ECE 449 - Intelligent Systems Engineering

Lab 3-D42: Fuzzy Logic Concepts

Lab date: *Thursday, October 28, 2021 -- 2:00 - 4:50 PM*
Room: *ETLC E5-013*
Lab report due: *Wednesday, November 10, 2021 -- 3:50 PM*

---

1. Objectives

The objectives of this lab are to become familiar with the basic concepts of fuzzy logic. These concepts include:

- defining membership functions and modifying them with linguistic terms
- performing various operations on fuzzy sets
- representing fuzzy sets using α-cuts
- constructing fuzzy relations, projections, and cylindrical extensions
- performing composition and using it in compositional rules of inference

2. Expectations

Complete the pre-lab, and hand it in before the lab starts. A formal lab report is required for this lab, which will be the completed version of this notebook. There is a marking guide at the end of the lab manual. If figures are required, label the axes and provide a legend when appropriate. An abstract, introduction, and conclusion are required as well, for which cells are provided at the end of the notebook. The abstract should be a brief description of the topic, the introduction a description of the goals of the lab, and the conclusion a summary of what you learned, what you found difficult, and your own ideas and observations.

3. Pre-lab

1. Why is defuzzification an important step when using fuzzy sets?

We also strongly recommend that you look over section 1 of the Python supplement to familiarize yourself with Jupyter notebooks and install the necessary libraries for future labs.

4. Introduction

Fuzzy logic is a form of logic in which the truth values of variables can range from the interval of 0 to 1, instead of exclusively 0 or 1. This can be used to solve problems in a more human-like reasoning way by allowing gradual membership in sets. These fuzzy sets form inputs and outputs to linguistic relations that can be easily constructed, such as:

IF *temp* **IS** HOT **THEN** *fan* **IS** HIGH

An important concept is the *linguistic variable*, which is a variable whose values are words. In the example above, the *linguistic variable* "temp" takes the value "HOT".
Employing fuzzy systems requires the user to first define membership functions that take values from 0 to 1, and are defined over the region of interest, called the *universe set*. One can apply linguistic modifiers (*hedges*) to modify the meaning of a fuzzy set, such as:

temp **IS** VERY HOT, rather than *temp* **IS** HOT

A *hedge* in this case is VERY.
Similar to crisp sets, the *union*, *intersection*, and *complement* operators can be performed on fuzzy sets. They may also be represented using a family of crisp sets, by using α*-cuts*.
Finally, similarly to crisp relations, which is the mapping between two crisp sets, fuzzy sets can form *relations* between two membership functions of different universes. These relations bring forth more operations, such as *cylindrical closure*, *sup-min composition*, *compositional rule of inference*, and *defuzzification*. *Cylindrical closure* is a fuzzy relation that corresponds to the cross-product domain of linguistic variables.

5. Background

Automatic monitoring stations are used to characterize the quality of the environment in the Arctic by collecting meteorological data at regular intervals. Because of how remote these locations are, the monitoring stations are designed to generate and store power from renewable resources, namely the sun and wind, to minimize the frequency of maintenance required. However, due to the polar nights and long winters, solar radiation reaching the ground during these times is very low or non-existent. Consequently, this can lead to long intervals during which there is no remaining power, and no data is collected. To avoid this, the duty cycle of the monitoring station can be adjusted in order to conserve power. A controller to determine the optimal duty cycle can be built using fuzzy logic based on two factors: *state of charge* (SOC) of the battery and *future average power* (P) from the renewable resources. For example, one such rule could be as follows:

IF *state of charge* **IS** LOW **AND** *future average power* **IS** MEDIUM **THEN** *duty cycle* ***IS*** *MEDIUM*

In the case where this rule would apply, the monitoring station could only take measurements for around half of its regular period to conserve power, and obtain data more frequently than what the previous method would offer. The next two labs will focus on this concept and work towards building a fuzzy controller to manage the power consumption of a monitoring station.

6. Experimental Procedure

If you have not yet installed the skfuzzy library, run the cell below.

In [1]:
```
%%bash
# "--user" is essential to install in local environment"
pip install --user -U scikit-fuzzy
```

```
Requirement already satisfied: scikit-fuzzy in ./.local/lib/python3.9/site-packages (0.4.2)
Requirement already satisfied: networkx>=1.9.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (2.3)
Requirement already satisfied: scipy>=0.9.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (1.7.1)
Requirement already satisfied: numpy>=1.6.0 in /opt/conda/lib/python3.9/site-packages (from scikit-fuzzy) (1.21.2)
Requirement already satisfied: decorator>=4.3.0 in /opt/conda/lib/python3.9/site-packages (from networkx>=1.9.0->scikit-fuzzy) (5.0.9)
```

Run the cell below to import the libraries required to complete this lab.

