

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №1 з дисципліни
«Методи та технології штучного інтелекту»

«Дослідження способів формування нечітких множин і операцій над ними»

Перевірив:
Шимкович В.М.

Виконав:
студент 3 курсу
групи ІІІ-11 ФІОТ
Прищеп В.С.

Київ-2023

Лабораторна робота №1

Дослідження способів формування нечітких множин і операцій над ними

Мета: Побудувати нечіткі множини з використанням різних типів функцій приналежності. Виконати найбільш поширені логічні операції над нечіткими множинами.

Індивідуальне завдання

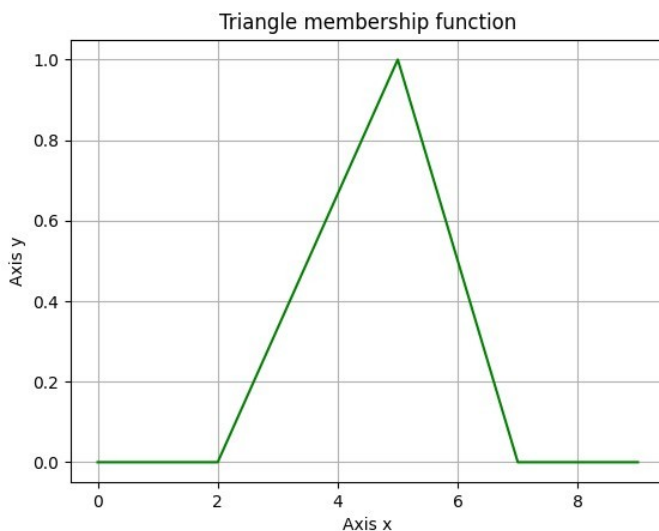
За допомогою мови python ми розробили програму, що виконує поставлене завдання. Для цього ми імпортуємо необхідні нам для роботи бібліотеки: numpy – для роботи з масивами даних, scikit-fuzzy – для роботи з нечіткими множинами та matplotlib – для візуалізації даних:

```
import numpy as np
import matplotlib.pyplot as plt
import skfuzzy as fuzz
```

1. Побудувати трикутну і трапецієподібну функцію приналежності.

#1. Triangle membership function

```
interval = np.arange(0, 10, 1)
membership_fn = fuzz.trimf(interval, [2, 5, 7])
_, axis = plt.subplots()
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('Triangle membership function')
plt.grid()
plt.show()
```



#1. Trapezoid membership function

```
interval = np.linspace(0, 10, 200)
```

```
membership_fn = fuzz.trapmf(interval, [1, 3, 6, 9])
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

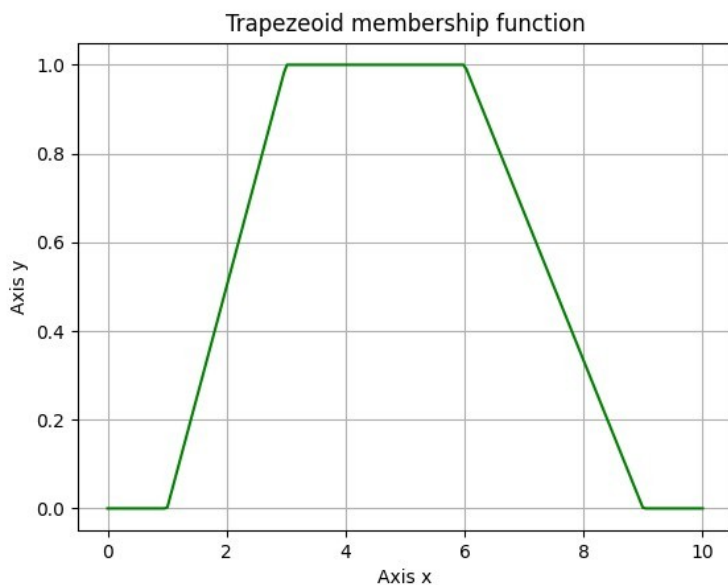
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Trapezeoid membership function')
```

```
plt.grid()
```

```
plt.show()
```



2. Побудувати просту і двосторонню функцію приналежності Гаусса, утворену за допомогою різних функцій розподілу.

