J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN(Print) 1226-0657
http://dx.doi.org/10.7468/jksmeb.2022.29.1.93 ISSN(Online) 2287-6081
Volume 29, Number 1 (February 2022), Pages 93–102

OBTAINING WEAKER FORM OF CLOSED SETS IN TOPOLOGICAL SPACE USING PYTHON PROGRAM

M. Vivek Prabu [a,*] and M. Rahini [b]

Abstract. The impact of programming languages in the research sector has helped lot of researchers to broaden their view and extend their work without any limitation. More importantly, even the complex problems can be solved in no matter of time while converting them into a programming language. This convenience provides upper hand for the researchers as it places them in a comfort zone where they can work without much stress. With this context, we have converted the research problems in Topology into programming language with the help of Python. In this paper, we have developed a Python program to find the weaker form of closed sets namely alpha closed set, semi closed set, pre closed set, beta closed set and regular closed set.

1. Introduction

Python programming language is highly powerful and more reliable. More importantly, it is user friendly as it can resolve complex problems also with few steps because of its simple programming structure. The impact of programming languages in the research sector has helped lot of researchers to broaden their view and extend their work without any limitation. More importantly, even the complex problems can be solved in no matter of time while converting them into a programming language. This convenience provides upper hand for the researchers as it places them in a comfort zone where they can work without much stress. With this context, we have converted the research problems in Topology into programming language with the help of Python. In our earlier research papers [7, 8], we constructed a python program to test whether the given collection of subsets constitute a topology or not. Further, we extended the coding to obtain the closure and interior of each subset

Received by the editors September 24, 2021. Accepted February 01, 2022.
2010 *Mathematics Subject Classification.* 54-04.
Key words and phrases. topology, Python program, interior, closure, dictionary, key, subset, set, alpha closed set, semi closed set, pre closed set, beta closed set, regular closed set.

[*]Corresponding author.

of the given set. In this paper, we have developed a Python program to find the weaker form of closed sets namely alpha closed set, semi closed set, pre closed set, beta closed set and regular closed set.

2. Preliminaries

In this section, we will be recollecting the preliminary definitions which remain as the preamble for our study.

2.1. Definition Consider an arbitrary subset of any set which satisfy the three specified axioms in topology, then the set together with its subset denominated as topological space. Assume A is the subset of the topological space, then we can say

(1) Whenever the closure of interior of closure of each subset A is contained in the respective subset A then the collection of such subsets constitutes α closed sets.

(2) Whenever the interior of closure of interior of each subset A is contained in the respective subset A then the collection of such subsets constitutes β closed sets.

(3) Whenever the interior of closure of each subset A is contained in the respective subset A then the collection of such subsets constitutes semi - closed set.

(4) Whenever the closure of interior of each subset A is contained in the respective subset A then the collection of such subsets constitutes pre - closed set.

(5) Whenever the closure of interior of each subset A is equal to the respective subset A then the collection of such subsets constitutes regular - closed set.

3. Algorithm

In our earlier research papers [7, 8], we constructed a python program to test whether the given collection of subsets constitute a topology or not. Further, we extended the coding to obtain the closure and interior of each subset of the given set. In this paper, we have developed a Python program to find the weaker form of closed sets namely alpha closed set, semi closed set, pre closed set, beta closed set and regular closed set. Initially one has to obtain a topology from the user and complete the process specified in our earlier paper entitled Application of Kuratowskis Closure Operator in Python Program [8], after which the following algorithm can be followed to find the weaker forms of closed sets.

Step 1: Assign A with each subset one by one.

To find α closed sets and β open sets:

Step 2: Initially assume the closure of interior of closure of A as the whole set.

Step 3: Collect the subsets whose closure of interior of closure is contained in the subset.

Step 4: Find the intersection of those subsets and store the same in the dictionary of the alpha closed set.

Step 5: Find the complement of α closed sets.

To find semi closed sets and semi open sets:

Step 6: Initially assume the interior of closure of A as the whole set.

Step 7: Collect the subsets whose interior of closure is contained in the subset.

Step 8: Find the intersection of those subsets and store the same in the dictionary of the semi closed set.

Step 9: Find the complement of semi closed sets.

To find pre closed sets and pre open sets:

Step 10: Initially assume the closure of interior of A as the whole set.

Step 11: Collect the subsets whose closure of interior is contained in the subset.

Step 12: Find the intersection of those subsets and store the same in the dictionary of the pre closed set.

Step 13: Find the complement of pre closed sets.

To find β closed sets and β open sets:

Step 14: Initially assume the interior of closure of interior of A as the whole set.

Step 15: Collect the subsets whose interior of closure of interior is contained in the subset.

Step 16: Find the intersection of those subsets and store the same in the dictionary of the equation equation $\beta-$ closed set.

Step 17: Find the complement of equation equation $\beta-$ closed sets.