In [2]:
```
%matplotlib inline

import numpy as np                          # General math operations
import matplotlib.pyplot as plt             # Data visualization
from mpl_toolkits.mplot3d import Axes3D     # 3D data visualization
import skfuzzy as fuzz                      # Fuzzy toolbox
```

Exercise 1: Membership functions
Consider a weather station with a battery that has a minimum SOC of 20% and a maximum SOC of 100%.

1. Define the universe set for SOC from 20 to 100, using 81 discrete elements.

In [38]:
```
X = np.linspace(20, 100, num = 81)      # Universe set for SOC
print("Universe Set: \n")
print(X)
```

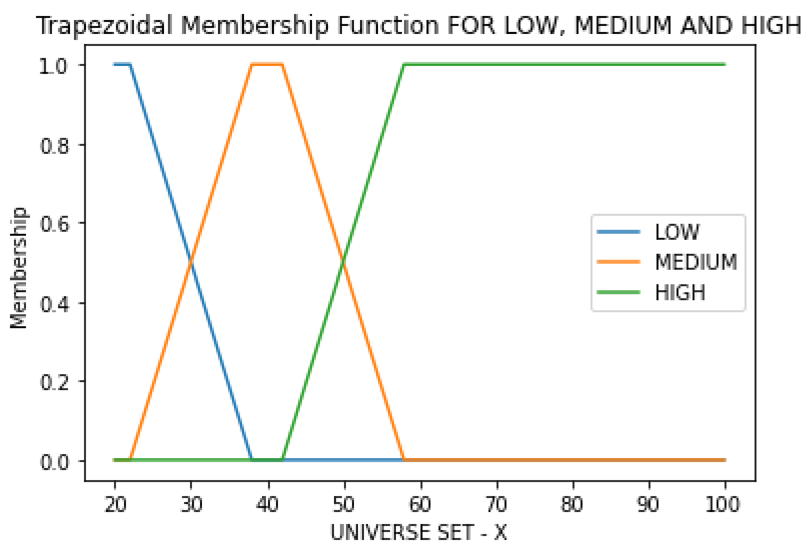
```
Universe Set: 

[ 20.  21.  22.  23.  24.  25.  26.  27.  28.  29.  30.  31.  32.  33.
  34.  35.  36.  37.  38.  39.  40.  41.  42.  43.  44.  45.  46.  47.
  48.  49.  50.  51.  52.  53.  54.  55.  56.  57.  58.  59.  60.  61.
  62.  63.  64.  65.  66.  67.  68.  69.  70.  71.  72.  73.  74.  75.
  76.  77.  78.  79.  80.  81.  82.  83.  84.  85.  86.  87.  88.  89.
  90.  91.  92.  93.  94.  95.  96.  97.  98.  99. 100.]
```

2. Plot the trapezoidal membership functions, LOW, MEDIUM, and HIGH, on one figure according to the parameters given below.

Fuzzy set	*a*	*b*	*c*	*d*
LOW	20	20	22	38
MEDIUM	22	38	42	58
HIGH	42	58	100	100

In [39]:

```
LOW = fuzz.trapmf(X, [20,20,22,38])
MEDIUM = fuzz.trapmf(X, [22,38,42,58])
HIGH = fuzz.trapmf(X, [42,58,100,100])

#Plotting MF FOR LOW
lowHandle ,= plt.plot(X,LOW, label = 'LOW')
#Plotting MF FOR MEDIUM
mediumHandle ,= plt.plot(X,MEDIUM, label = 'MEDIUM')
#Plotting MF FOR HIGH
highHandle ,= plt.plot(X,HIGH, label = 'HIGH')

plt.xlabel('UNIVERSE SET - X')
plt.ylabel('Membership')
plt.title('Trapezoidal Membership Function FOR LOW, MEDIUM AND HIGH')
plt.legend(handles = [lowHandle, mediumHandle, highHandle])
plt.show()
```

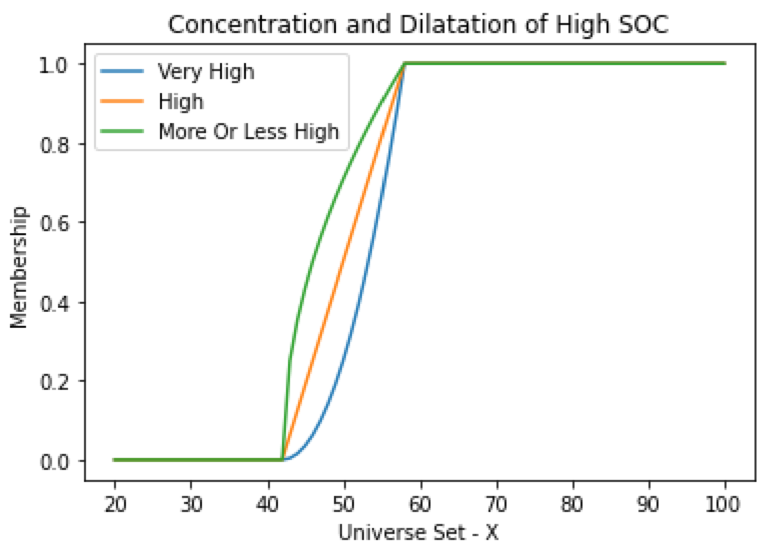


Exercise 2: Linguistic modifiers
Modify the fuzzy set HIGH SOC to VERY HIGH SOC and MORE OR LESS HIGH SOC.

