Name: Tania Rajabally

Roll No: 43

UID: 2017130047

Batch: C

Course Code: CEL71 (AI and Soft Computing Lab)

Date:06/10/2020

**Experiment No 6**

**Fuzzy Sets And Fuzzy Relations**

**Aim:** To implement Fuzzy Sets and Fuzzy Relations.

**Theory:**

Fuzzy sets are somewhat like sets whose elements have degrees of membership. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions (aka characteristic functions) of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1. In fuzzy set theory, classical bivalent sets are usually called crisp sets. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics.

A fuzzy set is a pair (U,m) where U is a set and m: U->[0,1] a membership function. The reference set U (sometimes denoted by or is called universe of discourse, and for each , the value m(x) is called the grade of membership of x in (U,m). The function is called the membership function of the fuzzy set .

In fuzzy mathematics, fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1 both inclusive. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1. Fuzzy logic is based on the observation that people make decisions based on imprecise and non-numerical information. Fuzzy models or sets are mathematical means of representing vagueness and imprecise information (hence the term fuzzy). These models have the capability of recognising, representing, manipulating, interpreting, and utilising data and information that are vague and lack certainty. Fuzzy logic has been applied to many fields, from control theory to artificial intelligence.

Cartesian Product:

Let A1, A2, ….., An be fuzzy sets in U1, U2, …Un, respectively.

The Cartesian product of A1, A2, ….., An is a fuzzy set in the space U1 x U2 x…x Un with the membership function as:

µA1 x A2 x…x An (x1, x2,…, xn) = min [µA1 (x1), µA2 (x2), …. µAn (xn)]

So, the Cartesian product of A1, A2, ….., An are donated by A1 x A2 x….. x An

Min-max Composition:

Fuzzy relation in different product space can be combined with each other by the operation called “Composition”. There are many composition methods in use , e.g. max-product method, max-average method and max-min method. But max-min composition method is best known in fuzzy logic applications.

Definition:

Max –min composition

Let R1 (x, y), where (x, y) belong to A × B and R2 (y, z) ,where(y, z) belong to B× C be the two relations.

The max- min composition is then the fuzzy set

R1○ R2 ={[(x,y), maxy{min { μ 1 (x,y), μ 2 (y.z)}}] x A,y B,c C}

Here μ 1 and μ 2 is membership function of a fuzzy relation on fuzzy sets.

**Procedure:**

Fuzzify all input values into fuzzy membership functions.

Execute all applicable rules in the rulebase to compute the fuzzy output functions.

De-fuzzify the fuzzy output functions to get "crisp" output values.

**Code:**

**import numpy as np**

**#Three fuzzy sets a, b, c**

**a = [1,0.2,0.5]**

**b = [0.9,0.4,0.9]**

**c = [0.1,0.2,0.7]**

**print("a:", a)**

**print("b:", b)**

**print("c:", c)**

**#a) Fuzzy Cartesion product (a,b) = min(x,y) for all x in a and y in b**

**def cart\_product(a,b):**

**c\_ab = []**

**for i in range(len(a)):**

**temp = []**

**for j in range(len(b)):**

**temp.append(min(a[i],b[j]))**

**c\_ab.append(temp)**

**# print(np.array(c\_ab))**

**return c\_ab**

**r=cart\_product(a,b)**

**r**

**#b) Fuzzy Catesian product between c and b**

**s=cart\_product(c,b)**

**s**

**#Max-min composition**

**# T = R o S**

**# R o S = {for all a(rows in R) {for all c (columns in S) : max{min{R(a, b), S(b, c) for all b (columns in R or rows in S)}} )**

