Module number: CS3052
Matriculation number: 200017941

Practical 1 – Turing Machines

Overview

For this practical we were required to build a Turing Machine simulator and a few Turing Machine designs. This practical is split into three parts: creating the simulator, designing Turing Machines and analyzing the time complexity of the simulator. I have only managed to do the first two parts so this report will only be focused on that.

Design

When it comes to the design choices, I have made use of the List data structure to store the states and symbols from the files, each in their own list. This is because lists offer several convenient functions which make the program more convenient such as the contains() function which can be used as part of the check to see whether the values in the list are valid.

Implementation

All the code is in the runtm.java file where it is split into two main parts. The first part is for checking to make sure that the tape and configuration files are both valid in terms of their syntax. This is done through a series of checks for each line where the said rules were followed. The second part is the actual operation of the Turing Machine. Two loops were used: a while loop followed by a for loop. The first one is used to make sure that the machine operates until there are no more transitions possible. The second is used to loop through all the transitions to see which ones match. Within this loop the machine operates simply by storing the current state and symbol in a variable and checking each of the possible transitions, then applying them. There are two conditionals which follow from this: one for if the current state is a accept state and another for if it is a reject state. These cause the program to end and the relevant values to be displayed.

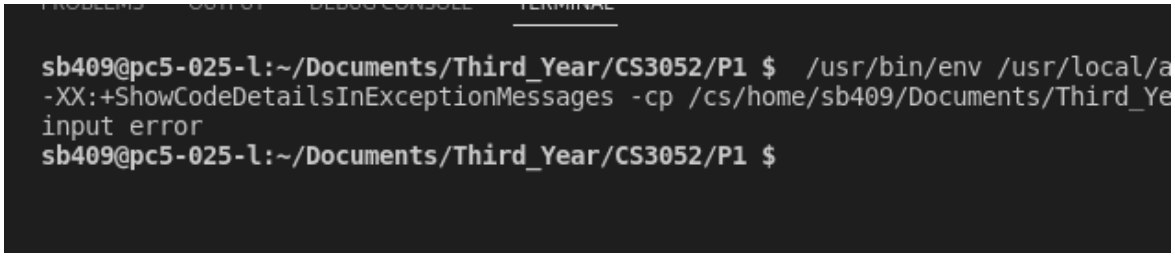
Testing

The initial testing phase involved running independent tests as I was developing the program. I chose a simple Turing Machine configuration and tape for this and made sure it simulated the Turing Machine correctly. Following this, after completely developing the program, I ran the tests from stacscheck. For some reason I was having issues with this as my program was not passing tests which it should be passing. So I verified this by running the tests from stacscheck independent. Most of the tests were successfully passed when I

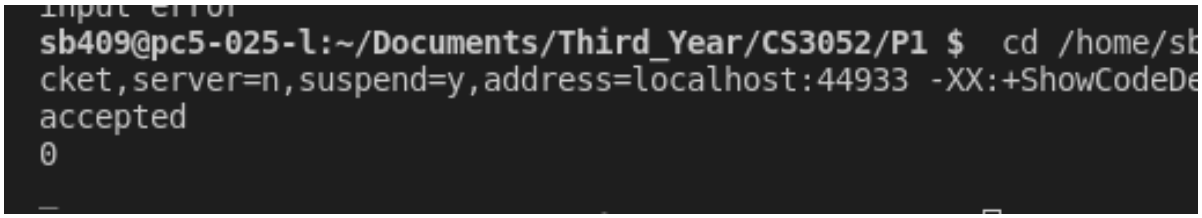
ran them independently, and I don't understand why that is the case. The tests below are all copies of the tests from stacscheck and are used to prove this claim.