#2. Gaussian membership function

```
interval = np.linspace(0, 10, 100)
```

```
membership_fn = fuzz.gaussmf(interval, 6, 1)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

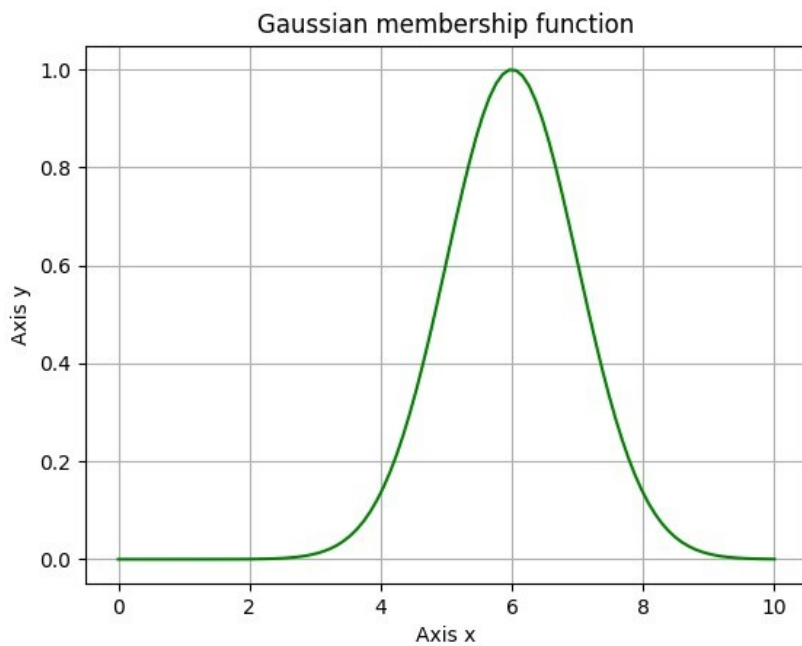
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Gaussian membership function')
```

```
plt.grid()
```

```
plt.show()
```



#2. Gaussian membership function of two combined Gaussian's

```
interval = np.linspace(-5, 15, 100)
```

```
membership_fn = fuzz.gauss2mf(interval, 4, 1, 8, 2)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

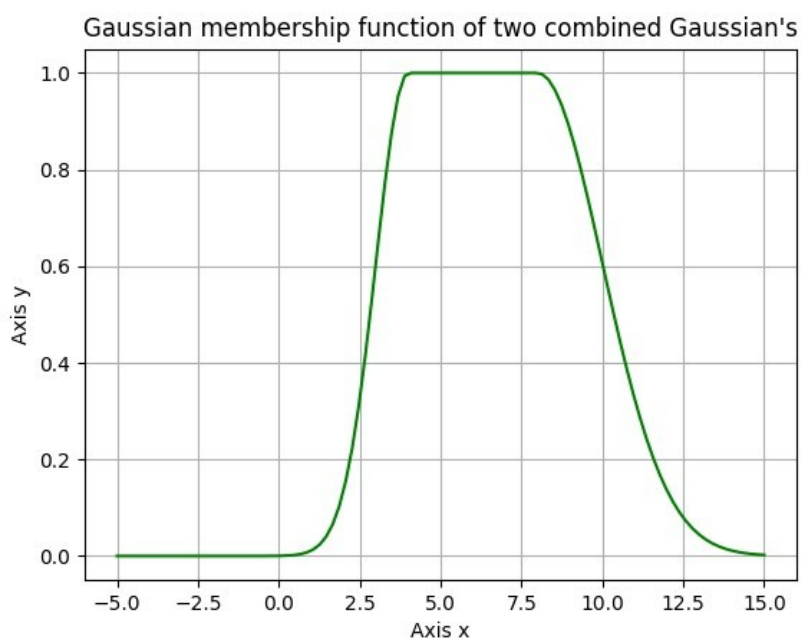
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Gaussian membership function of two combined Gaussian\'s')
```

```
plt.grid()
```

```
plt.show()
```



3. Побудувати функцію приналежності "узагальнений дзвін", яка дозволяє представляти нечіткі суб'єктивні переваги.

