Turing Machine Requirements Specification

CPTS322 Software Engineering I

Christian Penick

April 29, 2024

Table of Contents

[List of Figures 3](#_Toc164775151)

[Introduction 4](#_Toc164775152)

[Background 4](#_Toc164775153)

[Overview 7](#_Toc164775154)

[Environment 7](#_Toc164775155)

[Input and Output Devices 7](#_Toc164775156)

[Turing Machine Definition File 7](#_Toc164775157)

[States 8](#_Toc164775158)

[Alphabets 8](#_Toc164775159)

[Transition Functions 9](#_Toc164775160)

[Initial State 9](#_Toc164775161)

[Blank Character 9](#_Toc164775162)

[Final States 9](#_Toc164775163)

[Turing Machine Input String File 9](#_Toc164775164)

[Operation 10](#_Toc164775165)

[Commands 11](#_Toc164775166)

[Help User (H) 11](#_Toc164775167)

[Show Status (W) 11](#_Toc164775168)

[View Turing Machine (V) 11](#_Toc164775169)

[List Input Strings (L) 11](#_Toc164775170)

[Insert Input String (I) 12](#_Toc164775171)

[Delete Input String (D) 12](#_Toc164775172)

[Set Transitions (E) 12](#_Toc164775173)

[Truncate Instantaneous Descriptions (T) 12](#_Toc164775174)

[Run Turing Machine (R) 13](#_Toc164775175)

[Quit Turing Machine (Q) 14](#_Toc164775176)

[Exit Application (X) 14](#_Toc164775177)

[References 15](#_Toc164775178)

# List of Figures

Figure 1: Visualization of TM tape………………………………………….5

Figure 2: Graph Representation of TM…………………………………….6

Figure 3: TM Definition File Example………………………………………8

Figure 4: TM String File Example…………………………………………..9

# Introduction

This document serves to inform the software recipient of information regarding the Turing Machine (TM) software application Christian Penick will produce. This document will discuss TM theory and purpose as well as the operation of the TM software application.

The remainder of this document will describe the TM as an abstract mathematical model of a computer, the utility of the TM application, the environment of the TM application, and operation of the TM application.

# Background

A Turing Machine (TM) is an abstract mathematical model of a computer invented by Allen Turing in the 1930s and is thought to be the ultimate calculating mechanism despite its simple appearance.

A TM recognizes a language (strings of symbols from an alphabet). This alphabet, denoted by , is finite in length. The language is a list of all the possible strings of symbols that can be made from this finite alphabet. Thus, a language is infinite in length. It is important to note that a TM is only concerned with the syntax of the string of characters and is not concerned with the meaning of the string. This type of language is called a formal language. Because only the order of the string matters, we specify a formal language as a list of rules for constructing strings given a finite alphabet. For example, given an alphabet that contains the symbols ‘A’ and ‘B,’ we can create the following language: . This rule says our language consists of strings that contain any number of As followed by the same number of Bs. Because a TM can recognize a language, when given an arbitrary input string, there are three possible results: the TM will accept the input string (meaning the string is in the language the TM recognizes), the TM will not accept the input string (meaning the string is not in the language the TM recognizes), or the TM will run indefinitely (neither accepting nor rejecting the input string).

A TM consists of a tape (made up of cells) and a tape-head that can write to the tape and move left or right. A TM can be pictured as follows:

A diagram of a diagram

Description automatically generated

The tape consists of cells, each of which contains a character from the tape alphabet. The tape is infinite in one direction. At the beginning of TM operation, the input string is stored in the leftmost cells of the tape. Other cells initially contain the blank character, such as . It is important to note that the input alphabet is a subset of the tape alphabet and does not contain the blank character. The tape-head is able to read a character on tape, write a character over it, and move one cell left or right on the tape. The tape head starts at first cell and may never move left from that cell. The TM contains a finite list of states, one of which is the start state. The TM has 0 or more final states which cause it to halt when entered. State can be changed as part of move including tape head. Since TM can scan tape forwards and backwards, it does not necessarily halt when it encounters the first blank character.

Transition functions define how the tape-head moves and writes and how the state changed. The input to a transition function is the current state of the TM and the character in the next cell. The output from a transition function is the character to write to the current cell, which direction to move the tape-head (left or right), and the new state of the TM.

