**Activity 2**

Mathematical Foundation for Computer Application

— — — — — — — — — — — — — — — — — — — — —

**Name-** Naman Jain

**Class-** MCA AI & ML

**USN-** JUPGI202210715

Python code to check for equivalence relation on a set.

**Code:**

class Relation:

def checkEquivalence(self, A, R):

transitive = self.checkTransitive(R)

symmetric = self.checkSymmetric(R)

reflexive = self.checkReflexive(A, R)

return transitive and symmetric and reflexive

def checkTransitive(self, R):

# Empty relation is always transitive

if len(R) == 0:

return True

# Create a dictionary to

# store tuple as key value pair

tup = dict()

# Creating dictionary of relation

# where (a) is key and (b) is value

for i in R:

if tup.get(i[0]) is None:

tup[i[0]] = {i[1]}

else:

tup[i[0]].add(i[1])

for a in tup.keys():

# Set of all b's related with a

all\_b\_in\_aRb = tup.get(a)

if all\_b\_in\_aRb is not None:

# Taking all c's from each b one by one

for b in all\_b\_in\_aRb:

# Set of all c's related with b

all\_c\_in\_bRc = tup.get(b)

if a != b and all\_c\_in\_bRc is not None:

if not all\_c\_in\_bRc.issubset(all\_b\_in\_aRb):

# All c's related with each b must be

# subset of all b's related with a

return False

# For all aRb and bRc there exist aRc in relation R

return True

def checkSymmetric(self, R):

# Empty relation is always symmetric

if len(R) == 0:

return True

for i in R:

if (i[1], i[0]) not in R:

# If bRa tuple does not exists in relation R

return False

# bRa tuples exists for all aRb in relation R

return True

def checkReflexive(self, A, R):

# Empty relation on a non-empty

# relation set is never reflexive.

if len(A) > 0 and len(R) == 0:

return False

# Relation defined on an empty

# set is always reflexive.

elif len(A) == 0:

return True

for i in A:

if (i, i) not in R:

# If aRa tuple not exists in relation R

return False

# All aRa tuples exists in relation R

return True

# Driver code

if \_\_name\_\_ == '\_\_main\_\_':

# Creating a set A

A = {1, 2, 3, 4}

# Creating relation R

R = {(1, 1), (1, 3), (2, 2), (3, 3),

(3, 1), (3, 4), (4, 4), (4, 3)}

obj = Relation()

# R is not equivalence as for (1, 3) and

# (3, 4) tuples -> (1, 4) tuple is not present

if obj.checkEquivalence(A, R):

print("Equivalence Relation")

else:

print("Not a Equivalence Relation")