Turing Machine Application Requirements Specification

Christopher Wilkins

CptS 322 Spring 2014

Professor Corrigan

5/5/2014

**Table of Contents**

Revision History 1

1.0 Introduction 2

2.0 Background 2

3.0 Overview 3

4.0 Environment 4

5.0 Operation 8

**Revision History**

1. 5/52014 Final Submission

**1.0 Introduction**

This document describes in detail the requirements for creating and using the Turing Machine Application. This document gives a detailed description of a theoretical Turing Machine and then explains what is necessary to make an actual concrete Turing Machine simulator based on that theoretical model. This document is intended for developers who are interested in creating a Turing Machine Application. This document does not provide any design or implementation for a Turing Machine Application, but provides a detailed description of how it should work and how it should be used. This document is divided into four parts: background, overview, environment, and operation.

The background section describes the theoretical model of a Turing Machine. The section explains the various parts of the Turing Machine, their purpose, and their function. It describes the theoretical operation of the Turing Machine. The background also provides an explanation of the formal definition of a Turing Machine, including the syntax used to describe one.

The overview section describes the Turing Machine Application software and how it is used. It describes how the Turing Machine should be developed, compiled into an executable, and run. The required platforms installation steps are explained in the overview section. The overview section provides a method for the user to operate the Turing Machine Application to simulate a Turing Machine running on a user-provided input string.

The environment section explains and demonstrates the modes and methods of input and output used by the Turing Machine. It explains which devices the user should use to input information and commands into the Turing Machine. It also shows the data files that are used by the Turing Machine Application. It gives an example of both the Turing Machine Application definition file and the input string file. It shows both valid and invalid versions of these files and explains the criteria for the Turing Machine Application to accept these data files as valid input files.

The operation section gives an explanation of how a user invokes commands on the Turing Machine Application. It explains the command prompts in the Turing Machine, and gives an in depth explanation of each command available to the user. It has the requirements for the expected output for each command, as well as what is shown to the user on success or failure of each command. The operation section also provides all cases in which a command can fail due to invalid input from the user. It also explains the logic of how the Turing Machine Application determines whether or not an input string is accepted or rejected.

**2.0 Background**

A Turing Machine is a theoretical computer. It is the simplest form of a computer that is capable of executing any algorithm. A Turing Machine consists of several important parts: the tape, states, alphabet, and transitions.

The Turing Machine tape is infinitely long. It is made up of cells which contain individual characters. The Turing Machine can write characters to the tape at only one position. This position is called the tape head. The Turing Machine can move the tape head to the left or the right one cell at a time. The characters on the tape come from the Turing Machine's tape alphabet. The tape alphabet is a set of characters that can appear on the Turing Machine tape. The tape alphabet contains every character that can appear as an input string to the Turing Machine, and a character that represents blank cells on the Turing Machine. The Turing Machine also has an input alphabet which contains every character that can appear on an input string to the Turing Machine. The input alphabet is a subset of the tape alphabet. The tape is initially loaded with a finite length string known as an input string. This string consists of characters from only the input alphabet. The input string is placed on the tape with its first element at the first element of the Turing Machine tape. Every cell after the input string is a blank character. The input string cannot contain a blank character. The input string can be empty. This is represented by a tape filled with an infinite number of blank characters.

A Turing Machine can be described intuitively by a directed graph where the states are vertices and the transitions are the paths. A state is a position on the graph that the Turing Machine can be on. The state has a number of defined paths to other states or back onto itself. A transition (a path in the Turing Machine graph) represented by a character of the Turing Machine's alphabet. The Turing Machine determines which path to take based on what the current character is on the tape head of the Turing Machine tape. Each state can only have one transition per alphabet character.

The Turing Machine can be in only one state at any given time. The Turing Machine has one initial state and several halt states. The Turing Machine begins operation in the start state. If at any point the Turing Machine reaches a halt state, the Turing Machine ceases operation on the tape and does not perform any more transitions. If an input string is such that running transitions on it causes the Turing Machine to reach a halt state; that input string is considered to be accepted. If the Turing Machine runs indefinitely on the input string, or the Turing Machine reaches a state in which there are no valid transitions from the current tape, the input string is said to be rejected.