1. Plot HIGH, VERY HIGH, and MORE OR LESS HIGH on the same figure.

In [40]:

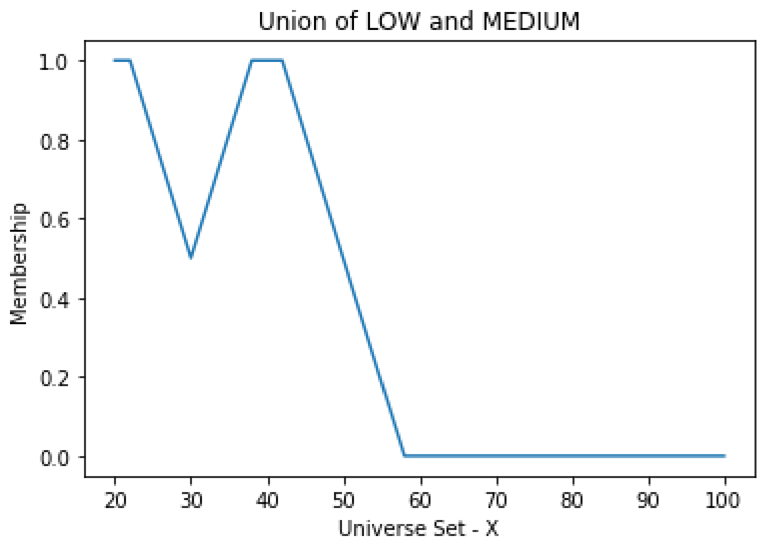
```
HIGH = fuzz.trapmf(X, [42,58,100,100])
veryHIGH = HIGH ** (2)
moreOrLessHIGH = HIGH ** (0.5)

veryHighHandle ,= plt.plot(X, veryHIGH, label = 'Very High')
highHandle ,= plt.plot(X,HIGH, label = 'High')
moreOrLessHighHandle ,= plt.plot(X, moreOrLessHIGH, label = 'More Or Less Hig
plt.xlabel('Universe Set - X')
plt.ylabel('Membership')
plt.title('Concentration and Dilatation of High SOC')
plt.legend(handles = [veryHighHandle, highHandle, moreOrLessHighHandle])
plt.show()
```



Exercise 3: Fuzzy set operations
On separate figures, plot the following fuzzy sets:

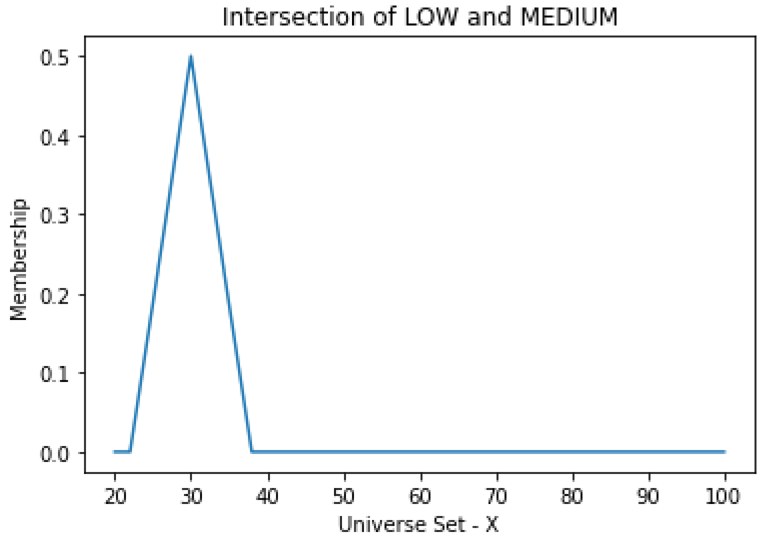
1. Union of LOW and MEDIUM

In [41]:
```
[Y, union]= fuzz.fuzzy_or(X, LOW, X, MEDIUM)          #Union of LOW and MEDIUM
plt.plot(Y,union)
plt.xlabel('Universe Set - X')
plt.ylabel('Membership')
plt.title('Union of LOW and MEDIUM')
plt.show()
```