Tests:

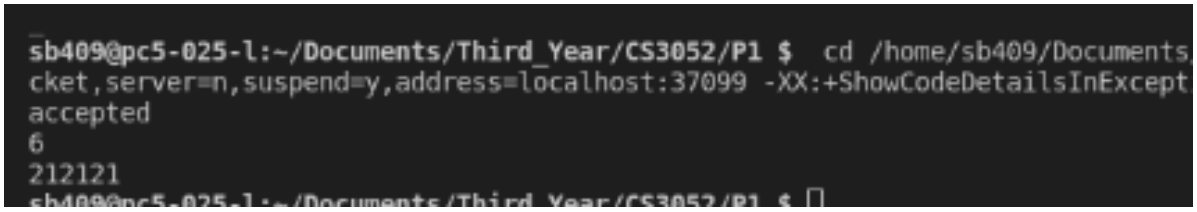
- bad.tape:



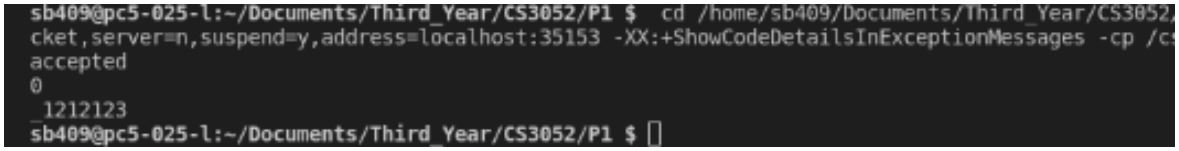
- blank.tape:



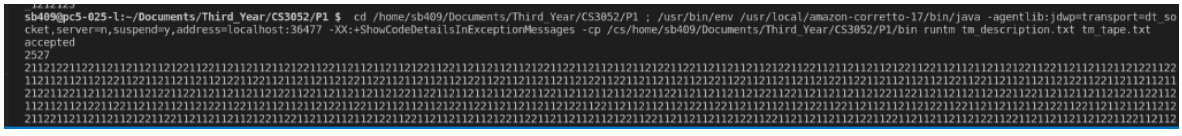
- left.tape:



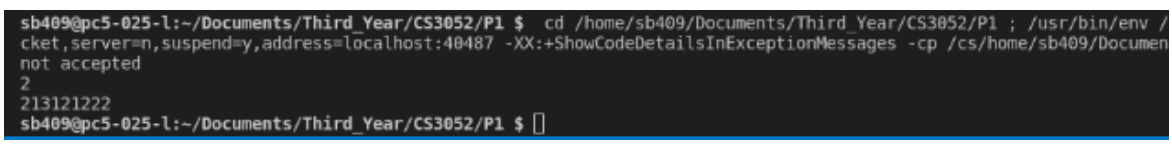
- left2.tape:



- long.tape:



- middle.tape:



- white.tape:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Y
cket,server=n,suspend=y,address=localhost:41465 -XX:+ShowCodeDetailsInExceptionMessa
accepted
12
212121121212
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badalphasize1.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badalphasize2.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badalphasize3.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badalphasize4.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badstateno1.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badstateno2.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

- badstatenp3.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

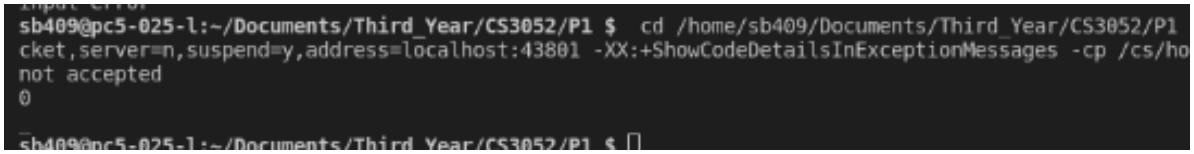
- badstateno4.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

