

**Central Department
of
Computer Science and Information Technology
Tribhuvan University**



**Lab Report
on
Implementation of Crisp set and its operations**

Submitted to:

Jagdish Bhatta

CDCSIT

Subject: Fuzzy Systems

Tribhuvan University

Submitted By:

Karna Bahadur Shrestha

MSc. CSIT 2020

Third Semester

Rollno 14

Date: 20 Jan 2022

Code:

```
#lab1 To implement crisp set and crisp set
#operations(union,intersection,complement,subset)
Domain={'a':1,'b':1,'c':1,'d':1,'e':1,'f':1,'g':1,'h':1,'i':1,}
def enter(name):
    list={}
    n=int(input("Enter the number of elements in set"+name))
    for i in range(n):
        name=input("Enter the name: ")
        while 1:
            value=int(input("Enter the value: "))
            if(value==0 or value==1):
                list[name]=value
                break;
            else:
                print("Value must be 0 or 1")
    return list

#putting the element of set to each other, making the elements of
#sets equal
def makeequal(dict1,dict2):
    for dict2_key in dict2:
        for i in range(len(dict1)):
            if(dict2_key in dict1)!=True:
                dict1[dict2_key]=0
    return dict1

def sortdict(A):
    sorted_keys = sorted(A.keys())
    sorteddict = {key:A[key] for key in sorted_keys}
    return sorteddict

def Union(A,B):
    Y_set=dict()
    for A_key, B_key in zip(A, B):
        A_value = A[A_key]
        B_value = B[B_key]

        if A_value >= B_value :
            if A_value==1 or B_value==1:
                Y_set[A_key] = A_value
        elif B_value>A_value:
            Y_set[B_key] = B_value

    print('Union of crisp Set is :', Y_set)

def Intersection(A,B):
    Y_set=dict()
    for A_key, B_key in zip(A, B):
        A_value = A[A_key]
```

```

        B_value = B[B_key]

        if A_value <= B_value :
            if A_value==1 and B_value==1:
                Y_set[A_key] = A_value
            elif B_value<=A_value:
                if A_value==1 and B_value==1:
                    Y_set[B_key] = B_value
        print('Intersection of crisp Set is :', Y_set)

def complement(A):
    Y_set=dict()
    for A_key, B_key in zip(A, Domain):
        A_value = A[A_key]
        B_value = Domain[B_key]

        if A_value-B_value !=0:
            Y_set[A_key] = 1
    return Y_set

def checksubset(A,B):
    countA=countB=0
    for A_key, B_key in zip(A,B):
        A_value = A[A_key]
        B_value = B[B_key]

        if B_value <= A_value :
            countA=countA+1
        if A_value<=B_value:
            countB=countB+1

    if(countA==len(B)):
        print("B is subset of A")
    if(countB==len(A)):
        print("A is subset of B")
    if((countA==len(B) or countB==len(A))!=True):
        print("Non are subset of each set")

A=enter("A")
B=enter("B")

# A = {"a": 0, "b": 1, "c": 1,'e':1}
# B= {"a": 0, "b": 1, "c": 1, 'd':0}

print('The First Crisp Set entered A:', A)
print('The Second Crisp Set entered B:', B)
print('-----')

A=makeequal(A,B)
B=makeequal(B,A)

```

```

A=sortdict(A)
B=sortdict(B)

print('The First Set after process A:', A)
print('The Second Set after process B:', B)
print('-----')
Union(A,B)
print('-----')
Intersection(A,B)
print('-----')

#checking A is subset of B or B is subset of A
checksubset(A,B)
print('-----')
#making the elements of A and B equal to domain for further process
A=makeequal(A,Domain)
B=makeequal(B,Domain)
print("Complement of A :",complement(A))
print("Complement of B :",complement(B))
print('-----')

```

Output:

Enter the number of elements in setA4
Enter the name: a
Enter the value: 2
Value must be 0 or 1
Enter the value: 0Enter the name: b
Enter the value: 1Enter the name: c
Enter the value: 1Enter the name: e
Enter the value: 1
Enter the number of elements in setB4
Enter the name: a
Enter the value: 0
Enter the name: b
Enter the value: 1Enter the name: c
Enter the value: 1Enter the name: d
Enter the value: 0

The domain set is considered as Domain: {'a': 1, 'b': 1, 'c': 1, 'd': 1, 'e': 1, 'f': 1, 'g': 1, 'h': 1, 'i': 1}

The First Crisp Set entered A: {'a': 0, 'b': 1, 'c': 1, 'e': 1}

The Second Crisp Set entered B: {'a': 0, 'b': 1, 'c': 1, 'd': 0}

The First Set after process A: {'a': 0, 'b': 1, 'c': 1, 'd': 0, 'e': 1}

The Second Set after process B: {'a': 0, 'b': 1, 'c': 1, 'd': 0, 'e': 0}

Union of crisp Set is : {'b': 1, 'c': 1, 'e': 1}

Intersection of crisp Set is : {'b': 1, 'c': 1}

B is subset of A

Complement of A : {'a': 1, 'd': 1, 'f': 1, 'g': 1, 'h': 1, 'i': 1}

Complement of B : {'a': 1, 'd': 1, 'e': 1, 'f': 1, 'g': 1, 'h': 1, 'i': 1}
