GROUP PROJECT REPORT

Program: Implement a Universal FA

Team: Javaccinos

Team Members: Phung K Ly, Shane Thomas, Donald Vu

Class: CS 3110.E01

I. Data Structure and Program design

Our group use Java to design this program. In this program, there are two main parts. First is reading all inputs (numbers of state, final states, alphabets, and transitions table) through an input file. We use a Scanner to read the information from a file and use arrays to store final states, alphabets, and transitions. That is all for the reading input file.

Next, the most important part is designing a universal FA. We store all the input testing strings into an array called input array. Then, we start to check each element from the input array to tell whether is accepted or not. For each element, we will check each character following the algorithm in the instruction. If the element contains a character that does not belong to the alphabet array, it will reject and come to the next element. Otherwise, we will check each character of the array and check if the array ends in the final state or not to determine that element is accepted or rejected

II. 10 Additional Testing Data Sets

FA1

abd1
101110000
1000
po920
10101010
1010
11111
100001
1010000
1110101

FA2

bw10
110001
10001110
1010
1000
0
11110000
010100
110100101110
10119

FA3
Abcdi093
29434jsf
osw45
093402
shfdncdk24
0AR345
2djsnA
Java12
Studio23
Universal12

FA4
hello!
1 1
10101010101010
1230000000
45!
.1
 100
 0
80932141079124709
1000000000000000000

III. Test String results

FA1

^	Reject
100	Accept
011	Reject
10abc1	Reject
0	Accept
1	Reject
0101011	Reject
11010	Accept
0001	Reject
1110	Accept
abd1	Reject
101110000	Accept
1000	Accept
po920	Reject
10101010	Accept
1010	Accept
11111	Reject

100001	Reject
1010000	Accept
1110101	Reject

FA2

^	Reject
1	Accept
000	Accept
101	Accept
111	Reject
01001	Accept
1011011	Reject
1011000	Reject
01010	Accept
1010101110	Reject
bw10	Reject
110001	Reject
10001110	Reject
1010	Accept
1000	Accept
0	Accept
11110000	Reject
010100	Accept
11110000	Reject
1110101	Reject

FA3

^	Reject
HelloWorld	Accept
abc	Accept
1st_Ex	Reject
Java	Accept
_finite_automaton	Reject
program	Accept
X3Y7	Accept
X=90	Reject
X*Y	Reject
Abcdi093	Accept
29434jsf	Reject
osw45	Accept
093402	Reject
shfdncdk24	Accept
0AR345	Reject

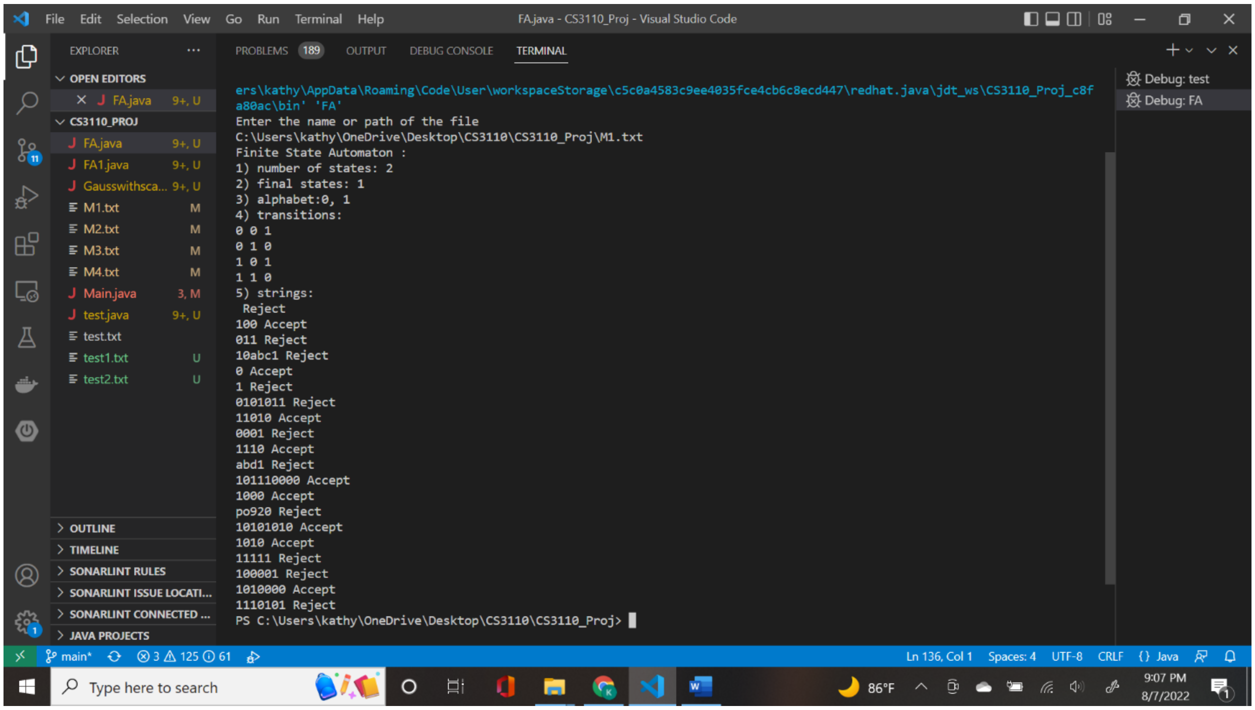
2djsnA	Reject
Java12	Accept
Studio23	Accept
Universal12	Accept

