SCOA Assignment No : 01

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

In [11]:

```
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)
```

```
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}
```

In [4]:

```
# Union of Two Fuzzy Sets
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)
```

```
Fuzzy Set Union is : {'a': 0.9, 'b': 0.9, 'c': 0.6, 'd': 0.6}
```

In [7]:

```
# Intersection of Two Fuzzy Sets
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)
```

```
Fuzzy Set Intersection is : {'a': 0.2, 'b': 0.3, 'c': 0.4, 'd': 0.5}
```

In [8]:

```
# Complement of Fuzzy Sets
for A_key in A:
    Y[A_key]= 1-A[A_key]

print('Complement of Fuzzy Set A is :', Y)

for B_key in B:
    Y[B_key]= 1-B[B_key]

print('Complement of Fuzzy Set B is :', Y)
```

```
Complement of Fuzzy Set A is : {'a': 0.8, 'b': 0.7, 'c': 0.4, 'd': 0.4}
Complement of Fuzzy Set B is : {'a': 0.09999999999999998, 'b': 0.09999999999
999998, 'c': 0.6, 'd': 0.5}
```

In [9]:

```
# Difference Between Two Fuzzy Sets
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)
```

```
Fuzzy Set Difference is : {'a': 0.09999999999999998, 'b': 0.0999999999999999
8, 'c': 0.6, 'd': 0.5}
```

In [15]:

```
# Cartesian product of Two Fuzzy Sets
import numpy as np
R = [[] for i in range(len(A))]
i = 0
for x in A:
    for y in B:
        R[i].append(min(A[x], B[y]))
    i += 1
print("Cartesian Product is",np.array(R),"\n")
```

```
Cartesian Product is [[0.2 0.2 0.2 0.2]
 [0.3 0.3 0.3 0.3]
 [0.6 0.6 0.4 0.5]
 [0.6 0.6 0.4 0.5]]
```

In [4]:

```
#max-min composition
r1 = int(input("Enter number of rows of first relation (R1): "))
c1 = int(input("Enter number of columns of first relation (R1): "))
rel1=[[0 for i in range(c1)]for j in range(r1)]
print("Enter the elments for R:")
for i in range(r1):
    for j in range(c1):
        rel1[i][j]=float(input())

r2 = int(input("Enter number of rows of second relation (R2): "))
c2 = int(input("Enter number of columns of second relation (R2): "))
rel2=[[0 for i in range(c2)]for j in range(r2)]
print("Enter the elments for R:")
for i in range(r2):
    for j in range(c2):
        rel2[i][j]=float(input())

print("\nR1 = ")
for i in range(r1):
    for j in range(c1):
        print(rel1[i][j],end=" ")
    print("\n")

print("\nR2 = ")
for i in range(r2):
    for j in range(c2):
        print(rel2[i][j],end=" ")
    print("\n")


col=0
comp=[]
for i in range(r1):
    comp.append([])
    for j in range(c2):
        l=[]
        for k in range(r2):
            l.append(min(rel1[i][k],rel2[k][j]))
        comp[i].append(max(l))

print("\nR1 composition R2 =")
for i in range(r1):
    for j in range(c2):
        print(comp[i][j],end=" ")
    print("\n")
```

```
Enter number of rows of first relation (R1): 2
Enter number of columns of first relation (R1): 2
Enter the elments for R:
0.6
0.3
0.2
0.9
Enter number of rows of second relation (R2): 2
Enter number of columns of second relation (R2): 3
Enter the elments for R:
1
```

0.5
0.3
0.8
0.4
0.7

R1 =
0.6 0.3

0.2 0.9

R2 =
1.0 0.5 0.3

0.8 0.4 0.7

R1 composition R2 =
0.6 0.5 0.3

0.8 0.4 0.7