Vivekanand Education Society's
Institute of Technology
(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)
Department of Information Technology

AIDS - 2 Lab
Experiment - 7

Aim: **To implement fuzzy set Properties**

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D20B
Subject	AIDS - 2
Grade:	

EXPERIMENT - 7

AIM: To implement fuzzy set Properties

THEORY:

Fuzzy sets are a fundamental concept in fuzzy logic, a branch of mathematics and artificial intelligence that deals with reasoning and decision-making under uncertainty and imprecision. Fuzzy sets provide a way to represent and work with degrees of membership or truth rather than strict binary values (true or false).In a traditional (crisp) set, an element either belongs to the set (true) or does not belong to the set (false). However, in a fuzzy set, each element can have a membership value between 0 and 1, indicating the degree to which it belongs to the set. These membership values represent the "fuzziness" or uncertainty associated with whether an element is part of the set.

Properties of Fuzzy Sets:

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

   $$(\underset{\sim}{A}')' = \underset{\sim}{A}$$

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.

   $$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup \underset{\sim}{A}$$

   $$\underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{B} \cap \underset{\sim}{A}$$

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.

   $$\underset{\sim}{A} \cup (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cup \underset{\sim}{C}$$

   $$\underset{\sim}{A} \cap (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cap \underset{\sim}{C}$$

4. **Distributivity**: Distributivity refers to a set of properties or rules that describe how fuzzy logic operators, such as AND (∧) and OR (∨), interact with each other.

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

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

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 $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$

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:

```
from copy import deepcopy
def checkElementHelper(x, S):
  for e in S:
    if x == e[1]: return e[0]
  return 0

global ClassA
ClassA=[[0.5,"Sakshi"], [0.1,"Mayuri"], [0.5,"Pushkaraj"], [0.4,"Aaman"], [0.5,"Prerna"]]

global ClassB
ClassB=[[0.2,"Sakshi"], [0.9,"Mayuri"], [0.3,"Pushkaraj"], [0.6,"Aaman"], [0.7,"Prerna"]]
```

A) Union[A(x), B(x)] = max(A (x) ,B (x))

```
#Union

def union(setA, setB):
  X, Y = deepcopy(setA), deepcopy(setB)
  Z = []
  for i in X:
    mb = checkElementHelper(i[1], Y)
    Z.append([max(mb, i[0]), i[1]])
    if mb != 0:
      Y.remove([mb, i[1]])
  Z = Z + Y
  return Z

print("Union Property", union(ClassA, ClassB))
```

```
Union Property [[0.5, 'Sakshi'], [0.9, 'Mayuri'], [0.5, 'Pushkaraj'], [0.6, 'Aaman'], [0.7, 'Prerna']]
```

B) Intersection[A(x), B(x)] = min(A (x) ,B (x))

```
#Intersection

def intersection(setA, setB):
  X, Y = deepcopy(setA), deepcopy(setB)
  Z = []
  for i in X:
    mb = checkElementHelper(i[1], Y)
    if min(mb, i[0]) != 0:
      Z.append([min(mb, i[0]), i[1]])
  return Z

print("Intersection Property", intersection(ClassA, ClassB))
```

```
Intersection Property [[0.2, 'Sakshi'], [0.1, 'Mayuri'], [0.3, 'Pushkaraj'], [0.4, 'Aaman'], [0.5, 'Prerna']]
```

C) Complement[A(x)] = 1-A(x)

```
#Complement

def complement(setA):
  Z = deepcopy(setA)
  for i in Z:
    i[0] = 1-i[0]
    if i[0] == 0:
      Z.remove(i)
  return Z

print("Complement A Property", complement(ClassA))
```

```
Complement A Property [[0.5, 'Sakshi'], [0.9, 'Mayuri'], [0.5, 'Pushkaraj'], [0.6, 'Aaman'], [0.5, 'Prerna']]
```

D) Bounded sum[A(x), B(x)]= min [1, A (x) + B (x)]

```
#Bounded Sum

def bounded_sum(setA, setB):
    X, Y = deepcopy(setA), deepcopy(setB)
    Z = []
    for i in X:
        mb = checkElementHelper(i[1], Y)
        Z.append([min(1, i[0] + mb), i[1]])
        if mb != 0:
            Y.remove([mb, i[1]])
    Z = Z + Y
    return Z

print("Bounded Sum Property", bounded_sum(ClassA, ClassB))
```

```
Bounded Sum Property [[0.7, 'Sakshi'], [1, 'Mayuri'], [0.8, 'Pushkaraj'], [1, 'Aaman'], [1, 'Prerna']]
```

E) Bounded diff[A(x), B(x)]= max[1, A (x) - B (x)]

```
#Bounded Difference

def bounded_difference(setA, setB):
    X, Y = deepcopy(setA), deepcopy(setB)
    Z = []
    for i in X:
        mb = checkElementHelper(i[1], Y)
        bounded_diff_result = max(0, i[0] - mb)
        Z.append([bounded_diff_result, i[1]])
    return Z

print("Bounded Difference Property:", bounded_difference(ClassA, ClassB))
```

```
Bounded Difference Property: [[0.3, 'Sakshi'], [0, 'Mayuri'], [0.2, 'Pushkaraj'], [0, 'Aaman'], [0, 'Prerna']]
```

<u>CONCLUSION</u>: We have successfully studied the properties of fuzzy sets. We have also implemented different operations on it like union, intersection and complement to further expand our understanding.