Turing Machine Simulator

Design

CPTS 322

Ryan Wilson

April 11, 2014

Table of contents:

List of figures

Revision history:

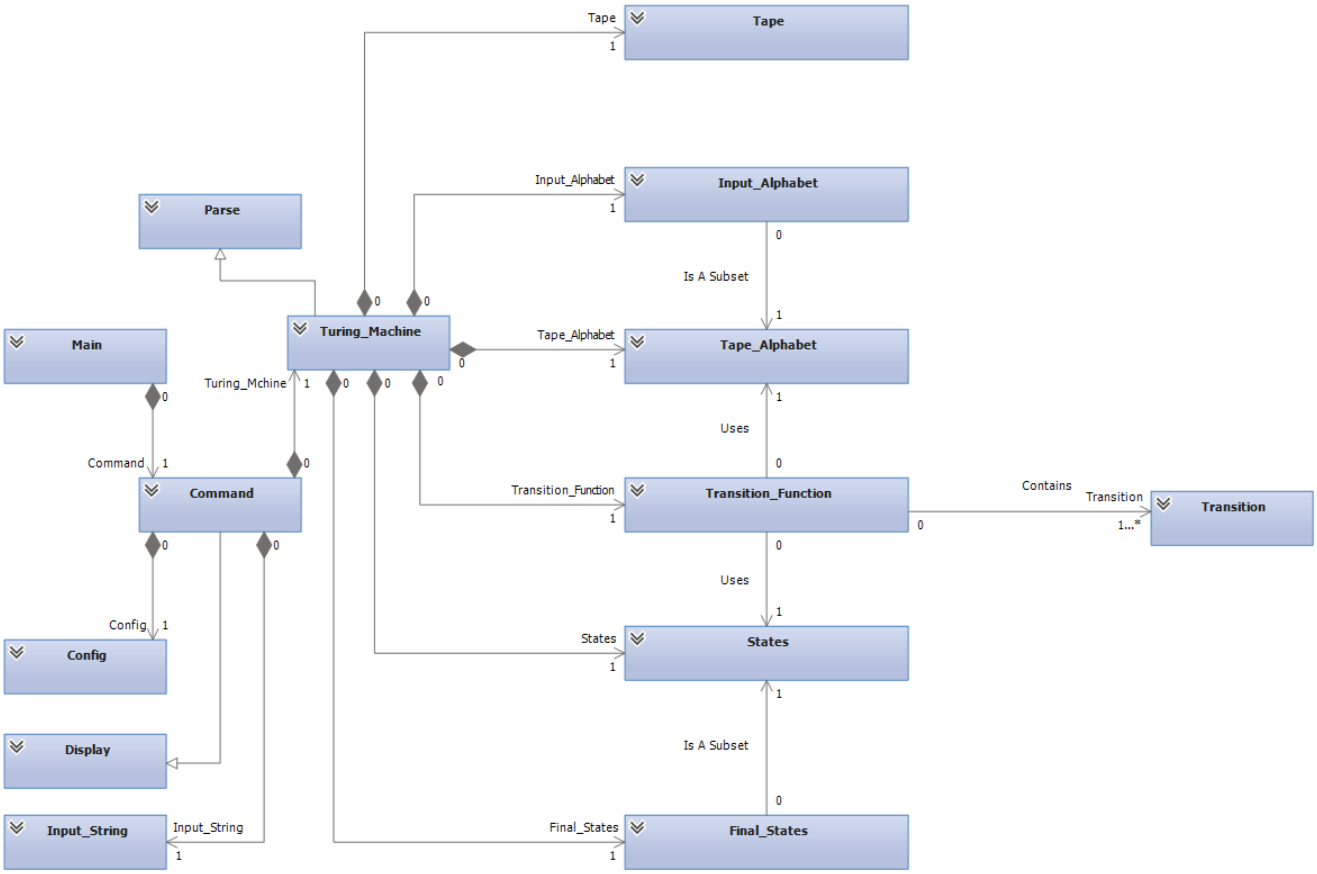
revision 5-4-14 added new classes to data dictionary to reflect changes that were made to application. Updated figures to reflect changes as well.

First draft no revisions 4-11-14

1.0 Introduction

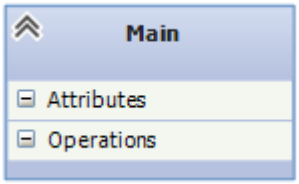
Design specifications for a Turing machine computer application intended for professionals in the theoretical studying of what a Turing machine is capable of. To know how this application is going to help, some background on what a Turing machine is and how it works. Followed by the purpose of the Turing machine application, hardware and software environment for development and installation. The environment will be clearly explained, as well as the operation of application. Finally reference and appendix.

2.0 Architecture



3.0 Data Dictionary

3.1 Main



Description:

Main is the starting point of the application. It will instantiate Turing_Machine, Command, and Configuration classes. It will also get the command line arguments for the name of the definition and input string file.

Associations:

The classes Turing_Machine, Command, and Configuration are components of Main constructing them.

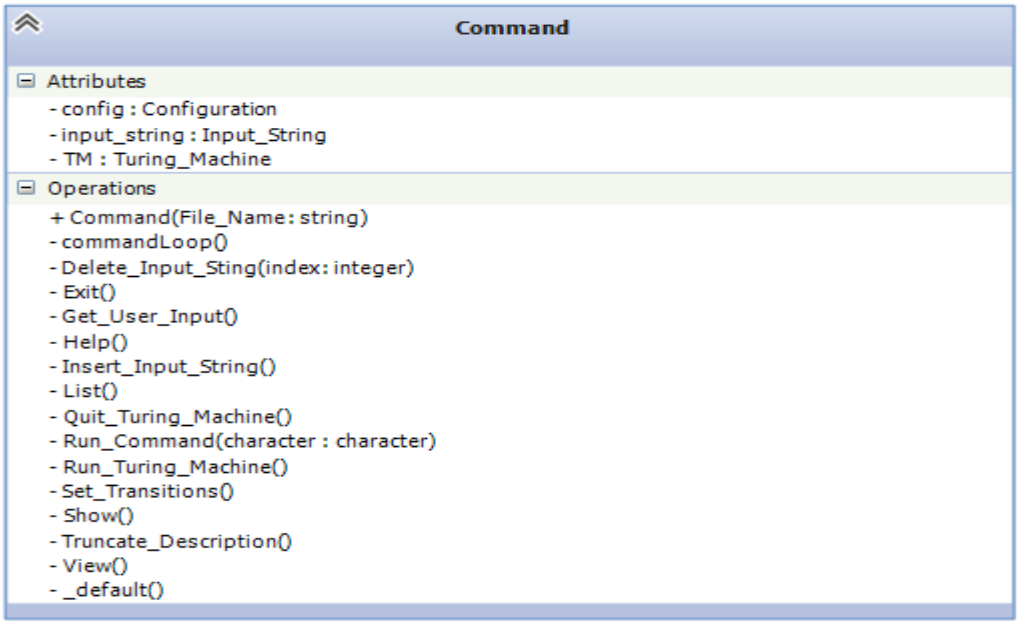
Attributes:

None

Methods:

None

3.2 Class Command



Class:

Command

Description:

The class Command contains all that is needed for the user interface.

