Testing Document

Washington State University – Tri Cities

CPT_S 322: Software Engineering Principles I

Project:

Turing Machine Console Application

Instructor:

Dr. Niel B. Corrigan

Created by:

Raul Martinez

Date:

April 21st, 2025

Table of Contents

Introduction

The purpose of this document is to highlight 3 different tests including their expected results as well as their actual results. This document aims to put the functionality of the Turing Machine Console Application to the test and report on its effectiveness, correctness, and functionality. This Turing Machine Console Application was designed by Raul Martinez at Washington State University – Tri Cities in Richland, WA. for the CPT_S 322 Software Engineering Principles I course with Dr. Niel B. Corrigan Visiting-Faculty. Dr. Niel B. Corrigan is acting as the sponsor of this application and specified all of the requirements this Turing Machine Console Application *must* adhere to.

This testing document is broken up into the following parts:

Title Page

Acts as cover for this document when in physical form and includes information about the document.

Table of Contents

Included for traceability of sections and clarity of this document.

Introduction

Highlights important information about the nature of this document and its purpose.

Test #1

Includes detailed information about the expectations for this test and its result.

Test #2

Includes detailed information about the expectations for this test and its result.

Test #3

Includes detailed information about the expectations for this test and its result.

Conclusion

Summary of the overall functionality of the Turing machine console application.

Test #1

Test Identification Test 1: Invalid Input Alphabet
Requirements The Turing Machine must reject strings not in the input alphabet.
Pass: X **Fail:**
Test Case: • Input string: “aab” • Definition files: anbn • Action: Insert the string “aab” into the Turing Machine and attempt to run it.
Expected Result: The Turing Machine should reject the input string “aab” because it is not part of the input alphabet anbn.
Actual Results: Input rejected as expected -> Pass

Test #2

Test Identification Test 2: Valid Input Alphabet
Requirements The Turing Machine must accept strings that are in the input alphabet.
Pass: X **Fail:**
Test Case: • Input string: “aabb” • Definition files: anbn • Action: Insert the string “aabb” into the Turing Machine and attempt to run it.
Expected Result: The Turing Machine should accept the input string “aabb” because it is part of the input alphabet anbn.
Actual Results: Input accepted as expected. -> Pass

Test #3

Test Identification Test 3: Testing the set command.
Requirements Ensure the Turing Machine can do multiple transitions at once.
Pass: X **Fail:**
Test Case: • Action: Open the 'e' Command • Action; Set the transitions per run to 100 • Input string: "aabb" • Definition file: anbn • Action: Insert the string "aabb" into the Turing Machine and attempt to run it with transitions per run set at 100
Expected Result: The run should execute successfully in 1 run and display the amount of transitions it took to accept this input string
Actual Results: The run executed successfully in 13 transitions -> Pass

Conclusion

Based on the variety of the test performed and the given results. The Turing Machine is able to distinguish and determine if input strings are Valid/ Invalid based on the definition and string files. As well as change the number of transitions per run successfully. These properties are some of the most foundational in the basic operation of this Turing Machine Console Application and ensuring they work is critical, which they do based on the tests performed and their actual results.