

Turing Machine Requirements and Specifications

CptS 322 Daniel Hanlen

May 5, 2014

Table of Contents

List of Figures.....	3
Revision History.....	3
1.0 Introduction.....	4
2.0 Background.....	4
3.0 Overview.....	4
4.0 Environment.....	5
4.1 Input and Output Devices.....	5
4.2 Turing Machine Definition File.....	5
4.3 Input String File.....	6
5.0 Operation.....	6
5.1 Invocation.....	6
5.1.1 Command Line.....	6
5.1.2 Configuration Settings.....	7
5.1.3 Opening Turing Machine.....	7
5.2 Commands.....	7
5.2.1 Help User.....	8
5.2.2 Show Status.....	8
5.2.3 View Turing Machine.....	8
5.2.4 List Input Strings.....	9
5.2.5 Insert Input String.....	9
5.2.6 Delete Input String.....	9
5.2.7 Set Transitions.....	10
5.2.8 Truncate Instantaneous Descriptions.....	10
5.2.9 Run Turing Machine.....	10
5.2.10 Quit Turing Machine.....	11
5.2.11 Exit Application.....	11
5.3 Termination.....	12
5.3.1 Closing Turing Machine.....	12
Reference.....	12

List of Figures

Illustration 1: Context Diagram.....5

Revision History

March 9, 2014 – Initial requirements compiled

1.0 Introduction

This requirements specification document is to outline and define the requirements criteria for the Turing machine which will be designed and implemented in CptS 322 at WSU-TC. The audience for this document is the instructor and myself, to serve as a guideline for what is expected in the completed product.

2.0 Background

This project will be centered around the creation of a Turing machine. A Turing machine is a mathematical model that is represented by a series of states, including a start state and any number of end states. Those states are then used in the processing of a tape, containing a series of characters. Based on what the current state is, the Turing machine can have different effects and movement on the tape. If there is no corresponding action, the machine rejects the tape. If the machine reaches a final state, the tape is said to be accepted. If the machine does not reach either of these conditions, the result is indeterminate.

A formal definition of a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$. Where Q is a finite set of states, Σ is a subset of Γ without B as a finite tape alphabet, δ is a transition function from Q and Γ to another Q , Γ , and a left or right movement, q_0 is a single initial state that the machine starts in, B is the blank symbol contained in Γ , and F is the set of final states.

3.0 Overview

The Turing machine developed will be used in accepting or rejecting strings represented by a series of characters given as input to the machine from a file. The program will be developed using C++ and will compile to be the Turing machine. It will be executed on the command line with one additional argument, a string containing the name of the Turing machine definition without any extension. This argument will be used both in finding the definition as well as the input file. I will code and test the program on the school's computer network, which runs Ubuntu. When run, the program will prompt for a command, and continue to do so, so long as the exit command isn't used. This simplifies the command structure of the program so that one loop is doing the decision making.

4.0 Environment

The following context diagram describes the environment the Turing machine will operate in:

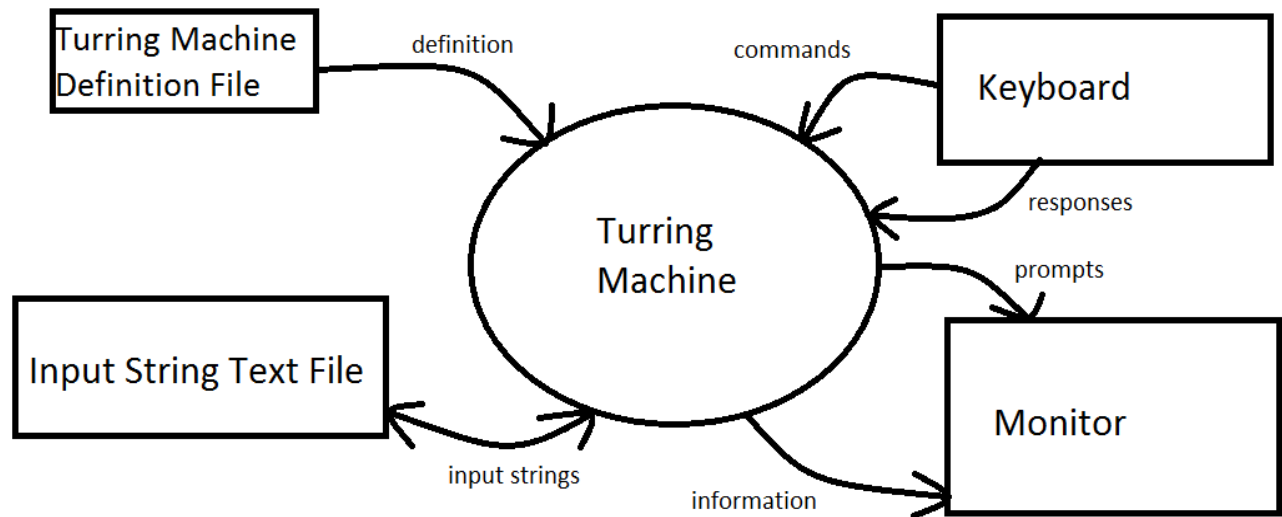


Illustration 1: Context Diagram

4.1 Input and Output Devices

The input devices are as follows:

- Two definition files that the program reads in
- Keyboard is used for user input to accept commands to run

The output devices are:

- Monitor
- File containing list of strings

4.2 Turing Machine Definition File

An example of a correct Turing machine is as follows:

STATES: s0 s1 s2 s3 s4

INPUT_ALPHABET: a b

TAPE_ALPHABET: a b X Y -

TRANSITION_FUNCTION:				
s0	a	s1	X	R
s0	Y	s3	Y	R
s1	a	s1	a	R
s1	b	s2	Y	L
s1	Y	s1	Y	R
s2	a	s2	a	L
s2	X	s2	X	R
s2	Y	s2	Y	L
s3	Y	s3	Y	R
s3	-	s4	-	R

INITIAL_STATE: s0

BLANK_CHARACTER: -

FINAL_STATES: s4

4.3 Input String File

A valid string definition file contains a series of strings, one per line, made up of characters from the tape alphabet. The character '\ ' is a reserved character, signifying an empty string. This must appear on a line by itself to be correct. If any strings are invalid according to the language read in from the Turing machine, those specific strings are removed from the list automatically, and an appropriate error message is displayed to the user.

Example file:

```
a
ab
\
aaabb
aaaaaaaaaaaaabbbbbbbb
aabb
ba
```

5.0 Operation

5.1 Invocation

5.1.1 Command Line

The program will be run on the command line. It will accept one, and only one, parameter to be passed. Any other attempt will pass a usage message out to the user and terminate. The input parameter is the

name of the Turing machine to load. If the Turing machine definition file is named “anbn.def” then the parameter to pass is “anbn”.

Example:

```
$> tm anbn
```

5.1.2 Configuration Settings

At beginning of operation, the following configuration settings are set to their default values:

- Whether or not help messages are provided to user for all prompts (default: no)
- Maximum number of transitions to perform at a time (default: 1)
- Maximum number of cells to left and right of tape head to display in instantaneous description (default: 32) (<aab[s0]aaa>)

5.1.3 Opening Turing Machine

When opening the Turing machine, the file shall be parsed for errors and completeness. Any errors result in an error message (referencing line numbers where applicable) printed out to the user.

5.2 Commands

In the following list, the letter of the command that is bold and underlined is the activation letter of the command.

Possible commands:

- **D**elte: delete input string from list
 - 1 based index for deletion
- e**X**it: exit the application
- **H**elp: help user with prompts or not (toggle)
 - command: print out list of commands before every command prompting
- **I**nsert: insert input string into list
- **L**ist: lists input strings
- **Q**uit: quit operation of Turing machine on input string
 - if no Turing machine is being operated on, print error message
- **R**un: Run turing machine on input string
 - detailed description later
- s**E**t: set the maximum number of transitions to perform at a time for run cmd
- sho**W**: show status of application
- **T**runcate: truncating instantaneous descriptions
- **V**iew: View Turing Machine
- blank line: nothing, reprint command

5.2.1 Help User

The **H**elp command allows the user to change the setting designating whether or not help messages are provided to user before all prompts for input:

- On the “Command: “ prompt:
 - List of commands and descriptions
- Help message for every prompt should provide information on possible responses by user to a particular prompt
- Help message does not need to be long, but it must convey more information to user than the prompt itself
- Value of setting is Yes or No
 - Default is No
- Serves as toggle between values without further input

5.2.2 Show Status

The **shoW** command displays the current status of the application to the user.

- Identifying information:
 - Course, semester, year, instructor name, author name, version of application
 - Configuration settings (help messages provided, max transitions, max chars in instant desc)
 - Name of T.M. (filename without extension. so first arg, full path if given works fine.)
 - Status of T.M. (Not run on input string, currently running, or completed operation)
 - If T.M is running, display input string
 - If T.M. is running, display total number of transitions that have been performed on it
 - If T.M has completed operation, display
 - Last Input string
 - Whether it was accepted, Rejected, or operation was terminated by user before normal completion
 - Total number of transitions performed to arrive at conclusion (can be 0)
- Status information displayed from show is not retained when user exits application

5.2.3 View Turing Machine

The **V**iew command displays the Turing machine definition currently loaded in a user-readable form

- Corresponds to what was read in the definition file
- This should NOT display the contents of the file verbatim.
- Display the following things:

- Description of T M from definition file must be displayed
- Formal specification of T M as $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ in order
 - you may spell out sigma, gamma, and delta
 - should use notation from formal specification as much as possible, including equal sign, commas, curly brackets, parentheses in delta
 - These characters may be used in sigma or gamma, but no special action needs to be taken in displaying characters as members of elements of alphabet

5.2.4 List Input Strings

The **List** command displays the list of input strings on which the Turing machine may be run

- Each input string should appear on it's own line
- input string should be numbered sequentially for reference in other commands, starting at 1
- If the list is empty, this should be indicated

5.2.5 Insert Input String

The **Insert** command allows the user to enter an input string from Σ^* and append it to list of input strings

- If input string is already in list, or contains a character not in Σ , error message is displayed and string discarded
- no error if no input string entered (just kick back up to COMMAND)
- If valid string, enter in list at end, and show message that it was
- Insert one string at a time

5.2.6 Delete Input String

The **Delete** command allows the user to delete an input string from the string list.

- Deletes one string at a time
- Renumbers all following strings in the list. no gaps are allowed, only consecutive numbers
- Input string to delete is indicated by number, using 1 as the base.
- If the user enters the number of a nonexistent string, an error message is displayed.
- entering no number backs out to COMMAND:

5.2.7 Set Transitions

The **sEt** command allows the user to change setting for maximum number of transitions to perform at a time during operation of T M on an input string

- Value of setting is a positive integer.
- Default is 1
- Current value should be displayed in the prompting string
- Entering no number backs out to COMMAND:
- Message displayed on setting changed
- If invalid setting is entered, an error msg is displayed, and setting is left unchanged

5.2.8 Truncate Instantaneous Descriptions

The **Truncate** command allows the user to change setting for maximum number of cells to the left and the right of the tape head that are displayed in instantaneous description.

- 32 is default
- Value of setting is a positive integer.
- Current value should be displayed in the prompting string
- Entering no number backs out to COMMAND:
- Message displayed on setting changed
- If invalid setting is entered, an error message is displayed, and setting is left unchanged

5.2.9 Run Turing Machine

The **Run** command allows the user to trace operation of the Turing machine on input string selected from a list of input strings

- If Turing Machine is not already running, User is prompted for number of input string upon which to run Turing machine
- No error if no number entered, terminating command
- If user selects a non existent input string, error message is displayed, and command is terminated
- Initial instantaneous description is displayed on monitor to user (before t.m has been run)
 - Example:
 - 0. [S0]AABB # zero is done only at the first
 - <numTransitions>. α_1 q α_2
- Each instantaneous description is numbered with the total number of transitions performed up to that point
- q is enclosed within square brackets []
- α_1 and α_2 are able to be truncated based on maximum number of cells to display

- a < is displayed at the beginning of the instantaneous description if α_1 is truncated
- a > is displayed at the ending of the instantaneous description if α_2 is truncated
- Application does not need to “wrap” α_1 and α_2 to make the instantaneous description readable
- After displaying initial instantaneous description, if initial state is also a final state, message displayed indicating that input string is accepted, message displayed without executing any transitions. command is terminated
- initial desc only done at first run
- If newly selected input string is not accepted at that point or T.M is already running on an input string, transitions may be performed.
- To trace operation of t m until it accepts or rejects input string, or user quits running t m.
- After performing up to maximum number of transitions specified by setting, current instantaneous description is displayed on the monitor
- If error on transition, print out prior step’s inst desc, and then an error
- Do not print out intervening inst desc when doing multiple steps
- The number of transitions performed may be less than the maximum if the input string is accepted or rejected
- If T M accepts input string, inst desc displayed will included a final state, and message should be provided indicating that input string is accepted
- If T M rejects input string, Inst Desc displayed has caused T M to crash, and message should be provided to user indicating that input string is rejected
- When printing accept or reject, print out total number of transitions performed as a part of message, along with the original input string

5.2.10 Quit Turing Machine

The **Quit** command terminates operation of a Turing machine on input string before completion.

- Total number of transitions performed up to that point should be displayed as part of message along with input string
 - Examples:
 - Input string aabb has not been accepted or rejected in 10 transitions
 - Not running on input string!
- If T M is not running on an input string, error message is displayed

5.2.11 Exit Application

The **eXit** command gives no opportunity for user to confirm or cancel termination of application. If they enter an x, it’s all over.

- If an input string was inserted or deleted from list during session, entire list is written to input

string file, replacing any original file

- If any problem writing file, produce an error message.
- If successful, write that it was successful
- Command terminates execution of application

5.3 Termination

5.3.1 Closing Turning Machine

When the program terminates, if the input string list has changed, the entire list of strings is written out to file. If an error occurs there is no way to recover, as the program continues to close.

Reference

Notes from instruction