Associations:

The class Command will be using an instantiation of a Turing_Machine and Configuration Class.

Attributes:

config : Configuration

The config is the instantiation of the configuration class used by this class.

input_string : Input_String

input_string is the instantiation of the Input_String class used by this class.

TM : Turing_Machine

TM is the instantiation of the Turing_Machine class used by this class.

Methods:

Command(File_Name : string)

This is a constructor for the class Command instantiating an instance of it as well as providing the file name for configuration class and Turing_Machine class.

_default()

This method is used to display a default message if the user inputs the wrong command.

Delete_Input_String()

The method Delete_Input_String will display a message to the user asking for the number associated with the string they wish to delete. It will then send a message to the class Input_String_List passing an integer representing the string to remove from the list.

Exit()

The method Exit will terminate operation of the application.

Get_User_Input() : Character

The method Get_User_Input() is used to get the user input, in the form of a character.

Help()

The method Help will send a message to a method in the class Configuration that will set one of it's attributes.

Insert_Input_String()

The method Insert_Input_String will bring up a new prompt waiting for a string. It will then send a message to the Input_String_List class adding that string to the list. If no string is entered then the prompt returns to Command:.

List()

The method List will send a message to Input_String_List class which will display a list of input strings.

Quit_Turing_Machine()

The method Quit_Turing_Machine will send a message to Turing_Machine class stopping operations on the input string.

Run_Command(character : character)

The method Run_Command receives a character from Get_Input method and runs one of the other methods.

Run_Turing_Machine()

The method Run_Turing_Machine sends a message to a method in Turing_Machine class performing operations on the input string.

Set_Transitions()

The method Set_Transitions opens a new prompt for the user to inter a number. It then sends a message to a method in Configuration class changing one of it's attributes. If no number is entered prompt returns to default.

Show()

The method Show sends a message to a methods in Turing_Machine class and Configuration class displaying information about the Application.

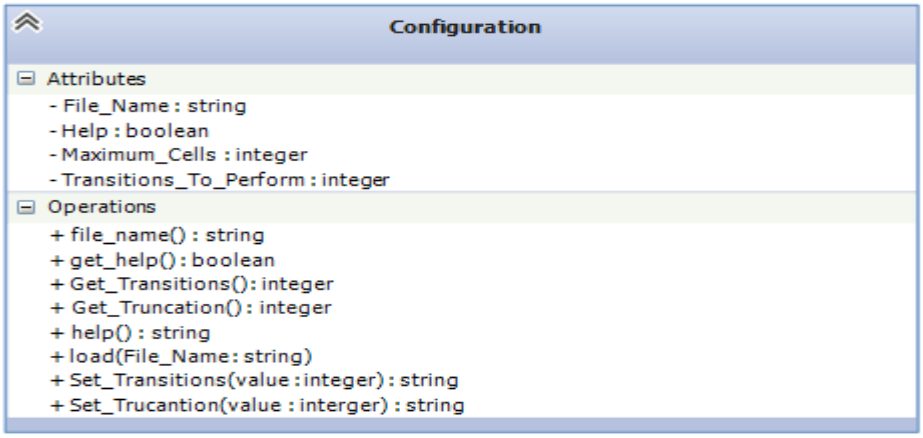
Truncate_Description()

The method Truncate_Description prompts for the user to enter a number. Then sends a message to a method in Configuration class changing one of it's attributes. If no number is entered returns to default prompt.

View()

The method View sends a message to a method in Turing_Machine class displaying information about the Turing Machine.

3.3 Class Configuration



Class:

Configuration

Description:

The class Configuration holds the configuration settings of the Turing Machine.

Associations:

The class Configuration is a used by the class Command to send and receive the configuration of the Turing machine.

Attributes:

File_Name : string

The attribute File_Name is the name entered from the command line. It's used to get the definition file and the input string file for the Turing Machine.

Help : Boolean = false

The attribute Help is a true false with the default set to false. This is used by other methods to check if the user needs extra information about a command.

Maximum_Cells : integer = 32

The attribute Maximum_Cells is the maximum number of characters to display to the left and right of the current character in the instantaneous description.

Transitions_To_Perform : integer = 1

The attribute Transitions_To_Perform is the maximum number of transition to perform on an input string.

Methods:

load(File_Name : string)

The method load sets the File_Name attribute to the string.

get_help() : Boolean

The method get_help returns the attribute Help.

Help() : string

The method Help() switches the Help attribute, and returns a message to display to the user that help is enabled or not.

file_name() : string

The method file_name returns the attribute File_Name.

Set_Transitions(value : integer) : string

The method Set_Transitions sets the Transitions_To_Perform attribute to the passed integer, and returns a message to display to the user saying it has change it to that number.

Get_Transitions() : integer

The method Get_Transitions returns the integer stored in the Transitions_To_Perform attribute.

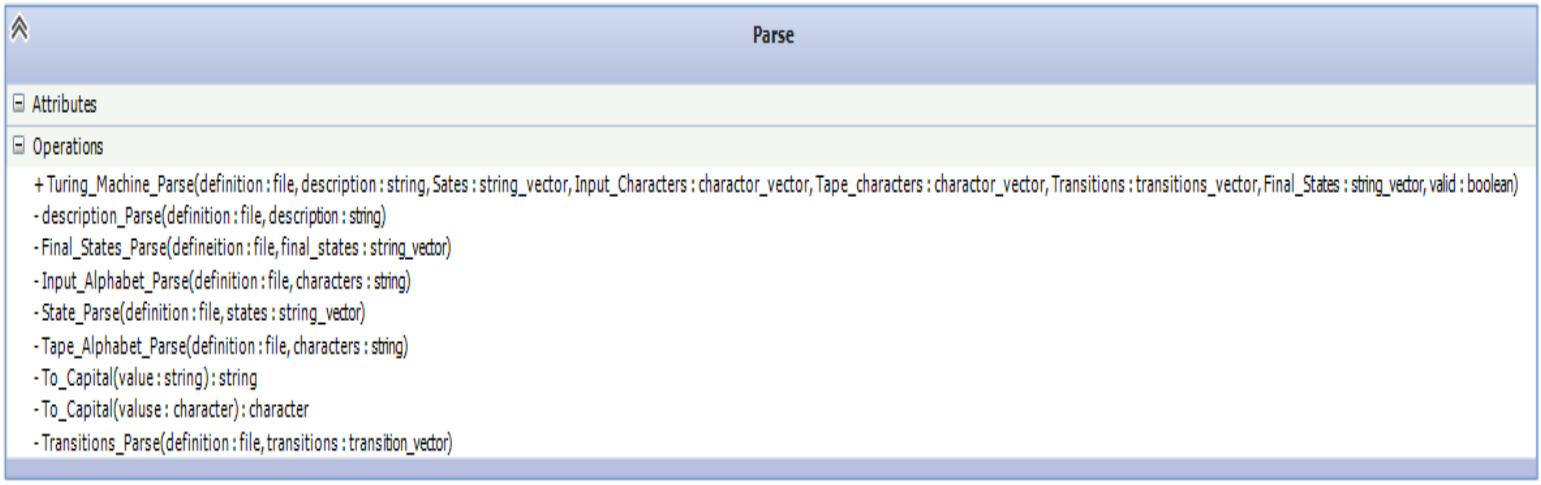
Set_Truncation(value : integer) : string

