Practical No: 03
Title:
Design and implement Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relations by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.

Problem Statement:
Design and implement Union, Intersection, Complement, and Difference operations on fuzzy sets. Create fuzzy relations by the Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.

Objective:
1. To implement Union, Intersection, Complement, and Difference operations on fuzzy sets.
2. To create fuzzy relations using the Cartesian product of fuzzy sets and perform max-min composition on fuzzy relations.
3. To demonstrate the application of these operations in fuzzy logic.

Outcome:
1. Successfully implement Union, Intersection, Complement, and Difference operations on fuzzy sets.

2. Create fuzzy relations using the Cartesian product of fuzzy sets and perform max-min composition on fuzzy relations.

Software Requirement
- Python (3.x recommended)
- Jupyter Notebook or any Python IDE

Hardware Requirement
A machine with sufficient RAM and processing power for model training (8GB RAM recommended)

Prerequisites:

What is Fuzzy Set ?

Fuzzy refers to something that is unclear or vague . Hence, Fuzzy Set is a Set where every key is associated with value, which is between 0 to 1 based on the certainty. This value is often called as degree of membership. Fuzzy Set is denoted with a Tilde Sign on top of the normal Set notation.

Operations on Fuzzy Set with Code:
1. Union :
Operations on Fuzzy Set with Code :
1. Union : Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Union of them, then for every member of A and B, Y will be:
degree_of_membership(Y)= max(degree_of_membership(A), degree_of_membership(B))

EXAMPLE:

```
# Example to Demonstrate the
# Union of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()
A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}
print('The First Fuzzy Set is :', A)
print('The Second Fuzzy Set is :', B)
for A_key, B_key in zip(A, B):
A_value = A[A_key]
B_value = B[B_key]
if A_value > B_value:
  Y[A_key] = A_value
else:
  Y[B_key] = B_value
print('Fuzzy Set Union is :’, Y)
```

Output

```
The First Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is : {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Union is : {'a': 0.9, 'b': 0.9, 'c': 0.6, 'd': 0.6}
```

3.Intersection:

Consider 2 Fuzzy Sets denoted by A and B, then let’s consider Y be the Intersection of them, then for every member of A and B, Y will be:
degree_of_membership(Y)= min(degree_of_membership(A), degree_of_membership(B))

EXAMPLE :

```
# Example to Demonstrate
# Intersection of Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()
A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}
print('The First Fuzzy Set is :', A)
print('The Second Fuzzy Set is :', B)
for A_key, B_key in zip(A, B):
  A_value = A[A_key]
  B_value = B[B_key]
if A_value < B_value:
  Y[A_key] = A_value
else:
  Y[B_key] = B_value

print('Fuzzy Set Intersection is :', Y)
```

Output

```
The First Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is : {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Fuzzy Set Intersection is : {'a': 0.2, 'b': 0.3, 'c': 0.4, 'd': 0.5}
```

3. Complement:
Consider a Fuzzy Sets denoted by A , then let’s consider Y be the Complement of it, then for every member of A , Y will be:
degree_of_membership(Y)= 1 - degree_of_membership(A)
EXAMPLE :

```
# Example to Demonstrate the
# Difference Between Two Fuzzy Sets
A = dict()
Y = dict()
A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
print('The Fuzzy Set is :', A)
for A_key in A:
  Y[A_key]= 1-A[A_key]
print('Fuzzy Set Complement is :', Y)
```

Output

```
The Fuzzy Set is : {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
Fuzzy Set Complement is : {'a': 0.8, 'b': 0.7, 'c': 0.4, 'd': 0.4}
```

4.Difference:

Consider 2 Fuzzy Sets denoted by A and B, then let’s consider Y be the Intersection of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= min(degree_of_membership(A), 1- degree_of_membership(B))

EXAMPLE:

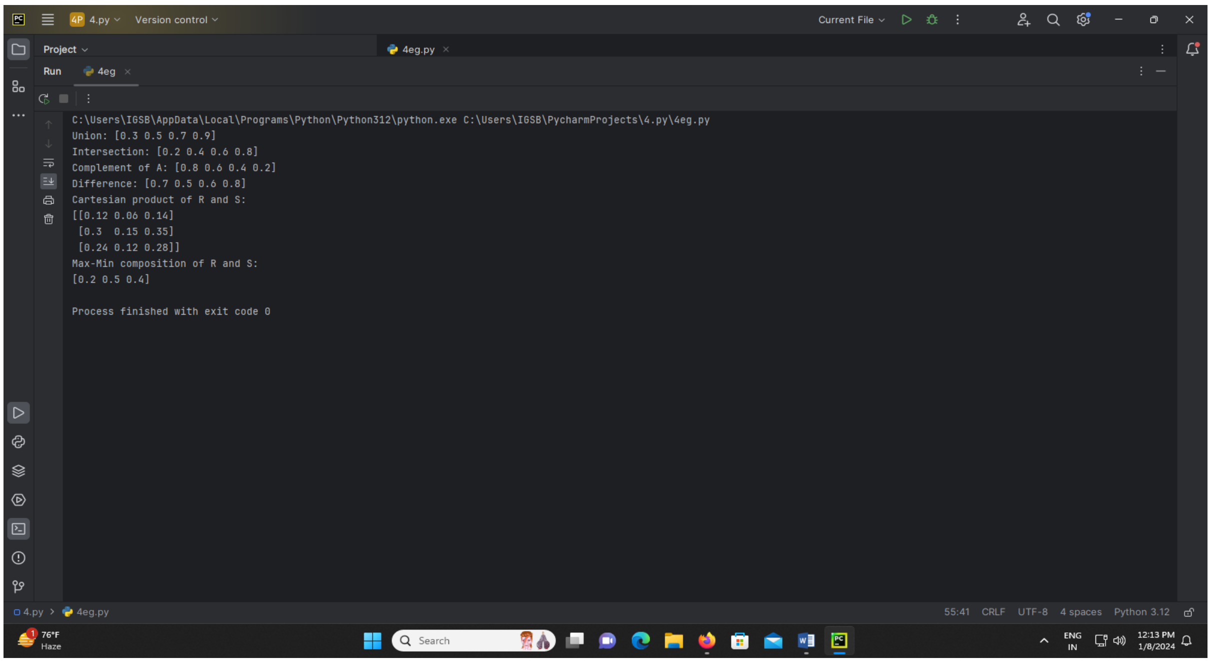
```
# Example to Demonstrate the
# Difference Between Two Fuzzy Sets
A = dict()
B = dict()
Y = dict()
A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}
print('The First Fuzzy Set is :', A)
print('The Second Fuzzy Set is :', B)
for A_key, B_key in zip(A, B):
 A_value = A[A_key]
 B_value = B[B_key]
 B_value = 1 - B_value
if A_value < B_value:
      Y[A_key] = A_value
else:
       Y[B_key] = B_value
print('Fuzzy Set Difference is :', Y)
```

Implement Union, Intersection, Complement and Difference operations on fuzzy sets. Also create fuzzy relations by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.

CODE:

```
import numpy as np

 # Function to perform Union operation on fuzzy sets

def fuzzy_union(A, B):

   return np.maximum(A, B)


# Function to perform Intersection operation on fuzzy sets

def fuzzy_intersection(A, B):

   return np.minimum(A, B)


# Function to perform Complement operation on a fuzzy set

def fuzzy_complement(A):

   return 1 - A


# Function to perform Difference operation on fuzzy sets

def fuzzy_difference(A, B):

   return np.maximum(A, 1 - B)


# Function to create fuzzy relation by Cartesian product of two fuzzy sets

def cartesian_product(A, B):

   return np.outer(A, B)


# Function to perform Max-Min composition on two fuzzy relations

def max_min_composition(R, S):

   return np.max(np.minimum.outer(R, S), axis=1)


# Example usage

A = np.array([0.2, 0.4, 0.6, 0.8])  # Fuzzy set A

B = np.array([0.3, 0.5, 0.7, 0.9])  # Fuzzy set B
```

```
# Operations on fuzzy sets
union_result = fuzzy_union(A, B)
intersection_result = fuzzy_intersection(A, B)
complement_A = fuzzy_complement(A)
difference_result = fuzzy_difference(A, B)

print("Union:", union_result)
print("Intersection:", intersection_result)
print("Complement of A:", complement_A)
print("Difference:", difference_result)

# Fuzzy relations
R = np.array([0.2, 0.5, 0.4])  # Fuzzy relation R
S = np.array([0.6, 0.3, 0.7])  # Fuzzy relation S

# Cartesian product of fuzzy relations
cartesian_result = cartesian_product(R, S)

# Max-Min composition of fuzzy relations
composition_result = max_min_composition(R, S)

print("Cartesian product of R and S:")
print(cartesian_result)

print("Max-Min composition of R and S:")
print(composition_result)
```



```
C:\Users\IGSB\AppData\Local\Programs\Python\Python312\python.exe C:\Users\IGSB\PycharmProjects\4.py\4eg.py
Union: [0.3 0.5 0.7 0.9]
Intersection: [0.2 0.4 0.6 0.8]
Complement of A: [0.8 0.6 0.4 0.2]
Difference: [0.7 0.5 0.6 0.8]
Cartesian product of R and S:
[[0.12 0.06 0.14]
 [0.3  0.15 0.35]
 [0.24 0.12 0.28]]
Max-Min composition of R and S:
[0.2 0.5 0.4]

Process finished with exit code 0
```

USE PYCHARM IDE

INSTALL NUMPY PACKAGE

Conclusion: Fuzzy logic provides a powerful framework for dealing with uncertainty and imprecision in data. By implementing operations on fuzzy sets and relations, we can model complex relationships and make more flexible decisions in various applications such as control systems, artificial intelligence, and pattern recognition. The Union, Intersection, Complement, and Difference operations allow us to manipulate fuzzy sets, while fuzzy relations and max-min composition enable us to represent and combine fuzzy relationships between sets. These operations and techniques expand the capabilities of traditional logic and set theory, offering a more nuanced approach to handling vague and ambiguous information