Turing Machine Design Document

CptS 322 Daniel Hanlen

May 5, 2014

Table of Contents

List of Figures

Revision History

April 11, 2014 – Initial design compiled

1.0 Introduction

The purpose of this document is to outline the design elements of the Turing Machine being designed for CPTS 322 at Washington State University Tri - Cities. The primary audience is the instructor and student involved. Some material will be drawn from a previous Requirements Specification, which covered everything the program should and should not do. In this document it will be shown how the program will accomplish this structurally, in addition to example behaviors of the program during operation. The first part of the program will have the UML diagrams for each class given. Next will be a detailed account of what the functions described in the UML do. Third will be examples of what the user interface will look like. Finally, example files for the Turing Machine and String Definitions will be given.

2.0 Architecture

The primary class is the main Turing Machine class. This is what contains the Tap, Tape Alphabet, Input Alphabet, States, Final States, and Transition Functions. The Transition Function also contains the Transition class.

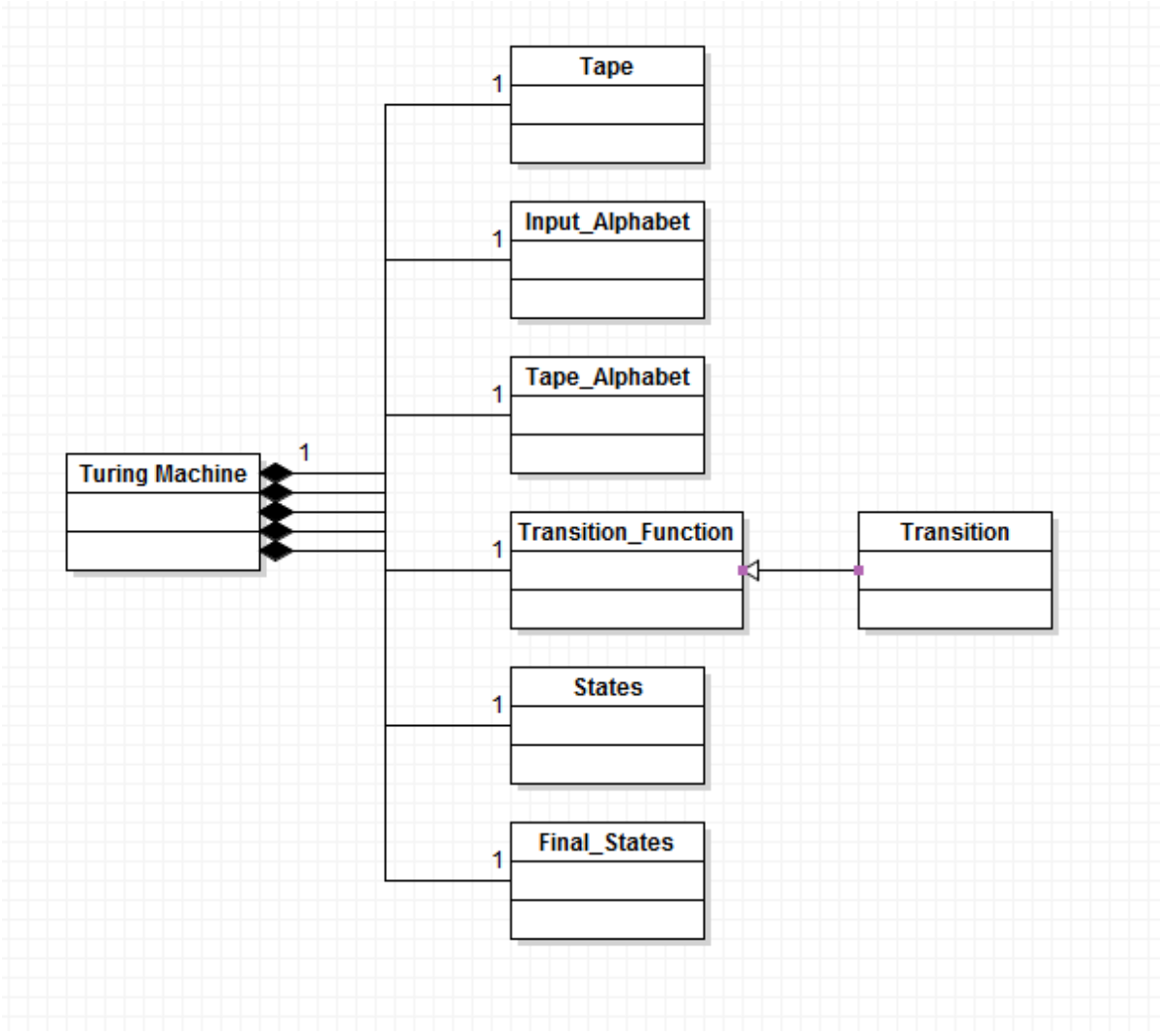

Illustration 1 Turing Machine

Of course, the details of those classes need to be defined as well.

