AIDS-2 LAB
EXP 7

Vaibhav Boudh
D20B   Roll No: 06

Aim: To implement fuzzy set Properties.

Theory:
Fuzzy set theory is an extension of classical set theory in which elements can have varying degrees of membership. Introduced by Lotfi A. Zadeh in 1965, this approach offers a mathematical framework for representing uncertainty in real-world problems. Unlike classical sets, where an element either fully belongs or does not belong (binary logic), fuzzy sets allow for partial membership. This partial membership is expressed through a membership function that assigns a value between 0 and 1 to each element.

Key Concepts:

1. Membership Function: A function that maps each point in the input space to a membership value (or degree of membership) between 0 and 1, represented as μ(x), where x is an element in the universe of discourse U.
2. Universe of Discourse: The set of all possible elements being considered.
3. Support: The set of all elements in the fuzzy set with a non-zero degree of membership.
4. Core: The set of all elements in the fuzzy set with a full degree of membership (i.e., μ(x) = 1).
5. Boundary: The set of all elements in the fuzzy set with a degree of membership greater than zero but less than one.

Properties of Fuzzy Sets:

1. Involution: Involution states that the complement of complement is set itself.

$(\underline{A}')' = \underline{A}$

2. Commutativity: Operations are called commutative if the order of operands does not alter the result. Fuzzy sets are commutative under union and intersection operations.

$\underline{A} \cup \underline{B} = \underline{B} \cup \underline{A}$

$\underline{A} \cap \underline{B} = \underline{B} \cap \underline{A}$

3. Associativity: Associativity allows to change the order of operations performed on operand, however relative order of operand can not be changed. All sets in the equation must appear in the identical order only. Fuzzy sets are associative under union and intersection operations.

$$\underline{A} \cup (\underline{B} \cup \underline{C}) = (\underline{A} \cup \underline{B}) \cup \underline{C}$$

$$\underline{A} \cap (\underline{B} \cap \underline{C}) = (\underline{A} \cap \underline{B}) \cap \underline{C}$$

4. Distributivity

$$\underline{A} \cup (\underline{B} \cap \underline{C}) = (\underline{A} \cup \underline{B}) \cap (\underline{A} \cup \underline{C})$$

$$\underline{A} \cap (\underline{B} \cup \underline{C}) = (\underline{A} \cap \underline{B}) \cup (\underline{A} \cap \underline{C})$$

5. Absorption: Absorption produces the identical sets after stated union and intersection operations.

$$\underline{A} \cup (\underline{A} \cap \underline{B}) = \underline{A}$$

$$\underline{A} \cap (\underline{A} \cup \underline{B}) = \underline{A}$$

6. Idempotency: Idempotency does not alter the element or the membership value of elements in the set.

$$\underline{A} \cup \underline{A} = \underline{A}$$

$$\underline{A} \cap \underline{A} = \underline{A}$$

7. Identity

$$\underline{A} \cup \phi = \underline{A}$$

$$\underline{A} \cap \phi = \phi$$

$$\underline{A} \cup X = X$$

$$\underline{A} \cap X = \underline{A}$$

8. Transitivity: If A ⊆ B and B ⊆ C then A ⊆ C

9. De Morgan's Law: De Morgan's Laws can be stated as the complement of a union is the intersection of the complement of individual sets and the complement of an intersection is the union of the complement of individual sets.

$$(\underline{A} \cup \underline{B})' = \underline{A}' \cap \underline{B}'$$

$$(\underline{A} \cap \underline{B})' = \underline{A}' \cup \underline{B}'$$

Implementation:

1) **Install the required libraries**

   !pip install scikit-fuzzy

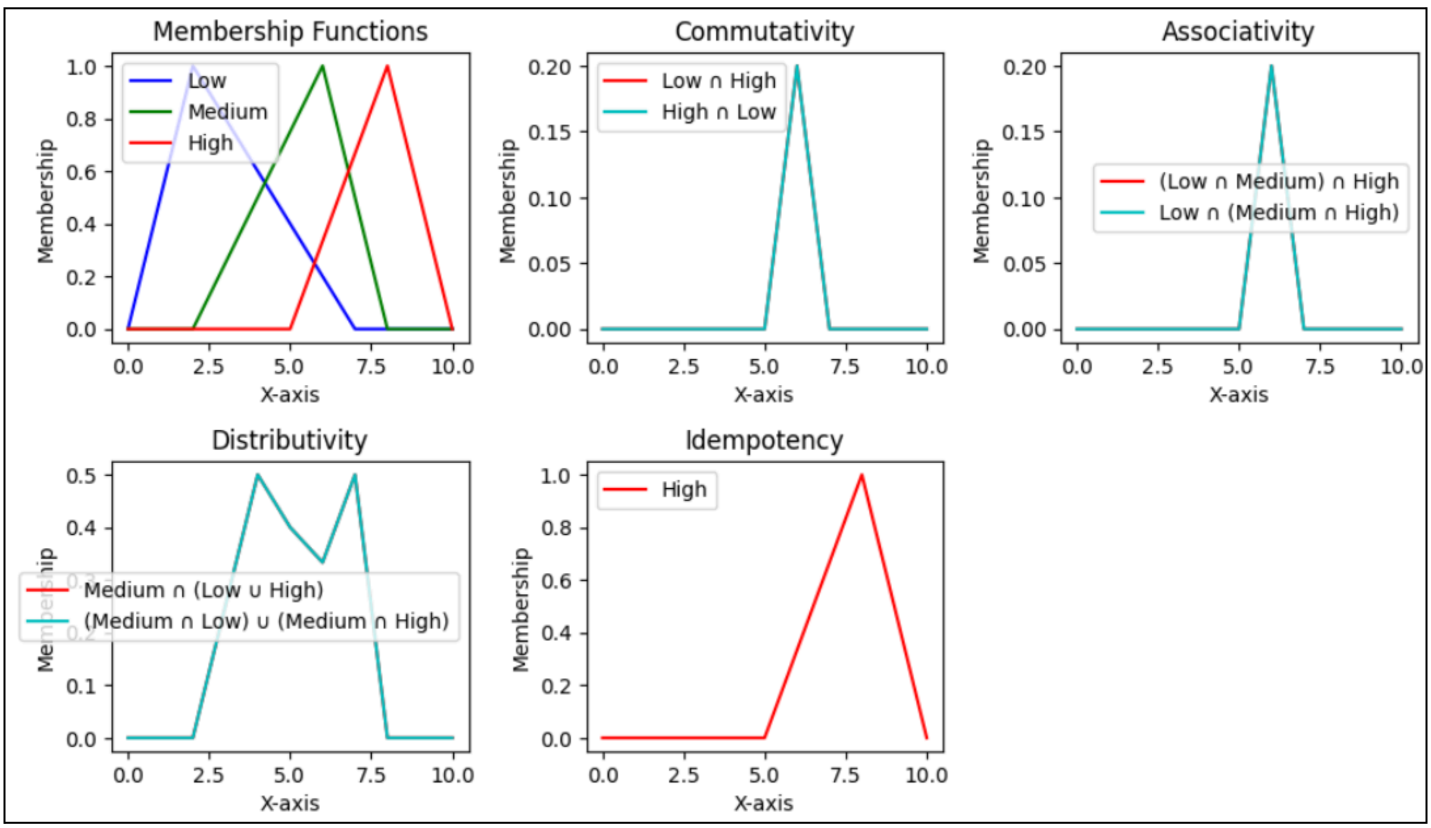
2) **Main logic**

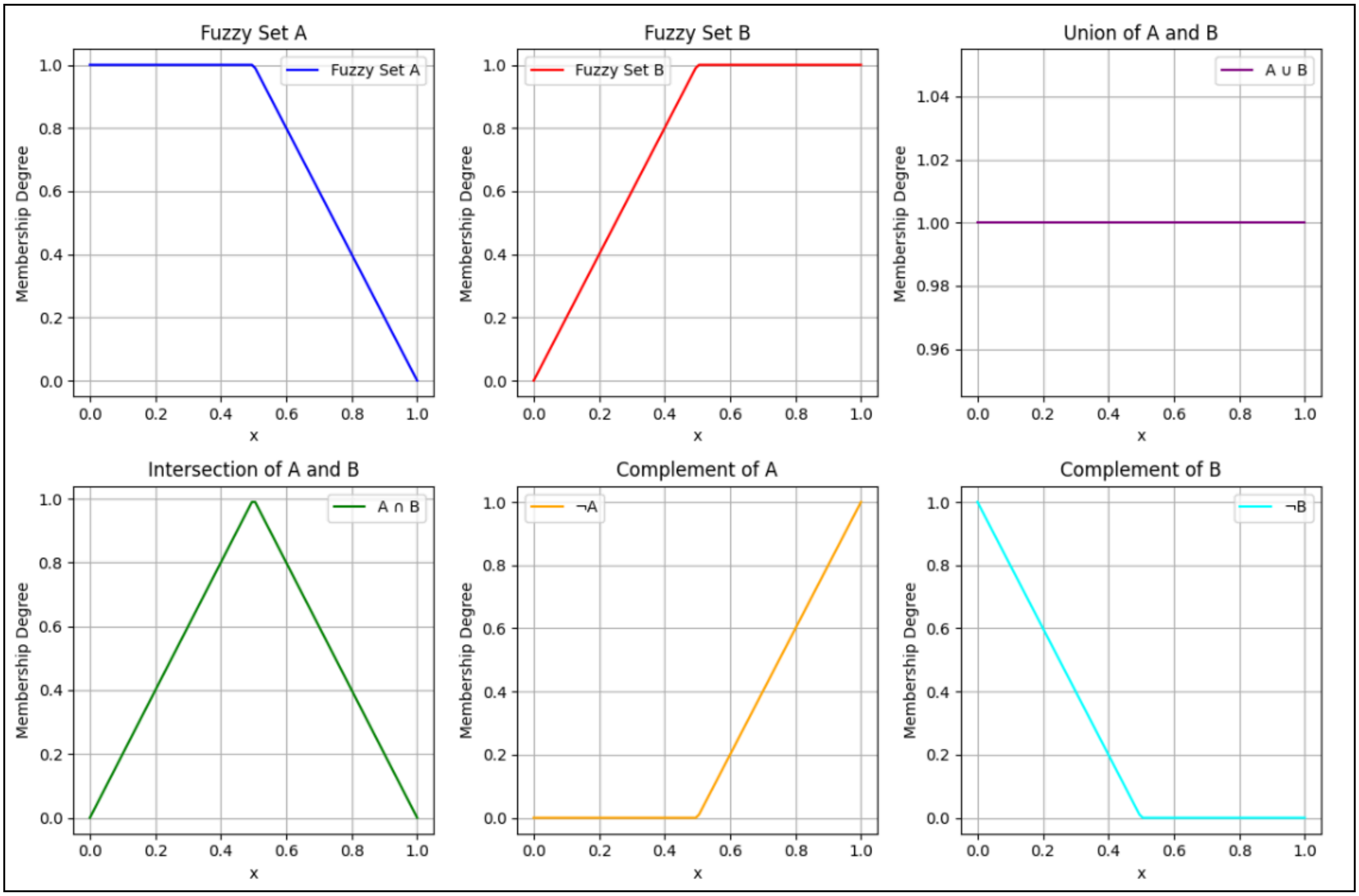
```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt

# Create a range of values for the x-axis
x = np.arange(0, 11, 1)

# Create membership functions
low = fuzz.trimf(x, [0, 2,7])
medium = fuzz.trimf(x, [2, 6, 8])
high = fuzz.trimf(x, [5, 8, 10])

# Fuzzy logic operations with modified values
def fuzzy_commutativity(A, B):
    return np.minimum(A, B), np.minimum(B, A)

def fuzzy_associativity(A, B, C):
    left = np.minimum(np.minimum(A, B), C)
    right = np.minimum(A, np.minimum(B, C))
    return left, right

def fuzzy_distributivity(A, B, C):
    left = np.maximum(np.minimum(A, B), np.minimum(A, C))
    right = np.minimum(A, np.maximum(B, C))
    return left, right

def fuzzy_idempotency(A):
    return A

# Apply fuzzy logic operations to membership functions
result1, result2 = fuzzy_commutativity(low, high)
result3, result4 = fuzzy_associativity(low, medium, high)
result5, result6 = fuzzy_distributivity(medium, low, high)
result7 = fuzzy_idempotency(high)

plt.figure(figsize=(10, 8))
plt.subplot(3, 3, 1)
plt.plot(x, low, 'b', linewidth=1.5, label='Low')
plt.plot(x, medium, 'g', linewidth=1.5, label='Medium')
plt.plot(x, high, 'r', linewidth=1.5, label='High')
plt.title('Membership Functions')
