DIT636 / DAT560 - Assignment 4: Fault-Based Testing and Finite-State Verification

Date Submitted:
Group Number:
Group Members:

Mutant Descriptions (Problem 1)

For each mutant:

- Indicate where the mutant is in the code (e.g., class::method::line number)
- Original code
- Mutated code
- Mutation operator applied
- Mutant category (invalid, equivalent, valid-but-not-useful, useful)
- Explain how the mutant is different from the original code and why it belongs to the indicated category.
- For each test that detects the mutant:
    - Test name and description of why that test detects the mutant.
    - Indicate whether the test was one submitted for Assignment 2, or if it was newly created for Assignment 3.
    - Alternatively, explain why the mutant is either invalid or equivalent.

Instructions for Running Tests (Problem 1)

This section is optional. You only need to fill it out if you used any external libraries other than JUnit itself, or did anything non-obvious when creating your unit tests.

Model Specification (Problem 2)

Overview of the Model

Give a brief description of the system and environment you have modeled. You may find it useful to include diagrams/visualizations of the transitions between states.

Assumptions

- State any assumptions you have made.

Requirements

1. List all critical requirements that you expect the real system (and model) to obey.

Safety Properties (Problem 2)

For each property:

- Property written as temporal logic (LTL or CTL)
- Property written in natural language
- Explanation of why this is a safety property
- List requirements that this property corresponds to

Liveness Properties (Problem 2)

For each property:

- Property written as temporal logic (LTL or CTL)
- Property written in natural language
- Explanation of why this is a liveness property
- List requirements that this property corresponds to