The Turing Machine operates by performing transitions on the tape. The Turing Machine has a transition function that contains the definition of every transition of the Turing Machine. A transition is made up of a source state, read character, destination state, a write character, and a move direction. The source state is the state that the Turing Machine must be in for the transition to be performed. The read character is the character that must be located at the position of the tape head on the tape for the transition to be performed. The destination state is the state that the Turing Machine will move into once the transition is performed. The write character is the character that is to be written to the tape at the position of the tape head when the transition is performed. The move direction is the direction that the tape head should be moved when a transition is performed. If the transition contains a left move when the tape head is at the first cell of the Turing Machine, the Turing Machine will crash and the input string will be rejected.

**3.0 Overview**

The purpose of the Turing Machine Application is to simulate, demonstrate, and provide a concrete implementation of the theoretical Turing Machine device. The Turing Machine Application can be used to simulate any possible Turing Machine. The Turing Machine Application allows the user to supply a Turing Machine definition that represents the Turing Machine that will be simulated.

The Turing Machine Application is intended to run on the Linux operating system using an x86 or x64 processor. The Turing Application is programmed in standard C++ using only standard C++ libraries. The Turing Machine Application is targeted toward the g++ C++ compiler. The application has no necessary development environment, however the development platform must support the g++ compiler and produce executables that run on the Linux operating system.

To run the Turing Machine Application, the user just needs the executable, named “tm”. This executable can be installed and run on any directory that the user has permission to access on a Linux computer. However, the user must supply a definition file and input string file that are both in the same directory. These can be provided to the Turing Machine with a full or relative path of the files.

Operating the Turing Machine Application to simulate the operation of a Turing Machine is performed by the application by executing a specified number of transitions per run. The user provides the Turing Machine with an input string to run on, and then using a command called “run”, the user will cause the Turing Machine to run a certain number of transitions on the input string as given by the maximum number of transitions configuration setting. The user can run transitions on an input string until the input string is either accepted or rejected. The user has control over several different configuration settings that affect how information is displayed to the user and how many transitions can be run at a time.

**4.0 Environment**

***Input and Output Devices***

1. The TMA uses a keyboard as an input device.
   1. The keyboard is used for entering commands
   2. Commands can be submitted for execution using the “Enter” key on the keyboard.
2. The TMA uses a monitor as an output device.
   1. Command prompts and command output are displayed on the monitor. This includes any messages to the user, command prompts, information requests, and status updates from the TMA.
3. The TMA uses two data files for input: TM definition file and TM input string file
   1. TM definition file contains the formal definition of the TM to load
   2. TM input string file contains a list of all input strings to load.

***Turing Machine Definition File***

The Turing Machine Definition file contains a complete rigorous definition of the turing machine that the application will simulate.

