Assignment No. 1

Title: Implementation of Mathematical Operations on Fuzzy Sets and Fuzzy Relations with Max–Min Composition

Introduction:

Fuzzy set theory is an extension of classical set theory where elements have degrees of membership ranging from 0 to 1, instead of a crisp belonging (either 0 or 1). This concept is widely used in decision-making, artificial intelligence, and control systems where uncertainties and vagueness exist. In this assignment, we implement basic operations on fuzzy sets, create fuzzy relations using the Cartesian product, and perform max-min composition on fuzzy relations.

Mathematical Operations on Fuzzy Sets

Let two fuzzy sets A and B be defined on a common universe X. The membership function of each element $x \in X$ is denoted by $\mu A(x)$ and $\mu B(x)$. The basic fuzzy set operations are:

1. Union: $\mu A \cup B(x) = \max(\mu A(x), \mu B(x))$
2. Intersection: $\mu A \cap B(x) = \min(\mu A(x), \mu B(x))$
3. Complement: $\mu A'(x) = 1 - \mu A(x)$

Fuzzy Relations and Cartesian Product

A fuzzy relation R between two fuzzy sets A (on universe X) and B (on universe Y) is represented by the Cartesian product:

$R(x,y) = \min(\mu A(x), \mu B(y))$ for all $(x,y) \in X \times Y$

This produces a fuzzy relation matrix where each entry represents the degree of association between elements of X and Y.

Max-Min Composition of Fuzzy Relations

Given two fuzzy relations R1 (between sets X and Y) and R2 (between sets Y and Z), the max-min composition $R = R1 \circ R2$ is defined as:

$R(x,z) = \max\_y [\min(R1(x,y), R2(y,z))]$

This operation combines two relations into a single relation representing indirect association between X and Z via Y.

Example Implementation

Consider fuzzy sets A and B defined over the universe {1, 2, 3}:

A = { (1, 0.2), (2, 0.7), (3, 1.0) }

B = { (1, 0.5), (2, 0.4), (3, 0.9) }

Union: $\mu A \cup B$ = { (1, 0.5), (2, 0.7), (3, 1.0) }

Intersection: $\mu A \cap B$ = { (1, 0.2), (2, 0.4), (3, 0.9) }

Complement of A: { (1, 0.8), (2, 0.3), (3, 0.0) }

Fuzzy relation R from A × B:

R = [[min(0.2,0.5), min(0.2,0.4), min(0.2,0.9)],

[min(0.7,0.5), min(0.7,0.4), min(0.7,0.9)],

[min(1.0,0.5), min(1.0,0.4), min(1.0,0.9)]]

Max-min composition can then be applied between two relations R1 and R2 to compute indirect associations.

Results:

- The program successfully computed fuzzy union, intersection, and complement.

- Fuzzy relations were constructed as a matrix-like structure.

- Max–min composition produced another fuzzy relation that combined the two given relations.

Conclusion:

This assignment demonstrated the practical implementation of fuzzy set operations and fuzzy relations. The results illustrate how fuzzy logic handles uncertainty and provides a foundation for applications in decision-making, expert systems, and artificial intelligence.

CODE –

```
import numpy as np

# ----- Define Fuzzy Sets -----
X = [1, 2, 3, 4, 5]
A = {1: 0.2, 2: 0.7, 3: 1.0, 4: 0.4, 5: 0.1}
B = {1: 0.6, 2: 0.2, 3: 0.9, 4: 0.3, 5: 0.5}

# ----- Fuzzy Set Operations -----
def union(A, B):
    return {x: max(A[x], B[x]) for x in A}

def intersection(A, B):
    return {x: min(A[x], B[x]) for x in A}

def complement(A):
    return {x: 1 - A[x] for x in A}

print("Union:", union(A,B))
print("Intersection:", intersection(A,B))
print("Complement of A:", complement(A))

# ----- Cartesian Product (Fuzzy Relation) -----
def cartesian(A, B):
    return {(x, y): min(A[x], B[y]) for x in A for y in B}

R1 = cartesian(A, B)
R2 = cartesian(B, A)  # another relation

print("\nFuzzy Relation R1 (A x B):")
for pair, val in R1.items():
    print(pair, ":", val)
print("\nFuzzy Relation R2 (B x A):")
for pair, val in R2.items():
    print(pair, ":", val)

# ----- Max-Min Composition -----
def max_min_composition(R1, R2, A_set, B_set, C_set):
    result = {}
    for x in A_set:
        for z in C_set:
            values = []
            for y in B_set:
                values.append(min(R1[(x,y)], R2[(y,z)]))
            result[(x,z)] = max(values)
    return result

Comp = max_min_composition(R1, R2, X, X, X)

print("\nMax-Min Composition (R1 ∘ R2):")
for pair, val in Comp.items():
    print(pair, ":", val)
```

OUTPUT –

```
Union: {1: 0.6, 2: 0.7, 3: 1.0, 4: 0.4, 5: 0.5}
Intersection: {1: 0.2, 2: 0.2, 3: 0.9, 4: 0.3, 5: 0.1}
```

Complement of A: {1: 0.8, 2: 0.30000000000000004, 3: 0.0, 4: 0.6, 5: 0.9}

Fuzzy Relation R1 (A x B):
(1, 1) : 0.2
(1, 2) : 0.2
(1, 3) : 0.2
(1, 4) : 0.2
(1, 5) : 0.2
(2, 1) : 0.6
(2, 2) : 0.2
(2, 3) : 0.7
(2, 4) : 0.3
(2, 5) : 0.5
(3, 1) : 0.6
(3, 2) : 0.2
(3, 3) : 0.9
(3, 4) : 0.3
(3, 5) : 0.5
(4, 1) : 0.4
(4, 2) : 0.2
(4, 3) : 0.4
(4, 4) : 0.3
(4, 5) : 0.4
(5, 1) : 0.1
(5, 2) : 0.1
(5, 3) : 0.1
(5, 4) : 0.1
(5, 5) : 0.1

Fuzzy Relation R2 (B x A):
(1, 1) : 0.2
(1, 2) : 0.6
(1, 3) : 0.6
(1, 4) : 0.4
(1, 5) : 0.1
(2, 1) : 0.2
(2, 2) : 0.2
(2, 3) : 0.2
(2, 4) : 0.2
(2, 5) : 0.1
(3, 1) : 0.2
(3, 2) : 0.7
(3, 3) : 0.9
(3, 4) : 0.4
(3, 5) : 0.1
(4, 1) : 0.2
(4, 2) : 0.3
(4, 3) : 0.3
(4, 4) : 0.3
(4, 5) : 0.1
(5, 1) : 0.2
(5, 2) : 0.5
(5, 3) : 0.5
(5, 4) : 0.4
(5, 5) : 0.1