To find regular closed sets and regular open sets:

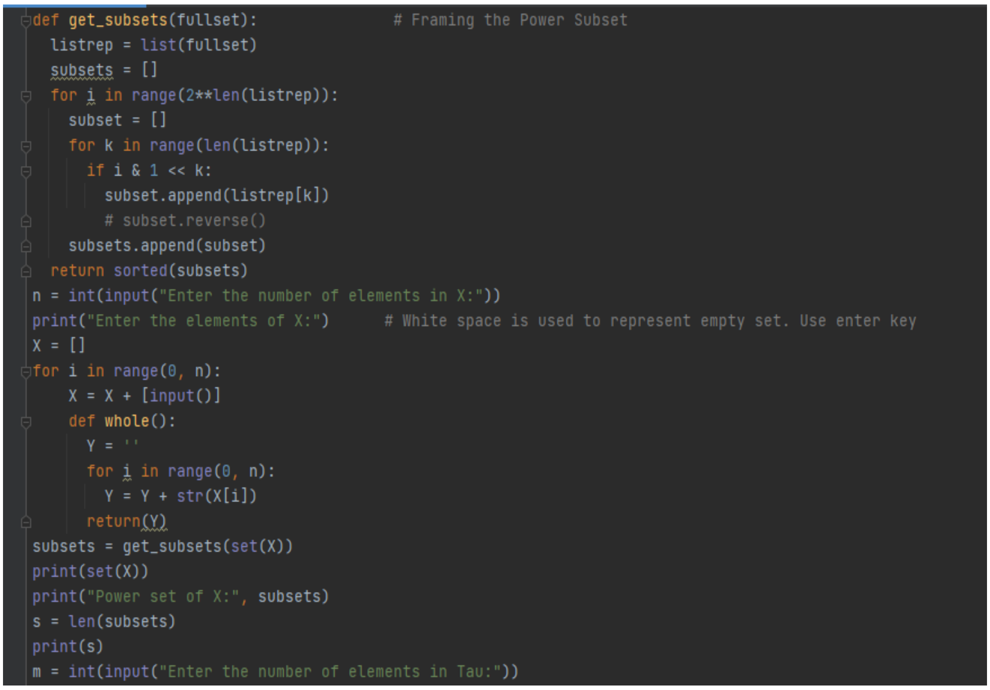
Step 18: Initially assume the closure of interior of A as the whole set.

Step 19: Collect the subsets whose closure of interior is equal to the subset and store the same in the dictionary of the regular closed set.

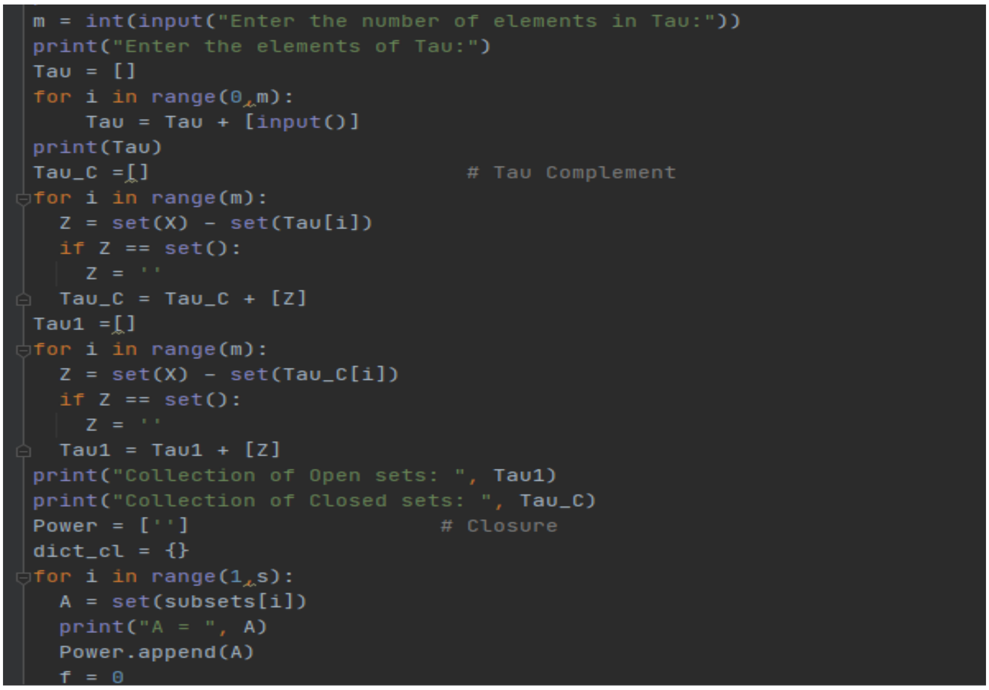
Step 20: Find the complement of regular closed sets.

Step 21: Terminate the program.

