Batch: **Roll No.:**

Experiment No. 7

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: : Fuzzification methods and operations on fuzzy set

Aim : To implement fuzzification methods and perform operations on fuzzy sets

Expected Outcome of Experiment:

CO4 : Apply basics of Fuzzy logic and neural networks

Books/ Journals/ Websites referred:

Pre Lab/ Prior Concepts:

Fuzzy logic:

The term fuzzy refers to things that are not clear or are vague. In the real world many times we encounter a situation when we cannot determine whether the state is true or false, their fuzzy logic provides very valuable flexibility for reasoning. In this way, we can consider the inaccuracies and uncertainties of any situation.

Characteristics of fuzzy logic:

1. This concept is flexible and we can easily understand and implement it.
2. It is used for helping the minimization of the logics created by humans.
3. It is the best method for finding the solution of those problems which are suitable for approximate or uncertain reasoning.
4. It always offers two values, which denote the two possible solutions for a problem and statement.
5. It allows users to build or create the functions which are non-linear of arbitrary complexity. **Definition of membership function (and fuzzy set):**

If X is a universe of discourse and x ϵ X, then a fuzzy set A in X is defined as a set of ordered pairs, that is

$$A = \{(x, \mu_A(x)) | x \in X\}$$

where A(x) is called the membership function for the fuzzy set A.
Note: A(x) map each element of X onto a membership grade (or membership value) between 0 and 1 (both inclusive).

Fuzzy set operations

1. Equal fuzzy sets

Two fuzzy sets A(x) and B(x) are said to be equal, if μA(x) = μB(x) for all x ∈ X. It is expressed as follows

A(x) = B(x), if μA(x)= μB(x)

2. Complement of fuzzy set A(x)

The complement is the opposite of the set. The complement of a fuzzy set is denoted by Ā(x) and is defined with respect to the universal set X as follows: Ā(x) = 1- A(x) for all x ε X

3. Intersections of fuzzy sets

Inter section of a fuzzy sets define how much of the element belongs to both sets. May have different degrees of membership in each set. The degree of membership is the lower membership in both sets of each element. Let A(x) and B(x) are two fuzzy sets, the intersection of is denoted by (A∩B)(x) and the membership function value is given as follows
μ (A∩B)(x)= min{μA(x),μB(x)}
Intersection is analogous to logical AND operation

4. Union of fuzzy sets

Union of fuzzy sets consists of every element that falls into either set. The value of the membership value is will be the largest membership value of the element in either set
Let A(x) and B(x) are two fuzzy sets for all x ∈ X, Union of fuzzy sets is denoted by (AUB)(x) and the membership function value is determined as follows μ (AUB)(x)= max{μA(x),μB(x)}

Fuzzy inference system

Fuzzy Inference System is the key unit of a fuzzy logic system having decision making as its primary work. It uses the “IF…THEN” rules along with connectors “OR” or “AND” for drawing essential decision rules.
Characteristics of Fuzzy Inference System:

- The output from FIS is always a fuzzy set irrespective of its input which can be fuzzy or crisp.
- It is necessary to have fuzzy output when it is used as a controller.

A defuzzification unit would be there with FIS to convert fuzzy variables into crisp variable

Implementation Details:

