Software Test Plan

# Test Plan Identifier

Turing Machine Assignment 5 Test Plan – State Diagram Graph and State Tree for TM ver. 1.1

Specify the unique identifier assigned to this test plan.

# Test Items

- Version 1.1 of the Turing Machine application code, updated to correct defects from assignment 4

- State Diagram graph which details the states of the Turing Machine, mostly focusing on user input

- State Tree Diagram as derived from the State Diagram Graph

- Round Trip test suite, derived from testing all paths of the State Diagram graph/State Tree Diagram

Identify the test items including their version/revision level. Also specify characteristics of their transmittal media that impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (e.g., programs must be transferred from tape to disk)

# Features to be Tested

State diagrams are developed for the application in this test, focused on user input and the transitions and states related to it.

Identify all software features and combinations of software features to be tested. Identify the test design specification associated with each feature and each combination of features.

# Features Not to be Tested

Features not to be tested in this test plan a detailed description of the file parsing and loading functionality. For this test, it has been abstracted.

Identify all features and significant combinations of features that are in the requirements document that will not be tested and the reasons.

# Approach

For this test plan, we will take the approach of creating state diagrams to visualize logic transitions in areas we believe are most prone to defects and correct them if any exist. As a group, we decided that this area was going to be the areas centered around user input.

For this test plan, no tests will be carried out in code, but application is walked through according to the transitions of the state diagram to ensure their validity.

Describe the overall approach to testing: extent of verification of requirements, use of test methods, use of test input files, simulated versus actual use scenarios, differences between test environment and actual use environment,

# Item Pass/Fail Criteria (optional based on risk level)

Specify the criteria to be used to determine whether each feature has passed or failed testing.

# Test Deliverables (optional based on risk level)

Minimum deliverables:

* This document
* Actual test results

# Procedure Steps (optional based on risk level)

The steps required to perform the testing. Typically documents in a checklist. See the attached table below as one example.

# Test Log

Provide an entry for each test run including the version of the software tested, date, tester, associated files used or generated.

# Test Summary

Summary of results with references to any unresolved problem reports.