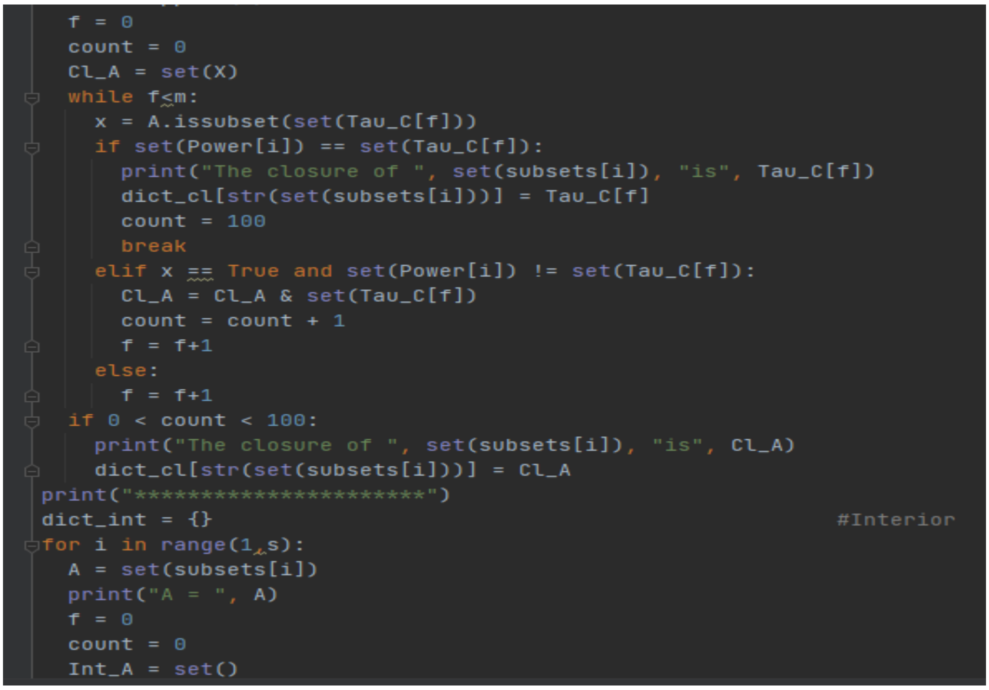
4. Coding



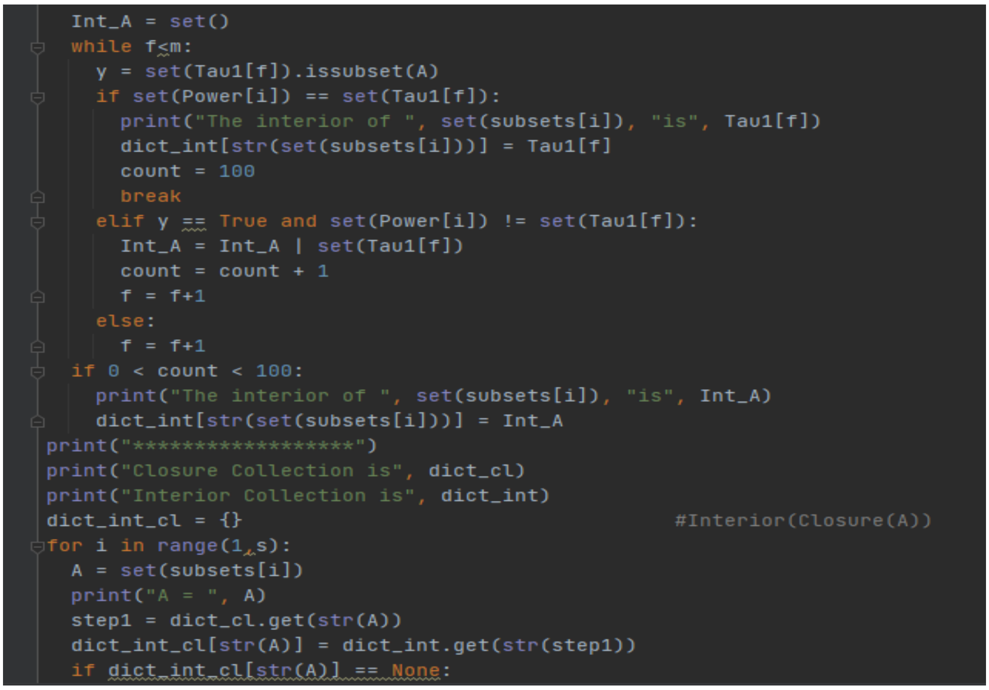
```
def get_subsets(fullset):             # Framing the Power Subset
    listrep = list(fullset)
    subsets = []
    for i in range(2**len(listrep)):
        subset = []
        for k in range(len(listrep)):
            if i & 1 << k:
                subset.append(listrep[k])
                # subset.reverse()
        subsets.append(subset)
    return sorted(subsets)
n = int(input("Enter the number of elements in X:"))
print("Enter the elements of X:")      # White space is used to represent empty set. Use enter key
X = []
for i in range(0, n):
        X = X + [input()]
        def whole():
            Y = ''
            for i in range(0, n):
                Y = Y + str(X[i])
            return(Y)
subsets = get_subsets(set(X))
print(set(X))
print("Power set of X:", subsets)
s = len(subsets)
print(s)
m = int(input("Enter the number of elements in Tau:"))
```