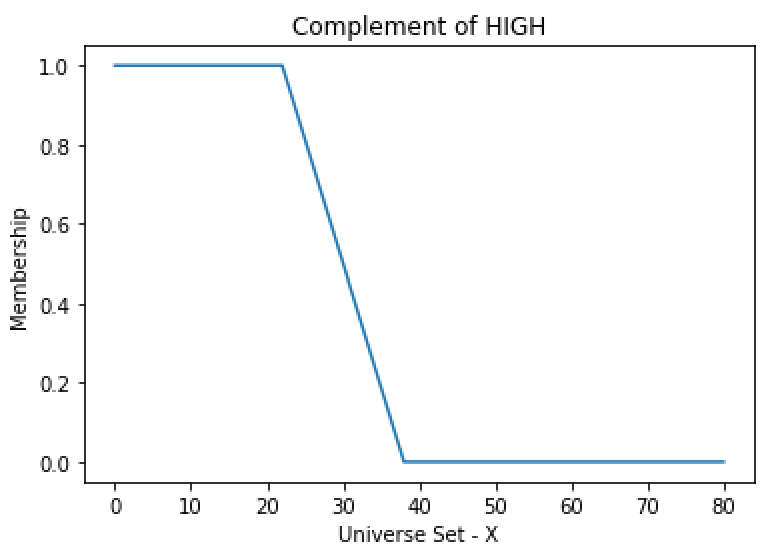
2. Intersection of LOW and MEDIUM

In [42]:
```
[Z, intersection]= fuzz.fuzzy_and(X, LOW, X, MEDIUM)     #Inersection of LOW
plt.plot(Z,intersection)
plt.xlabel('Universe Set - X')
plt.ylabel('Membership')
plt.title('Intersection of LOW and MEDIUM')
plt.show()
```



3. Complement of HIGH

In [43]:
```
complement = fuzz.fuzzy_not(HIGH)     #Complement of HIGH
plt.plot(complement)
plt.xlabel('Universe Set - X')
plt.ylabel('Membership')
plt.title('Complement of HIGH')
plt.show()
```

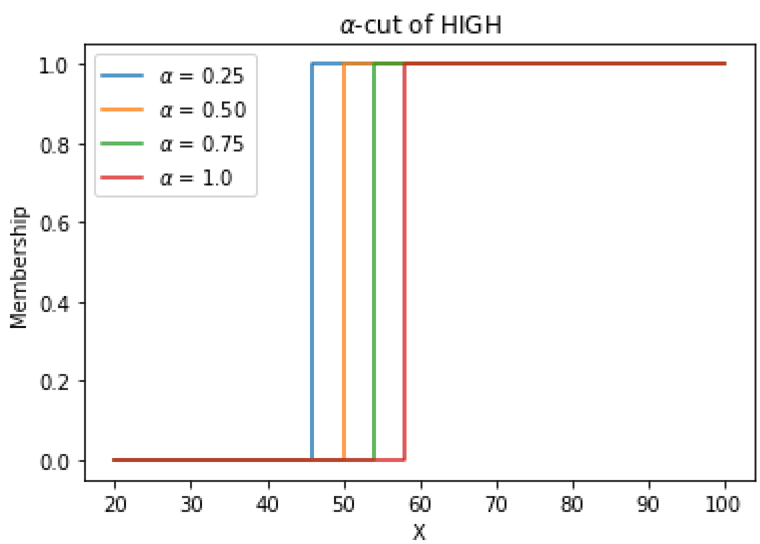


Exercise 4: α-cuts

Using the HIGH SOC fuzzy set.

1. Plot the individual α-cuts for α = {1.0, 0.75, 0.50, 0.25} on the same figure.

In [3]:
```
X = np.linspace(20, 100, num = 81)      # Universe set for SOC
HIGH = fuzz.trapmf(X, [42,58,100,100])

alphaCut1 = (HIGH >= 0.25)  # alpha-cut with alpha = 0.25
aCutHandle1 ,= plt.step(X, alphaCut1, where = 'post', label = r'$\alpha$ = 0.

alphaCut2 = (HIGH >= 0.50) # alpha-cut with alpha = 0.50
aCutHandle2 ,= plt.step(X, alphaCut2, where = 'post', label = r'$\alpha$ = 0.

alphaCut3 = (HIGH >= 0.75) # alpha-cut with alpha = 0.75
aCutHandle3 ,= plt.step(X,alphaCut3, where = 'post', label = r'$\alpha$ = 0.7

alphaCut4 = (HIGH >= 1.0) # alpha-cut with alpha = 1.0
aCutHandle4 ,= plt.step(X,alphaCut4, where = 'post', label = r'$\alpha$ = 1.0

plt.xlabel('X')
plt.ylabel('Membership')
plt.title(r'$\alpha$-cut of HIGH')
plt.legend(handles = [aCutHandle1,aCutHandle2, aCutHandle3, aCutHandle4])
plt.show()
```

