C# Application to Deal with Neutrosophic g(α)-Closed Sets In Neutrosophic Topology

S. Saranya  
*Kongunadu Arts and Science College*

M. Vigneshwaran  
*Kongunadu Arts and Science College*

S. Jafari  
*College of Vestsjælland South*

Follow this and additional works at: [https://digitalcommons.pvamu.edu/aam](https://digitalcommons.pvamu.edu/aam)

Part of the Computer Sciences Commons, Geometry and Topology Commons, and the Logic and Foundations Commons

**Recommended Citation**
Saranya, S.; Vigneshwaran, M.; and Jafari, S. (2020). C# Application to Deal with Neutrosophic g(α)-Closed Sets In Neutrosophic Topology, Applications and Applied Mathematics: An International Journal (AAM), Vol. 15, Iss. 1, Article 12. Available at: [https://digitalcommons.pvamu.edu/aam/vol15/iss1/12](https://digitalcommons.pvamu.edu/aam/vol15/iss1/12)

This Article is brought to you for free and open access by Digital Commons @PVAMU. It has been accepted for inclusion in Applications and Applied Mathematics: An International Journal (AAM) by an authorized editor of Digital Commons @PVAMU. For more information, please contact hvkoshy@pvamu.edu.
C# Application to Deal with Neutrosophic $g\alpha$-Closed Sets
In Neutrosophic Topology

1S. Saranya, 2M. Vigneshwaran and 3S. Jafari

1,2PG and Research Department of Mathematics
Kongunadu Arts and Science College
Coimbatore - 641 029, India

3Department of Mathematics
College of Vestjaelland South
Herrestraede 11, 4200 Slagelse, Denmark

1saranyamaths1107@gmail.com; 2vignesh.mat@gmail.com; 3jafaripersia@gmail.com

Received: September 13, 2019; Accepted: May 2, 2020

Abstract

In this paper, we have developed a C# Application for finding the values of the complement, union, intersection and the inclusion of any two neutrosophic sets in the neutrosophic field by using .NET Framework, Microsoft Visual Studio and C# Programming Language. In addition to this, the system can find neutrosophic topology, neutrosophic $\alpha$-closed sets and neutrosophic $g\alpha$-closed sets in each resultant screens. Also, this computer-based application produces the complement values of each neutrosophic closed sets.

Keywords: .NET Framework; Microsoft Visual Studio; C# application; Neutrosophic set operations; Neutrosophic topology; Neutrosophic $\alpha$-closed set; Neutrosophic $g\alpha$-closed set

MSC 2010 No.: 03E72, 03F55, 54A40, 68W99, 68U99

226
1. Introduction

Smarandache (1998) introduced the neutrosophic set and the neutrosophic topology. By using the concept of neutrosophic set, Salama and Alblowi (2012) introduced the neutrosophic topological spaces with its basic definitions of complement, union, intersection and the inclusion of two neutrosophic sets. Salama et al. (2014) has developed some software programs for dealing with neutrosophic sets. In the same year, Salama et al. has designed and implemented a neutrosophic data operations by using object oriented programming. Arokiarani et al. (2017) has introduced neutrosophic α-closed sets in neutrosophic topological spaces. Vigneshwaran and Saranya (2018) has introduced b∗gα-closed sets in topological spaces. Saranya and Vigneshwaran (2019) has introduced neutrosophic b-closed sets, neutrosophic gα-closed sets, neutrosophic ∗gα-closed sets and neutrosophic b∗gα-closed sets in neutrosophic topological spaces and also the authors developed a new C# application to deal with neutrosophic α-closed sets and neutrosophic ∗gα-closed sets in neutrosophic topology.

Based on the concepts of neutrosophic sets, several researchers introduced various types of neutrosophic sets in the neutrosophic topological spaces. For the introduction of new sets in the field of neutrosophic topological spaces, the researchers manually find the clouser and interior of neutrosophic sets or various types of clouser and interior of neutrosophic sets. The choice of different kinds of closure and interior operator is purely based on the definition of the neutrosophic sets. Also, the researchers manually compare the neutrosophic sets by using the set inclusion concept. To introduce a new a type of neutrosophic set or various types of generalized neutrosophic sets in neutrosophic topological space is time consuming, since these sets has to be found manually which it requires more manpower. Moreover, the researchers have to find the union, intersection and complement of such neutrosophic sets manually.

The present study introduces the C# application to reduce the manual calculations for finding the values of the complement, union, intersection and the inclusion of two neutrosophic sets in a neutrosophic field. We have developed a C# application by using .NET Framework, Microsoft Visual Studio and C# Programming Language. In this application, the user can calculate the values of neutrosophic topology, neutrosophic α-closed set, and neutrosophic gα-closed set values in each resultant screens. Also, it generates the values of the complement sets.

2. Results on Neutrosophic gα-Closed Sets via C# Application

In this section we will show the working process of C# application for finding the values of the complement, union, intersection and the inclusion of any two neutrosophic sets. Also, it produces the values of neutrosophic topology(τ), neutrosophic α-closed set and neutrosophic gα-closed set values in neutrosophic topological spaces. The complements of neutrosophic α-closed set and neutrosophic gα-closed set values will be displayed at the end of the results of each sets.