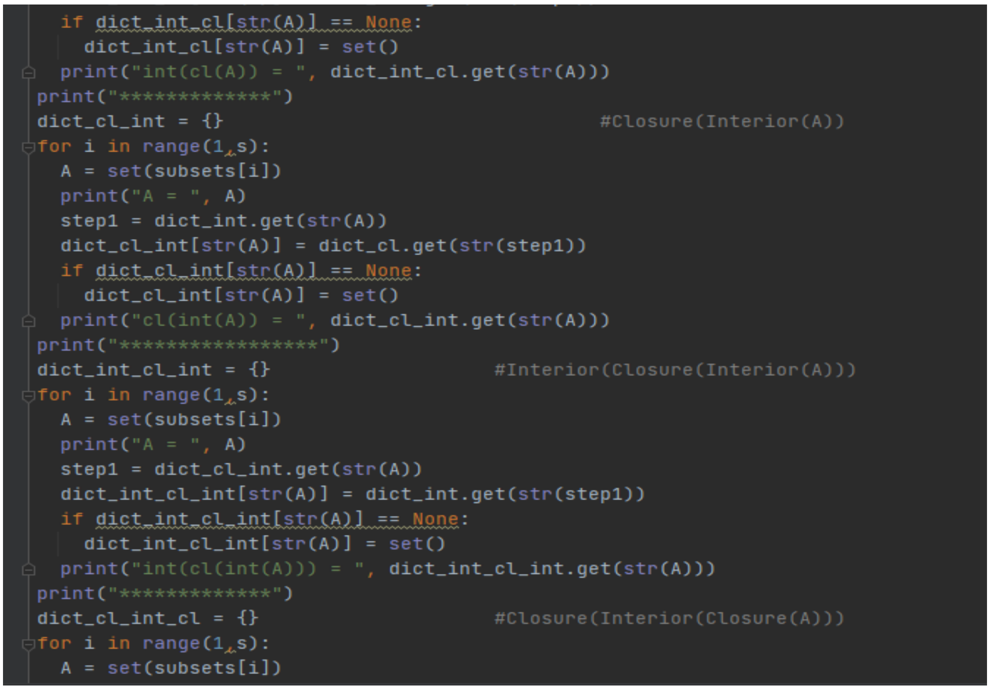
```
m = int(input("Enter the number of elements in Tau:"))
print("Enter the elements of Tau:")
Tau = []
for i in range(0,m):
    Tau = Tau + [input()]
print(Tau)
Tau_C =[]                          # Tau Complement
for i in range(m):
  Z = set(X) - set(Tau[i])
  if Z == set():
    Z = ''
  Tau_C = Tau_C + [Z]
Tau1 =[]
for i in range(m):
  Z = set(X) - set(Tau_C[i])
  if Z == set():
    Z = ''
  Tau1 = Tau1 + [Z]
print("Collection of Open sets: ", Tau1)
print("Collection of Closed sets: ", Tau_C)
Power = ['']                      # Closure
dict_cl = {}
for i in range(1,s):
  A = set(subsets[i])
  print("A = ", A)
  Power.append(A)
  f = 0
```



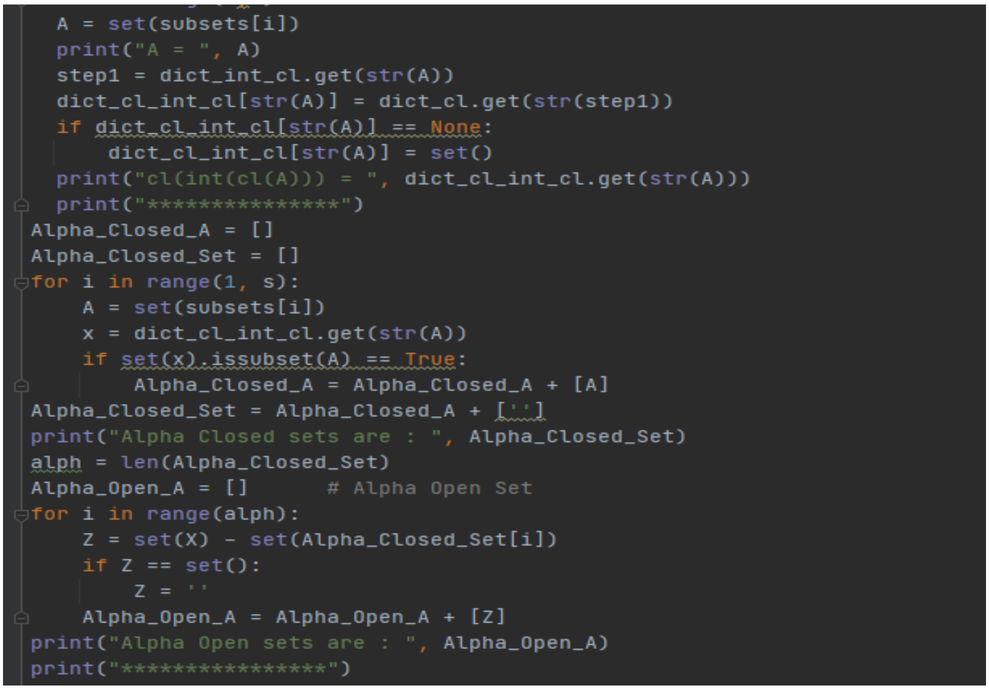
```
  f = 0
  count = 0
  Cl_A = set(X)
  while f<m:
    x = A.issubset(set(Tau_C[f]))
    if set(Power[i]) == set(Tau_C[f]):
      print("The closure of ", set(subsets[i]), "is", Tau_C[f])
      dict_cl[str(set(subsets[i]))] = Tau_C[f]
      count = 100
      break
    elif x == True and set(Power[i]) != set(Tau_C[f]):
      Cl_A = Cl_A & set(Tau_C[f])
      count = count + 1
      f = f+1
    else:
      f = f+1
  if 0 < count < 100:
    print("The closure of ", set(subsets[i]), "is", Cl_A)
    dict_cl[str(set(subsets[i]))] = Cl_A
print("************************")
dict_int = {}                                                   #Interior
for i in range(1,s):
  A = set(subsets[i])
  print("A = ", A)
  f = 0
  count = 0
  Int_A = set()
```