#3.Generalized Bell membership function

```
interval = np.linspace(-10, 20, 100)
```

```
membership_fn = fuzz.gbellmf(interval, 2, 2, 5)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

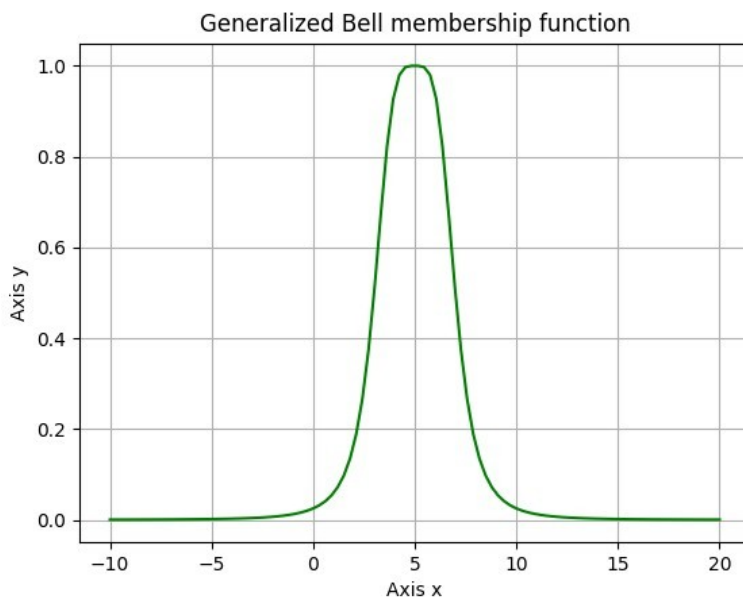
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Generalized Bell membership function')
```

```
plt.grid()
```

```
plt.show()
```



4. Побудувати набір сігмоїдних функцій: основну односторонню, яка відкрита зліва чи справа; додаткову двосторонню; додаткову несиметричну.

#4.Basic sigmoid membership function

```
interval = np.linspace(0, 10, 100)
```

```
membership_fn = fuzz.sigmf(interval, 3, -3)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

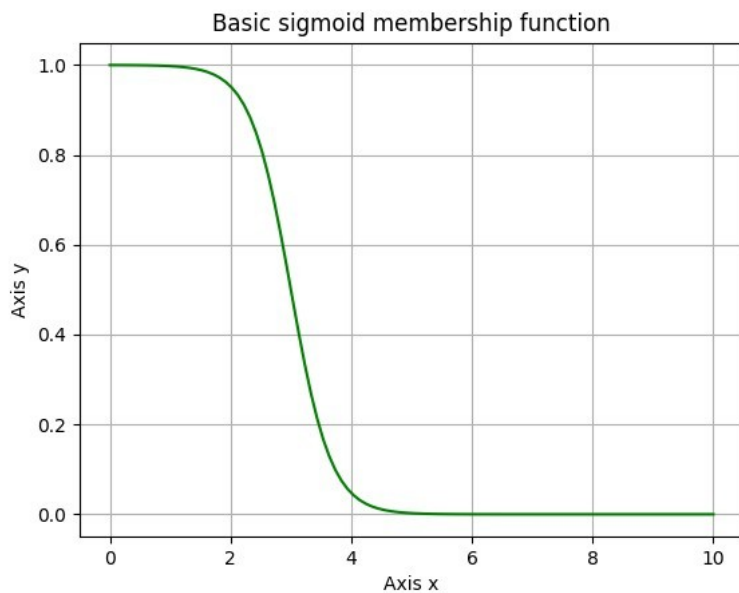
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Basic sigmoid membership function')
```

```
plt.grid()
```

```
plt.show()
```