A diagram of a diagram

Description automatically generated If we think of the states as nodes of a graph, and the transition functions as edges of a graph, we can visualize a TM as a graph. For example:

Formally, a Turing machine is specified by M = (Q, , , , , B, F), where

1. Q is a finite set of states
2. is a finite input alphabet
3. is a finite tape alphabet
4. Delta is a transition function from
5. is the initial state
6. is the blank symbol
7. is the set of final states

# Overview

A TM attempts to accept or reject a user provided input string. The TM application development occurred on an x64 PC using the Linux Ubuntu operating system. The user must install the TM application to a computer using the Linux Ubuntu operating system. The user will download the TM executable and execute the application from the command line. The user will provide the name that is used for both a file that defines the TM to use as well as the file that contains input strings the user wants to test on the previously defined TM.

# Environment

## Input and Output Devices

The TM application employs a keyboard as the input device and a monitor as the output device.

## Turing Machine Definition File

The Turing Machine definition file must have the extension “.def.” A valid TM definition file contains eight components/key words which may appear in any case/combination of cases. Whitespaces are to be used as a delimiter between tokens (groups of characters). Whitespaces consist of ‘\n’, ‘\t’, spaces, and any combination is to be considered whitespace. Because the definition file is free format, these whitespaces may be placed arbitrarily; however, as noted, they must be used to delimit groups of characters.

A screenshot of a computer

Description automatically generatedThe components of the definition file must appear in this order: TM description, states, input alphabet, tape alphabet, transition function, initial state, blank character, and final states. The description of each component following the TM description begins with the name of the component (if it has a space in the name, replace it with an underscore), and followed by a colon. Every token in a component of the TM definition must be separated with a space. Using this syntax, a valid TM definition files appears as such:

### States

A state in Q is named as a string of upper- and lower-case letters, digits 0-9, underscores, and/or hyphens. A state is case sensitive and must be unique. No limit on length of state name or number of states.

### Alphabets

Input alphabet and tape alphabet will consist only of printable characters from Ascii character set with exception of “\”, “[“, “]”, “<”, “>”. White space characters are excluded from input alphabet and tape alphabet. Characters in input alphabet and tape alphabet must be unique since both are sets. Every character in the input alphabet must be in the tape alphabet. The blank symbol must not be in input alphabet.

### Transition Functions

Any number of transitions may be defined in the set of transition functions. Each transition must use states in set of states and tape characters in tape alphabet, otherwise the transition is undefined, and the TM will crash during execution. Transition must also define direction to move tape head (left or right), using the characters ‘L’ or ‘R’ in upper or lower case. At most one transition from a given state on a given tape character may be defined because the TM is deterministic. No transitions may be defined to leave the final state. Every transition must be fully defined (cannot have any absent read/write characters, source/destination states or move direction)

### Initial State

Initial state must be in set of states.

### Blank Character

The blank symbol must be a tape character in tape alphabet.

### Final States

Any state in set of states may be defined to be a final state in set of final states. States in set of final states must be unique.

## Turing Machine Input String File

The input string file must have the extension “.str.’ The input string file contains input strings for use during operation of TM application. Each line of the input string file is an input string composed of characters from the input alphabet. The empty string is specified by reserved character ‘\’ appearing on a line by itself. If any line is empty or contains a character not in set of input alphabet, the application will discard the line and will display an error message on the console. Duplicate input strings are discarded by application. There is no limit on the length of string in .str file or on the number of strings.

# Operation

The user has access to numerous commands when running the TM application. This section aims to help the user understand how to start the program, perform standard use of the application, and how to close the application.

Invocation occurs on the command line in a directory holding the TM application. Invocation begins with the name of the application followed either an absolute or relative path of the TM definition file, without the extension ‘.def.’ No other command line variations are accepted. If a definition file of that name does not exist, cannot be opened, or does not contain a valid TM definition, execution will be terminated. An input string file of the same name is not required for TM application invocation.

There are two configuration settings the user has access to in the TM application. The first configuration setting specifies the maximum number of transitions to perform at a time during operation of the TM on an input string. The default number of transitions to perform is one. To modify this configuration setting, input ‘E’ when prompted by the application for a command. The second configuration setting specifies the max number of cells to the left and right of the tape head to display in instantaneous descriptions during operation of TM on an input string. The default number of cells displayed on both sides is thirty-two. To modify this configuration setting, input ‘T’ when prompted by the application for a command. The user may adjust these two configuration settings both before the TM has begun running on an input string and during the TM application’s tracing of the input string.