The method Set_Truncation sets the Maximum_Cells attribute to the passed integer, and returns a message to display to the user saying it has changed it to that number.

Get_Truncation() : integer

The method Get_Truncation returns the integer stored in the Maximum_Cells attribute.

3.4 Class Parse



Class:

Parse

Description:

The class Parse controls all the parsing of the files for this application.

Associations:

The class Parse is a base class of the class Turing_Machine.

Attributes:

Base class for Turing_Machine class.

Methods:

Description_Parse(definition: file, out description : string)

The method Description_Parse uses the open definition file and performs operations to get the description storing it in the description string.

Final_States_Parse(definition: file, out final_states : string_vector)

The method Final_States_Parse uses the open definition file and performs operations to get the final states storing them in the final_states string_vector.

Input_Alphabet_Parse(definition: file, out characters : character_vector)

The method Input_Alphabet_Parse uses the open definition file and performs operations to get the input alphabet storing them in the characters character_vector.

States_Parse(definition: file, out States : string_vector)

The method States_Parse uses the open definition file and performs operations to get the states storing them in the States string_vector.

Tape_Alphabet_Parse(definition: file, out characters : character_vector)

The method Tape_Alphabet_Parse uses the open definition file and performs operations to get the tape alphabet storing them in the characters character_vector.

Transition_Parse(definition: file, out transitions : transition_vector)

The method Transition_Parse uses the open definition file and performs operations to get the transitions storing them in the transitions transition_vector.

Turing_Machine_Parse(definition: file, out description : string,

out States : string_vector, out input_characters : character_vector,

out tape_characters : character_vector, out transitions : transition_vector,

out final_states : string_vector, out valid : boolean)

The method Turing_Machine_Parse runs the other methods in this class to successfully parse the definition file passed in to it. Any errors in parsing and the valid is set to false. Any errors during the parse process will display messages to the user.

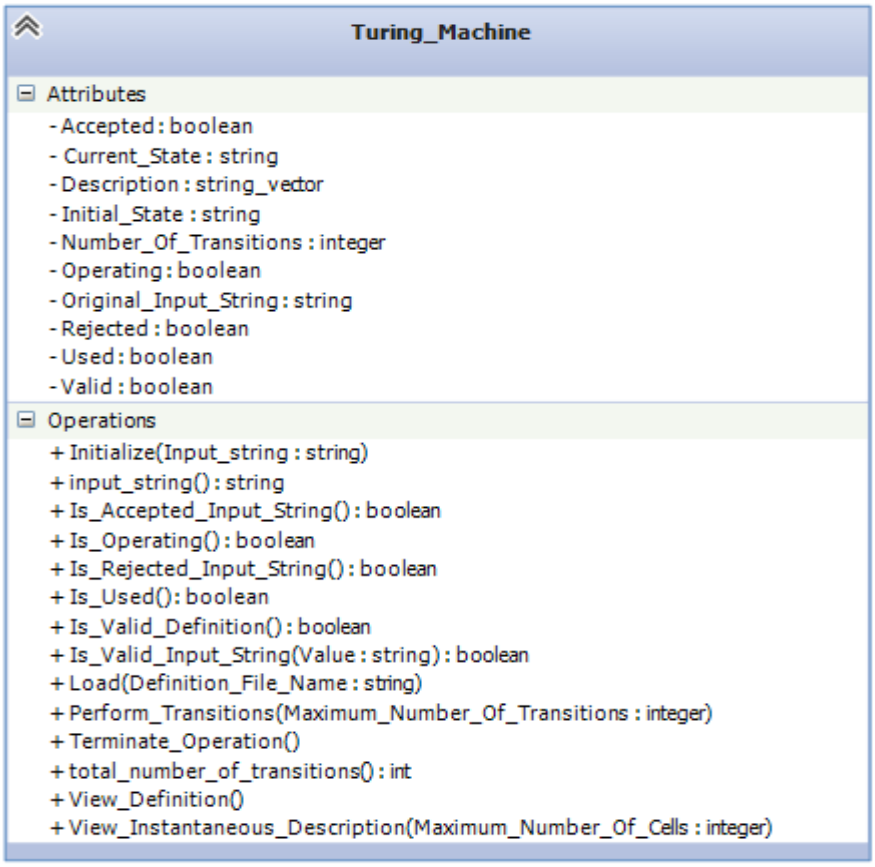
To_Capital(value : string) : string

The method To_Capital returns the string value in all uppercase letters.

To_Capital(value : character) : character

The method To_Capital returns the character value in uppercase letter.

3.5 Class Turing_Machine



Class:

Turing_Machine

Description:

The class Turing_Machine contains the core methods for the operations of the application.

Associations:

The class Turing_Machine is a component of the class Command, receiving messages delegated to it by the user interface.

Attributes:

Accepted : Boolean

The attribute Accepted is for if the Turing Machine has accepted the input string.

Current_State : string

The attribute Current_State holds the current state the Turing Machine is in for a input string.

Description : string

The attribute Description is passed to the Turing_Machine_Parse method which then fills it with the

description from the definition file.

Initial_State : string

The attribute Initial_State holds the initial state of the Turing Machine.

Number_Of_Transitions : integer

The attribute Number_Of_Transitions holds the number of transitions run on an input string.

Operating : Boolean

The attribute Operating will be true if the Turing Machine is operating on an input string, other wise false.

Original_Input_String : string

The attribute Original_Input_String will contain a copy of the input string passed to do operations on. This is because the input string may not be the same after performing operations on it.

Rejected : Boolean

The attribute Rejected will be true if the Turing Machine rejected the input string, other wise false.

Used : Boolean = false

The attribute Used is if the Turing Machine has ran an input string or not. Set to true once the Turing Machine runs an input string.

