**J**ournal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.4 : 2024
ISSN : **1906-9685**


# OBTAINING COMPLEMENT OF OPEN SETS IN TOPOLOGICAL SPACE USING C PROGRAMMING

**R.Savithiri, A.Manonmani**, *Department of Mathematics, L.R.G. Govt. Arts college(w), Tirupur, Tamil Nadu, India.* savithiri7965@gmail.com

**Abstract:** Computer languages in the research field has helped lot to save researchers time to extend their work. In general, Difficult problems can be solved while converting them into a programming language. In this context, we constructed a C program to test the collection of open sets forms a topological space and then finds the closed set for each set in the collection.
**Keywords:** topology, C program, open set, closed set.

## 1.Introduction

The C programming language was developed in the Bell Labs of AT&T by an employee called Dennis Ritchie[1] between 1969 and 1973 while working on Unix operating system. C programming is a fundamental and popular language. C is stable, universally supported, and has reliable tooling. Particularly, it is simple and easy language because of its simple programming structure. Also C library will compile and run almost anywhere, can be developed anywhere, and will require minimal dependencies. Computer languages in the research field has helped lot to save researchers time to extend their work. In general, Difficult problems can be solved while converting them into a programming language. In this context, we constructed a C program to test the collection of open sets forms a topological space and then finds the closed set for each set in the collection.

## 2. Preliminaries

**Definition 2.1:** A topology[2] on a set X is a collection τ of subsets of X having the following properties:
1) ∅ and X are in τ.
2) The union of the elements of any subcollection of τ is in τ.
2) The intersection of the elements of any finite subcollection of τ is in τ.
A set X for which a topology τ has been specified is called a topological space.
**Definition 2.2:** If X is a topological space with topology τ , we say that a subset U of X is an open set of X if U belongs to the collection τ. Using this, a topological space is a set X together with a collection of subsets of X, called open sets[2], such that ∅ and X are both open, and such that arbitrary unions and finite intersections of open sets are open.
**Definition 2.3:** A subset A of a topological space X is said to be closed[[2] if the set X-A is open.

## 3. Algorithm

Step 1: Define the collection of sets.
Step 2: Check if the empty set and the set itself are included in the collection.
Step 3: Check closure under finite intersections: For every pair of sets, verify that their intersection is also in the collection.
Step 4: Check closure under arbitrary unions: For every pair of sets, verify that their union is also in the collection.
Step 5: If all conditions are met, the given collection of sets forms a topological space; otherwise it does not.

Step 6: If the given collection of sets forms a topological space, then find the complement set for each set in the collection.

```
// Checking if the universe is included in the collection
for (int i = 0; i < numSets; i++)
{
    if (sets[i][0] == '\0')
{
        printf(" Error: The universe is not included in the collection.\n");
        return 0;
    }
}
// Checking closure under finite intersections
for (int i = 0; i < numSets; i++)
{
    for (int j = i + 1; j < numSets; j++)
{
        if (!(sets[i][0] == '\0' || sets[j][0] == '\0'))
{
            int intersectionEmpty = 1;
            for (int k = 0; k < 4; k++)
{
                if ((sets[i][k] != '\0' && sets[j][k] == '\0') ||
                    (sets[i][k] == '\0' && sets[j][k] != '\0'))
```

```
{
                intersectionEmpty = 0;
                break;
              }
            }
            if (intersectionEmpty)
{
              printf("Error: Not closed under finite intersections.\n");
              return 0;
            }
          }
        }
      }
                                    // Checking closure under arbitrary unions
      for (int i = 0; i < numSets; i++)
{
        for (int j = i + 1; j < numSets; j++)
{
          if (!(sets[i][0] == '\0' || sets[j][0] == '\0'))
{
            bool unionEmpty = 1;
            for (int k = 0; k < 4; k++)
{
              if (sets[i][k] != '\0' | sets[j][k] != '\0')
{
                unionEmpty = 0;
                break;
              }
            }
            if (unionEmpty)
{
              printf("Error: Not closed under arbitrary unions.\n");
              return 0;
            }
          }
        }
      }
```

```
                                  // If both conditions are satisfied, it's a topological space
    return 1;
}

                                  // Function to find the closed set for a given set
void findClosedSet(int numSets, char sets[][5], char universe[], char set[])
{
    if (set[0] == '\0')
{
        printf("The closed set for {} is the entire space {%s} \n", universe);
    }
else {
        printf("The closed set for {%s} is {%s}.\n", set, universe);
    }
}

int main()
{
        int numSets = sizeof(sets) / sizeof(sets[0]);
                                  // Verify if the collection forms a topological space
    if (isTopologicalSpace(numSets, sets, universe))
{
                                  // Find closed sets for each set in the collection
        for (int i = 0, i < numSets, i++)
{
            findClosedSet(numSets, sets, universe, sets[i]);
        }
    }
    return 0,
}
```

**Output:1**
Collection of open sets : {{} ,{c},{a,b,c}}
Collection of closed sets : {{} ,{a,b},{a,b,c}}
**Output:2**
Collection of open sets : {{} ,{c}}
Error: The universe is not included in the collection.

**References**

[1] E.Balagurusamy : Programming in ANSI C, Sixth edition, McGraw-Hill Education,Europe.
[2] James R Munkres: Topology. Second edition, published by Pearson Education, 2000.
[3] Manoj Kumar Srivastav & Asoke Nath: Mathematical model of various statements of C- type language, International Journal of Advanced Computer Research 3(13), pp. 79-87, 2013.
[4] G.F.Simmons, Introduction to Topology and Modern Analysis, Tata McGraw-Hill Edition, New Delhi(2004).
[5] Valsamma K.M & Anilkumar V :C Programming for Mathematical computing, University of Calicut, kerala.
[6] M.Vivek Prabu & M.Rahini: Obtaining weaker form of closed sets in topological space using python program, J. Korean soc. Math. Educ. Ser.B, Vol:29 , pp. 93-102, Feb 2022.