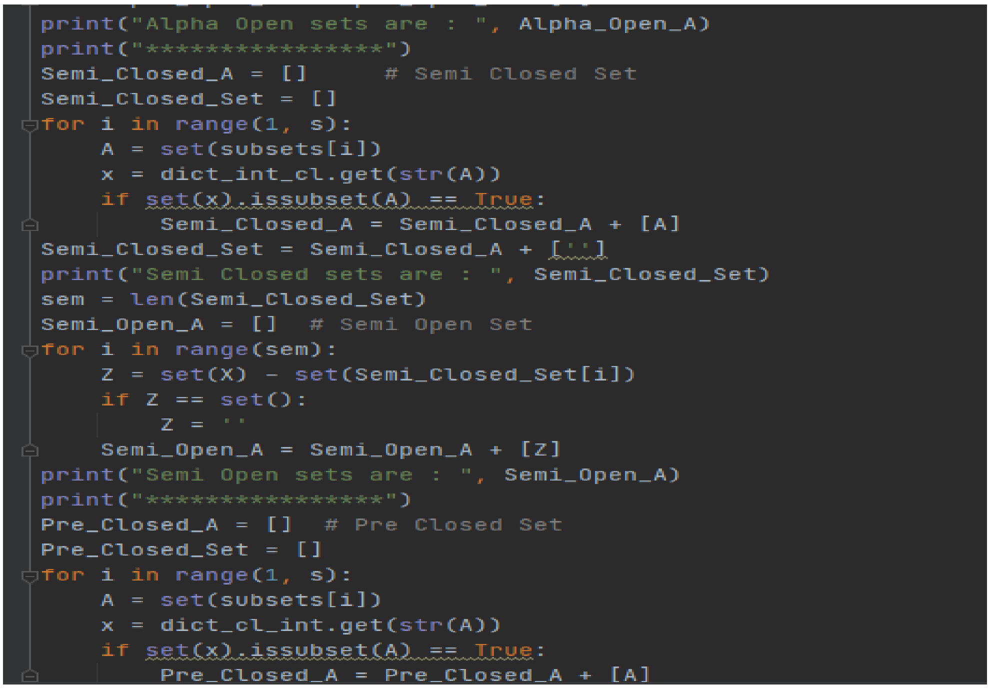
```
  Int_A = set()
  while f<m:
    y = set(Tau1[f]).issubset(A)
    if set(Power[i]) == set(Tau1[f]):
      print("The interior of ", set(subsets[i]), "is", Tau1[f])
      dict_int[str(set(subsets[i]))] = Tau1[f]
      count = 100
      break
    elif y == True and set(Power[i]) != set(Tau1[f]):
      Int_A = Int_A | set(Tau1[f])
      count = count + 1
      f = f+1
    else:
      f = f+1
  if 0 < count < 100:
    print("The interior of ", set(subsets[i]), "is", Int_A)
    dict_int[str(set(subsets[i]))] = Int_A
print("*****************")
print("Closure Collection is", dict_cl)
print("Interior Collection is", dict_int)
dict_int_cl = {}                                    #Interior(Closure(A))
for i in range(1,s):
  A = set(subsets[i])
  print("A = ", A)
  step1 = dict_cl.get(str(A))
  dict_int_cl[str(A)] = dict_int.get(str(step1))
  if dict_int_cl[str(A)] == None:
```