Valid : Boolean

The attribute Valid will be passed as an argument to the method Turing_Machine_Parse being set to false if any errors occur during parsing.

Methods:

Initialize(Input_String : string)

The method Initialize invokes the Initialize method of the class Tape.

Is_Accepted_Input_String() : Boolean

The method Is_Accepted_Input_String returns the Accepted attribute.

Is_Operating() : Boolean

The method Is_Operating returns the Operating attribute.

Is_Rejected_Input_String() : Boolean

The method Is_Rejected_Input_String return the Rejected attribute.

Is_Used() : Boolean

The method Is_Used returns the Used attribute.

Is_Valid_Definition() : Boolean

The method Is_Valid_Definition return the Valid attribute.

Is_Valid_Input_String(value : string) : Boolean

The method Is_Valid_Input_String returns true if the value is a valid input string, otherwise false.

Perform_Transition(Maximum_Number_Of_Transitions : integer)

The method Perform_Transition will begin performing transitions up to Maximum_Number_Of_Transitions.

Terminate_Operation()

The method Terminate_Operation will halt operations on the input string.

Load(Definition_File_Name : string)

The method Load uses the passed string to load the definition file.

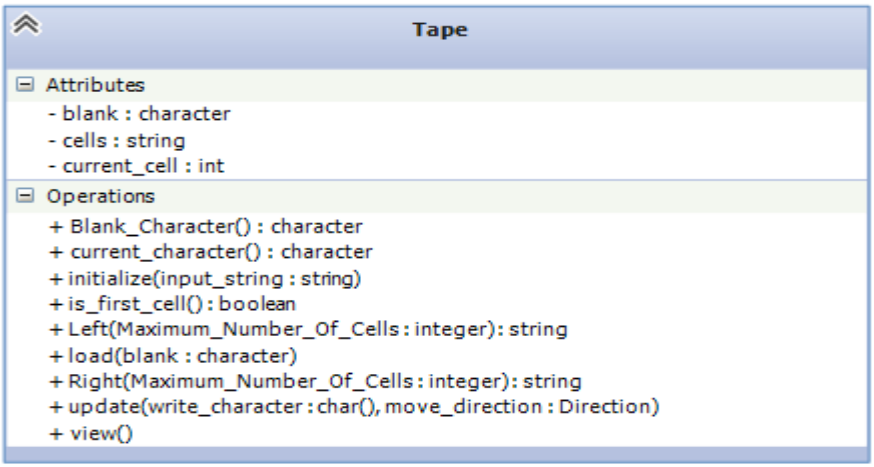
View_Definition():

The method View_Definition will display the definition attribute to the monitor.

View_Instantaneous_Description(Maximum_Number_Of_Cells : integer)

The method View_Instantaneous_Description will display the instantaneous description to the monitor up to the Maximum_Number_Of_Cells.

3.6 Class Tape



Class:

Tape

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 number 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 a part of a Turing machine.

Associations:

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

Attributes:

Cells : string = “”

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

Current_Cell : integer = 0

The index of the current cell on the Turing machine tape is stored in the attribute Current_Cell.

Blank :character = "

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

Methods:

Current_Character() : Character

The method Current_Character returns the character of the current cell of the Turing machine.

Initialize(Input_String : string)

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

Is_First_Cell() : Boolean

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

load(Blank : character)

The method load will set any default values and the Blank attribute to Blank.

Update(Write_Character : character, 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 fist cell. If a right move is specified from the last cell , a blank character is appended to the tape. If no storage is available for this character, an out of storage error will be thrown. Assuming that the update may be performed, the character to write on the tape is stored in the current 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.

```
Method update(Write_Character: character, Move_Direection : direction) is
begin
    if Move_Direction = L and Current_Cell = 0
        then return;
    end if
    if Move_Direction = R and Current_Cell = Cells.length() - 1 then
        Cells.append(blank);
    end if
    Cells[Current_Cell] := Write_Character;
    if Move_Direction = L then
        Current_Cell := Current_Cell – 1;
    else
        Current_Cell := Current_Cell + 1;
    end if;
end update;
```

Left(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 that 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.

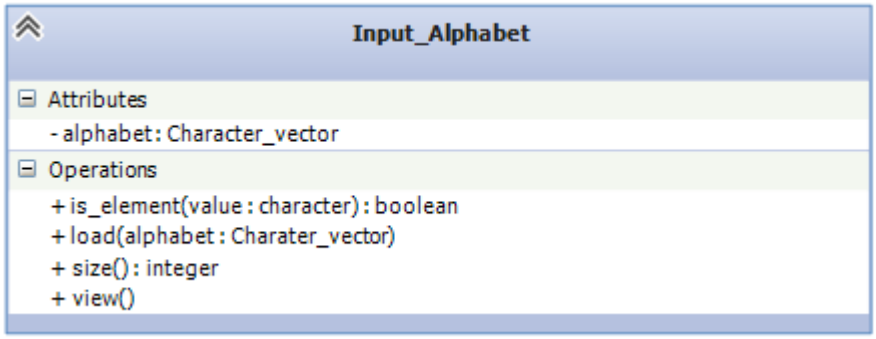
Right(Maximum_Number_Of_Cells : integer) : string

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 that 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 end of the string.

View()

The method View displays the blank character of the Turing machine.

3.7 Class Input_Alphabet



Class:

Input_Alphabet

Description:

The class Input_Alphabet consists of the input alphabet of the Turing machine. Input alphabet exists only as part of a Turing machine.

Associations:

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

Attributes:

Alphabet : character_vector

The input alphabet of the Turing machine is stored in the attribute Alphabet.

Methods:

load(alphabet : character_vector)

The method load will set the alphabet attribute with the passed character_vector alphabet.

Is_Element(value : character) : Boolean

The method Is_Element returns a value of true if the parameter value is an element of the alphabet.

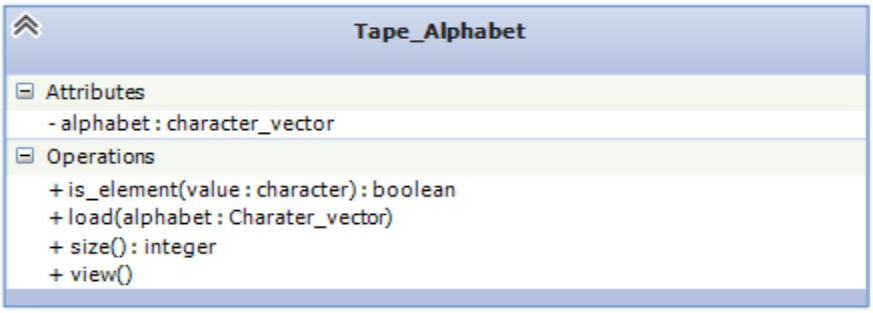
Size() : integer

The method Size returns the number of elements in alphabet.