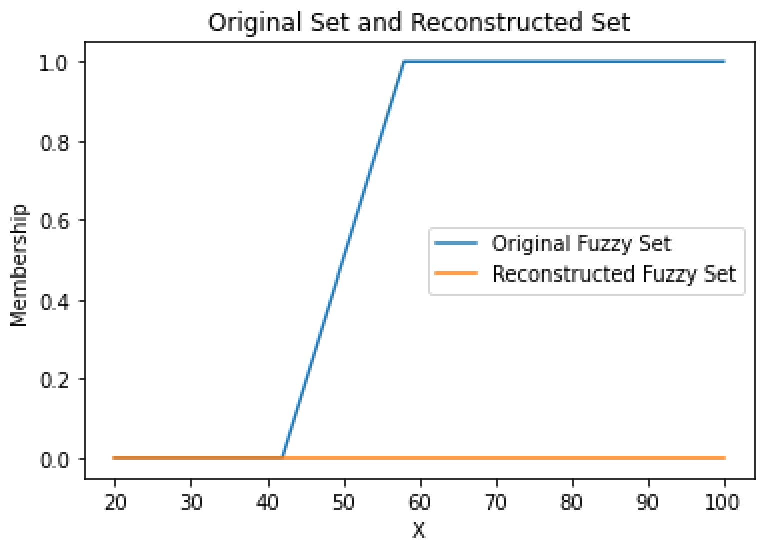


2. Plot the original fuzzy set and its α-cut reconstruction on the same figure.

HINT: The **np.amax()** function is helpful in reconstructing the fuzzy set.

In [37]:

```
X = np.linspace(20, 100, num = 81)      # Universe set for SOC
HIGH = fuzz.trapmf(X, [42,58,100,100])
handle1 ,= plt.plot(X, HIGH, label = 'Original Fuzzy Set')

val1 = np.amax(0.25 * alphaCut1)
val2 = np.amax(0.50 * alphaCut2)
val3 = np.amax(0.75 * alphaCut3)
val4 = np.amax(1.0 * alphaCut4)
new = fuzz.trapmf(X,[val1,val2,val3,val4])
handle2 , = plt.plot(X,new, label = 'Reconstructed Fuzzy Set')

plt.xlabel('X')
plt.ylabel('Membership')
plt.title(r'Original Set and Reconstructed Set')
plt.legend(handles = [handle1,handle2])
plt.show()
```



3. Comment on the quality of the α-cut reconstruction.

The α -cut reconstruction obtained is of a poor quality. As we can see from the figure above, the original set is in the form of a trapezoid, however, the reconstructed fuzzy set is in the form of a straight line. So, we observe that the reconstruction does not even closely resemble the original fuzzy set and hence it can be said that the reconstruction is poor.

Exercise 5: Relations - Cylindrical closure

Based off of typical meteorological data, the locations in which the monitoring stations are situated can only provide future average power from 0W to 100W.

1. Define the universe set for future average power from 0 to 100, using 101 discrete elements.

In [44]:

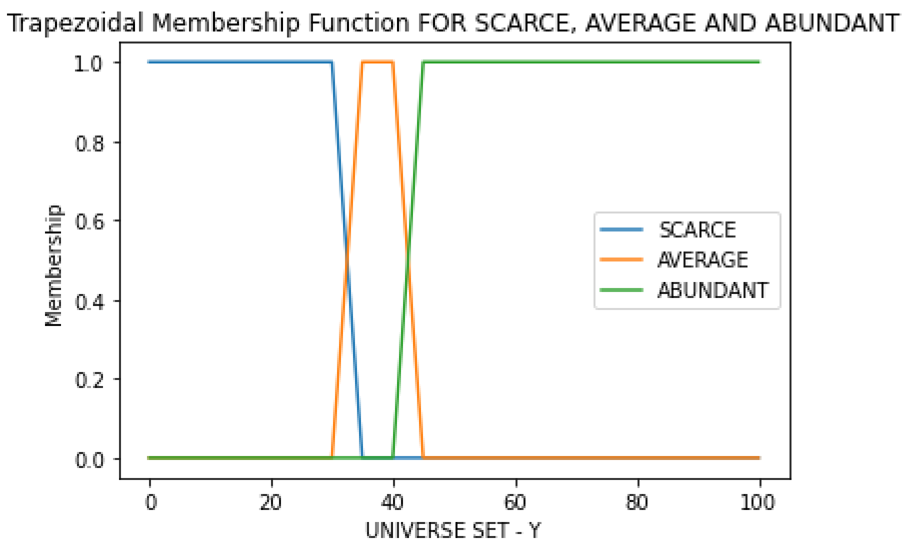
```
Y = np.linspace(0, 100, num = 101)       # Universe set for future average po
print("Universe Set: \n")
print(Y)
```

```
Universe Set: 

[  0.   1.   2.   3.   4.   5.   6.   7.   8.   9.  10.  11.  12.  13.
  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.
  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.  40.  41.
  42.  43.  44.  45.  46.  47.  48.  49.  50.  51.  52.  53.  54.  55.
  56.  57.  58.  59.  60.  61.  62.  63.  64.  65.  66.  67.  68.  69.
  70.  71.  72.  73.  74.  75.  76.  77.  78.  79.  80.  81.  82.  83.
  84.  85.  86.  87.  88.  89.  90.  91.  92.  93.  94.  95.  96.  97.
  98.  99. 100.]
```