The following formulas introduced by Salama and Alblowi (2012) have considered in the coding of C# application to produce the values of the corresponding operations in neutrosophic sets.
Let $L$ and $M$ be two neutrosophic sets of the form,

$$L = \{ <x, mv(L(x)), iv(L(x)), nmv(L(x)) > \; \forall \; x \in X \},$$

$$M = \{ <x, mv(M(x)), iv(M(x)), nmv(M(x)) > \; \forall \; x \in X \},$$

(1) $L' = \{ <x, nmv(L(x)), 1 - iv(L(x)), mv(L(x)) > \; \forall \; x \in X \},$

(2) $L \subseteq M \iff mv(L(x)) \leq mv(M(x)), iv(L(x)) \leq iv(M(x)), nmv(L(x)) \geq nmv(M(x)) \; \forall \; x \in X,$

(3) $L \cup M = \{ x, max[mv(L(x)), mv(M(x))], min[iv(L(x)), iv(M(x))], min[nmv(L(x)), nmv(M(x))] \; \forall \; x \in X \},$

(4) $L \cap M = \{ x, min[mv(L(x)), mv(M(x))], max[iv(L(x)), iv(M(x))], max[nmv(L(x)), nmv(M(x))] \; \forall \; x \in X \}.$

**Algorithm: Neutrosophic Topology**

<table>
<thead>
<tr>
<th>input</th>
<th>$0_N, 1_N, L, M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>complement of $L$ and $M$; union of $L$ and $M$; intersection of $L$ and $M$; inclusion of $L$ and $M$; neutrosophic Topology</td>
</tr>
</tbody>
</table>

**STEPS:**

Step 1: check $0_N$ and $1_N$ is valid.

Step 2: $L$ and $M$ should be a neutrosophic set.

Step 3: calculate the complement of $L$ and $M$.

Step 4: calculate the union of $L$ and $M$.

Step 5: calculate the intersection of $L$ and $M$.

Step 6: check the inclusion of $L$ and $M$.

Step 7: if the union and intersection conditions are satisfied, go to Step 8 or else repeat Step 2.

Step 8: compute the neutrosophic topology for the assigned data.
In the above initial resultant screen / user screen, the user has to enter the values of $0_N$, $1_N$, $L$ and $M$ to get the values of its complement sets, union of $L$ and $M$ sets, intersection of $L$ and $M$ and the subset of $L$ or $M$. Finally, the user will get the corresponding neutrosophic topology.
The above figure shows the entered values of the initial resultant screen. In this, some of the values are not entered by the user. For this incomplete data, the dialog box asks the user to enter all the values.
The above figure shows the entered values of the initial resultant screen. Here some of the values are not properly entered by the user. For this incorrect data, the dialog box asks the user to enter the values in the non-standard unit interval 0 and 1. In this the user did not follow the conditions to enter L and M. Both L and M should be a neutrosophic values.
The above figure shows the results of the complement of two neutrosophic sets \([L' \text{ and } M']\), union of two neutrosophic sets \([L \cup M]\), intersection of two neutrosophic sets \([L \cap M]\) and the inclusion of two neutrosophic sets \([L \subseteq M]\). Also, it shows the result of neutrosophic topology.
Algorithm: Neutrosophic $\alpha$-Closed Set

<table>
<thead>
<tr>
<th>Input</th>
<th>Neutrosophic set $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Neutrosophic $\alpha$-closed set; Neutrosophic $\alpha$-open set $[D]$</td>
</tr>
</tbody>
</table>

**STEPS:**

Step 1: check $C$ is valid.

Step 2: find $Ncl(C)$, if $Ncl(C)$ satisfies the neutrosophic closure condition then goto step-3 or else repeat Step 1.

Step 3: find $Nint[Ncl[C]]$, if $Nint[Ncl[C]]$ satisfies the neutrosophic interior of neutrosophic closure condition then go to Step 4 or else repeat Step 1.

Step 4: find $Ncl[Nint[Ncl[C]]]$. If $Ncl[Nint[Ncl[C]]]$ satisfies the neutrosophic closure of neutrosophic interior of neutrosophic closure condition, then go to Step 5 or else repeat Step 1.

Step 5: if $N\alpha cl[C] = C$, then produce neutrosophic $\alpha$-closed set or else repeat Step 1.

Step 6: compute the neutrosophic $\alpha$-open set $[D]$ for the assigned data.

**Figure 6.** Neutrosophic $\alpha$ Closed Set - Flow Chart [N$\alpha$CS - FC]
The above figure shows the entered data set $C$ is not satisfies the definition of neutrosophic $\alpha$-closed sets. For this, the user has to enter some other neutrosophic set $C$. 