1. Turing machine definition file name is provided by the user as a command line argument.
   1. Application looks for a file with the name “param.def” where “param” is the command like argument supplied by the user.
   2. Definition file is a human-readable text file.
   3. Can contain any amount of white space between tokens.
2. The file contains 8 sections: STATES, INPUT\_ALPHABET, TAPE\_ALPHABET, TRANSITION\_FUNCTION, INITIAL\_STATE, BLANK\_CHARACTER, and FINAL\_STATES.
   1. All section names are also the headings for the sections within the file, except for the description section.
      1. The headings given above are case sensitive and must appear as shown above with a colon following each heading.
      2. Description section does not have a heading.
      3. The sections must occur in the order given in 4.2.3
      4. Each section of the turing machine definition can be separated by any number of white space characters.
3. The description section is text block containing a description of the turing machine described by the definition file.
   1. It can be formatted in anyway, including any number of white space characters.
   2. The description must be saved in the application with the exact same formatting as in the TM definition file.
   3. The description consists of all text in the file read up until the “STATES:” heading is found.
4. The STATES section provides a list of state names used by the turing machine. Each state is separated by any number of white space characters.
   1. The state names are checked to verify that they do not contain any reserved characters.
   2. Each state name is a string consisting of one or more characters.
   3. The STATES section should contain at least one state.
5. The INPUT\_ALPHABET section consists of all of the characters that can appear in an input string.
   1. The characters are single ASCII characters separated by any number of white space characters.
      1. If a string is read (more than one character without whitespace in between) an error will be displayed and the definition file will be considered invalid.
   2. The characters are checked to verify that they do not contain any reserved characters. If they do contain reserved characters, an error is displayed.
   3. If the “TAPE\_ALPHABET:” heading is encountered before any input character is read, the definition file will be considered invalid.
6. The TAPE\_ALPHABET section consists of all of the characters than can appear anywhere on the tape alphabet.
   1. Characters can include the entire input alphabet and the blank character.
   2. The characters are single ASCII characters separated by any number of white space characters.
   3. These characters are checked to verify that they do not contain any reserved characters. If they do contain reserved characters, an error is displayed.
   4. If the “INITIAL\_STATE:” heading is encountered before any input character is read, the definition file will be considered invalid.
7. The INITITIAL\_STATE section consists of a single string that is the name of the state that the Turing Machine will begin operation in.
   1. This should only consist of one string item separated by white space.
   2. The next token in the file after the state should be the next section header. If the “BLANK\_CHARACTER:” heading is encountered before any string is read, the definition file will be considered invalid.
8. The BLANK\_CHARACTER section consists of a single character that is the character that will be used to fill in empty cells on the turing machine tape.
   1. This character is verified to ensure that it is an element of the turing machine's tape alphabet.
   2. This section should only consist of an individual character separated by white space.
   3. The next token in the file after this should be the next section header. If “FINAL\_STATES” is not the next header in the file, the file will be considered invalid and a message will be displayed notifying the user.
9. The FINAL\_STATES section provides a list of state names as strings that the turing machine will halt when reached.
   1. The state names are checked to verify that they do not contain any reserved characters.
   2. Each state is checked to verify that it is an element of the turing machine's states.
   3. This heading should be the last heading in the file.
10. Well-formed definition file

This Turing machine accepts the language of one or more a's followed by the same number of b's.

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 a L

s2 Y s2 Y L

s3 Y s3 Y R

s3 - s4 – R

INITIAL\_STATE: s0

BLANK\_CHARACTER: -

FINAL\_STATES: s4

1. Malformed definition file

This Turing machine accepts the language of one or more a's followed by the same number of b's.

STATES: s0 s1 s2 s3 s4

INPUT\_ALPHABET: a b z

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 a L

s2 Y s2 Y L

s3 Y s3 Y R

s3 - s4 – R

BLANK\_CHARACTER: -

INITIAL\_STATE: s0

FINAL\_STATES: s4

* 1. The input alphabet has a character “z” that does not appear in the tape alphabet.
  2. The BLANK\_CHARACTER section appears before the INITIAL\_STATE section when it should appear after.

***Instantaneous Description***

1. The instantaneous description is a snapshot of how the tape of the turing machine and its current state.
2. The instantaneous description is a string consisting of an abbreviated version of the Turing Machine's tape and he current state of the Turing Machine.
3. The instantaneous description looks like the following:
   1. < aa [s0] bb >
   2. [s0] is the name of the current state and is positioned at the current location of the tape head.
   3. The character to the left of the current state is the character at the location of the tape head.
   4. The < and > characters show that the instantaneous description is currently being truncated to only show two characters to the left and the right of the tape head.

***Reserved Characters***

1. The reserved characters are characters used by the TMA and should not be used as state names, or in input strings unless otherwise specified
2. List of reserved characters:
   1. < (Used for truncation in instantaneous description)
   2. > (Used for truncation in instantaneous description)
   3. [ (Used for current state in instantaneous description)
   4. ] (Used for current state in instantaneous description)
   5. / (Used to denote empty string)
   6. - (Used to denote blank tape character)

***Input String File***

The input string file is a file containing every input string that the user wishes to load into the Turing Machine Application at startup.

1. The name of the input string file is the same as the Turing Machine definition name with an extension of “.str” instead of “.def”.
   1. If the input string file is not in the same directory as the Turing Machine Application executable, the user must provide a full or relative path to the file.
2. The input string file is loaded when the application starts up.