2. Plot the trapezoidal membership functions, SCARCE, AVERAGE, and ABUNDANT in one figure, according to the parameters given below.

Fuzzy set	*a*	*b*	*c*	*d*
SCARCE	0	0	30	35
AVERAGE	30	35	40	45
ABUNDANT	40	45	100	100

In [45]:
```
SCARCE = fuzz.trapmf(Y, [0,0,30,35])
AVERAGE = fuzz.trapmf(Y, [30,35,40,45])
ABUNDANT = fuzz.trapmf(Y, [40,45,100,100])

#Plotting MF FOR SCARCE
scarceHandle , = plt.plot(Y,SCARCE, label = 'SCARCE')
#Plotting MF FOR AVERAGE
averageHandle , = plt.plot(Y,AVERAGE, label = 'AVERAGE')
#Plotting MF FOR ABUNDANT
abundantHandle , = plt.plot(Y,ABUNDANT, label = 'ABUNDANT')

plt.xlabel('UNIVERSE SET - Y')
plt.ylabel('Membership')
plt.title('Trapezoidal Membership Function FOR SCARCE, AVERAGE AND ABUNDANT')
plt.legend(handles = [scarceHandle,averageHandle,abundantHandle])
plt.show()
```

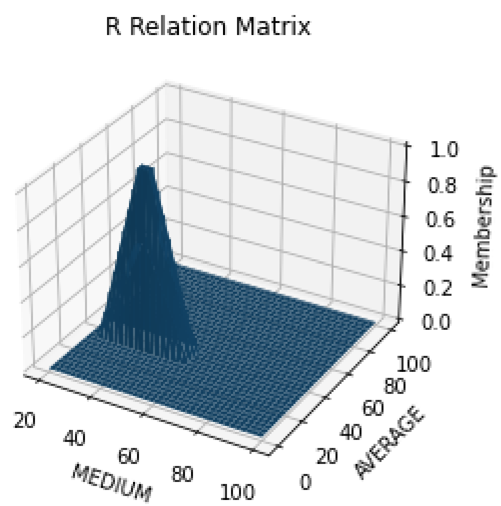


3. Using *Larsen implication*, define the relation $R(MEDIUM,\ AVERAGE)$. Plot the relation matrix.

In [46]:

```
X = np.linspace(20, 100, num = 81)
Y = np.linspace(0, 100, num = 101)
MEDIUM = fuzz.trapmf(X, [22,38,42,58])
AVERAGE = fuzz.trapmf(Y,[30,35,40,45])

larsen = fuzz.relation_product(MEDIUM, AVERAGE)  # Larsen implication

fig = plt.figure()
[gX, gY]= np.meshgrid(X, Y, indexing = 'ij')
ax = fig.gca(projection = '3d')
ax.plot_surface(gX, gY, larsen)
ax.set_xlabel('MEDIUM')
ax.set_ylabel('AVERAGE')
ax.set_zlabel('Membership')
ax.set_title('R Relation Matrix')
plt.show()
```

```
/tmp/ipykernel_102/873607703.py:10: MatplotlibDeprecationWarning: Calling g
ca() with keyword arguments was deprecated in Matplotlib 3.4. Starting two
minor releases later, gca() will take no keyword arguments. The gca() funct
ion should only be used to get the current axes, or if no axes exist, creat
e new axes with default keyword arguments. To create a new axes with non-de
fault arguments, use plt.axes() or plt.subplot().
  ax = fig.gca(projection = '3d')
```



4. What is the meaning of the individual rows of the relation matrix? What does the first row mean?

```
The relation matrix represents a fuzzy implication relation where, each
individual row of the relation matrix corresponds to a matching operation and
provides a stronger (local) implication. This means that MEDIUM SOC is
coupled with AVERAGE future average powerthrough a min operation, in this
case it is larsen, so it means through a min product operation.
```

```
Thus, each individual row corresponds to a value from a row in the membership
function MEDIUM coupled (through a min product operation) with different
values in each column from the membership function AVERAGE. Thus, the first
row means, that the values in the first row in the membership function MEDIUM
are coupled (through a min product operation)  with the first column in the
membership function AVERAGE.
```