View()

The method View displays the input alphabet of the Turing machine.

3.8 Class Tape_Alphabet



Class:

Tape_Alphabet

Description:

The class Tape_Alphabet contains the tape alphabet of the Turing machine. The tape alphabet exists only as a part of a Turing machine.

Associations:

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

Attributes:

Alphabet : character_vector

The tape alphabet of the Turing machine is stored in the attribute Alphabet.

Methods:

load(alphabet : character_vector)

The method load will set the alphabet attribute with the passed character_vector alphabet.

Is_Element(value : character) : Boolean

The method Is_Element returns a value of true if the parameter value is an element of the alphabet.

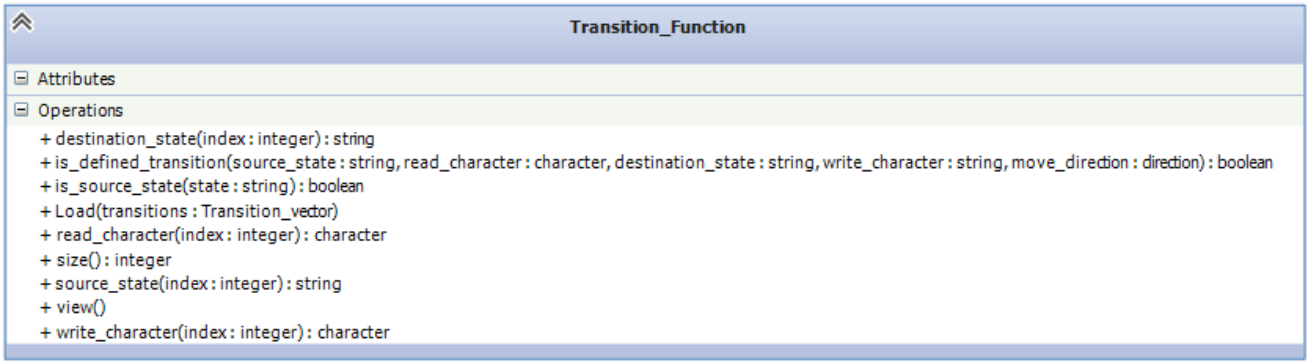
Size() : integer

The method Size returns the number of elements in alphabet.

View()

The method View displays the tape alphabet of the Turing machine.

3.9 Class Transition_Function



Class:

Transition_Function

Description:

The transition function of a Turing machine tell what to do with an input character and a state.

Associations:

The class Transition_Function contains a vector of transitions and is a component of the class Turing_Machine, receiving messages delegated to it by the Turing machine.

Attributes:

None

Methods:

Destination_State(index : integer) : string

The method Destination_State returns the state that is in a transition indicated by index.

Is_Defined_Transition(Source_State : string, Read_Character: character,
out Destination_State : string, out write_Character : string,
out Move_Direction : Direction) : Boolean

The method Is_Defined_Transition returns a value true if the source state and read character are in the transition then sets the destination state, write character, and move direction to what is stored in the transition, otherwise it returns a value of false.

Is_Source_State(Source_State : string) : Boolean

The method Is_Source_State returns a value of true if the source state parameter is a source state in the transition, otherwise it returns a value of false.

Read_Character(index : integer) : character

The method Read_Character returns the character that is in a transition indicated by index.

Size() : integer

The method Size returns the number of transitions of the Turing machine.

Source_State(index : integer) : string

The method Source_State returns the state that is in a transition indicated by index.

Load(Transitions : transition_vector)

The method load populates the transitions contained with in.

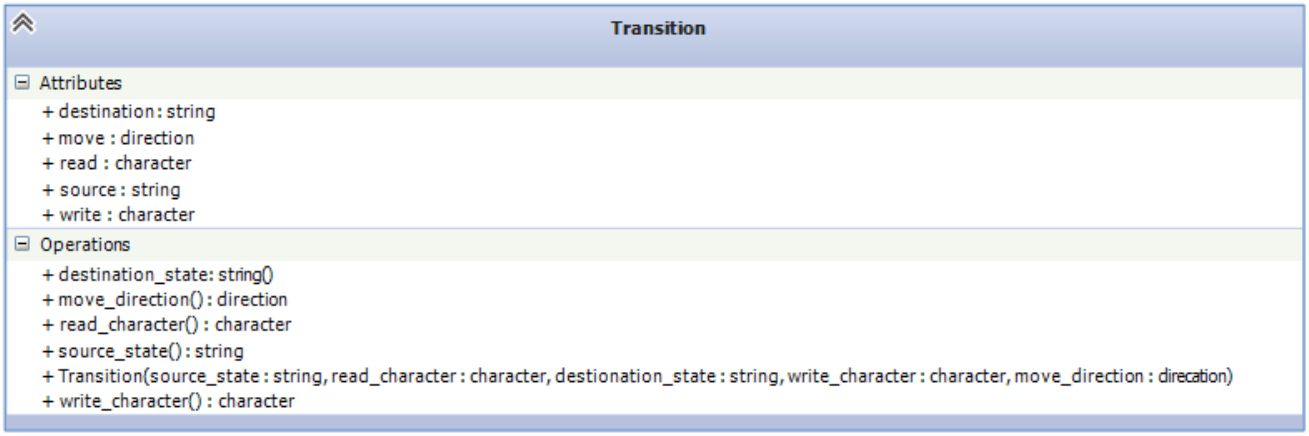
View()

The method View displays a list of information about all transitions contained with in.

Write_Character(index : integer) : character

The method Write_Character return the character that is in a transition indicated by index.

3.10 Class Transition



Class:

Transition

Description:

A transition is how a Turing machine decides what to do with the tape head, and what state to move to. Transition exist only as part of a transition function.

Associations:

The class Transition_Function contains a collection of transition class.

Attributes:

Destination : string

The destination state of a transition is stored in the attribute Destination.

Move : Direction

The direction of a transition to move the tape head is stored in the attribute Move.

Read : Character

The character to read from the current cell of the tape is stored in the attribute Read.

Source : string

The source state of a transition is stored in the attribute Source.

Write : character

The character to write to the current cell of the tape is stored in the attribute Write.

Methods:

Destination_State() : String

The method Destination_State returns the attribute Destination.

Move_Direction() : Direction

The method Move_Direction return the attribute Move.

Read_Character() : character

The method Read_Character returns the attribute Read.

Source_State() : string

The method Source_State returns the attribute Source.