#4. Difference of two sigmoid membership functions

```
interval = np.linspace(-5, 15, 100)
```

```
membership_fn = fuzz.dsigmf(interval, 1, 5, 7, 6)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

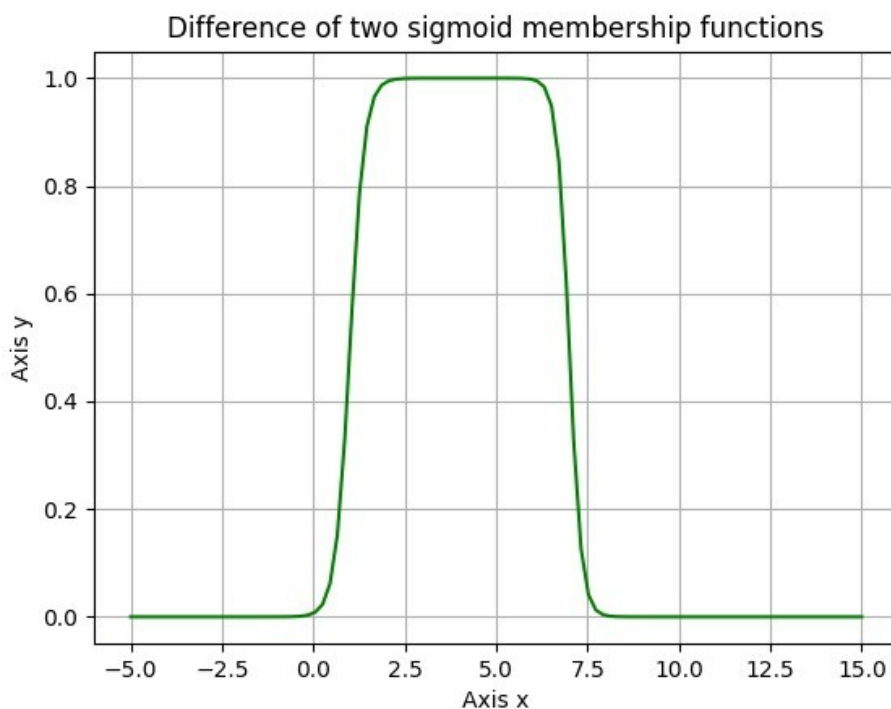
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Difference of two sigmoid membership functions')
```