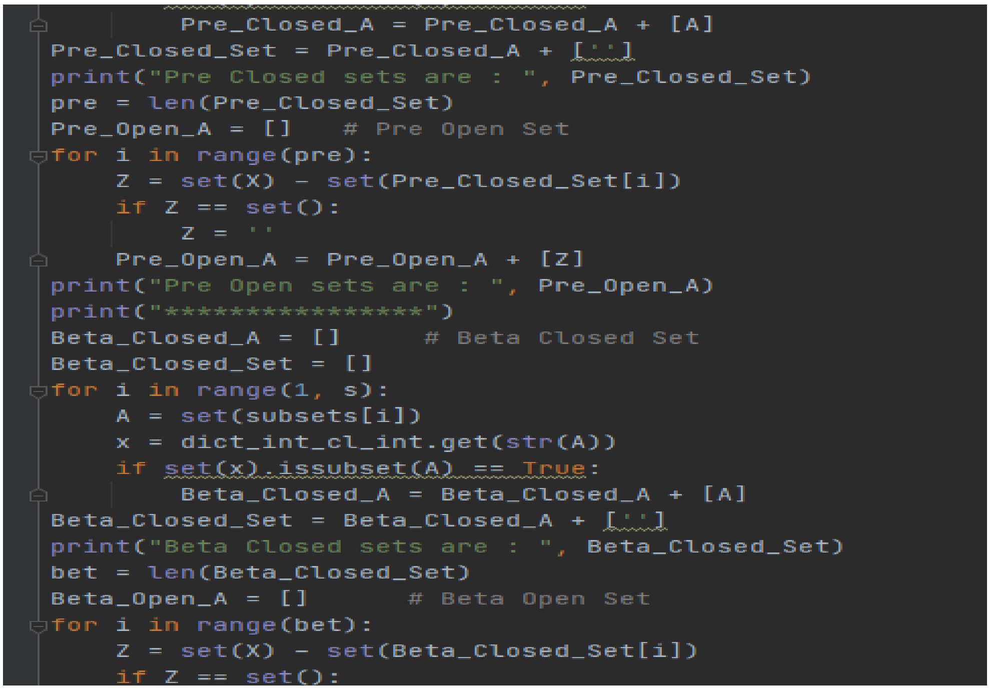
```
  if dict_int_cl[str(A)] == None:
    dict_int_cl[str(A)] = set()
  print("int(cl(A)) = ", dict_int_cl.get(str(A)))
print("*************")
dict_cl_int = {}                                 #Closure(Interior(A))
for i in range(1,s):
  A = set(subsets[i])
  print("A = ", A)
  step1 = dict_int.get(str(A))
  dict_cl_int[str(A)] = dict_cl.get(str(step1))
  if dict_cl_int[str(A)] == None:
    dict_cl_int[str(A)] = set()
  print("cl(int(A)) = ", dict_cl_int.get(str(A)))
print("*****************")
dict_int_cl_int = {}                    #Interior(Closure(Interior(A)))
for i in range(1,s):
  A = set(subsets[i])
  print("A = ", A)
  step1 = dict_cl_int.get(str(A))
  dict_int_cl_int[str(A)] = dict_int.get(str(step1))
  if dict_int_cl_int[str(A)] == None:
    dict_int_cl_int[str(A)] = set()
  print("int(cl(int(A))) = ", dict_int_cl_int.get(str(A)))
print("*************")
dict_cl_int_cl = {}                     #Closure(Interior(Closure(A)))
for i in range(1,s):
  A = set(subsets[i])
```