FA4

7	Accept
-7	Reject
007	Reject
3.14	Reject
103	Accept
24930000	Accept
0	Accept
01	Reject
00	Reject
0101	Reject
hello!	Reject
1 1	Reject
10101010101010	Accept
1230000000	Accept
45!	Reject
.1	Reject
100	Reject
0	Reject
80932141079124709	Accept
10000000000000000000	Accept

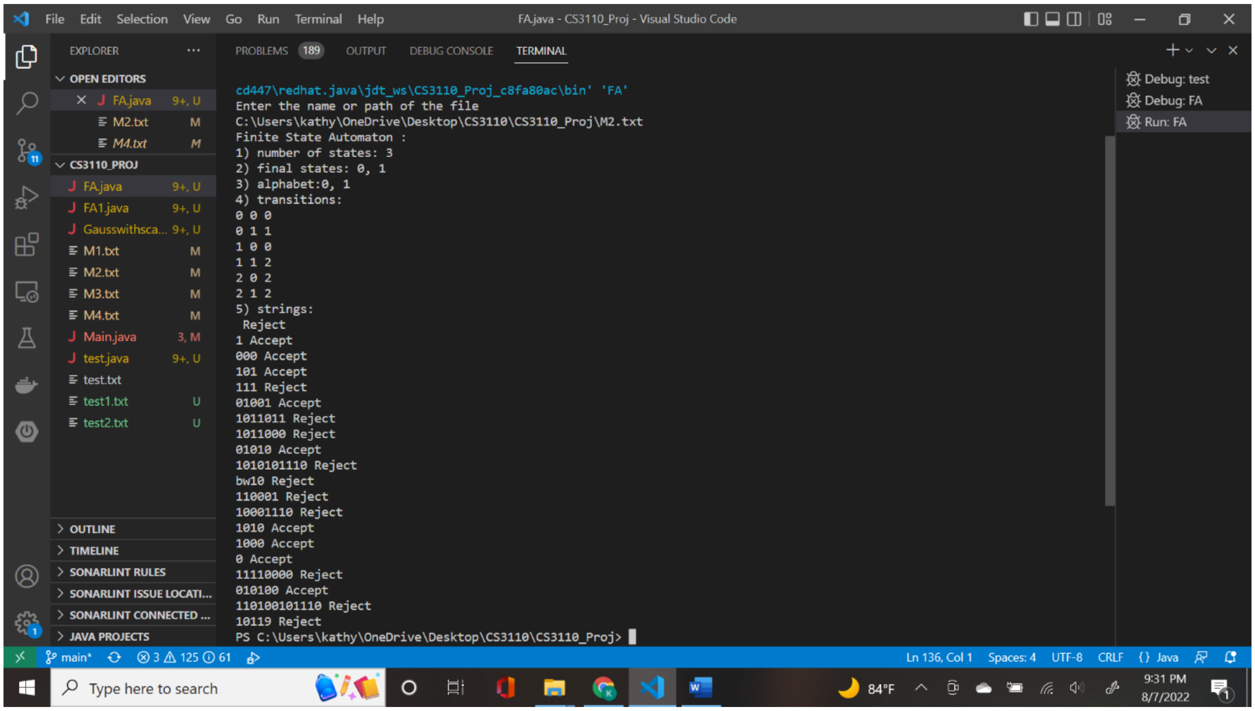
IV. Screenshots for capturing test runs

FA1



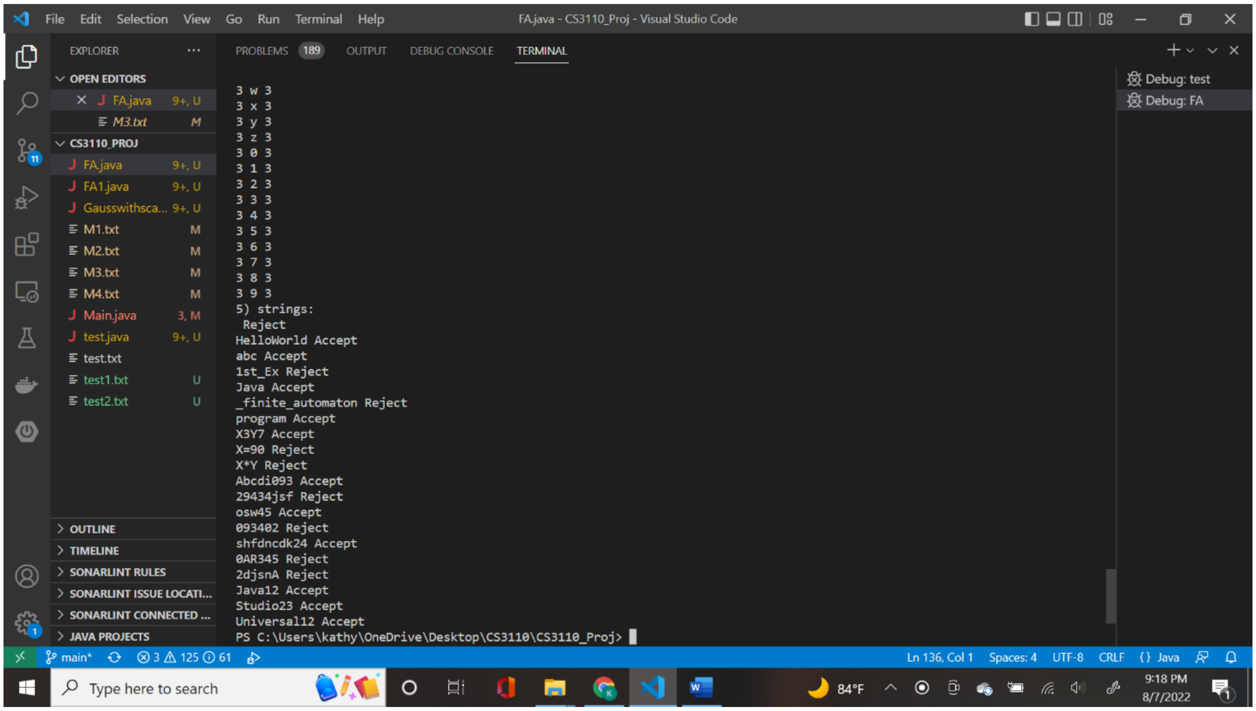
```
ers\kathy\AppData\Roaming\Code\User\workspaceStorage\c5c0a4583c9ee4035fce4cb6c8ecd447\redhat.java\jdt_ws\CS3110_Proj_c8f
a80ac\bin' 'FA'
Enter the name or path of the file
C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj\M1.txt
Finite State Automaton :
1) number of states: 2
2) final states: 1
3) alphabet:0, 1
4) transitions:
0 0 1
0 1 0
1 0 1
1 1 0
5) strings:
 Reject
100 Accept
011 Reject
10abc1 Reject
0 Accept
1 Reject
0101011 Reject
11010 Accept
0001 Reject
1110 Accept
abd1 Reject
101110000 Accept
1000 Accept
po920 Reject
10101010 Accept
1010 Accept
11111 Reject
100001 Reject
1010000 Accept
1110101 Reject
PS C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj>
```