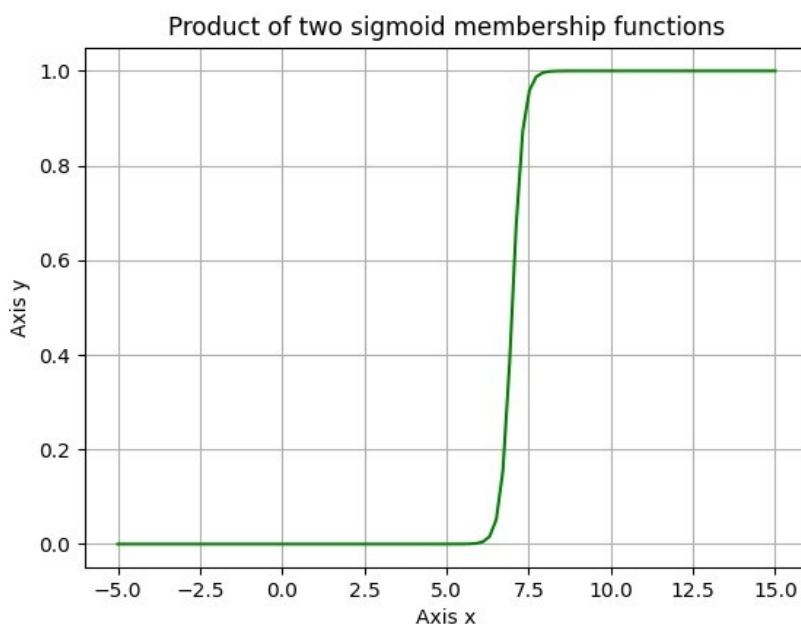
```
plt.grid()
```

```
plt.show()
```



#4. Product of two sigmoid membership functions

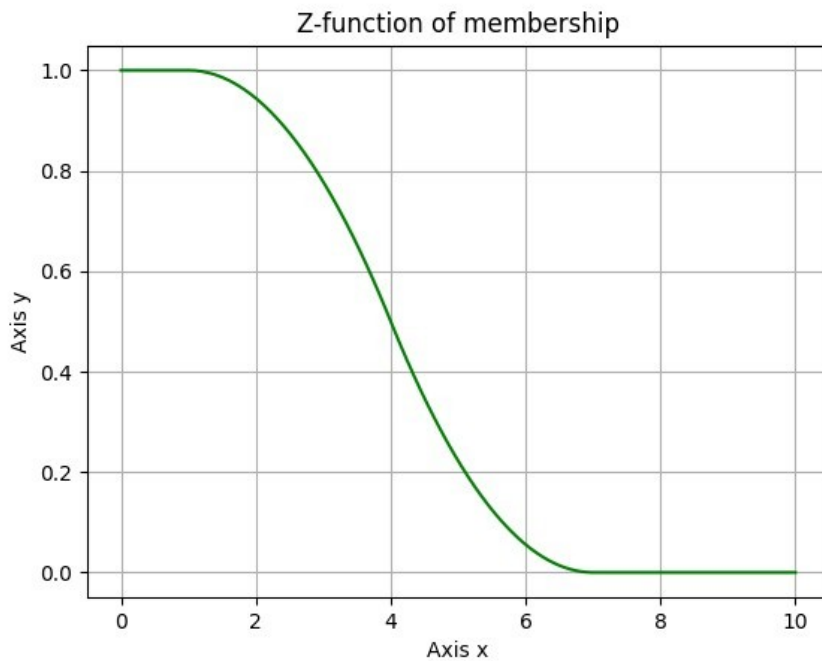
```
interval = np.linspace(-5, 15, 100)
membership_fn = fuzz.psigmf(interval, 1, 5, 7, 6)
_, axis = plt.subplots()
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('Product of two sigmoid membership functions')
plt.grid()
plt.show()
```



5. Побудувати набір поліноміальних функцій приналежності (Z-, PI- і S-функцій).

#5.Z-function of membership

```
interval = np.linspace(0, 10, 100)
membership = fuzz.zmf(interval, 1, 7)
_, axis = plt.subplots()
axis.plot(interval, membership, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('Z-function of membership')
plt.grid()
plt.show()
```



#5.PI-function of membership

```
interval = np.linspace(0, 10, 100)
```

```
membership_fn = fuzz.pimf(interval, 1, 3, 6, 9)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
```

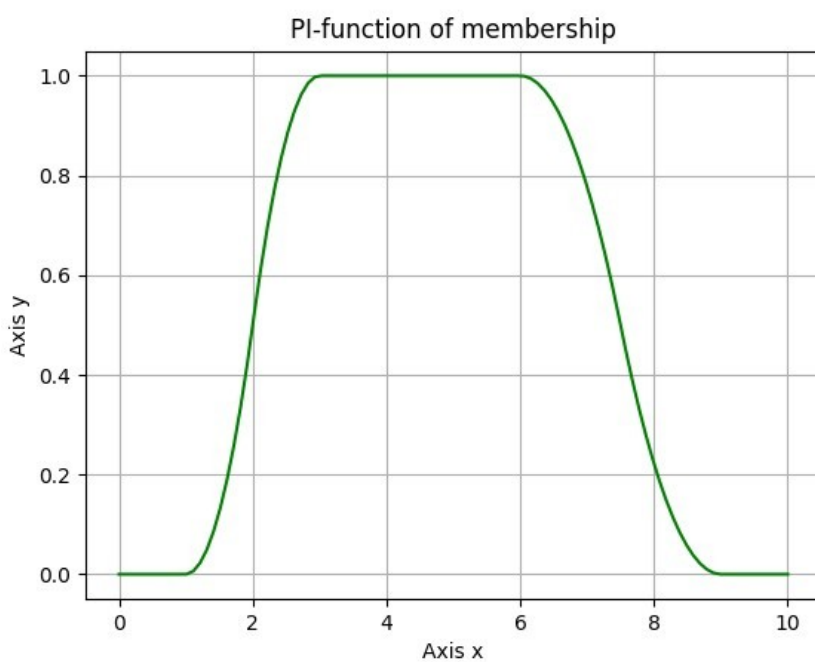
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('PI-function of membership')
```

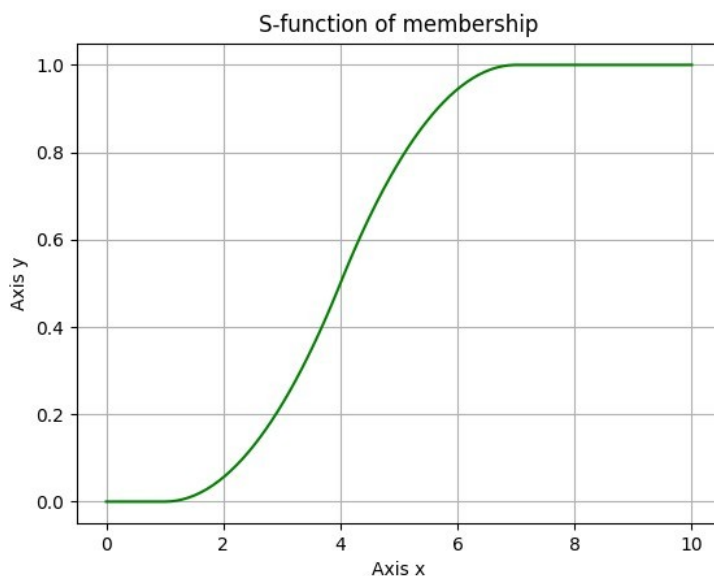
```
plt.grid()
```