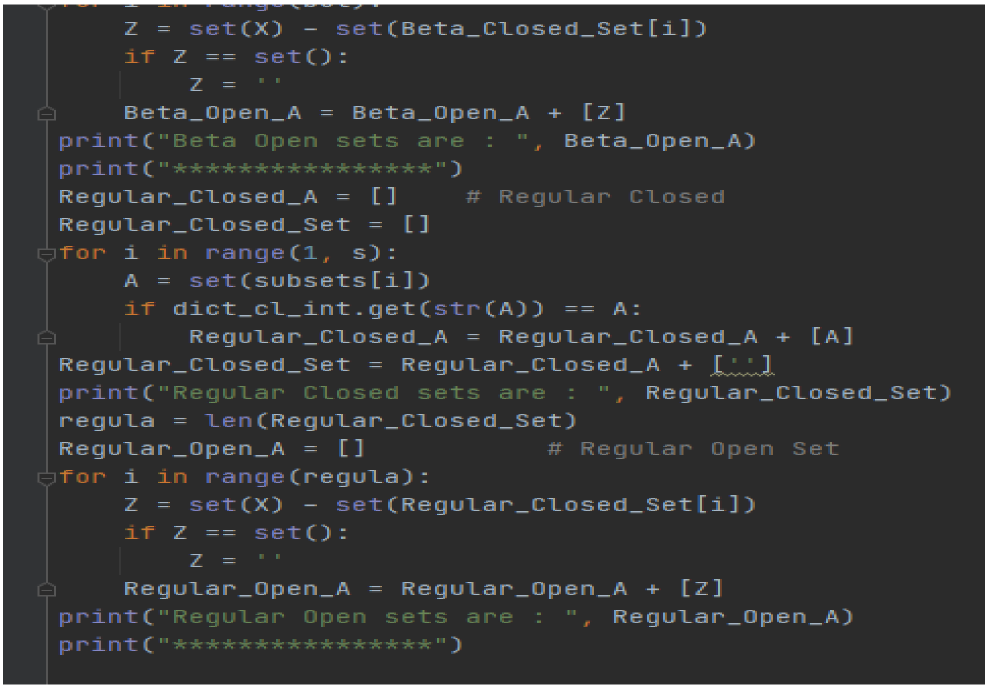
```
 A = set(subsets[i])
 print("A = ", A)
 step1 = dict_int_cl.get(str(A))
 dict_cl_int_cl[str(A)] = dict_cl.get(str(step1))
 if dict_cl_int_cl[str(A)] == None:
     dict_cl_int_cl[str(A)] = set()
 print("cl(int(cl(A))) = ", dict_cl_int_cl.get(str(A)))
 print("***************")
Alpha_Closed_A = []
Alpha_Closed_Set = []
for i in range(1, s):
    A = set(subsets[i])
    x = dict_cl_int_cl.get(str(A))
    if set(x).issubset(A) == True:
        Alpha_Closed_A = Alpha_Closed_A + [A]
Alpha_Closed_Set = Alpha_Closed_A + ['']
print("Alpha Closed sets are : ", Alpha_Closed_Set)
alph = len(Alpha_Closed_Set)
Alpha_Open_A = []       # Alpha Open Set
for i in range(alph):
    Z = set(X) - set(Alpha_Closed_Set[i])
    if Z == set():
        Z = ''
    Alpha_Open_A = Alpha_Open_A + [Z]
print("Alpha Open sets are : ", Alpha_Open_A)
print("****************")
```



```
print("Alpha Open sets are : ", Alpha_Open_A)
print("****************")
Semi_Closed_A = []      # Semi Closed Set
Semi_Closed_Set = []
for i in range(1, s):
    A = set(subsets[i])
    x = dict_int_cl.get(str(A))
    if set(x).issubset(A) == True:
        Semi_Closed_A = Semi_Closed_A + [A]
Semi_Closed_Set = Semi_Closed_A + ['']
print("Semi Closed sets are : ", Semi_Closed_Set)
sem = len(Semi_Closed_Set)
Semi_Open_A = []  # Semi Open Set
for i in range(sem):
    Z = set(X) - set(Semi_Closed_Set[i])
    if Z == set():
        Z = ''
    Semi_Open_A = Semi_Open_A + [Z]
print("Semi Open sets are : ", Semi_Open_A)
print("****************")
Pre_Closed_A = []  # Pre Closed Set
Pre_Closed_Set = []
for i in range(1, s):
    A = set(subsets[i])
    x = dict_cl_int.get(str(A))
    if set(x).issubset(A) == True:
        Pre_Closed_A = Pre_Closed_A + [A]
```



```
        Pre_Closed_A = Pre_Closed_A + [A]
Pre_Closed_Set = Pre_Closed_A + ['']
print("Pre Closed sets are : ", Pre_Closed_Set)
pre = len(Pre_Closed_Set)
Pre_Open_A = []   # Pre Open Set
for i in range(pre):
    Z = set(X) - set(Pre_Closed_Set[i])
    if Z == set():
        Z = ''
    Pre_Open_A = Pre_Open_A + [Z]
print("Pre Open sets are : ", Pre_Open_A)
print("****************")
Beta_Closed_A = []     # Beta Closed Set
Beta_Closed_Set = []
for i in range(1, s):
    A = set(subsets[i])
    x = dict_int_cl_int.get(str(A))
    if set(x).issubset(A) == True:
        Beta_Closed_A = Beta_Closed_A + [A]
Beta_Closed_Set = Beta_Closed_A + ['']
print("Beta Closed sets are : ", Beta_Closed_Set)
bet = len(Beta_Closed_Set)
Beta_Open_A = []       # Beta Open Set
for i in range(bet):
    Z = set(X) - set(Beta_Closed_Set[i])
    if Z == set():
```



```
    Z = set(X) - set(Beta_Closed_Set[i])
    if Z == set():
        Z = ''
    Beta_Open_A = Beta_Open_A + [Z]
print("Beta Open sets are : ", Beta_Open_A)
print("****************")
Regular_Closed_A = []     # Regular Closed
Regular_Closed_Set = []
for i in range(1, s):
    A = set(subsets[i])
    if dict_cl_int.get(str(A)) == A:
        Regular_Closed_A = Regular_Closed_A + [A]
Regular_Closed_Set = Regular_Closed_A + ['']
print("Regular Closed sets are : ", Regular_Closed_Set)
regula = len(Regular_Closed_Set)
Regular_Open_A = []            # Regular Open Set
for i in range(regula):
    Z = set(X) - set(Regular_Closed_Set[i])
    if Z == set():
        Z = ''
    Regular_Open_A = Regular_Open_A + [Z]
print("Regular Open sets are : ", Regular_Open_A)
print("****************")
```