## Commands

### Help User (H)

Enter ‘H’ at command prompt to display a list of all commands, including the name of the command, a description of the command, and its character to type at the command line.

### Show Status (W)

Enter ‘W’ at command prompt to display the status of the application to the user. Information includes current configuration settings, name of definition file, status of TM (three possible states), course, semester, year, instructor, author, version, and status of list of input strings (whether it will be written to file or not). If TM is running, it will display the input string and total number of transition performed so far. If TM has completed operation on an input string, the command will display the last input string, and whether it was accepted, rejected, or quite before answer, and the total number of transitions performed.

### View Turing Machine (V)

Enter ‘V’ at command prompt to display the formal definition of the TM as M = (Q, Σ, Γ, Δ, Q0, B, F) either using Greek symbols, or spelled out names of the Greek symbols, but not using the keywords from the definition file. The TM description follows the formal definition.

### List Input Strings (L)

Enter ‘L’ at command prompt to display a list of input strings upon which the TM may be run. Input strings will be numbered sequentially, allowing the user to reference a certain string in the Delete Input String command.

### Insert Input String (I)

Enter ‘I’ at command prompt to enter an input string and append it to the list of input strings. If the input string is already in list or contains a character not specified in the TM alphabet, the string will be discarded. If string is inserted, the list of strings will be written out to file. If no input string is entered, the list of input strings remains unchanged.

### Delete Input String (D)

Enter ‘D’ at command prompt to be prompted for the number of the line the user would like to delete. This command can cause a renumbering of the strings as displayed by the List Input Strings command. If string is deleted, the list of strings will be written out to file. If no number is provided, the list of input strings remains unchanged.

### Set Transitions (E)

Enter ‘E’ at command prompt to change the maximum number of transitions to perform at a time during operation of TM on an input string. The value must be specified as a positive integer (must be greater than 0). The current value will be displayed for the user. The user may simply type *enter* to not change the value of this configuration setting.

### Truncate Instantaneous Descriptions (T)

Enter ‘T’ at command prompt to change the setting for maximum number of cells to left and right of tape head to display in instantaneous descriptions during operation of TM on an input string. The current value will be displayed for the user. The user may simply type *enter* to not change the value of this configuration setting.

### Run Turing Machine (R)

Enter ‘R’ at command prompt to trace the operation of the TM on an input string selected from the list (using its corresponding line number). If a non-existent input string is specified, the command will be terminated. If only *enter* is typed, the user will be prompted to enter a command again. If a valid number is entered, the initial instantaneous description will be displayed. No intermediate instantaneous descriptions will be displayed. After performing up to the max number of transitions, the current instantaneous description will be displayed. If the TM accepts or rejects the input string, the total number of transitions performed is displayed along with the original input string and result (accepted or rejected). If the input string was rejected (the TM crashed), then the instantaneous description right before the crash will be displayed. If the original input string was not accepted within the specified maximum number of transitions, subsequent calls to the **run** **(R)** command will continue to perform up to the maximum number of transitions, stopping either at acceptance or rejection. An input string is accepted when the TM enters a final state. An input string is rejected when a transition for the current source state and read character can not be found, or a transition function causes the tape head to move to the left of the first cell. If the TM accepts the original input string, the Boolean **accepted** is set to true. If the Turing Machine rejects the original input string, the Boolean **rejected** is set to true. A user may quit operation of the TM through the **quit** command so that the next call to **run (R)** will again prompt the user for the line number of the input string they would like to run the TM on.

### Quit Turing Machine (Q)

Enter ‘Q’ at command prompt to terminate operation of the TM on an input string before completion. If the TM is not running on an input string, an error will be displayed, and the user will be prompted again for a command. If the TM is running on an input string, the following information will be displayed: input string, total number of transitions performed, and message indicating that the string is neither accepted nor rejected. The user will then be prompted for a command.

### Exit Application (X)

Enter ‘X’ at command prompt to cancel termination of the application. This command provides no opportunity for the user to confirm or cancel termination of the application. If an input string was discarded, inserted, or deleted from list during session, entire list is written to input string file, replacing the original file and a message will be provided indicating that input string file was successfully written, otherwise, an error message will be displayed.

# References

Corrigan, Neil. *Dr. CPT\_S 322 Class Notes*. 2024.