Figure 7. Screenshot of Not Satisfies the Definition of Neutrosophic $\alpha$-Closed Set
Figure 8. Existence of Neutrosophic $\alpha$ Closed Set [NoCS] via C# Application
Table 1. Neutrosophic $\alpha$-Closed Sets

<table>
<thead>
<tr>
<th>$N_\alpha CS$</th>
<th>Membership</th>
<th>Indeterminacy</th>
<th>Non-Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.5, 0.5, 0.6)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_2$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.5, 0.7, 0.7)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_3$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.9, 0.9, 0.9)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_4$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.1, 0.2, 0.9)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_5$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.4, 0.4, 0.4)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_6$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.5, 0.4, 0.4)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_7$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.5, 0.9, 0.4)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_8$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.2, 0.9, 0.4)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_9$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.35, 0.46, 0.39)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$C_{10}$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0, 0.9, 0.4)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
</tbody>
</table>

By using this application we have checked out the above neutrosophic sets as neutrosophic $\alpha$-closed set in neutrosophic topological spaces.

**Algorithm: Neutrosophic $g\alpha$-Closed Set**

<table>
<thead>
<tr>
<th>input</th>
<th>neutrosophic set $E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>neutrosophic $g\alpha$-closed set; neutrosophic $g\alpha$-open set $[F]$</td>
</tr>
</tbody>
</table>

**STEPS:**

Step 1: check $E$ is valid.

Step 2: check $E \subseteq D$ then go to Step 3, otherwise repeat Step 1.

Step 3: find $Ncl(E)$. If $Ncl(E)$ satisfies the neutrosophic closure condition, then go to Step 4 or else repeat Step 1.

Step 4: find $Nint[Ncl(E)]$. If $Nint[Ncl(E)]$ satisfies the neutrosophic interior of neutrosophic closure condition, then go to Step 5 or else repeat Step 1.

Step 5: find $Ncl[Nint[Ncl(E)]]$. If $Ncl[Nint[Ncl(E)]]$ satisfies the neutrosophic closure of neutrosophic interior of neutrosophic closure condition, then go to Step 6 or else repeat Step 1.

Step 6: calculate $N\alpha cl[E]$.

Step 7: if $N\alpha cl[E] \subseteq D$, then produce neutrosophic $g\alpha$-closed set, else repeat Step 1.

Step 8: compute the neutrosophic $g\alpha$-open set $[F]$ for the assigned data.
Figure 9. Neutrosophic $g\alpha$ Closed Set - Flow Chart [NgoCS - FC]

Figure 10. Existence of Neutrosophic $g\alpha$ Closed Set $[NgoCS]$ via $C\#$ Application
Table 2. Neutrosophic $g_{\alpha}$-Closed Sets

<table>
<thead>
<tr>
<th>$N_{g_{\alpha}C.S}$</th>
<th>Membership</th>
<th>Indeterminacy</th>
<th>Non-Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>(0.3, 0.213, 0.3)</td>
<td>(0.6594, 0.1, 0.517)</td>
<td>(0.671, 0.627, 0.5137)</td>
</tr>
<tr>
<td>$E_2$</td>
<td>(0.3, 0.4, 0.3)</td>
<td>(0.0, 0)</td>
<td>(0.5, 0.5, 0.5)</td>
</tr>
<tr>
<td>$E_3$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.1, 0.1, 0)</td>
<td>(0.6, 0.6, 0.5)</td>
</tr>
<tr>
<td>$E_4$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.11, 0.1, 0)</td>
<td>(0.6, 0.6, 0.5)</td>
</tr>
<tr>
<td>$E_5$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.5, 0.1, 0.5)</td>
<td>(0.6, 0.6, 0.5)</td>
</tr>
<tr>
<td>$E_6$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.5, 0.1, 0.5)</td>
<td>(0.66, 0.6, 0.5)</td>
</tr>
<tr>
<td>$E_7$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.68, 0.1, 0.52)</td>
<td>(0.66, 0.63, 0.5)</td>
</tr>
<tr>
<td>$E_8$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.69, 0.1, 0.57)</td>
<td>(0.67, 0.67, 0.57)</td>
</tr>
<tr>
<td>$E_9$</td>
<td>(0.3, 0.2, 0.3)</td>
<td>(0.659, 0.1, 0.57)</td>
<td>(0.671, 0.627, 0.57)</td>
</tr>
<tr>
<td>$E_{10}$</td>
<td>(0.3, 0.213, 0.3)</td>
<td>(0.6594, 0.1, 0.57)</td>
<td>(0.671, 0.627, 0.57)</td>
</tr>
</tbody>
</table>

By using this application we have checked out the above neutrosophic sets as neutrosophic $g_{\alpha}$-closed set in neutrosophic topological spaces.

3. Conclusion and Future Work

In this paper, the new C# application has been introduced and discussed its working process via .NET Framework, Microsoft Visual Studio and C# programming language in the neutrosophic field. Also, the existence of the complement of a neutrosophic set, union of two neutrosophic sets, intersection of two neutrosophic sets, the inclusion of two neutrosophic sets, neutrosophic topology, neutrosophic $\alpha$-closed set and neutrosophic $g_{\alpha}$-closed set in neutrosophic topological spaces has been presented in each figure. The implementation of this present application would help researchers to enhance and promote further studies on continuous functions, open maps, and closed maps, respectively.

Acknowledgment:

The authors wish to thank the reviewers for their careful reviews which improve the presentation of this paper considerably.

REFERENCES