**def maxMinComp(r1,r2):**

**t = []**

**#for each x1**

**for i in range(len(r1)):**

**temp = []**

**#for all z**

**for k in range(len(r2[0])):**

**max\_li = []**

**#combination of all y**

**for j in range(len(r1)):**

**max\_li.append(min(r1[j],r2[j][k]))**

**m = max(max\_li)**

**temp.append(m)**

**t=temp**

**return t**

**# print(np.array(t))**

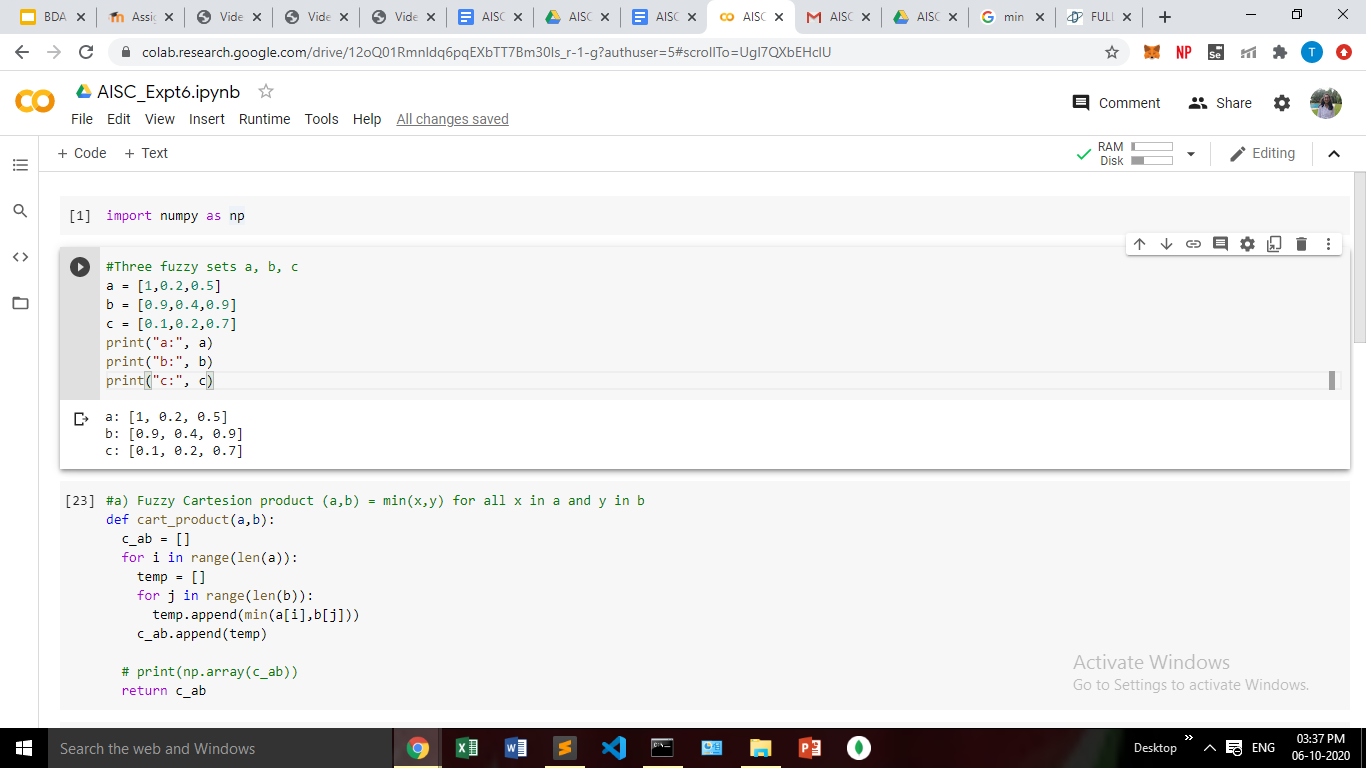
