**Turing Machine Lab 1/31/2018 Hudson J Whittaker**

Problem Summary:

Create a program to simulate a Turing Machine.

Properties

- List of instructions (five numbers that define tape)

- Tape (can move in either direction even past the length of the tape, can read and set current value)

- State (starts at zero, changed by instructions)

Problem Requirements:

* Input can come from standard in or file specified at command line
* Tape is at most 32767 characters long
* Can hold at most 100 instructions
* Tape has no spaces between numbers
* Tape has only positive integers
* Instructions contain spaces between integers
* Outputs final tape and final state

System Design:

**Lab1 Class**

Purpose: get filename and run program

Methods:

Main()

Ask user for file name

Give that to TuringMachine through the run() method

**TurMac Class**

purpose: do what it is told

field variables:

tape : StringLog of numbers

state : integer

instructions : object

Methods:

Constructors

public turMac()

sets state to 0, tells Tape class to start at pos 0, gets tape to set tape

public turMac(int state, int pos, int tape)

Observers

public int getTape()

calls Tape class for current tape value

public int getPos()

calls Tape class for current tape position

public int getState()

calls Instructions class for current state

public int getInstr()

Transformers

public void setState(int state)

get state from instructions

public void replaceTape(int replace)

calls Tape class to change current tape value

public void moveTape(int pos)

calls Tape class to change tape position

**Instructions** Class

purpose: find set of instructions based off of the state and tape; return new state, new tape, tape movement

field variables:

curState : int

curTape : int

cups : int

Methods:

Constructors

public instructions(int position, int tape, int state)

Observers

public int getTape()

public int getState()

public int getInstr(int state, int tape)

finds correct set of instructions based on the given state and tape

public int getPos()

Transformers

public void setState(int state)

uses getInstr(int state, int tape) to find new state to set (3rd column)

public void setTape(int tape)

uses getInstr(int state, int tape) to find new tape to set

(4th column)

public void setPos(int pos)

uses getInstr(int state, int tape) to find new position to set (5th column)

**Instruction Class**

Purpose: make Instructions object using String input

Field variables:

curState, curVal, newState, newVal, direction: integer

Methods:

Constructor

Public Instruction(String instructions)

Scans through string and sets each number to respective variable

Observers

Equals(int state, int val)

getState()

getVal()

getNewState()

getNewVal()

getDir()

Transformer

toString()

**Tape Class**

purpose: contain tape array and control tape position

field variables:

position: int

tape: array of int

Observers

public int getVal(int pos)

needs a requested address to access correct element

public int getPos()

Transformers

public void setVal(int val)

will set the current tape[] address to the given value

public void setPos(int pos)

will change tape[] current position by 1 based on the given value

Testing Plan:

**Trivial Test Cases**

Set/getState

Reject state and break while loop if < 0

Set/getVal of tape

Change pos of tape and check new pos

Send tape dir = 0 and check that it doesn’t move

**Simple Test Cases**

|  |
| --- |
| Case 1: Simple. Read in value and go to accepting state |
| Input: |
| 0 1 1 1 |
| 0 0 -1 0 0 |
| Output: |
| 0111 and accepting |
| Case 2: Simple. Read in value, change and go to accepting state |
| Input: |
| 0 1 1 1 |
| 0 0 -1 1 0 |
| Output: |
| 1111 and accepting |

**Standard Test Case**

|  |
| --- |
| Case 3: Standard. Read in value, change state, change tape, repeat |
| Input: |
| "111110010101011000111101" |
| 0 1 1 0 1 |
| 0 0 -2 0 0 |
| 1 1 2 0 1 |
| 1 0 -2 0 0 |
| 2 1 3 0 1 |
| 2 0 -2 0 0 |
| 3 1 4 0 1 |
| 3 0 -2 0 0 |
| 4 1 5 0 1 |
| 4 0 -2 0 0 |
| 5 1 6 1 -1 |
| 5 0 6 0 -1 |
| 6 0 7 0 -1 |
| 7 0 -1 1 0 |
| Output: |
| "000100010101011000111101" and accepting |

**Error Test Case**

|  |
| --- |
| Case 1: Error. Tells user that negative tape cannot be entered and changes it to positive. |
| Input: |
| 0 1 -1 1 |
| 0 0 -1 0 0 |
| Output: |
| 0111 and accepting |

Management Plan:

Expected:

Tape = 1 hour

Instruction + Instructions = 2 hours

TuringMachine = 2 hours

Lab1 = 1 hour

Debug and test = 1 hour

Write SDR = 1 hour

Result:

Tape = 3 hours

Instruction + Instructions = 2 hours

TuringMachine = 4 hours

Lab1 = 2 hours

Debug and test = 1 hour

Write SDR = 1 hour

Conclusion:

Total Expected (8) / Total Result (13)

= about 62% more time than expected