected:<5>. Actual:<8>. TestAdapter mismatch: Model vs Implementation

WriteTestSuiteWithCheckers* produced
Assert.AreEqual failed. Expected:<7>. Actual:<4>. TestAdapter mismatch: Model vs Implementation

Digging deeper, by *right clicking* and *view test results details*, we can refer to the Standard Console Output of every test result.

We can see that for **SendTestSuiteWithCheckers***, every erroneous state of **8 (DATA_SENT)** is produced after the call to **sendACKAdapter()**. The expected/correct state is **5 (ACK_SENT)**.

We can see that for **WriteTestSuiteWithCheckers***, every erroneous state of **4 (DATA_RECEIVED)** is produced after the call to **receiveACKAdapter()**. The expected/correct state is **7 (ACK_RECEIVED)**.

Therefore, from these error messages, we can deduce that the implementation of the **sendACK()** and **receiveACK(out byte[])** are buggy and do not change the state of the client correctly.

The test case IDs that exposes the bugs are every **SendTestSuiteWithCheckers*** and **WriteTestSuiteWithCheckers***.