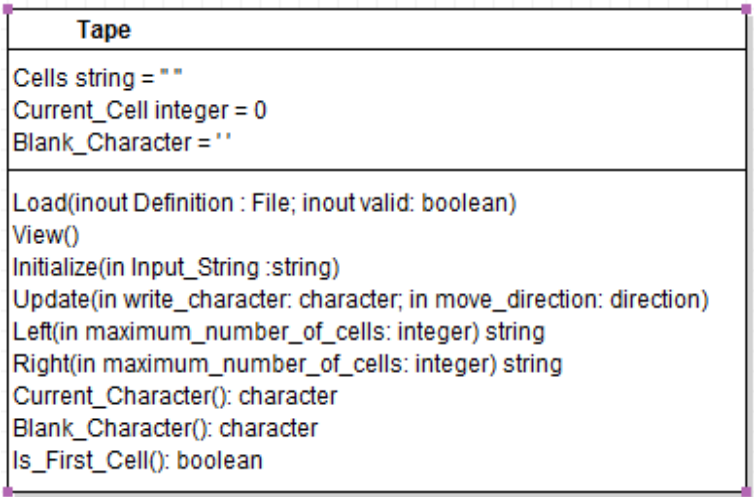

Illustration 2 Tape

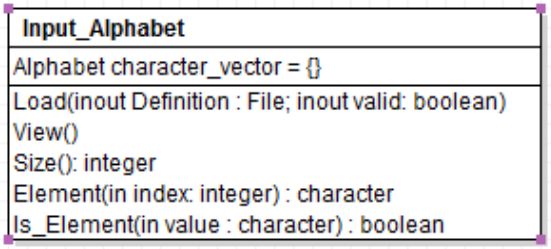

Illustration 3 Input Alphabet

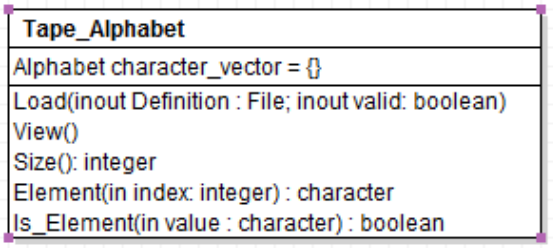

Illustration 4 Tape Alphabet

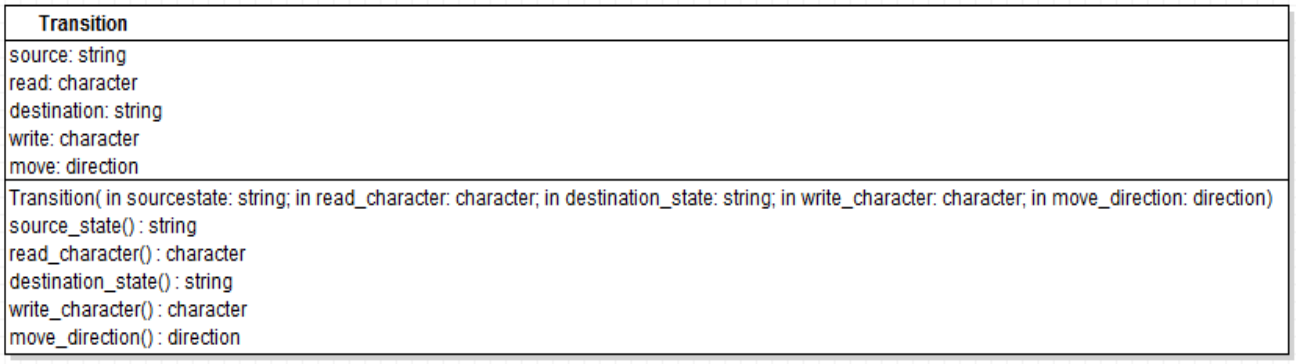

Illustration 5 Transition

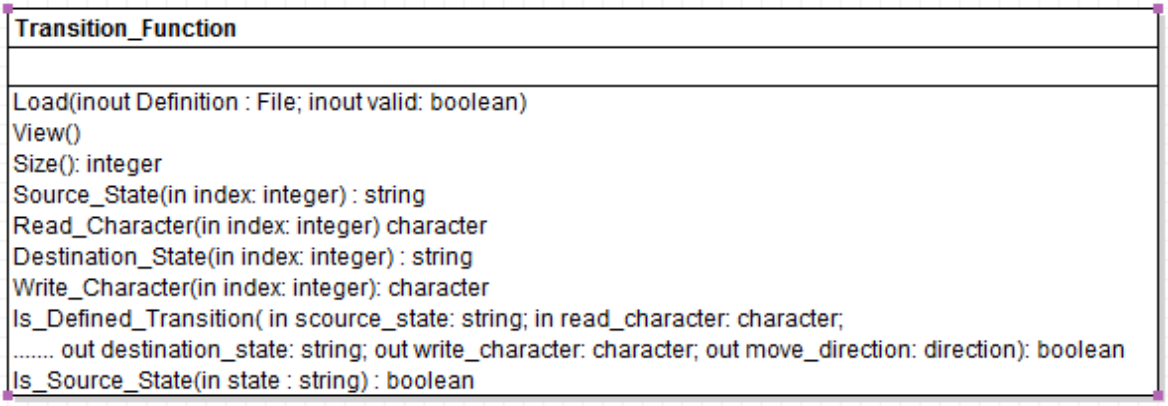

Illustration 6 Transition_Function

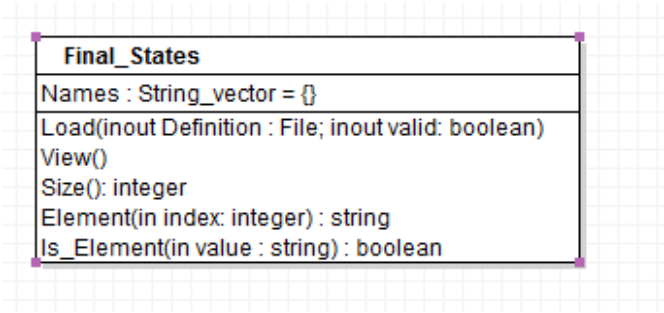

Illustration 7 Final States

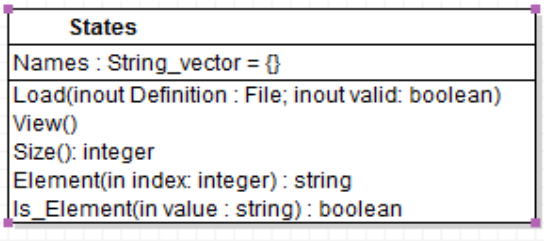

Illustration 8 States

3.0 Data Dictionary

In this section, details will be provided for each class in the Turing Machine.

3.1 Tape

3.1.1 Description

The Tape of a Turing machine consists of an ordered sequence of cells, indexed starting at 0, which

may grow to any size needed up to the limit of storage during operation of the machine on an input string. Each cell contains a character in the tape alphabet. An input string is stored in the lowest numbered tape cells at the beginning of operation, and all other tape cells initially contain the blank character. The current cell starts at the first cell on the tape. In performing a transition of the Turing machine, the character contained in the current cell may be read and written, and the current cell may be moved one cell to the left or right. The tape exists only as part of a Turing machine.

3.1.2 Associations

The class Tape is a component of the class Turing_Machine, receiving messages delegated to it by the Turing machine.

3.1.3 Attributes

cells: String = “ ”

The attribute cells is a dynamically growing character string containing the Turing machine tape. Whenever necessary, it may be extended by appending a blank character.

current_cell : Integer = 0

The index of the current cell on the Turning machine tape is stored in the attribute current_cell.

blank : Character = ' '

The blank character of the Turing machine is contained in the attribute blank.

3.1.4 Methods

load(inout definition : File, inout valid: Boolean)

The method load reads the blank character from the Turing machine definition file. If the blank character is reserved or not printable, or the next keyword does not follow it in the file, an error message is displayed and valid is set to false.