Unlike the Turing Machine definition file, the input string file can be changed by the Turing Machine Application. Throughout operation of the Turing Machine Application, users may

1. insert
   1. or delete input strings.

When the Turing Machine Application is terminated using the “Quit” command,

1. the input string file is overwritten with the current state of the input string list within the Turing Machine Application.
2. The input string file contains one input string per line.
3. A well-formed input string contains only characters that are part of the input alphabet of the Turing Machine as defined in the Turing Machine definition file.
4. An input string may be composed solely of the reserved character “\”, which represents an empty input string.
   1. An input string file may not contain any reserved characters except for the character representing an empty string.
5. The input strings in the input string file are each separated by a new line.
6. When the Turing Machine reads the input string file, it will report any strings that are malformed.
   1. If a string is malformed, it is not inserted into the input string list of the application.

If the input string file does not exist, or it does not contain any well-formed input strings,

1. the application will report this as an error message to the user
   1. and the input string list in the application will be empty.
2. the application should proceed as normal without attempting to open the input string file.
3. A new input string file will then be created or overwritten when the program exits normally.
4. If the new input string file fails to be created or overwritten, the Turing Machine application will continue to exit.

The user will need to insert input strings into the input string list before he or she can run transitions using the run command.

An example of a well-formed input string file:

a

ab

\

aaabb

aaaaaaaaabbbbbbbbb

aabb

aaaaaabbbbbb

The input strings are separated by a new line and each input string contains only characters specified in the Turing Machine definition file.

An example of a badly-formed input string file:

a

a\b

\

aaa[bb

aaaaaaaaabbbbbbbbb

aabb

aaaaaabbbbbb

This input string file contains several reserved characters. For example, it contains the opening bracket (“[“) character which is reserved for instantaneous descriptions. The file also contains the reserved character for empty strings (“\”) inside a non-empty input string.

**5.0 Operation**

***Invocation***

***Command Line***

The Turing Machine application features a command prompt with the label “command:” where a command can be entered.

1. The user enters a command by typing a letter representing the command and then pressing the “enter” key on their keyboard.
   1. The application then either displays the information associated with that command,
   2. or prompts the user for additional information in order to properly exit the command.

The following sections will explain each of the possible commands that the user can access on the Turing Machine application.

1. All commands consist of a single letter and they are case insensitive.

***Opening of Turing Machine***

The Turing Machine Application is a command line application, so it must be initialized through a console or shell application of the user's operating system.

1. The name of the Turing Machine Application executable is “tm”.
2. The Turing Machine Application accepts one command line argument.
3. This command line argument is the name of the Turing Machine definition file name.
4. The Turing Machine can be opened by typing:

“./tm tmDef”

where “tmDef” is the name of the Turing Machine definition.

On startup, the Turing Machine Application will look for two files with the name given by the Turing Machine definition name on the command line.

1. The first file is one with an extension of “.def” which contains the Turing Machine definition.
2. The second file has an extension of “.str” and contains the input string list.
3. If both files are found and do not contain errors, the user will be taken to the command prompt.
   1. Otherwise an error message will be displayed related to the error that occurred when trying to load these files.

***Configuration Settings***

The Turing Machine features three configuration settings. These settings are: maximum number of transitions, help on/off, and truncation.

1. These settings each have a default setting which is set automatically when starting the Turing Machine application.
2. These settings can be changed through the commands of the Turing Machine application.
3. However, any changes to these settings are not permanent and they will revert back to their original values each time the Turing Machine application is run.

The maximum number of transitions configuration setting controls how many transitions that the Turing Machine application performs each time it is run on an input string.

1. The default value for this configuration setting is 1 transition. This means that when the run command is used, the Turing Machine will attempt to run one transition on the input string. This is a maximum number, so the actual number may be lower if the Turing Machine reaches a halt state or the input string is rejected.
2. This setting can be changed by using the Set command described below.

The help on/off configuration setting controls whether or not help information is displayed on the input prompts of the Turing Machine application.

1. Help information includes
   1. instructions
   2. and example inputs for each help prompt.
2. For the command prompt,
   1. a list of all commands,
   2. the character to activate,
   3. and a description is shown.
3. When this setting is turned on, help prompts are displayed for each command prompt.
4. When this setting is turned off, no help prompts are displayed for any commands.
5. This configuration setting can be toggled using the help command (described below).
   1. The default setting is off.