Exercise 6: Sup-min composition

Three monitoring stations positioned at different locations are checked, accordingly, the membership values of SOC fuzzy sets are assigned. The findings are expressed as a relation, $LocationSOC(location, state\ of\ charge)$, and defined using the following matrix:

$$LocationSOC = \begin{bmatrix} 0.84 & 0.08 & 0 \\ 0.03 & 0.5 & 0.08 \\ 0 & 0.1 & 0.8 \end{bmatrix}$$

Each row in the matrix corresponds to a monitoring station, and the columns give the membership values in the fuzzy sets, LOW, MEDIUM, and HIGH, respectively. For example, the last monitoring station has a 0 LOW, 0.1 MEDIUM, and a 0.8 HIGH SOC.
Additionally, it was determined how the SOC of the monitoring station corresponds to the future average power of its location. This is represented by the relation, $SOCPower(state\ of\ charge, future\ average\ power)$, found below.

$$SOCPower = \begin{bmatrix} 1 & 0.3 & 0 \\ 0.2 & 0.5 & 0.3 \\ 0 & 0.5 & 1 \end{bmatrix}$$

1. Determine the max-min composition $c_1 = LocationSOC \circ SOCPower$ and $c_2 = SOCPower^T \circ LocationSOC^T$ and print the resulting matrices.

In [47]:
```
LocationSOC = np.array([[0.84, 0.08,0], [0.03, 0.5,0.08],[0,0.1,0.8]])
transpose_LocationSOC = LocationSOC.transpose()   #Transpose of LocationSOC

SOCPower = np.array([[1,0.3,0], [0.2,0.5,0.3],[0,0.5,1]])
transpose_SOCPower = SOCPower.transpose()   #Transpose of SOCPower

c1 = fuzz.maxmin_composition(LocationSOC, SOCPower)    #max-min composition c
c2 = fuzz.maxmin_composition(transpose_SOCPower,transpose_LocationSOC)   #max

print("c1:\n",c1)
print("\n")
print("c2:\n",c2)
```

```
c1:
 [[0.84 0.3  0.08]
 [0.2  0.5  0.3 ]
 [0.1  0.5  0.8 ]]


c2:
 [[0.84 0.2  0.1 ]
 [0.3  0.5  0.5 ]
 [0.08 0.3  0.8 ]]
```

2. How can you interpret these relations?

```
The relation c1 is a fuzzy set resulting from a Cartesian product forming a
relation R : X × Y → [0, 1] (where X and Y are the Universe of Discourse for
SOC and future average power respectively) followed by a modelling using a t-
norm operation - min.
Thus, we can say that c1 is relation that gives us an association of the
Location of SOC with the Power of SOC for each monitoring station whereas c2
is the transpose of c1 and gives an association of Power of SOC with the
Location of SOC for each monitoring station.
```

Exercise 7: Compositional rule of inference
Another monitoring station was checked and found to have a SOC of 28%. Use a compositional rule of inference to determine the future average power fuzzy set based on the knowledge of a monitoring station with LOW SOC in a location with SCARCE future average power.

1. Express the item as a fuzzy singleton on the SOC universe set.

In [48]:
```
X = np.linspace(20, 100, num = 81)   #SOC universe set
S = np.zeros(81)    # Define fuzzy singleton
S[28] = 1
print("Fuzzy Singleton:\n",S)
```

```
Fuzzy Singleton:
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

2. Use *Mamdani implication* to define the relation between LOW and SCARCE.

In [49]:
```
X = np.linspace(20, 100, num = 81)
Y = np.linspace(0, 100, num = 101)
LOW = fuzz.trapmf(X, [20,20,22,38])
SCARCE = fuzz.trapmf(Y, [0,0,30,35])
mamdani = fuzz.relation_min(LOW,SCARCE)
print("Mamdani Implication to define the relation between LOW and SCARCE:\n",
```

```
Mamdani Implication to define the relation between LOW and SCARCE:
 [[1. 1. 1. ... 0. 0. 0.]
 [1. 1. 1. ... 0. 0. 0.]
 [1. 1. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

3. Use the relation from exercise 9 to derive the associated fuzzy set. Print this fuzzy set as a vector.

In [50]:
```
fuzzySet = fuzz.maxmin_composition(S,mamdani)
print(fuzzySet)
```

```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]]
```

Exercise 8: Defuzzification

Determine the crisp value of the fuzzy set obtained from the compositional rule of inference applied in the previous exercise. Use the Mean of Maxima (MOM) defuzzification method.