1. Implement the following membership functions and visualize the same

a. Triangular.
b. Trapezoidal
c. Gaussian
d. Generalized
e. Sigmoid

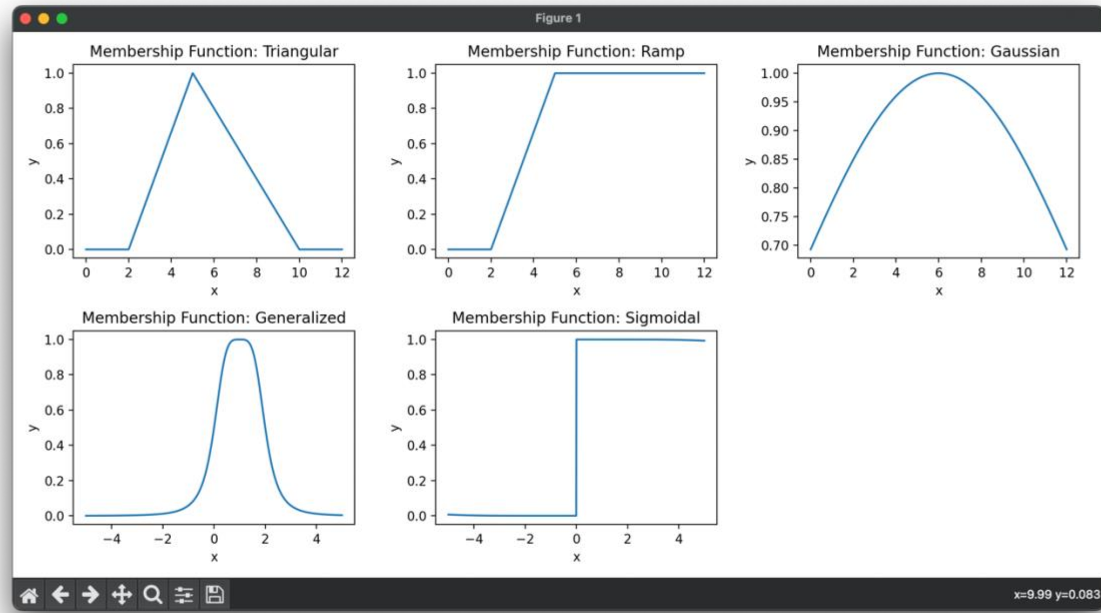
```
import matplotlib.pyplot as plt
import numpy as np
import math

def plotGraph(x, y, title):
    plt.plot(x, y)
    plt.xlabel("x")
    plt.ylabel("y")
    plt.title(f'Membership Function: {title}')
    plt.show()

def triangular(a, b, c, X):
    Y = []
    for x in X:
        if x < a:
            Y.append(0)
        elif x >= c:
            Y.append(0)
        elif a <= x < b:
            Y.append((x - a) / (b - a))
        elif b <= x < c:
            Y.append((c - x) / (c - b))
    return Y
```

```
def ramp(a, b, X):
    Y = []
    for x in X:
        if x <= a:
            Y.append(0)
        elif a < x <= b:
            Y.append((x - a) / (b - a))
        else:
            Y.append(1)
    return Y

def gaussian(arr, c, sigma):
    Y = []
    for x in arr:
        out = math.e ** (-0.5 * (((x - c) / sigma) ** 2))
        Y.append(out)
    return Y

def generalized(arr, a, b, c):
    Y = []
    for x in arr:
        out = 1 / (1 + (abs((x - c) / a)) ** (2 * b))
        Y.append(out)
    return Y

def sigmoidal(arr, a, c):
    Y = []
    for x in arr:
```

```python
        out = 1 / (1 + (math.e ** (-a / (x - c))) )
        Y.append(out)
    return Y

def plotGraph(x, y, title, position):
    plt.subplot(2, 3, position)
    plt.plot(x, y)
    plt.xlabel("x")
    plt.ylabel("y")
    plt.title(f'Membership Function: {title}')

X = np.linspace(0, 12, 1000)

Y1 = triangular(2, 5, 10, X)
Y2 = ramp(2, 5, X)
Y3 = gaussian(X, 6, 7)

X2 = np.linspace(-5, 5, 1000)
Y4 = generalized(X2, 1, 2, 1)
Y5 = sigmoidal(X2, 25, 0)

plt.figure(figsize=(12, 6))

plotGraph(X, Y1, 'Triangular', 1)
plotGraph(X, Y2, 'Ramp', 2)
plotGraph(X, Y3, 'Gaussian', 3)
plotGraph(X2, Y4, 'Generalized', 4)
plotGraph(X2, Y5, 'Sigmoidal', 5)
```

```
plt.tight_layout()  # Ensure proper spacing between subplots
plt.show()
```

Output:



2. Implement the fuzzy operations- Union, intersection, compliment and visualize the same.

```
import matplotlib.pyplot as plt

A = {"a": 0.6, "b": 0.3, "c": 0.4, "d": 0.9}
B = {"a": 0.3, "b": 0.9, "c": 0.5, "d": 0.7}

print("Fuzzy Set A:", A)
print("Fuzzy Set B:", B)
```