The truncate configuration setting controls how many characters on the tape to the left and the right of the tape head on the instantaneous description are displayed.

1. The default value for this configuration setting is 3. This means that when the instantaneous description is displayed, the maximum number of tape character to be displayed on either side of the tape head will be 3. If there are more than 3 characters, the tape will be truncated and a '<' will be shown on the left or a '>' will be shown on the right.
2. This configuration setting can be changed by the truncate command as described below.

***Help Command***

The help command allows the user to toggle the display of help prompts for the Turing Machine application.

1. This command can be activated by typing “h” at the command prompt and pressing enter on the keyboard.
2. This will display a different message depending on whether or not help is turned on or off by executing this command.
   1. If help is turned off when the command is run, the configuration setting for help will be set to “on” and the message “Help is enabled.” will display.
   2. If help is turned on when the command is run, the configuration setting for help will be set to “off” and the message “Help is disabled” will be displayed.

***Show Command***

The Turing Machine’s Show command displays detailed information about the Turing Machine application and the currently loaded Turing Machine definition.

1. It can be accessed by entering “w” at the command prompt.
2. It displays the application’s
   1. author,
   2. version number,
   3. class name,
   4. instructor,
   5. and date of creation.
3. It also shows
   1. the name of the current Turing Machine definition,
   2. the status of the Turing Machine (whether or not it is or has run on an input string),
   3. the number of transitions performed (if any),
   4. and finally the configuration settings of the Turing Machine.
4. After displaying this information, the user is immediately taken back to the command prompt without further input.

***View Command***

The View command displays the current Turing Machine definition. Once the command is entered at the command prompt, no further information is requested from the user.

1. It can be accessed by typing “v” at the command prompt and pressing enter on the keyboard.
2. It displays the description of the Turing Machine: a list of all states, the input alphabet (sigma), the tape alphabet (gamma), a list of final states, the blank character, and a starting state.
3. After displaying this information, the user is immediately taken back to the command prompt without further input.

The description displayed is the definition of the Turing Machine as given by the Turing Machine Definition file.

***List Command***

The list command shows the list of input strings that are currently loaded into the Turing Machine application. Once the command is entered at the command prompt, no further information is requested from the user.

1. It can be accessed by typing “l” at the command prompt and pressing enter on the keyboard.
2. When “l” is entered, an enumerated list (beginning at 1) of all input strings is displayed.
3. This list consists of all input strings that have either been read from the input string file or added by the user by using the “insert” command. The number of each input string represents the number that the input string is accessed when using the “run” and “delete” commands.
4. After displaying this information, the user is immediately taken back to the command prompt without further input.

***Insert Command***

The insert command allows the user to enter a new input string.

1. It can be accessed by entering “i” at the command prompt and pressing enter on the keyboard.
2. If an empty input string is entered (ie pressing “enter” at the input string prompt without typing anything else), the user will immediately be taken back to the command prompt silently without requiring any other input.
3. Once the user has initiated the insert command, the application will then prompt for an input string, which reads “Enter input string: “.
4. The user must then type the input string that he or she wishes to enter. After typing the input string, the user will press enter on their keyboard to submit the input string for validation.
5. The Turing Machine application validates the input string by checking to see if each character in the input string is present in the input alphabet of the Turing Machine, as defined in the Turing Machine definition file.
   1. The input string may not contain any characters other than characters which appear in the input alphabet of the Turing Machine definition.
   2. The input string may not contain the empty character.
   3. The Turing Machine must not contain any reserved characters. These include '<', '>', '[', ']'.
   4. The Turing Machine may contain a “/”, if it is the only character in the input string. The “/” character represents an empty input string and is a reserved character in the Turing Machine.
6. If the input string passes validation, it will be inserted into the Turing Machine and a positive message affirming that it has been added will appear onscreen.
   1. This message is: “Input string has successfully been inserted.” and the input string will be inserted into the input string list.
7. If the input string does not pass validation, the user will be given a negative message notifying them that it is an invalid input string, and the input string will not be inserted into the input string list.
   1. This message is “Input string was not valid and has not been inserted.”