**#c) c.r using max min composition**

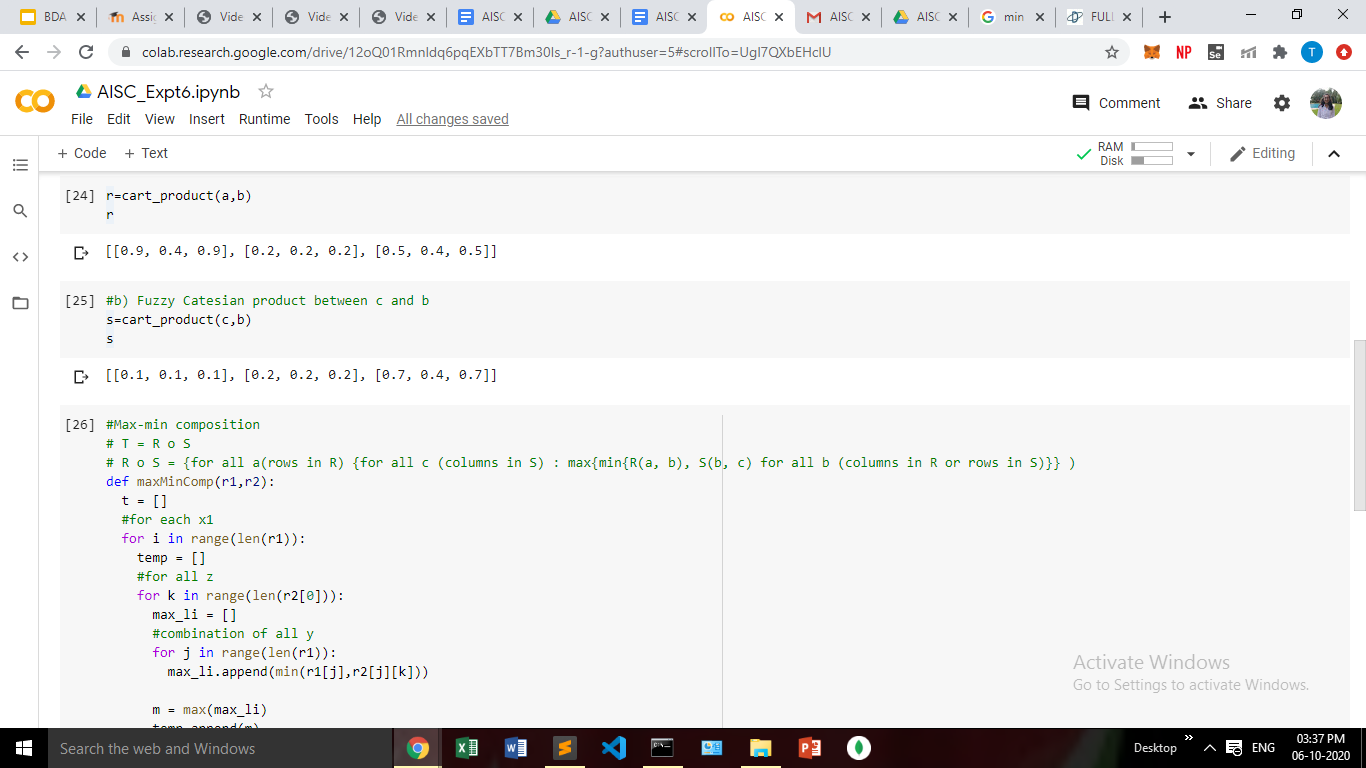
**np.array(maxMinComp(c, r))**

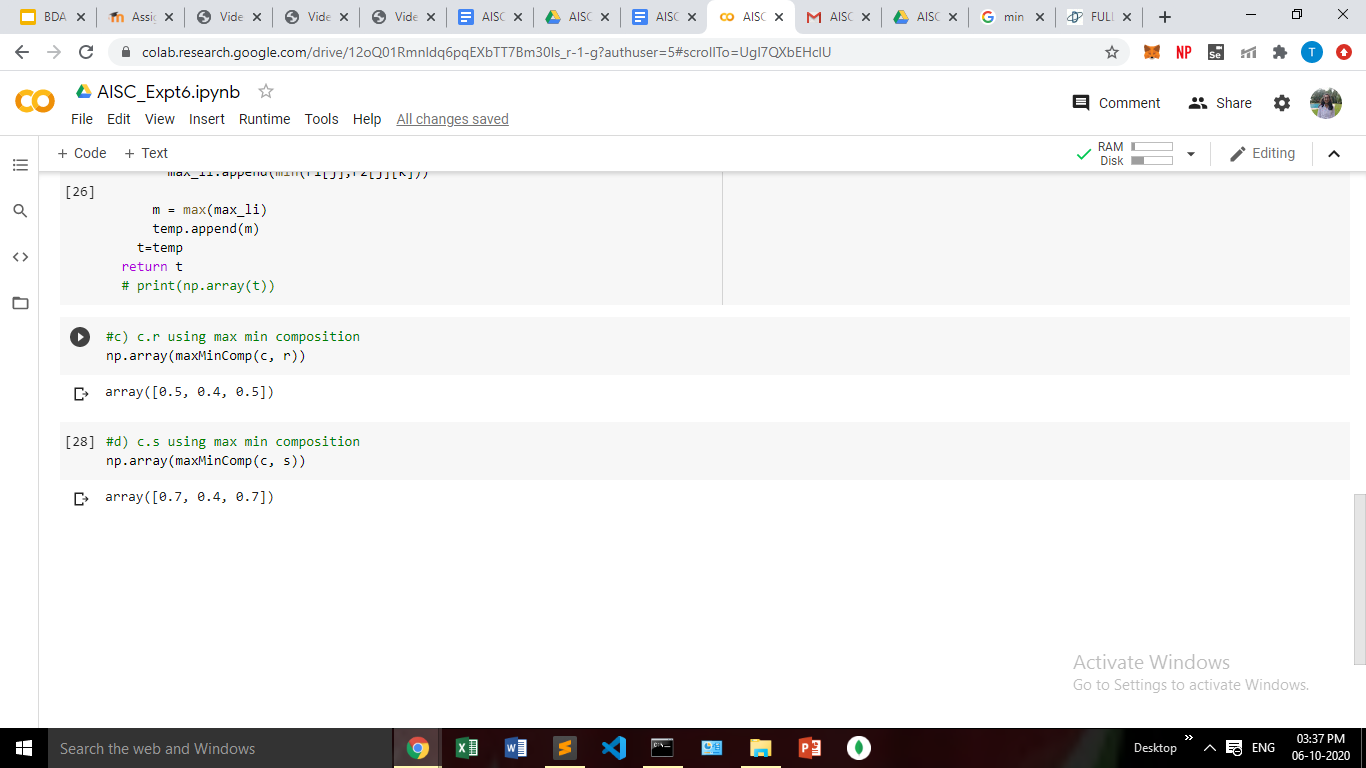
**#d) c.s using max min composition**

**np.array(maxMinComp(c, s))**

**Output:**

****

****

****

**Conclusion:**

In this experiment, we have found out the fuzzy relation between the sets. We found the fuzzy cartesian product and min max composition of the sets.