```
plt.show()
```



#5.S-function of membership

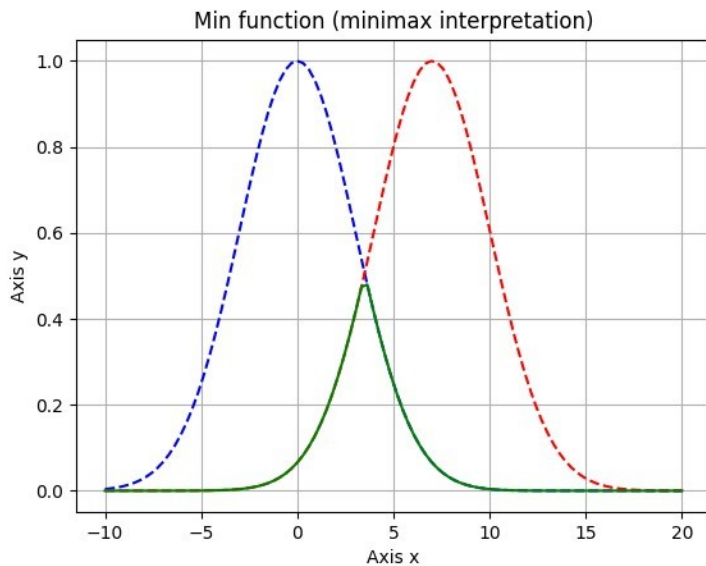
```
interval = np.linspace(0, 10, 100)
membership_fn = fuzz.smf(interval, 1, 7)
_, axis = plt.subplots()
axis.plot(interval, membership_fn, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('S-function of membership')
plt.grid()
plt.show()
```



6. Побудувати мінімаксну інтерпретацію логічних операторів з використанням операцій пошуку мінімуму і максимуму.

#6.Min function (minimax interpretation)

```
interval = np.linspace(-10, 20, 100)
membership_fn1 = fuzz.gaussmf(interval, 0, 3)
membership_fn2 = fuzz.gaussmf(interval, 7, 3)
min_fn = np.fmin(membership_fn1, membership_fn2)
_, axis = plt.subplots()
axis.plot(interval, membership_fn1, 'b', linestyle='--', linewidth=1.5)
axis.plot(interval, membership_fn2, 'r', linestyle='--', linewidth=1.5)
axis.plot(interval, min_fn, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('Min function (minimax interpretation)')
plt.grid()
plt.show()
```



#6.Max function (minimax interpretation)

