SOURCE CODE GUIDE

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

This folder consists of all the files part of the project. There are two sub folders- one contains the python files using which the Angluin's L* and Rivest-Schapire variant of L* was implemented and the other folder contains the code for the robot environment model.

The PDF file declaration contains the declaration statement signed.

The PPT used in the video is also attached in the name Video PPT.

A copy of the final report is also attached. The naming convention for the final report is the same as mentioned on the keats page.

2 PYTHON FILES FOLDER

The best IDE to run these python files would be Visual Studio Code. The required packages to be installed to run these files are as follows:

- Pandas
- Numpy
- Random
- Math
- String

Once, the packages are installed, the DFA, (represented as a Pandas Data Frame) which is to be tested is to be un-commented. Care should be taken to uncomment only one DFA.

By default, upon running it without making any change in the code, the DFA which accepts even number of ‘a’s and ‘b’s will be used for learning and the corresponding number of membership queries will be printed. The same applies for Angluin’s L* and Rivest and Schapire’s improvement on L*. Please refer to the following figure as to how the DFA is represented and how another DFA can be used for testing.
The following is the default setting.

```
# Constants

# DFA to accept three consecutive 'a's

# ACTUALTRANSITION={0:{'':0,'a':1, 'b':0},
#       1:{'':1,'a':2, 'b':0},
#       2:{'':2,'a':3, 'b':0},
#       3:{'':3,'a':1,'b':3}}
# ACTUALINITIAL=0
# ACTUALSTATES=[0,1,2,3]
# ACTUALFINAL=[3]

# DFA to accept even number of 'a's and 'b's

ACTUALTRANSITION={0:{'':0,'a':1, 'b':2},
     1:{'':1,'a':0, 'b':3},
     2:{'':2,'a':3, 'b':0},
     3:{'':3,'a':2,'b':1}
}
ACTUALINITIAL=0
ACTUALSTATES=[0,1,2,3]
ACTUALFINAL=[0]
```

Figure 1 : Code for DFA which accepts even number of a's and b's

If we want to test for the DFA that accepts three consecutive ‘a’s, we need to change the setting as follows

```
# Constants

# DFA to accept three consecutive 'a's

ACTUALTRANSITION={0:{'':0,'a':1, 'b':0},
    1:{'':1,'a':2, 'b':0},
    2:{'':2,'a':3, 'b':0},
    3:{'':3,'a':1,'b':3}}
ACTUALINITIAL=0
ACTUALSTATES=[0,1,2,3]
ACTUALFINAL=[3]

# DFA to accept even number of 'a's and 'b's

# ACTUALTRANSITION={0:{'':0,'a':1, 'b':2},
#       1:{'':1,'a':0, 'b':3},
#       2:{'':2,'a':3, 'b':0},
#       3:{'':3,'a':2,'b':1}
# }
# ACTUALINITIAL=0
# ACTUALSTATES=[0,1,2,3]
# ACTUALFINAL=[0]
```

Figure 2: Code for DFA that accepts three consecutive a's

The Output will be displayed in the terminal. The Output for the DFA which accepts even number of ‘a’s and ‘b’s is as follows. There are three items displayed in the output.

- The learned DFA
- The number of times the membership query is called
- Observation table