Regardless of whether or not the input string passes validation, the user will be taken immediately back to the command prompt without requiring any further input.

***Delete Command***

The delete command allows the user to delete an input string from the application.

1. It can be accessed by entering “d” at the command prompt and pressing enter on the keyboard.
2. If the user types nothing and presses enter at this prompt, the user will be taken immediately back to the command prompt without further input being required.
3. The user will then be prompted for the index of the input string to delete. The index is the number of the input string as shown in the List command.
   1. This prompt is “Delete input string index: ”.
4. Once the user enters an integer for the input string, the application will then validate the index to check to see if it is correct.
5. If the index entered does not pass validation a message will be displayed notifying the user that he or she entered an invalid index number.
   1. The message displayed will be “Invalid input string index.”
6. Validation of the index will not pass if the index is greater than the maximum number of strings in the input string list, or if it is less than or equal to 0, or the index entered is not an integer.
7. If the index entered passes validation, the input string will be removed from the input string list and a message will appear on the screen informing the user that the input string has been deleted.
   1. This message will be “Input string [input string] has been removed from the input string list.”

Regardless of whether or not the index was validated, the user will be taken back to the command prompt immediately without requiring any further input.

***Set Command***

The set command allows the user to set the maximum transitions to run configuration setting of the Turing Machine application. This command allows the user to change the maximum number of transitions configuration setting.

1. It can be accessed by typing “e” at the command prompt and pressing enter.
2. On entering this command, the user will be prompted for the maximum transitions to perform.
   1. This prompt is “Maximum number of transitions to be performed (currently: [current maximum transitions]: ”
3. It will also show the current value that this configuration setting is set to.
4. If the user presses enter at this prompt without entering anything, he or she will be taken immediately back to the command prompt without further input.
5. At this prompt, the user will enter a number which is the maximum number of transitions to perform.
6. The input from the user will then be checked to ensure that it is a valid number of transitions that can be performed. The input to this prompt from the user consists of an integer greater than 0.
7. If the user entered a number less than or equal to 0, or anything that is not an integer, the setting will not be changed and the user will receive a message informing him or her.
   1. The message displayed will be: “That is an invalid number of transitions.”
8. If the user enters a value that is equal to the current number value of the configuration setting, no message will be displayed and the user will be taken immediately back to the command prompt without requiring any further input.
9. If the input passes validation, the setting will be changed and a message will be displayed notifying the user of this.
   1. The message displayed is: “Maximum number of transitions to perform has been changed to [user input]”.

Regardless of whether or not the input passes validation, the user will be taken back to the command prompt without requiring any further input.

***Truncate Command***

The truncate command allows the user to change the configuration setting for the maximum number of tape characters to be displayed to the left and the right of the tape head when an instantaneous description is displayed.

1. This command can be accessed by typing “t” at the command prompt and pressing enter.
2. Upon activating this command, the user will be prompted for the maximum number of transitions to perform.
   1. The prompt displayed is: “Number of characters in instantaneous description (current=[current number]): ”
3. It will also show the current value that this configuration setting is set to.
4. The input from the user to this prompt consists of an integer representing the number of characters that should be shown to the left and the right of the tape head on the instantaneous description.
5. After the user enters this value, it will be validated to ensure it is a valid number of characters to be displayed.
6. If the user enters a number less than or equal to 0 or enters non-numeric characters the setting will not be changed and the user will be notified that their input is invalid.
   1. This message is: “Invalid truncation value.”
7. If the user enters a value that is equal to the current number value of the configuration setting, no message will be displayed and the user will be taken immediately back to the command prompt without requiring any further input.
8. If the input from the user passes validation, the truncation setting will be changed and the user will be informed that is was changed.
   1. This message is: “Truncation value has been changed to [user input].”

Regardless of whether or not the input passes validation, the user will be taken back to the command prompt without requiring any further input.

***Run Command***

The run command will cause the Turing Machine to begin running on an input string.