```
interval = np.linspace(-10, 20, 100)
```

```
membership_fn1 = fuzz.gaussmf(interval, 0, 3)
```

```
membership_fn2 = fuzz.gaussmf(interval, 7, 3)
```

```
max_fn = np.fmax(membership_fn1, membership_fn2)
```

```
_, axis = plt.subplots()
```

```
axis.plot(interval, membership_fn1, 'b', linestyle='--', linewidth=1.5)
```

```
axis.plot(interval, membership_fn2, 'r', linestyle='--', linewidth=1.5)
```

```
axis.plot(interval, max_fn, 'g', linewidth=1.5)
```

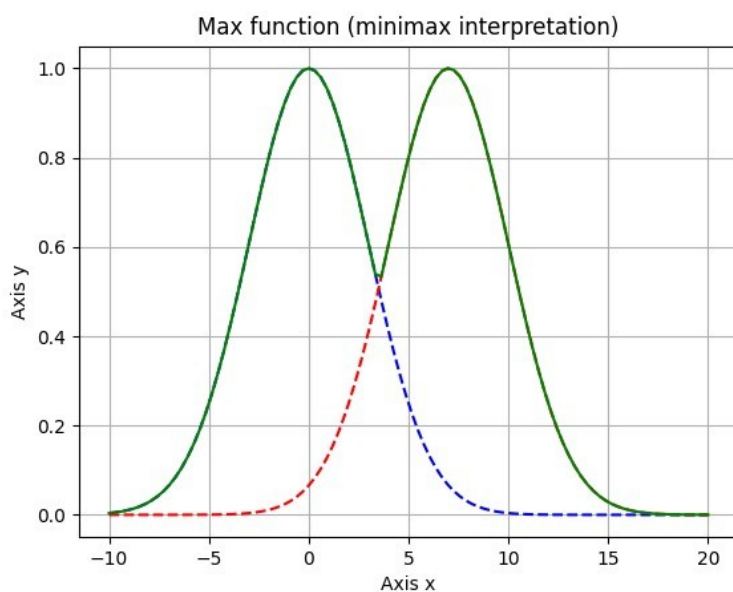
```
axis.set_xlabel('Axis x')
```

```
axis.set_ylabel('Axis y')
```

```
axis.set_title('Max function (minimax interpretation)')
```

```
plt.grid()
```

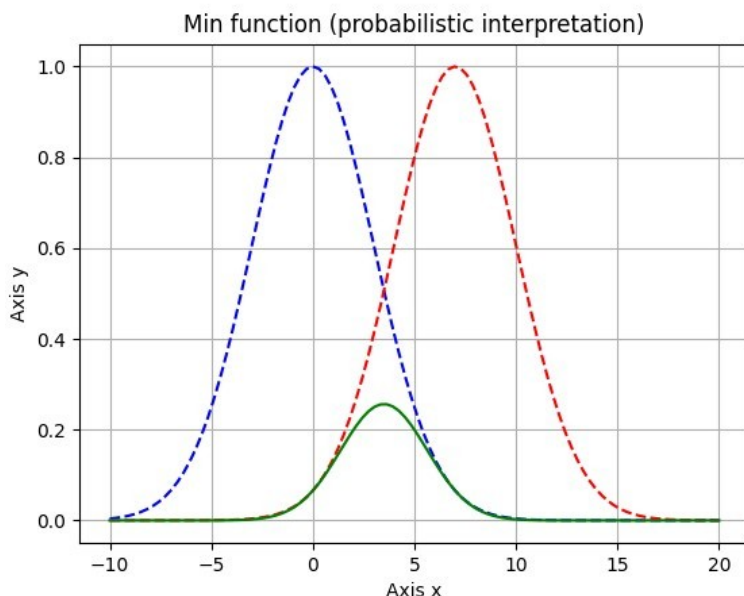
```
plt.show()
```



7. Побудувати вірогідну інтерпретацію кон'юнктивну і диз'юнктивних операторів.

#7.Min function (probabilistic interpretation)

```
interval = np.linspace(-10, 20, 100)
membership_fn1 = fuzz.gaussmf(interval, 0, 3)
membership_fn2 = fuzz.gaussmf(interval, 7, 3)
prod = membership_fn1 * membership_fn2
_, axis = plt.subplots()
axis.plot(interval, membership_fn1, 'b', linestyle='--', linewidth=1.5)
axis.plot(interval, membership_fn2, 'r', linestyle='--', linewidth=1.5)
axis.plot(interval, prod, 'g', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title("Min function (probabilistic interpretation)")
plt.grid()
plt.show()
```



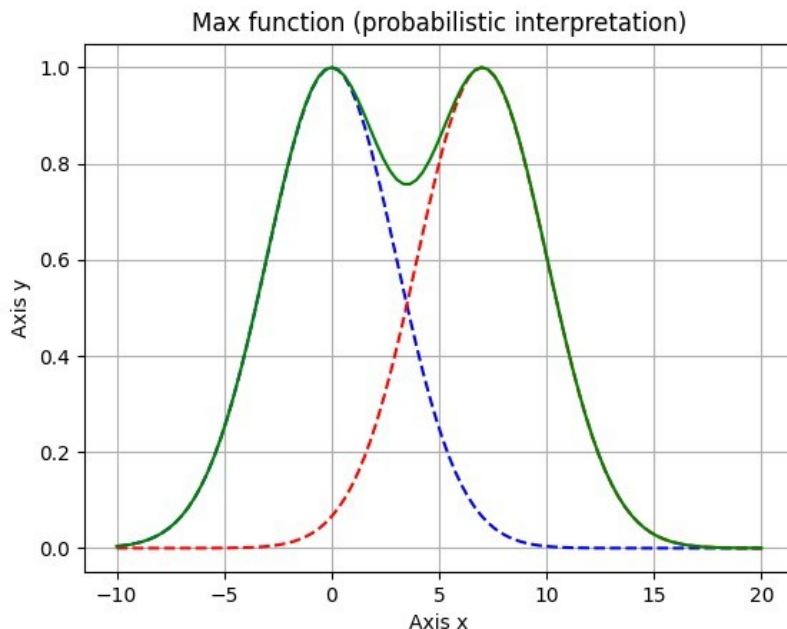
#7.Max function (probabilistic interpretation)