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   Python

PS C:\Users\ashkm\OneDrive\Desktop\KINGS COLLEGE STUDY MATERIAL\PROJECT\Project\Supplementary files\Python files> & C:/Users/ashkm/anaconda
/envs/Data__Mining/python.exe "c:/Users/ashkm/OneDrive/Desktop/KINGS COLLEGE STUDY MATERIAL/PROJECT/Project/Supplementary files/Python file
/RivestSchapire.py"
Learned DFA is as follows
[{0: {'': 0, 'a': 1, 'b': 2}, 1: {'': 1, 'a': 0, 'b': 3}, 2: {'': 2, 'a': 3, 'b': 0}, 3: {'': 3, 'a': 2, 'b': 1}}, 0, [0], [0, 1, 2, 3]]
The number of times the Membership Query is called
168
The Upper half of the Observation table
       a  b
    1  0  0
a   0  1  0
b   0  0  1
ab  0  0  0
The Lower half of the obsercation table
       a  b
aa   1  0  0
ba   0  0  0
bb   1  0  0
aba  0  0  1
abb  0  1  0
PS C:\Users\ashkm\OneDrive\Desktop\KINGS COLLEGE STUDY MATERIAL\PROJECT\Project\Supplementary files\Python files>
```

Figure 3: Sample output for the DFA which accepts even number of a's and b's

3 MATLAB SIMULINK

Matlab and Simulink has to be installed in the system in order to run the program. Click on the model1.slx file in the folder, the MATLAB will be launched. If the page containing the Simulink flowchart has not opened after MATLAB has been launched, it is advised to click the model1.slx file again inside MATLAB.

After the Simulink flowchart is visible, the run button on top has to be selected to start the simulation.

After the run button, the following two buttons are to be pressed. Forward/Reverse denotes ‘a’ and left /right denotes ‘b’. The button has to be pressed according to the input. Eg. If it is “aabb”, forward/reverse button has to be selected twice. And left/right twice, in the same order.

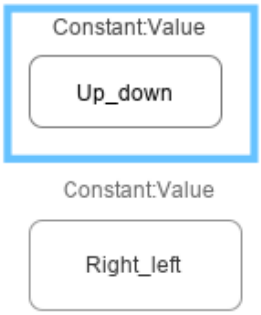

Figure 4: Simulink Input Buttons

Output, whether, the robot is in final state or not, is denoted by the sensor display.

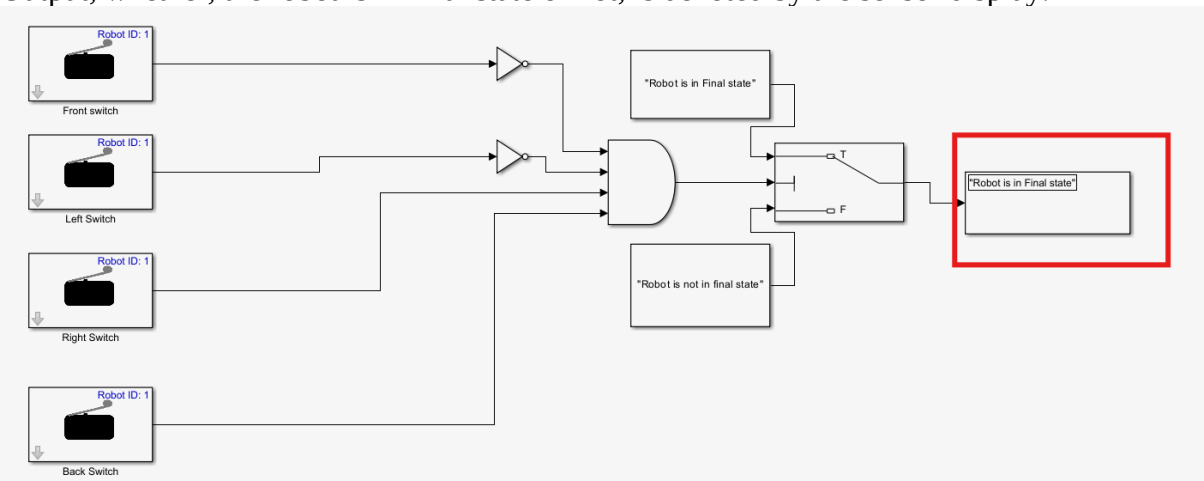

Figure 5: Result of the membership query

Also, in the virtual environment, this block is added to identify the final position.

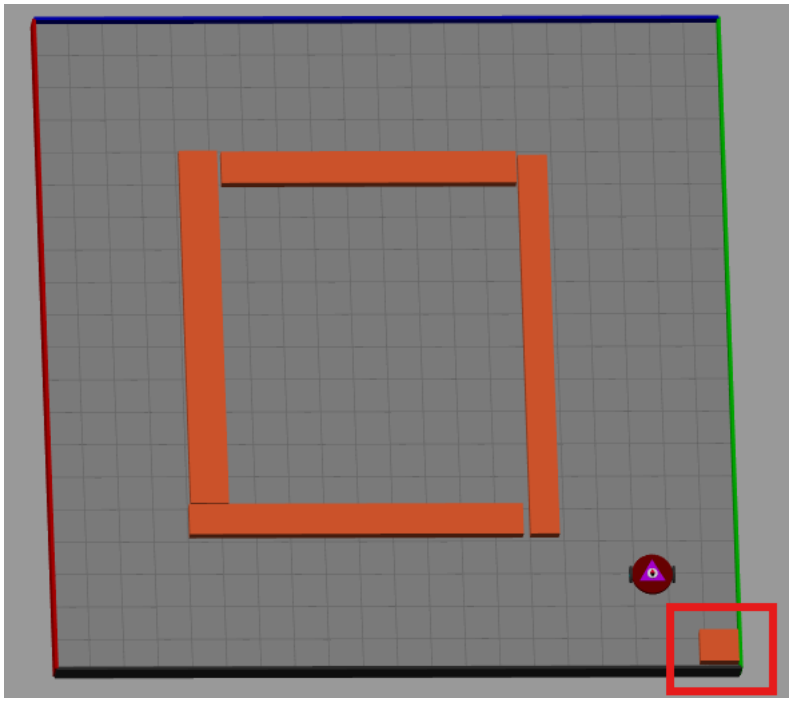

Figure 6: Robot virtual environment