FA2



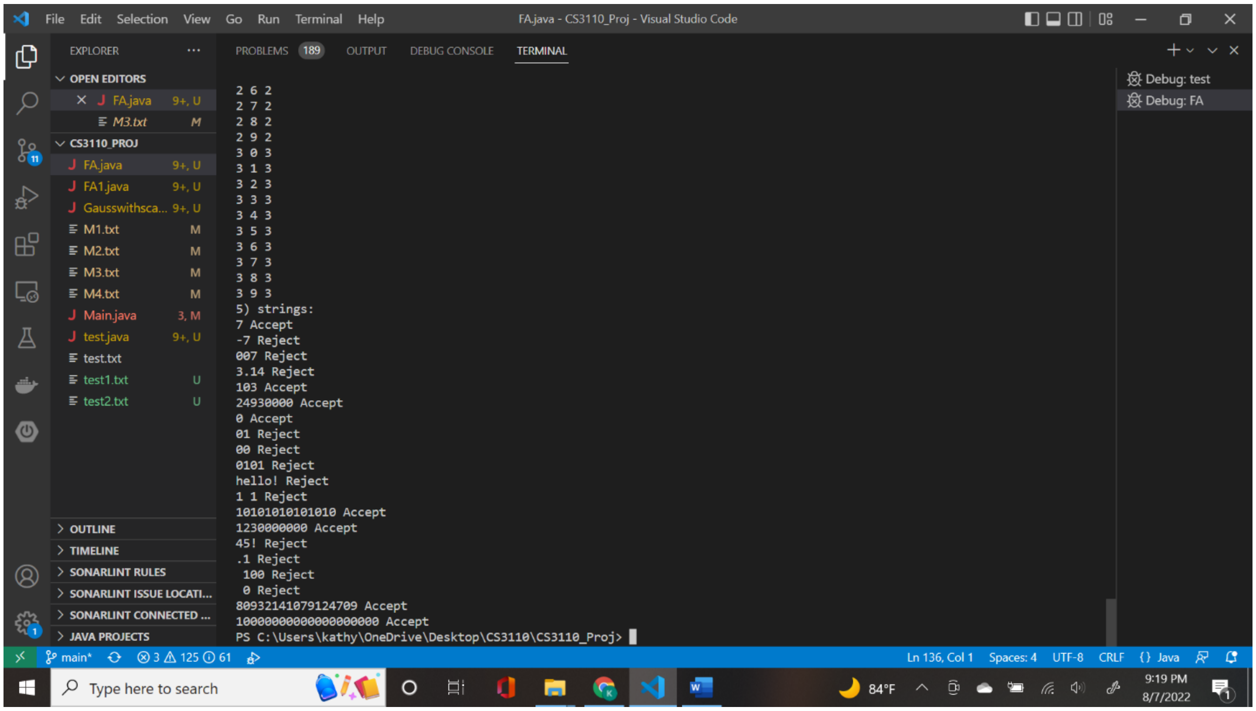
```
cd447\redhat.java\jdt_ws\CS3110_Proj_c8fa80ac\bin' 'FA'
Enter the name or path of the file
C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj\M2.txt
Finite State Automaton :
1) number of states: 3
2) final states: 0, 1
3) alphabet:0, 1
4) transitions:
0 0 0
0 1 1
1 0 0
1 1 2
2 0 2
2 1 2
5) strings:
 Reject
1 Accept
000 Accept
101 Accept
111 Reject
01001 Accept
1011011 Reject
1011000 Reject
01010 Accept
1010101110 Reject
bw10 Reject
110001 Reject
10001110 Reject
1010 Accept
1000 Accept
0 Accept
11110000 Reject
010100 Accept
110100101110 Reject
10119 Reject
PS C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj>
```

FA3



```
3 w 3
3 x 3
3 y 3
3 z 3
3 0 3
3 1 3
3 2 3
3 3 3
3 4 3
3 5 3
3 6 3
3 7 3
3 8 3
3 9 3
5) strings:
 Reject
HelloWorld Accept
abc Accept
1st_Ex Reject
Java Accept
_finite_automaton Reject
program Accept
X3Y7 Accept
X=90 Reject
X*Y Reject
Abcdi093 Accept
29434jsf Reject
osw45 Accept
093402 Reject
shfdncdk24 Accept
0AR345 Reject
2djsnA Reject
Java12 Accept
Studio23 Accept
Universal12 Accept
PS C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj>
```

FA4



```
2 6 2
2 7 2
2 8 2
2 9 2
3 0 3
3 1 3
3 2 3
3 3 3
3 4 3
3 5 3
3 6 3
3 7 3
3 8 3
3 9 3
5) strings:
7 Accept
-7 Reject
007 Reject
3.14 Reject
103 Accept
24930000 Accept
0 Accept
01 Reject
00 Reject
0101 Reject
hello! Reject
1 1 Reject
10101010101010 Accept
1230000000 Accept
45! Reject
.1 Reject
 100 Reject
 0 Reject
80932141079124709 Accept
10000000000000000000 Accept
PS C:\Users\kathy\OneDrive\Desktop\CS3110\CS3110_Proj>
```

V. Summary

In the conclusion, we think this project is really interesting. The most challenging part of this project is the FA3. When other FA (FA1, FA2, and FA4) works only with digits, it is easy to create a universal FA method to work with them. Because we just need to use integer type to work with all arrays from the input. However, the FA3 is in the other scenario when we have to deal with characters (A-Z and a-z). Thus, we change our minds to change some array (alphabet, transition) into string type. This caused a lot of struggle at first. However, in the end, we can fix all of them to make our universal FA method works great.