```
interval = np.linspace(-10, 20, 100)
membership_fn1 = fuzz.gaussmf(interval, 0, 3)
membership_fn2 = fuzz.gaussmf(interval, 7, 3)
prod = membership_fn1 * membership_fn2
algSum = membership_fn1 + membership_fn2 - prod
_, axis = plt.subplots()
axis.plot(interval, membership_fn1, 'b', linestyle='--', linewidth=1.5)
axis.plot(interval, membership_fn2, 'r', linestyle='--', linewidth=1.5)
axis.plot(interval, algSum, 'g', linewidth=1.5)
```

```

axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title("Max function (probabilistic interpretation)")
plt.grid()
plt.show()

```



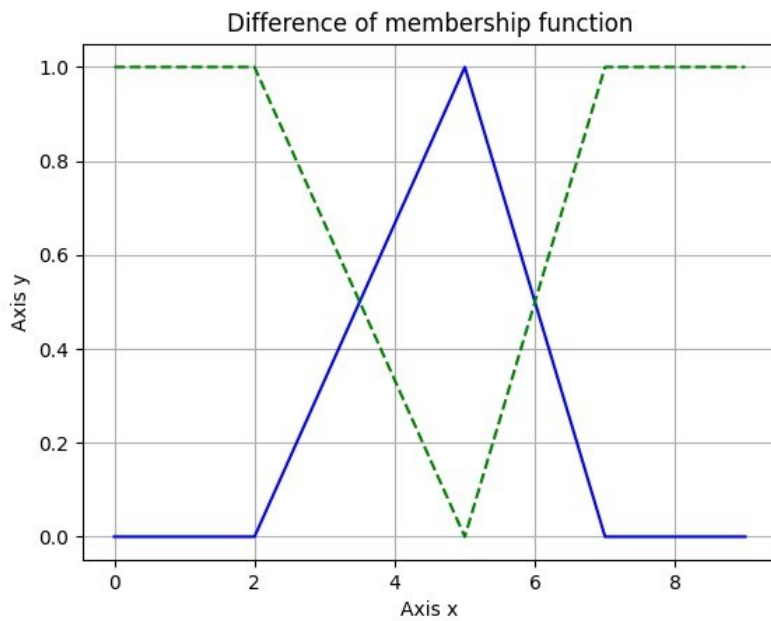
8. Побудувати доповнення нечіткої множини, яке описує деяке розмите судження і представляє собою математичний опис вербального вираження, який заперечує це нечітка множина.

#8.Difference of membership function

```

interval = np.arange(0, 10, 1)
membership_fn = fuzz.trimf(interval, [2, 5, 7])
diff_plot = 1 - membership_fn
_, axis = plt.subplots()
axis.plot(interval, membership_fn, 'b', linewidth=1.5)
axis.plot(interval, diff_plot, 'g', linestyle='--', linewidth=1.5)
axis.set_xlabel('Axis x')
axis.set_ylabel('Axis y')
axis.set_title('Difference of membership function')
plt.grid()
plt.show()

```



Висновок: Під час виконання лабораторної роботи я розглянув різні типи функцій приналежності (трикутні, трапецієподібні, Гауссівські, "узагальнений дзвін", сігмоїдні, поліноміальні). Також виконав основні логічні операції над нечіткими множинами (об'єднання, перетин та заперечення) та їх мінімаксні та вірогідні інтерпретації.