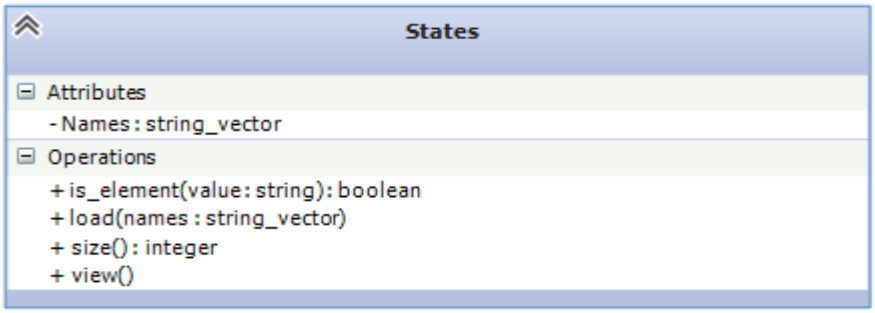
Transition(source_state : string, read_character : character, destination_state : string,
write_character : character, move_direction : Direction)

This is the constructor for the class Transition, it creates an instance of the class setting all the attributes.

Write_Character() : character

The method Write_Character returns the attribute Write.

3.11 Class States



Class:

States

Description:

The states of a Turing machine determine what to do with a particular input character. States exist only as part of a Turing machine.

Associations:

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

Attributes:

Names : string_vector

A collection of states in the form of strings is stored in the attribute Names.

Methods:

Is_Element(value : string) : Boolean

The method Is_Element returns a value of true if the parameter value is a state in the attribute Names, otherwise it returns a value of false.

load(names : string_vector)

The method load sets the attribute names.

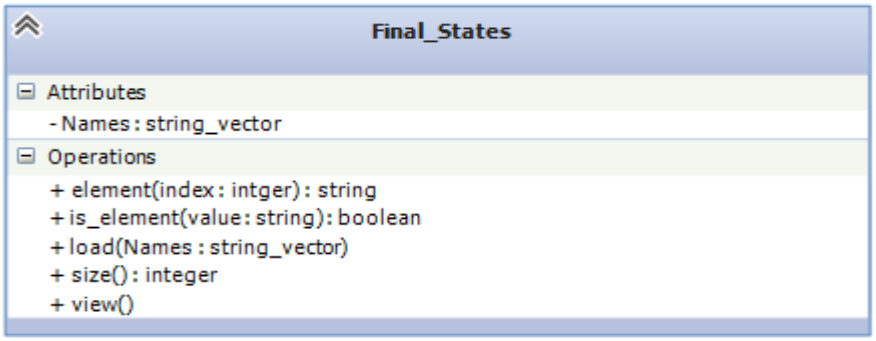
View()

The method View displays a list of all the states stored in the attribute Names.

Size() : integer

The method size returns the number of elements in the string_vector Names.

3.12 Class Final_States



Class:

Final_States

Description:

The final states of a Turing machine are the set of states that determine if the input string is accepted. Final states exist only as part of a Turing machine.

Associations:

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

Attributes:

Names : string_vector

A collection of final states in the form of strings is stored in the attribute Names.

Methods:

Element(index : integer) : string

The method Element returns the final state determined by index from the attribute Name.

Is_Element(value : string) : Boolean

The method Is_Element returns a value of true if the parameter value is a final state in the attribute Names, otherwise it returns a value of false.

load(names : string_vector)

The method load sets the attribute Names.

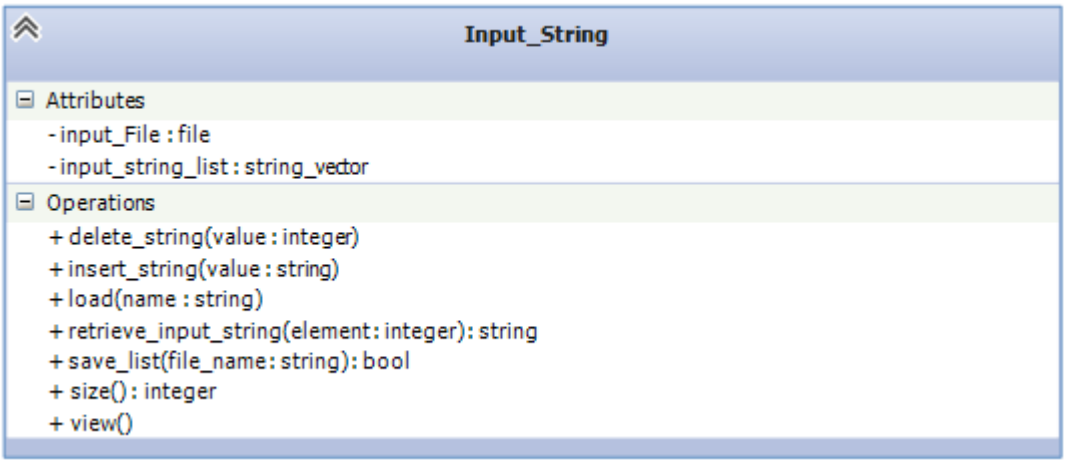
Size() : integer

The method size return the number of elements in Names.

View()

The method View displays a list of all the states stored in the attribute Names.

3.13 Class Input_String



Class:

Input_String

Description:

The Input_string class stores the list of strings to use on the Turing machine as well as the methods used to operate on them.

Associations:

The input_string class is used by the command class to get and send messages to and from the Turing machine class. Also inherits from the parse class.

Attributes:

input_File : file

The attribute input_File is used to open the file and store the list of strings to it.

input_string_list : string_vector

The attribute input_string_list is used to store the list of input strings used by the Turing machine.

Methods:

delete_string(value : integer)

The method delete_string is used to delete the string in the list associated with value.

insert_string(value : string)

The method insert_string is used to insert a string into the list.

load(name : string)

The method load uses the string passed in to open the input string file and load all strings from it.

retrieve_input_string(element : integer) : string

The method retrieve_input_string returns the string at the index of element.

save_list(file_name : string) : Boolean

The method save_list saves the list to file and returns true if successful.

Size() : integer

The method size returns the number of elements in input_string_list.

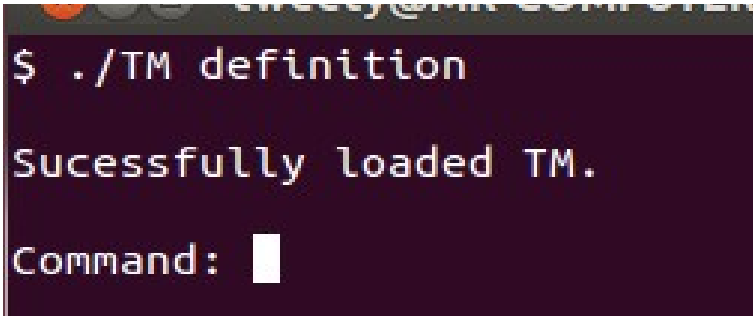
View()

The method view displays the elements of input_string_list.

