Functional Analysis – Decision Table and State Diagram

June/2025 Shmuel Gershon / Michael Stahl

Overview

In this exercise we continue our validation work on the T-Tweak web-based service.

In case you forgot, re-read the opening paragraphs of Exercise 2.

This exercise:

This is a follow-up of part1 of the Decision table/State diagram exercise.

You can see the definition of all the APIs through a Swagger UI ("Swagger allows you to describe the structure of your APIs so that machines can read them" https://swagger.io/docs/specification/2-0/what-is-swagger/). You can also try using the APIs from the Swagger interface.

The following URL presents the specification of our service's APIs: http://t-tweak.gershon.info/docs

Your mission:

File **EX4b_state_machine.pdf** contains our solution to the "storage" state diagram. Write the test cases needed to achieve 0-switch coverage of this state machine.

a. A test is a set of calls to the API with different commands and index or string values.
b. Use the provided **EX4b_state_diagram_tests.csv** file (and guide) to define your tests.

Guide

- The provided **EX4b_state_diagram_tests.csv** file has 5 columns. The first 3 are following the three fields that the *storage* API accepts. Note that the **command** column is always used, while the **index** and **string** columns are used only for some commands and should be empty when not needed. Play with the t-tweak interface to learn when and how to use these fields.

  The 4th column is for you, to write anything you want. This can help you track what you did and is optional (you can leave it empty or write in it).

The 5th column is for numbering the commands. This will help us to communicate about your solution if we need to. It is not used for anything else but is mandatory.

- The provided **EX4b_state_diagram_tests.csv** file contains examples for testing the sample state machine below. This should give you an idea of what your tests should look like. Read the text in the comments column to understand better.

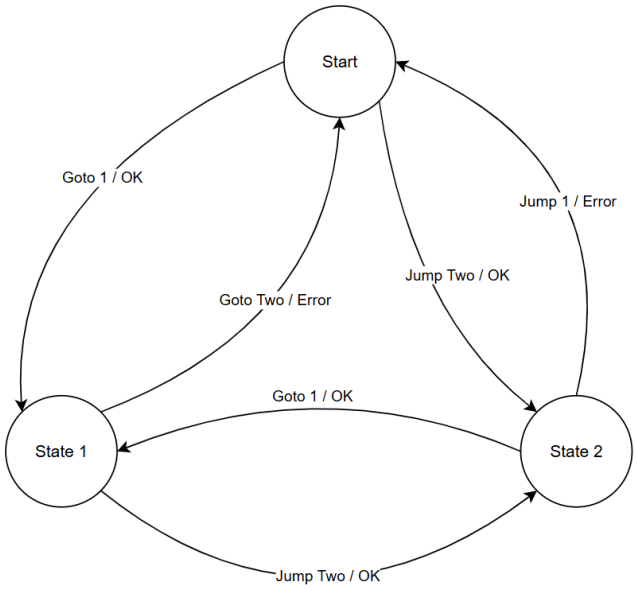
  Delete the sample tests (but not the headers!) and add your solution.

- Pay attention to the quality of your csv file! Correct column header names; no redundant space before and after commands (spaces before and after String are considered part of your string); integer numbers in the Index column… etc. etc.

  Work on the file in Excel and save-as **CSV (MS-DOS)** csv. This will save it with the right encoding.

  Don't lose points on technical issues with your submitted file!

  The sample state machine:



- All tests start at the StandBy state and must end in the StandBy test.

  This is a generally good rule for testing: at the end of any test leave the system-under-test in a known state, so that tests can run in any order. In real life there may be exceptions to this rule, but not in this exercise.

- You need to cover all the transitions that are in the state machine diagram, and you **should not** cover transitions that are not in the diagram. For example: the t-tweak storage state machine ignores undefined commands (it just stays in the same state). It ignores some defined commands, depending on its state. These transitions are not described in the diagram we provided, and therefore you should NOT test them.

How to submit:

Submit the **EX4b_state_diagram_tests.csv** file with your tests. NOT in a zip file please.

Grading
This weight of this part of the exercise in the final grade is 3 points. It is not a mandatory exercise.

You will lose points for inefficient testing (traversing more than the minimal required time each transition or testing transitions that are not in the diagram).
You will lose points for missing a transition that is shown in the diagram (not have a test that traverses this transition).

The deadline for submission is 2/Jul 23:55 .

Late submission penalty:
- One day: -1 point (out of the 3)
- Two days: -2 points (out of the 3)

You can't submit later than 2 days after the deadline.

Appeals will be accepted up to a week after you get the grades.