plt.xlabel('X-axis')
plt.ylabel('Membership')
```

```
plt.legend()

plt.subplot(3, 3, 2)
plt.plot(x, result1, 'r', linewidth=1.5, label='Low ∩ High')
plt.plot(x, result2, 'c', linewidth=1.5, label='High ∩ Low')
plt.title('Commutativity')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()

plt.subplot(3, 3, 3)
plt.plot(x, result3, 'r', linewidth=1.5, label='(Low ∩ Medium) ∩ High')
plt.plot(x, result4, 'c', linewidth=1.5, label='Low ∩ (Medium ∩ High)')
plt.title('Associativity')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()

plt.subplot(3, 3, 4)
plt.plot(x, result5, 'r', linewidth=1.5, label='Medium ∩ (Low ∪ High)')
plt.plot(x, result6, 'c', linewidth=1.5, label='(Medium ∩ Low) ∪ (Medium ∩ High)')
plt.title('Distributivity')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()

plt.subplot(3, 3, 5)
plt.plot(x, result7, 'r', linewidth=1.5, label='High')
plt.title('Idempotency')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()

# Define the universe of discourse
x = np.linspace(0, 1, 100)

# Define two fuzzy sets A and B
def fuzzy_set_A(x):
    return np.maximum(0, np.minimum(1, (1 - x) * 2))

def fuzzy_set_B(x):
    return np.maximum(0, np.minimum(1, (x * 2)))

# Calculate fuzzy sets A and B
A = fuzzy_set_A(x)
B = fuzzy_set_B(x)

# Union of A and B
```

```
A_union_B = np.maximum(A, B)

# Intersection of A and B
A_intersection_B = np.minimum(A, B)