view()

The method view displays the blank character of the Turing Machine.

initialize(in input_string: String)

The method initialize sets the Turing machine tape to the input string followed by a blank character, replacing the previous contents of the tape. The current cell is set to the first cell on the tape, indicated by the index 0.

update(in write_character: Character, in move_direction: Direction)

The method update first determines if the update of the Turing machine tape is possible. The method returns if a left move is specified from the first cell, replacing the previous character in that cell. To move the current cell one cell to the left, the index is decremented, or to move the current cell one cell to the right, the index is incremented.

left(in maximum_number_of_cells : Integer) : String

The method left returns a character string of up to the maximum number of cells from the Turing

machine tape to the left of the current cell, excluding that cell. The length of the string will be less than the maximum if there are fewer cells to the left of the current cell. If the string is truncated from the tape, the reserved character '<' will be added to the beginning of the string.

right(in maximum_number_of_cells : Integer) : String

The method right returns a character string of up to the maximum number of cells from the Turing machine tape to the right of the current cell, including that cell. The length of the string will be less than the maximum if there are fewer cells to the right of the current cell. If the string is truncated from the tape, the reserved character '>' will be added to the ending of the string.

current_character() : Character

The method current_character returns the character contained in the current cell on the Turning machine tape.

blank_character() : Character

The method **blank_character** return the blank character of the Turning machine.

is_first_cell() : Boolean

The method **is_first_cell** returns a value of true if the current cell on the Turing machine tape is the first cell, indicated by the index 0. Otherwise, it returns a value of false.