1. The run command is triggered by typing “r” at the command prompt.
2. If the Turing Machine is not currently running on an input string, the user will be prompted for additional information.
   1. The user is prompted for the index of the input string to run transitions on. The input string is given by the user as the index of the input string as shown on the listing of the List command.
   2. This index must be less than or equal to the number of input strings in the input string list and greater than or equal to zero.
   3. The input string index provided by the user is validated to ensure that it is a valid index. Once the user enters an integer for the input string, the application will then validate the index to check to see if it is correct. If the index entered is greater than the maximum number of strings in the input string list, or if it is less than or equal to 0, or the index entered is not an integer, then a message will be displayed notifying the user that he or she entered an invalid index number.
      1. The message displayed will be “Invalid input string index.”
   4. If the index entered passes validation, the input Turing Machine will begin running transitions on the input string.
3. If the user has already began operation on the input string on a previous execution of the run command, the user will not be prompted for any information.
4. The Turing Machine proceed will attempt to perform the maximum number of transitions as specified by the maximum number of transitions configuration setting.

When the Turing Machine Application operates on a new input string, it must perform several states to begin operation.

1. The Turing Machine loads the input string onto the tape.
2. The Turing Machine then sets its current state to the initial state as defined in the Turing Machine definition file.
3. The Turing Machine tracks the total number of transitions performed, so this value is reset to zero.
4. The Turing Machine then begins to perform transitions. It attempts to perform as many transitions as are specified by the maximum number of transitions configuration setting.

When the run command is run successfully:

1. it will display an instantaneous description of how the tape of the Turing Machine looked before running transitions and an instantaneous description of how the tape looked after running transitions.

***Instantaneous Description***

The instantaneous description is a string consisting of an abbreviated version of the Turing Machine's tape and he current state of the Turing Machine. The instantaneous description looks like the following:

< aa [s0] bb >

where [s0] is the name of the current state and is positioned at the current location of the tape head. The character to the left of the current state is the character at the location of the tape head.

1. The < and > characters show that the instantaneous description is currently being truncated to only show two characters to the left and the right of the tape head.

The Turing Machine performs transitions by reading the character on the tape at the tape head, looking for a transition from the current state for this tape character.

1. If there is a valid transition:
   1. The Turing Machine writes the write character defined in the transition,
   2. Updates current state to the transition's destination state
   3. Moves the tape head in the direction specified by the transition.
2. If there is not valid transition from the current state given the current character on the tape at the tape head, the Turing Machine will crash and the input string will be rejected.
3. If the input string is rejected, a message will be displayed to the user: “Input string rejected in [transitions performed]” where “[transitions performed]” is the total number of transitions performed on the Turing Machine so far.
4. If the Turing Machine reaches a halt state before the maximum number of transitions have occurred, a message will display, informing the user that the input string has been accepted.
   1. This message should be “Input string accepted in [transitions performed]”.
5. If the input string is not accepted or rejected by the time it has run the maximum number of transitions, nothing will be displayed to the user.
6. To run more transitions, the user must use the run command again.

***Quit Command***

The quit command is used to stop the Turing Machine from running on an input string.

1. This command can be accessed by typing “q” at the command prompt.
2. When this command is entered, the Turing Machine will cease to run on the Turing Machine.
3. The total number of transitions that were performed will be displayed onscreen.
4. This command requires no further input from the user, so the user will be taken back to the command prompt without requiring any further input.
5. If the Turing Machine is not currently running on an input string, a message will be displayed telling the user that nothing was quit.
   1. This message is: “The Turing Machine is not currently running on an input string.”

***Exit Command***

The exit command is used to exit the Turing Machine application.

1. This command can be accessed by typing “x” at the command prompt.
2. Upon entering this command, the Turing Machine will attempt to write the current input string list to the input string file.
   1. The input string file is given by the name entered on the command prompt at launch of the application plus “.str”.
3. The Turing Machine will overwrite the existing input string file on exit.
4. Each input string in the current input string list will be written, with one input string per line.
5. A message will display telling the user whether or not the input string file was successfully written.
   1. If the input string is successfully written a message “Input string file successfully written.” will be displayed.
   2. Otherwise a message “Could not write input string file.” will be displayed.
6. Regardless of whether or not the input string file was successfully written, the application will exit and the user will be taken immediately back to their operating system without requiring any further input.