4.0 User Interface

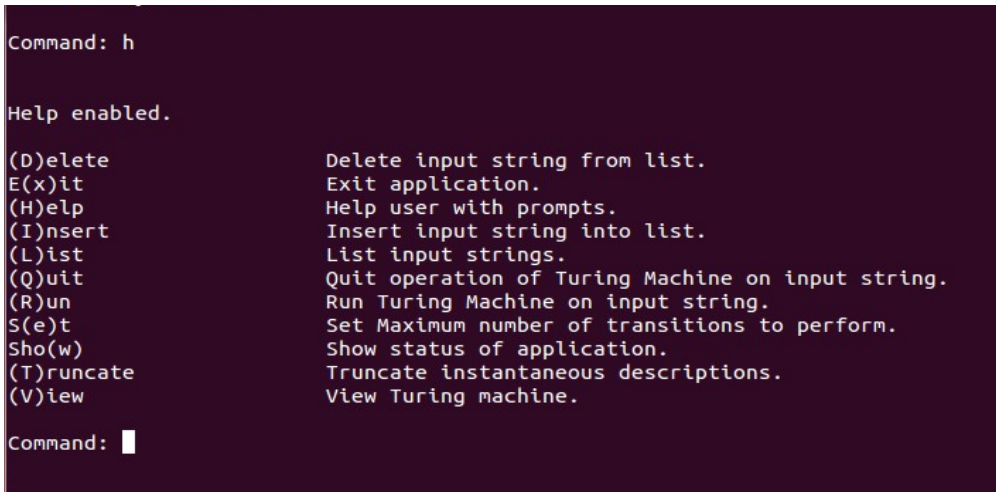
4.1 Command Line Invocation

The user will type the name of the application followed by the name of the definition file not including the .def extension.



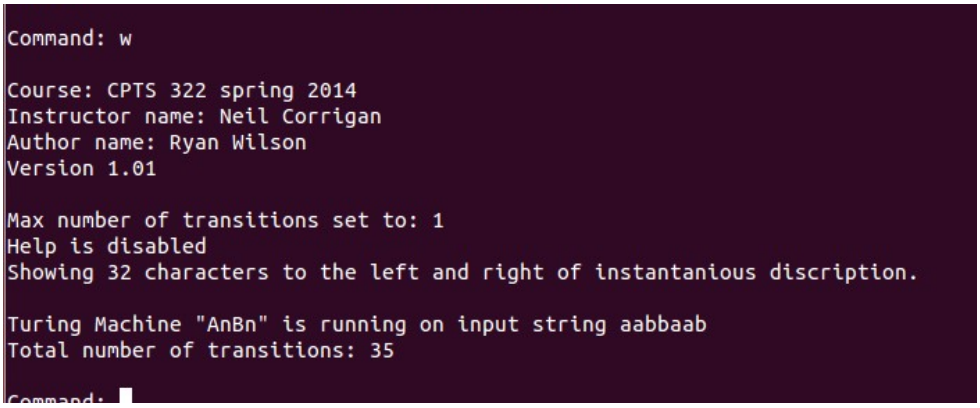
4.2 Help Command

This command enables help for the user.



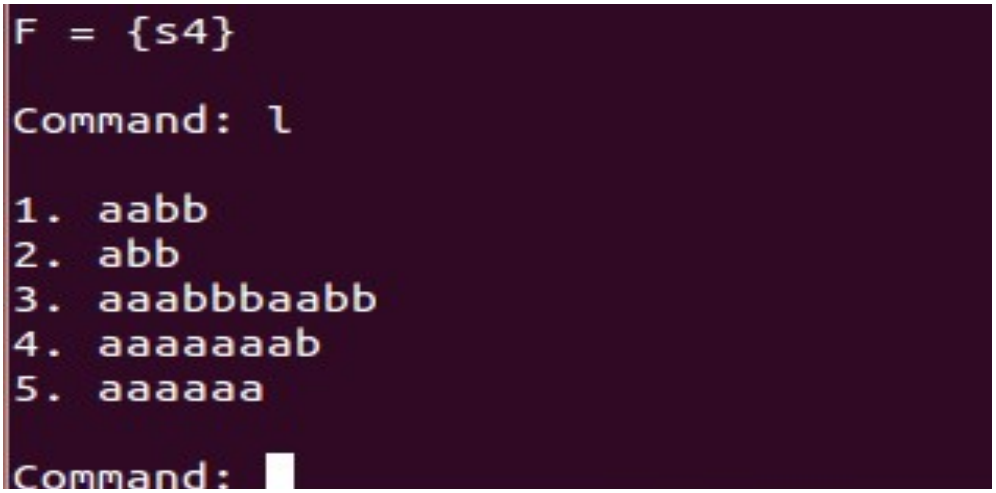
4.3 Show Command

This command shows information about the application.



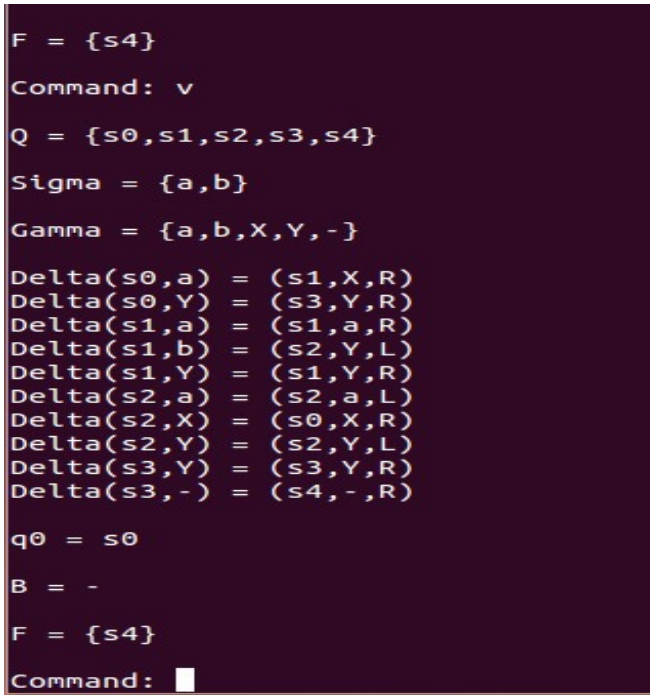
4.4 List Command

This command lists the input strings in the Turing machine.



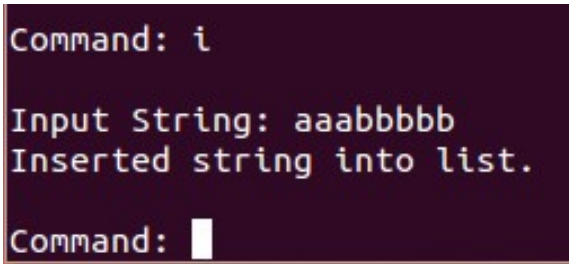
4.5 View Command

This command displays information about the Turing machine.