3.2 Final_State

3.2.1 Description

Final_State is used to hold the string naming the final state of the application.

3.2.2 Associations

A member of Final_States, it can belong to Final_States many times.

3.2.3 Attributes

name : String

The name of the final state

numberOfFinalStates : interface

the total number of final states in existance

3.2.4 Methods

Operator = () : Final_State

We want to use a deep copy on this

Get_Name(out string)

Get the name of this final state

Set_Name(string)

Set the name of this final state

3.3 Input_Alphabet

3.3.1 Description

Input_Alphabet is used to contain the alphabet allowed to an input string.

3.3.2 Associations

Input_Alphabet is contained in Turing_Machine.

3.3.3 Attributes

alphabet: Character array

contains the characters allowed in the input alphabet

3.3.4 Methods

load(inout definition : File, inout valid: Boolean)

The method load reads the input characters from the Turing machine definition file.

view()

This method views the input alphabet for use in printing the TM

size() : int

The method size returns how many characters are in the input alphabet

element(value : int) : character

The method element returns the character at the given position in the array

is_element(value : char) : boolean

The method is_element returns true if the given character can be found in the alphabet, false if not.

3.4 Tape_Alphabet

3.4.1 Description

Tape_Alphabet is used to contain the alphabet allowed to be used on the tape.

3.4.2 Associations

Tape_Alphabet is contained in Turing_Machine.

3.4.3 Attributes

alphabet: Character array

contains the characters allowed in the tape alphabet

3.4.4 Methods

load(inout definition : File, inout valid: Boolean)

The method load reads the tape characters from the Turing machine definition file.

view()

This method views the tape alphabet for use in printing the TM

size() : int

The method size returns how many characters are in the tape alphabet

element(value : int) : character

The method element returns the character at the given position in the array

is_element(value : char) : boolean

The method is_element returns true if the given character can be found in the alphabet, false if not.

4.0 User Interface

Below are examples on how the output will look for every user output.

4.1 Command Line Invocation

>tm anab

Successfully loaded anbn.def and anbn.str

Command: _

4.2 Help Command

Comand: h

Help enabled.

Command list:<list Commands below>

Delete – deletes a string

eXit – exit the program, saving the list of strings if any changes were made

Help – toggle help on/off

Insert – insert a string into the end of the list of strings

List – list the strings available to run

Quit – stop execution of the TM

Run – run the turring machine or continue to run the TM

sEt – set the number of transitions to perform in one run command

shoW – show information about this application

Truncate – truncate the number of characters displayed in the instantaneous description of the TM

View – view the formal definition of the TM

Command: _

4.3 Show Command

Command: w

CptS322 Spring 2014

Instructor: Dr. Corrigan
Author: Daniel Hanlen

Help is off
String truncated at 32 characters
1 Transition per run

Version: 0.1

TM: AnBn TM is currently running on "aabb" with 5 transitions

Command:_

4.4 View Command

Prints out formal definition of the Turing machine

4.5 List Command

Command: l
List of input strings:
1: ab
2: aab
3: aabb
Command: _

4.6 Insert Command

Command: i
Enter string: aabba
String inserted at position 4

Command: _

4.7 Delete Command

Command: d

Enter index # to delete from list: 3

String 'aabbb' has been deleted

Command: _

4.8 Set Command

Command: e

Number of transitions (1): 0

Number of transitions must be positive

Command: _

4.9 Truncate Command

Command: t

Character display limit (32): -1

Display limit must be a positive integer.

Command: _

4.10 Run Command

Command: r

Input string #: 3

0. [s0]aabb

1. x[s1]abb

Command: _

4.11 Quit Command

Command: q

Quiting Turing machine processing.

Input string aabb has not been accepted or rejected in 10 transitions.

Command: _

4.12 Exit Command

Command: x

Exiting program. The list of strings has been written to anbn.str

> _

5.0 Files

Below are example files for the Turring Machine Definition File and the Input String File.

5.1 Turing Machine Definition File

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

5.2 Input String File

abbababbabaaabbbabbaba bababa abbab

abcabcaaccccbb bcaaacccbbbbbbbbb
ccabbaabbcccabbcacbbacababcacbca
bacbabcaabccbacbacbacbabcabcabcaacbcba

References

Class notes

Class handouts