

Model-based testing

October 18th, 2023

Ana Paiva, José Campos

In this recitation class, we are going to explore ‘Model-based Software Testing’, *a black-box testing technique*.

1. State Machine

Suppose there is an air conditioning system named **ColdHot** which has the following requirements:

- When the user turns it on, the machine is in an idle state. This is the initial state.
- If the user turns it off, the machine is off. If the user turns it on again, the machine is back to idle.
- If it is too hot, then the cooling process starts. It goes back to idle when the defined temperature is reached.
- If it is too cold, then the heating process starts. It goes back to idle when the defined temperature is reached.

Draw a minimal state machine to represent these requirements. (Tip: you may use draw.io to draw the state machine.) How many states, transitions, and events are there? Write your answer in a txt file.

2. Transition Tree

Derive the transition tree of the **ColdHot**’s state machine. How many different paths are there? Write your answer in a txt file.

3. Transition Table

Derive the transition table of the **ColdHot**’s state machine using the transition tree. How many “normal” paths and sneaky paths can be tested based on this transition table? Write your answer in a txt file.

4. (optional) QF-Test exercise

Background

[QF-Test tool](#) is a cross-platform software tool for the GUI test automation. QF-Test supports GUI capture & replay which allows testers to run an application and record the interaction between a user and the application. The sequence of steps/actions performed by the user are recorded, e.g., clicks, and the tool can then automatically replay the exact same interactive steps any number of times without requiring a human intervention. This supports fully automatic regression testing of graphical user interfaces.

Note: on Windows, in case you do not have administrator privileges you could follow [these instructions](#) to install the tool.

Exercise

- Choose one use case of the many available on the [FEUP's webpage](#), e.g., login, search, etc. It must be composed of at least 3 different states.
- Perform 'Model-based Software Testing', i.e., draw the state machine, derive the transition tree, and transition table.
- Implement the derived set of test cases using the [QF-Test tool](#). Note that all states and transitions must be exercised at least one.
- Implement one "sneak path" (i.e., a path in the state machine that should not exist) as a test case using the [QF-Test tool](#).

5. What should you submit/deliver?

Zip the

- State machine derived in Section 1 and the txt file.
- Transition tree derived in Section 2 and the txt file.
- Transition table derived in Section 3 and the txt file.
- (optional) State machine, transition tree, and transition table derived in Section 4. [QF-Test](#) project.

and submit it [here](#) (M.EIC's moodle) or [here](#) (MESW's moodle).

Deadline: End of the recitation class.

Grades: available on October 30, 2023.

Miscellaneous

- [Videos about the testing tool QF-Test](#)
- [QF-Test youtube channel](#)
- [Functional Testing with QF-Test](#)
- [Capture and replay with QF-Test](#)