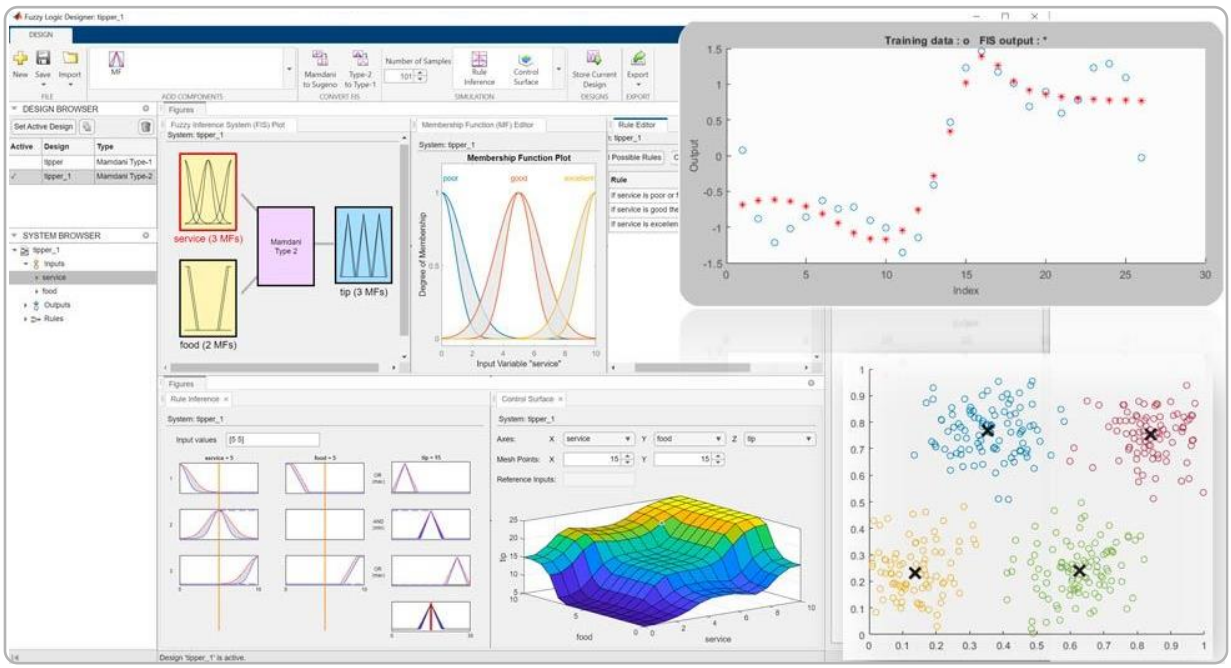
Cairo University
Faculty of Computers & Artificial Intelligence
Department of Computer Science
Soft Computing Course - 2022

Assignment 3 – Fuzzy Logic Toolbox

About the problem:

A "Fuzzy Logic Toolbox" provides functions and apps for designing and simulating fuzzy logic systems. It lets the user specify and configure input variables, output variables, membership functions, rules and defuzzification methods. (Example: MATLAB's Fuzzy Logic Toolbox™)



What you are required to do:

Build a simple fuzzy logic toolbox and test it on a well-known problem. In this fuzzy logic toolbox, the user can:

1. Define a new fuzzy logic system.
2. Define the system's variables.
3. Define each variable's range and fuzzy sets.
4. Define the input variables' crisp values.
5. Define the rules.
6. Get the predicted output.

Example:

```
Fuzzy Logic Toolbox
===================
1- Create a new fuzzy system
2- Quit

1

Enter the system's name and a brief description:
------------------------------------------------
Project Risk Estimation
The problem is to estimate the risk level of a project based on the project
funding and the technical experience of the project's team members.

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

1

Enter the variable's name, type (IN/OUT) and range ([lower, upper]):
(Press x to finish)
---------------------------------------------------------------------
proj_funding      IN    [0, 100]
exp_level   IN    [0, 60]
risk  OUT   [0, 100]
x

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

4

CAN'T START THE SIMULATION! Please add the fuzzy sets and rules first.
```

```
Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

2

Enter the variable's name:
--------------------------
exp_level

Enter the fuzzy set name, type (TRI/TRAP) and values: (Press x to finish)
-----------------------------------------------------
beginner    TRI   0 15 30
intermediate      TRI   15 30 45
expert      TRI 30 60 60
x

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

2

Enter the variable's name:
--------------------------
proj_funding

Enter the fuzzy set name, type (TRI/TRAP) and values: (Press x to finish)
-----------------------------------------------------
very_low    TRAP  0 0 10 30
low   TRAP  10 30 40 60
medium      TRAP  40 60 70 90
high  TRAP  70 90 100 100
x

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.
```

```
2

Enter the variable's name:
-------------------------
risk

Enter the fuzzy set name, type (TRI/TRAP) and values: (Press x to finish)
-----------------------------------------------------
low   TRI   0 25 50
normal      TRI   25 50 75
high  TRI 50 100 100
x

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

3

Enter the rules in this format: (Press x to finish)
IN_variable set operator IN_variable set => OUT_variable set
-------------------------------------------------------------
proj_funding high or exp_level expert => risk low
proj_funding medium and exp_level intermediate => risk normal
proj_funding medium and exp_level beginner => risk normal
proj_funding low and exp_level beginner => risk high
proj_funding very_low and_not exp_level expert => risk high
x

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.

4

Enter the crisp values:
-----------------------
proj_funding: 50
exp_level: 40
```

```
Running the simulation…
Fuzzification => done
Inference => done
Defuzzification => done

The predicted risk is normal (37.5)

Main Menu:
==========
1- Add variables.
2- Add fuzzy sets to an existing variable.
3- Add rules.
4- Run the simulation on crisp values.
Close

Fuzzy Logic Toolbox
===================
1- Create a new fuzzy system
2- Quit
2
```

Note: In this example, the user's input is in **gray**.

Important remarks to help you solve the problem:

1. You will need to **create the appropriate classes and data structures** to store the system's data.
2. The system may contain **multiple output variables.**
3. The expected fuzzy set shapes are either **triangular or trapezoidal.**
4. In the "run simulation" step, you will need to **perform fuzzification, inference and defuzzification** using the weighted average method.
5. **Show error messages** if there is missing input or if the user enters invalid input.
6. **Test your program** on the example in the lab.
7. If you build a **graphical user interface** for the fuzzy logic toolbox and implement reading from and writing to a file, you will get **bonus grades.**

Assignment submission notes:

- The **maximum** number of students in a team is **4** and the **minimum** is **3**.
- The **deadline will be announced, and no late submission** is allowed.
- Please submit **one compressed folder**. The folder name should follow this structure: ID1_ID2_ID3_GROUP
- **Cheating** students will take **negative grades** and no excuses will be accepted. If you have any problems during the submission, contact your TA but don't, under any circumstances, give your code to or take the code from your friends.

Grading Criteria: (5 marks)

Program & Code Structure	1
Fuzzification (Memberships)	1.5
Inference	1
Defuzzification (Output)	1.5

Good luck