**AIDS Lab**

**EXPERIMENT NO. 7**

**Aim**: To demonstrate properties of fuzzy sets.

**Theory**:

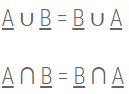
Properties of a fuzzy set helps us to simplify many mathematical fuzzy set operations. Sets are collections of unordered, district elements. We can perform various fuzzy set operations on the fuzzy set. It is recommended to readers to first navigate through the fuzzy set operations for better understanding of properties of the fuzzy set. Most of the properties of crisp sets hold for fuzzy sets also.

Properties of Fuzzy Sets:

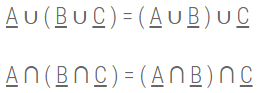
1. Involution: Involution states that the complement of complement is set itself.



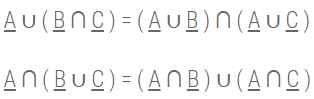
1. Commutativity: Operations are called commutative if the order of operands does not alter the result. Fuzzy sets are commutative under union and intersection operations.



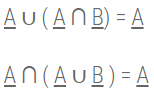
1. Associativity: Associativity allows to change the order of operations performed on operand, however relative order of operand can not be changed. All sets in the equation must appear in the identical order only. Fuzzy sets are associative under union and intersection operations.



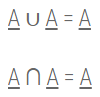
1. Distributivity



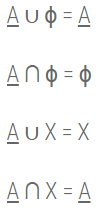
1. Absorption: Absorption produces the identical sets after stated union and intersection operations.



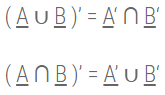
1. Idempotency: Idempotency does not alter the element or the membership value of elements in the set.



1. Identity



1. Transitivity: If A ⊆ B and B ⊆ C then A ⊆ C
2. 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.



**Code and Output**:

Fuzzy sets can be considered as an extension and gross oversimplification of classical sets. It can be best understood in the context of set membership. Basically it allows partial membership which means that it contains elements that have varying degrees of membership in the set.

| from copy import deepcopy def checkElementHelper(x, S):  for e in S:  if x == e[1]: return e[0]  return 0  global ClassA ClassA=[[0.7,"Ninad"], [0.2,"Krishna"], [0.4,"Shalaka"], [0.3,"Isha"], [0.5,"Jisha"]]  global ClassB ClassB=[[0.3,"Ninad"], [0.4,"Krishna"], [0.3,"Shalaka"], [0.2,"Isha"], [0.1,"Jisha"]] |
| --- |

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)) |
| --- |



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)) |
| --- |



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)) |
| --- |



Fuzzy Sum

| def fuzzy\_sum(setA, setB):  X, Y = deepcopy(setA), deepcopy(setB)  Z = []  for i in X:  mb = checkElementHelper(i[1], Y)  if mb != 0:  m = round(i[0] + mb - (i[0] \* mb), 3)  Z.append([m, i[1]])  Y.remove([mb, i[1]])  else:  Z.append(i)  Z = Z + Y  return Z  print("Fuzzy Sum Property", fuzzy\_sum(ClassA, ClassB)) |
| --- |



Fuzzy Product

| *# Algebraic Product* def fuzzy\_product(setA, setB):  X, Y = deepcopy(setA), deepcopy(setB)  Z = []  for i in X:  mb = checkElementHelper(i[1], Y)  if mb != 0:  m = i[0] \* mb  Z.append([m, i[1]])  Y.remove([mb, i[1]])  return Z  print("Fuzzy Product Property", fuzzy\_product(ClassA, ClassB)) |
| --- |



Bounded Sum

| def bounded\_sum(setA, setB):  X, Y = deepcopy(setA), deepcopy(setB)  Z = []  for i in X:  mb = checkElementHelper(i[1], Y)  if mb != 0:  m = i[0] + mb  Z.append([min(1, m), i[1]])  Y.remove([mb, i[1]])  else:  Z.append(i)  Z = Z + Y  return Z  print("Bounded Sum Property", bounded\_sum(ClassA, ClassB)) |
| --- |



Bounded Difference

| def bounded\_diff(setA, setB):  X, Y = deepcopy(setA), deepcopy(setB)  Z = []  for i in X:  mb = checkElementHelper(i[1],Y)  if mb != 0:  m = round(i[0] - mb, 3)  Z.append([max(0, m), i[1]])  Y.remove([mb, i[1]])  else:  Z.append(i)  return Z  print("Bounded Difference Property", bounded\_diff(ClassA, ClassB)) |
| --- |



**Conclusion**:

Thus we studied an overview of what a Fuzzy Set is and its properties.