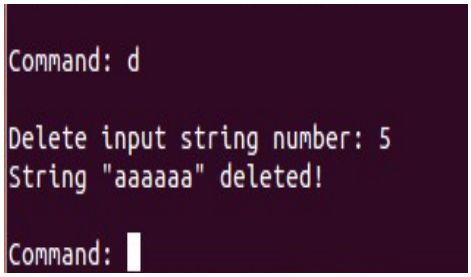
4.6 Insert Command

This command prompts for user to input a string then inserts it into the list.



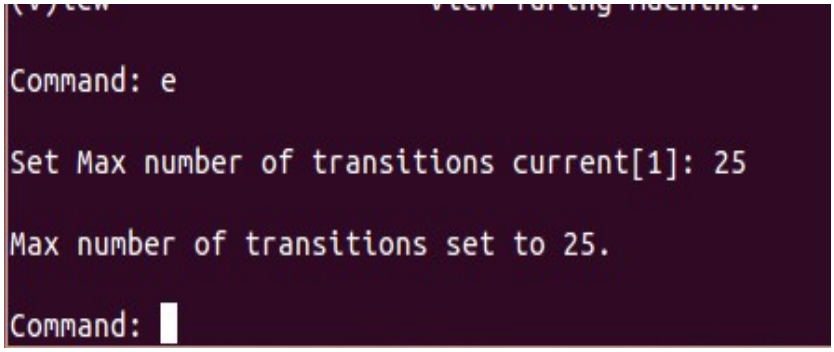
4.7 Delete Command

This command prompts the user for a number that represents th string they wish to delete.



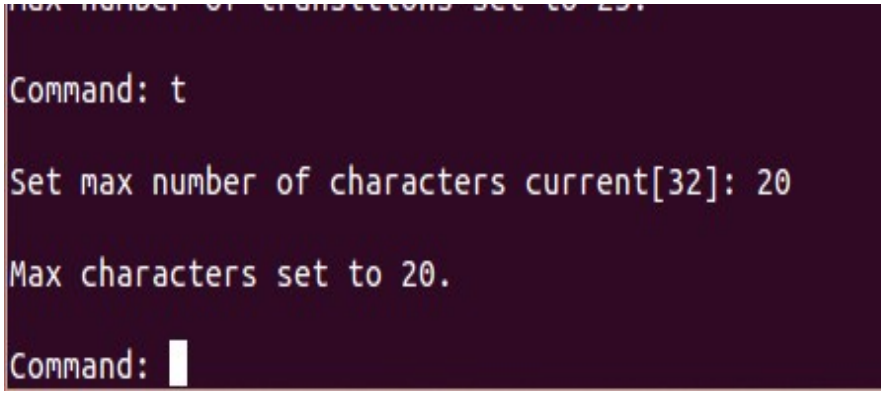
4.8 Set Command

This command prompts the user for a the number of transitions to perform.



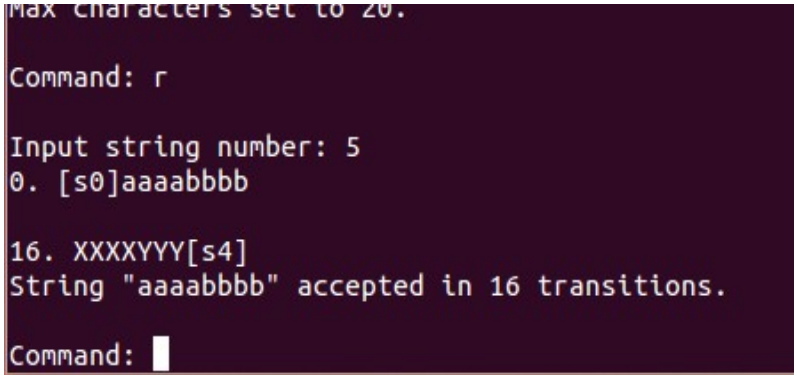
4.9 Truncate Command

This command prompts the user for the number of maximum cells to display to the left and right of the instantaneous description.



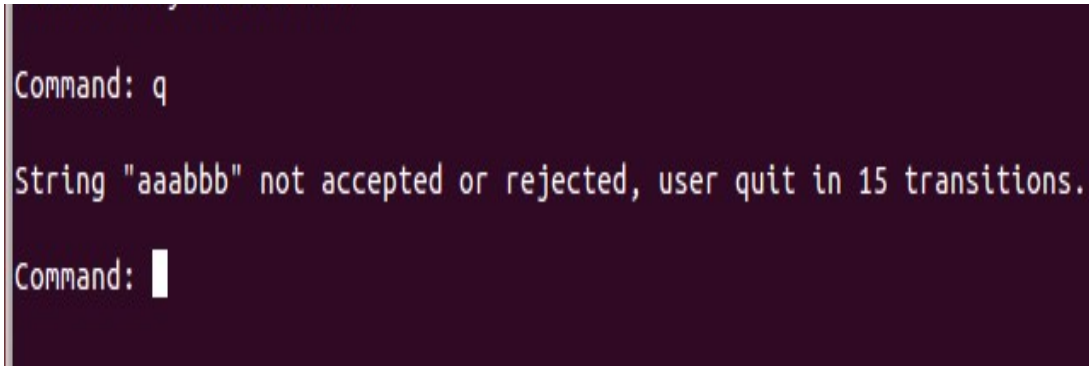
4.10 Run Command

This command if Turing machine not currently running on a string, will prompt for the user to enter a number representing the string they wish to run. If it is running it will run the Turing machine up to the maximum number of transitions set in the configuration.



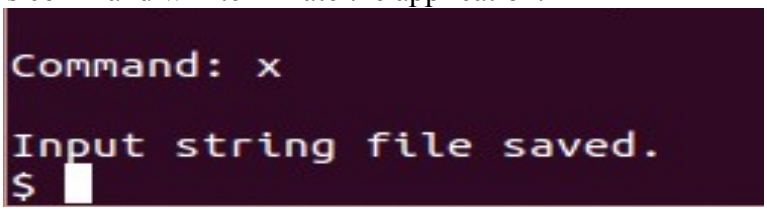
4.11 Quit Commands

This command stops operations on an input string.



4.12 Exit Command

This command will terminate the application.



5.0 Files

To files are used by this application, the definition file denoted by the extension .def and the input string list file denoted by the extension .str. The input string list file is optional, but the definition file is require.

5.1 Definittion File

tm.def

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 s0 X R

s2 Y s3 Y R

s3 Y s3 Y R

s3 - s4 – R

INITIAL_STATE: s0

BLANK_CHARACTER: -

FINAL_STATES: s4

5.2 Input String File

tm.str

a

aaabbb

ababab

\

bbaa

cds

\\

\ab

aaabbb

a a abbb

References

Appendix