# Complement of A and B
A_complement = 1 - A
B_complement = 1 - B

# Plotting
plt.figure(figsize=(12, 8))

# Plot fuzzy set A
plt.subplot(2, 3, 1)
plt.plot(x, A, label='Fuzzy Set A', color='blue')
plt.title('Fuzzy Set A')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()

# Plot fuzzy set B
plt.subplot(2, 3, 2)
plt.plot(x, B, label='Fuzzy Set B', color='red')
plt.title('Fuzzy Set B')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()

# Plot union of A and B
plt.subplot(2, 3, 3)
plt.plot(x, A_union_B, label='A ∪ B', color='purple')
plt.title('Union of A and B')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()

# Plot intersection of A and B
plt.subplot(2, 3, 4)
plt.plot(x, A_intersection_B, label='A ∩ B', color='green')
plt.title('Intersection of A and B')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()

# Plot complement of A
```

```
plt.subplot(2, 3, 5)
plt.plot(x, A_complement, label='¬A', color='orange')
plt.title('Complement of A')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()

# Plot complement of B
plt.subplot(2, 3, 6)
plt.plot(x, B_complement, label='¬B', color='cyan')
plt.title('Complement of B')
plt.xlabel('x')
plt.ylabel('Membership Degree')
plt.grid(True)
plt.legend()


plt.tight_layout()
plt.show()
```





Conclusion: Thus we have understood about the Fuzzy set and implemented its properties.