1. Print the resulting future average power of the location.

In [52]:
```
crispValue = fuzz.defuzz(Y, fuzzySet, 'mom') # Mean Of Maximum defuzzificatic
print("Resulting future average power of the location:",crispValue)
```

```
Resulting future average power of the location: 50.0
```

Abstract

A fuzzy relation R can be defined in a similar fashion as (crisp) relation, however the values of R are taken from the entire interval [0, 1] instead of the binary values {0, 1}, i.e. R : X × Y → [0, 1]. In addition to related/unrelated, fuzzy relation can also express a degree of relationship in sense of relation R.These fuzzy sets form inputs and outputs to linguistic relations that can be easily constructed.

In order to use fuzzy sets, we first define membership functions that take values from 0 to 1, and are defined over the universe set. Linguistic variable is a variable whose values are words and one can apply linguistic modifiers/hedges to modify the meaning of a fuzzy set, such as: fan speed IS VERY FAST, rather than fan speed IS FAST where the hedge is VERY.

Also, the union, intersection, and complement operators can be performed on fuzzy sets just like on crisp relations. α -cuts is a family of crisp sets used to represent such operations - union, intersection and complement.

Fuzzy sets can also form relations between two membership functions of different universe of discourse. Such relations culminate operations, such as cylindrical closure, sup-min composition, compositional rule of inference, and defuzzification. Here, cylindrical closure is a fuzzy relation that corresponds to the cross-product domain of linguistic variables and defuzzification is the process of obtaining a single number from the output of the aggregated fuzzy set.

Introduction

In this lab, we had the following objectives:
1. Defining membership functions and modifying them with linguistic terms
This was done in exercise 1,2 and 5 where trapeziodal membership functions were plotted for LOW, MEDIUM and HIGH SOC (Exercise 1) and SCARCE,AVERAGE and ABUNDANT future average power (Exercise 5). The linguistic terms for the membership function HIGH SOC were also modified using VERY HIGH AND MORE OR LESS HIGH through dilation and concerntration of fuzzy sets ((Exercise 2).

2. Performing various operations on fuzzy sets
In exercise 3, various operations on fuzzy sets were performed such as union and intersection between LOW AND MEDIUM and the complement of HIGH.

3.Representing fuzzy sets using α -cuts
In exercise 4, α - cuts were plotted for the HIGH SOC fuzzy set with α = {1.0, 0.75, 0.50, 0.25} and the set was reconstructed using α- cuts and the quality of reconstruction was observed.

4. Constructing fuzzy relations, projections, and cylindrical extensions
In exercise 5, using Larsen implication, we defined the relation $R(MEDIUM, AVERAGE)$ and plotted the relation matrix. We observed the relation matrix to conclude what each row in such a matrix represents.

5.Performing composition and using it in compositional rules of inference
Finally, through exercises 6, 7 and 8, we performed Sup-min composition (exercise 6) to obtain a max-min composition between two membership functions and we then determined the future average power fuzzy set based on the knowledge of a monitoring station with LOW SOC in a location with SCARCE future average power using a compositional rule of inference and a fuzzy singleton. Lastly, using defuzzification - Mean of Maxima (MOM) defuzzification method, we found the future average power of the location.

Conclusion

In this lab, we became familiar with the basic concepts of fuzzy logic. Firstly, trapezoidal membership functions were plotted for different fuzzy sets, then, a membership function for the fuzzy set HIGH SOC was modified with linguistic terms using dilation and concentration. We then perfomed different fuzzy operations such as union, intersection and complement. α -cuts were plotted for the HIGH SOC and the fuzzy set was reconstructed using the α -cuts. We observed a poor reconstruction as it was a straight line while the HIGH SOC was in the form of a trapezoid.

We also constructing fuzzy relations using larsen implication and cylindrical extensions and plotted a 3D relation matrix for MEDIUM and AVERAGE. Composition was performed and it was used in compositional rules of inference such as Sup-min composition, to obtain a max-min composition between two membership functions.

Finally,we determined the future average power fuzzy set based on the knowledge of a monitoring station with LOW SOC in a location with SCARCE future average power using a compositional rule of inference and a fuzzy singleton and using the Mean of Maxima (MOM) defuzzification method, we found the future average power of the location as 50.0

Lab 1 Marking Guide

Exercise	Item	Total Marks	Earned Marks
	$Pre - lab$	10	
	$Abstract$	3	
	$Introduction$	3	
	$Conclusion$	4	
1	$Membership\ functions$	10	
2	$Linguistic\ modifiers$	5	
3	$Fuzzy\ operations$	10	
4	$Alpha\ cuts$	10	
5	$Fuzzy\ relations$	15	
6	$Sup - min\ composition$	15	
7	CRI	5	
8	$Defuzzification$	10	
	TOTAL	100	