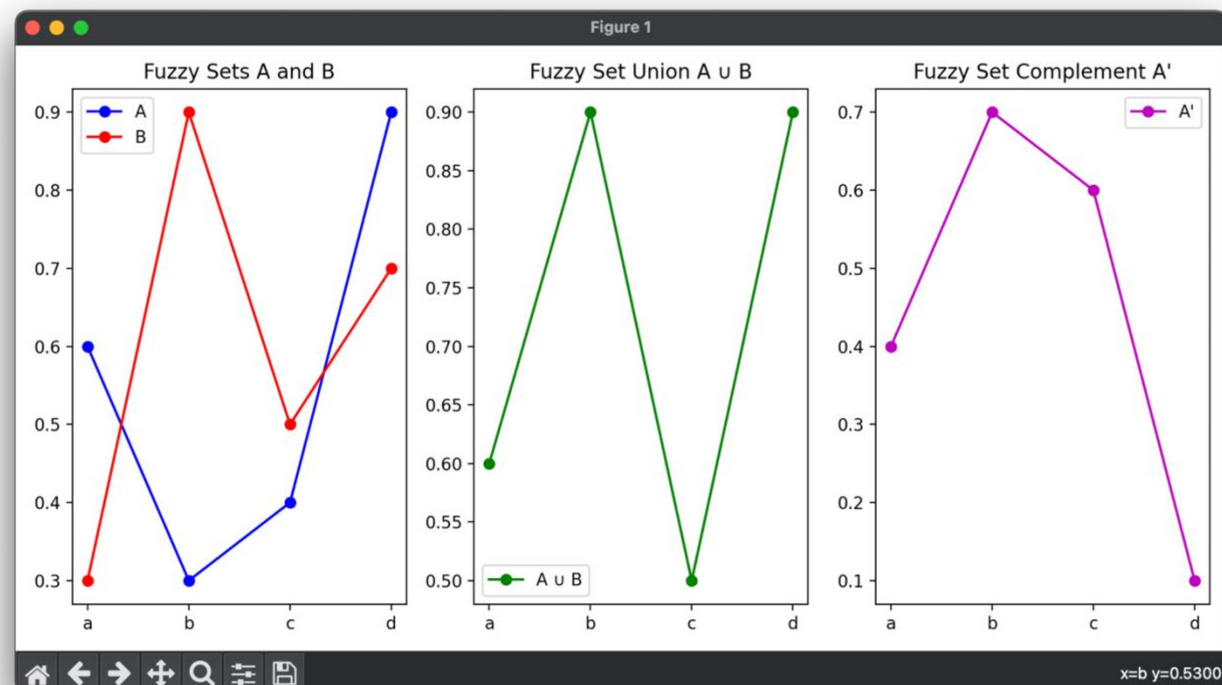
```
union = dict()
intersect = dict()
complement = dict()

# Union
for key in A:
    if key in B:
        a_val = A[key]
        b_val = B[key]
        union[key] = max(a_val, b_val)
    else:
        union[key] = A[key]

print("Fuzzy Set Union of A and B: ", union)

# Intersection
for key in A:
    if key in B:
        a_val = A[key]
        b_val = B[key]
        intersect[key] = min(a_val, b_val)

print("Fuzzy Set Intersection of A and B: ", intersect)

# Complement
for key in A:
    a_val = A[key]
    complement[key] = round((1 - a_val), 1)
```

```
print("Fuzzy Set Complement of A: ", complement)

# Visualize Fuzzy Sets and Operations
x = list(A.keys())

y_A = list(A.values())
y_B = list(B.values())
y_union = list(union.values())
y_intersect = list(intersect.values())
y_complement = list(complement.values())

plt.figure(figsize=(10, 5))

plt.subplot(131)
plt.plot(x, y_A, marker='o', label='A', color='b')
plt.plot(x, y_B, marker='o', label='B', color='r')
plt.title("Fuzzy Sets A and B")
plt.legend()

plt.subplot(132)
plt.plot(x, y_union, marker='o', label='A ∪ B', color='g')
plt.title("Fuzzy Set Union A ∪ B")
plt.legend()

plt.subplot(133)
plt.plot(x, y_complement, marker='o', label="A'", color='m')
plt.title("Fuzzy Set Complement A'")
```

```
plt.legend()


plt.tight_layout()

plt.show()
```

Output:

```
❯ python3 -u "/Users/pargatsinghdhanjal/Desktop/Soft Computing/tempCodeRunnerFile.py"
Fuzzy Set A: {'a': 0.6, 'b': 0.3, 'c': 0.4, 'd': 0.9}
Fuzzy Set B: {'a': 0.3, 'b': 0.9, 'c': 0.5, 'd': 0.7}
Fuzzy Set Union of A and B:  {'a': 0.6, 'b': 0.9, 'c': 0.5, 'd': 0.9}
Fuzzy Set Intersection of A and B:  {'a': 0.3, 'b': 0.3, 'c': 0.4, 'd': 0.7}
Fuzzy Set Complement of A:  {'a': 0.4, 'b': 0.7, 'c': 0.6, 'd': 0.1}
2023-10-19 14:33:49.315 Python[5562:6124580] WARNING: Secure coding is automatically en
of this application. Opt-in to secure coding explicitly by implementing NSApplicationDe
```



Post Lab Descriptive Questions :

1. Implement the fuzzy membership functions to define the fuzzy set for the following course evaluation crisp set
   (Hint: descriptors for fuzzy set can be bad, poor, average, good, very good, excellent)

   $EX = Marks \geq 90$

   $A = 80 \leq Marks < 90$

   $B = 70 \leq Marks < 80$

   $C = 60 \leq Marks < 70$

   $D = 50 \leq Marks < 60$

   $P = 35 \leq Marks < 50$

   $F = Marks < 35$

```
def marks(arr):
  o = []
  for mark in arr:
    if mark >= 90:
      o.append("Excellent")
    elif mark >= 80:
      o.append("Very good")
    elif mark >= 70:
      o.append("Good")
    elif mark >= 60:
      o.append("Average")
    elif mark >= 50:
      o.append("Poor")
    elif mark >= 35:
      o.append("Bad")
    else:
```

```
        o.append("Fail")
    return o

arr = [80, 50, 12, 45, 12, 35, 78]
res = marks(arr)

print(arr)
print(res)
```

Output

```
❯ python3 -u "/Users/pargatsinghdhanjal/Desktop/Soft Computir
[80, 50, 12, 45, 12, 35, 78]
['Very good', 'Poor', 'Fail', 'Bad', 'Fail', 'Bad', 'Good']
```

Date: ________

Signature of faculty in-charge