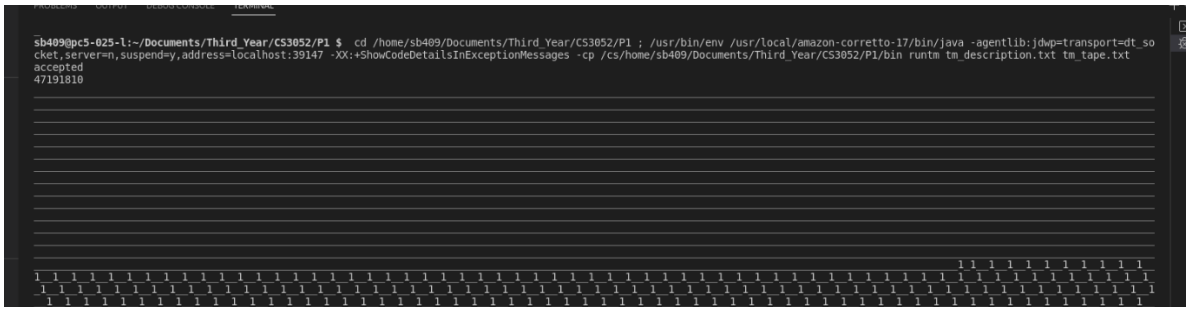
- badtrans1.tm:

```
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $  cd /home/sb409/Documents/Third_Year/CS3052/P1 ; /usr/bin
cket,server=n,suspend=y,address=localhost:35765 -XX:+ShowCodeDetailsInExceptionMessages -cp /cs/home/sb409/D
input error
sb409@pc5-025-l:~/Documents/Third_Year/CS3052/P1 $
```

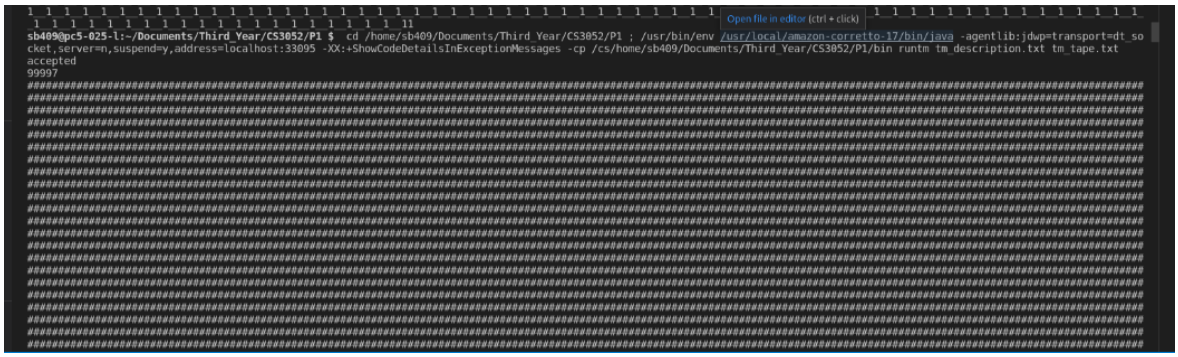
- accept.tm:



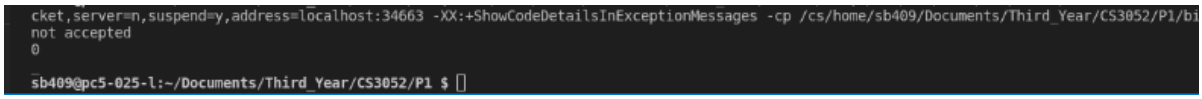
- busy5.tape and busy5.tm:



- manystates.tm:



- manysymols.tm:



However, there are some tests which I wasn't able to pass for example: manytrans.tm and manytrans2.tm. For these tests my program just didn't produce any output and just kept running.

Evaluation

For the most part, from all the testing that I did, my simulator seems to be working fine. But, performance when it comes to larger and more complex Turing Machines could be better as my program seems to be taking a bit too long.

Conclusion

Overall, I managed to program a simulator which performs like a Turing Machine. I found it a bit of a challenge to implement all the required rules since there were so many. Careful consideration had to be made when testing to verify whether the simulator performs as intended as there are several different possible cases which could occur.