Output:

```
Collection of Open sets: [{’a’}, {’b’}, {’b’, ’a’}, {’b’, ’c’, ’a’}, ”]
Collection of Closed sets: [{’b’, ’c’}, {’c’, ’a’}, {’c’}, ”, {’b’, ’c’, ’a’}]
Dictionary of Closure = {”{’a’}”: {’c’, ’a’}, ”{’b’}”: {’b’, ’c’}, ”{’b’, ’a’}”: {’c’, ’b’, ’a’}, ”{’c’}”: {’c’}, ”{’c’, ’a’}”: {’c’, ’a’}, ”{’b’, ’c’}”: {’b’, ’c’}, ”{’b’, ’c’, ’a’}”: {’b’, ’c’, ’a’}}
Dictionary of Interior = {”{’a’}”: {’a’}, ”{’b’}”: {’b’}, ”{’b’, ’a’}”: {’b’, ’a’}, ”{’c’}”: set(), ”{’c’, ’a’}”: {’a’}, ”{’b’, ’c’}”: {’b’}, ”{’b’, ’c’, ’a’}”: {’b’, ’c’, ’a’}}
Alpha Closed sets are: [{’a’, ’c’}, {’b’, ’a’, ’c’}, {’b’, ’c’}, {’c’}, ”]
Alpha Open sets are: [{’b’}, ”, {’a’}, {’b’, ’a’}, {’b’, ’a’, ’c’}]
****************
Semi Closed sets are: [{’a’}, {’a’, ’c’}, {’b’}, {’b’, ’a’, ’c’}, {’b’, ’c’}, {’c’}, ”]
Semi Open sets are: [{’b’, ’c’}, {’b’}, {’a’, ’c’}, ”, {’a’}, {’b’, ’a’}, {’b’, ’a’, ’c’}]
****************
Pre Closed sets are: [{’a’, ’c’}, {’b’, ’a’, ’c’}, {’b’, ’c’}, {’c’}, ”]
Pre Open sets are: [{’b’}, ”, {’a’}, {’b’, ’a’}, {’b’, ’a’, ’c’}]
***************
Beta Closed sets are: [ {’a’}, {’a’, ’c’}, {’b’}, {’b’, ’a’, ’c’}, {’b’, ’c’}, {’c’}, ”]
Beta Open sets are: [{’b’, ’c’}, {’b’}, {’a’, ’c’}, ”, {’a’}, {’b’, ’a’}, {’b’, ’a’, ’c’}]
****************
Regular Closed sets are: [{’a’, ’c’}, {’b’, ’a’, ’c’}, {’b’, ’c’}, ”]
Regular Open sets are: [{’b’}, ”, {’a’}, {’b’, ’a’, ’c’}]
****************
```

5. Conclusion

In this paper, we have developed a python program to find the weaker forms of closed sets as an extension of our earlier research articles [7,8]. From the various dictionaries, we have obtained the required subsets which fulfill the definition of various weaker forms of closed sets like alpha closed sets, semi closed sets, pre closed sets, beta closed sets and regular closed sets. We shall further extend this coding to find the respective closures of these closed sets, which shall pave way for the topologists to generalize these weaker forms and obtain much more weaker forms of closed sets

Acknowledgment

The referees have reviewed the paper very carefully. The authors express their deep thanks for the comments. The first author was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(MEST)(2012008507).

References

1. E. Balagurusamy: Introduction to Computing & Problem Solving Python. first edition, published by McGraw Hill Education (India) Private Ltd., 2016.
2. Belal Nairat: On Some Properties of β-open sets. International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 13, Number 6 (2018).
3. A. Bogdanchikov, M Zhaparov & R Suliyev: Python to learn programming. April 2013, published by Journal of Physics conference series 423 (2013) 012027, doi: 10.1088/17426596/423/1/012027.
4. James R. Munkres: Topology. Second edition, published by Pearson Education, 2000.
5. S.A. Kulkarni: Problem solving and python programming. Second edition, published by Yes Dee Publishing Pvt. Ltd, 2017.
6. Pratap Kumar Saha & Amar Kumar Banerjee: Semi open sets in bispaces. CUBO A Mathematical Journal, Vol.17, No. 01, (99106). March 2015.

7. M. Vivek Prabu & M. Rahini: Usage of Python Program to Test the Topological Axioms. (Communicated)
8. M. Vivek Prabu & M. Rahini: Application of Kuratowskis Closure Operator in Python Program. (Communicated)

[a]Assistant Professor, PG and Research Department of Mathematics, (Autonomous), G.N.Mills (Po), Coimbatore - 641029, Tamil Nadu, India.
Email address: vivekprabu_m@kongunaducollege.ac.in

[b]Ph.D Research Scholar, PG and Research Department of Mathematics, Kongunadu Arts and Science College (Autonomous), G.N.Mills (Po), Coimbatore - 641029, Tamil Nadu, India.
Email address: rahini1697@gmail.com