

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

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

„Дослідження способів формування нечітких множин і операцій над ними”

Виконав(ла)

ІІ-14 Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірив

Шимкович Володимир Миколайович
(прізвище, ім'я, по батькові)

Київ 2023

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

Постановка задачі: за допомогою пакетів моделювання або мови програмування високого рівня:

1. Побудувати трикутну і трапецієподібну функцію приналежності.
2. Побудувати просту і двосторонню функцію приналежності Гаусса, утворену за допомогою різних функцій розподілу.
3. Побудувати функцію приналежності "узагальнений дзвін", яка дозволяє представляти нечіткі суб'єктивні переваги.
4. Побудувати набір сігмоїдних функцій: основну односторонню, яка відкрита зліва чи справа; додаткову двосторонню; додаткову несиметричну.
5. Побудувати набір поліноміальних функцій приналежності (Z-, PI- і S-функцій).
6. Побудувати мінімаксну інтерпретацію логічних операторів з використанням операцій пошуку мінімуму і максимуму.
7. Побудувати вірогідну інтерпретацію кон'юнктивну і диз'юнктивних операторів.
8. Побудувати доповнення нечіткої множини, яке описує деяке розмите судження і представляє собою математичний опис вербального вираження, який заперечує це нечітка множина.

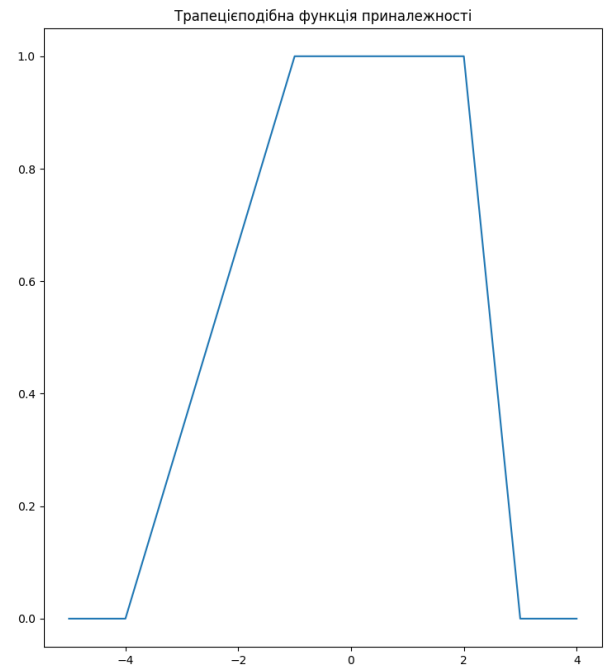
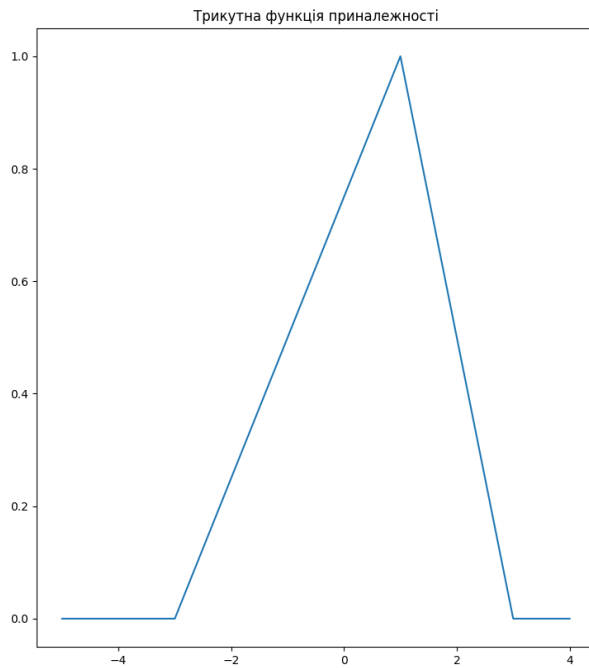
При виконанні пунктів 1 - 8 індивідуального завдання, значення змінних a, b, c, d і т.д. необхідно вибирати довільним чином.

9. Оформіть звіт по лабораторній роботі.

Виконання завдання.

Для виконання завдання була обрана мова програмування високого рівня Python.

1. Побудувати трикутну і трапецієподібну функцію приналежності.



```
# 1. Побудувати трикутну і трапецієподібну функцію приналежності

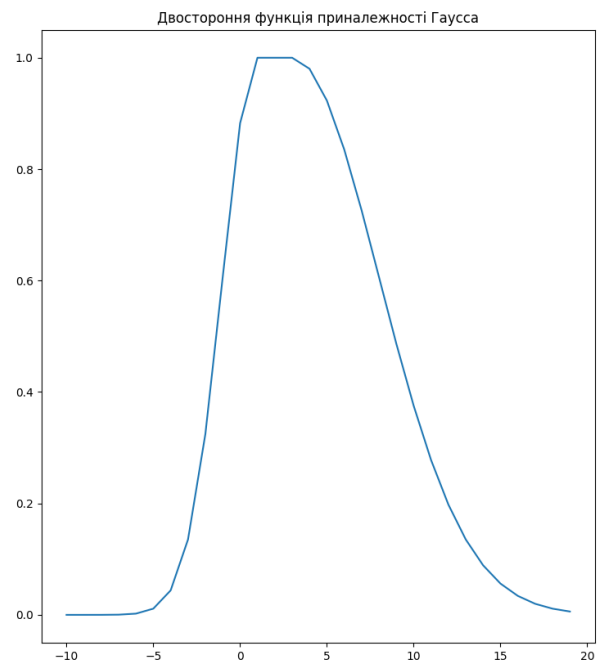
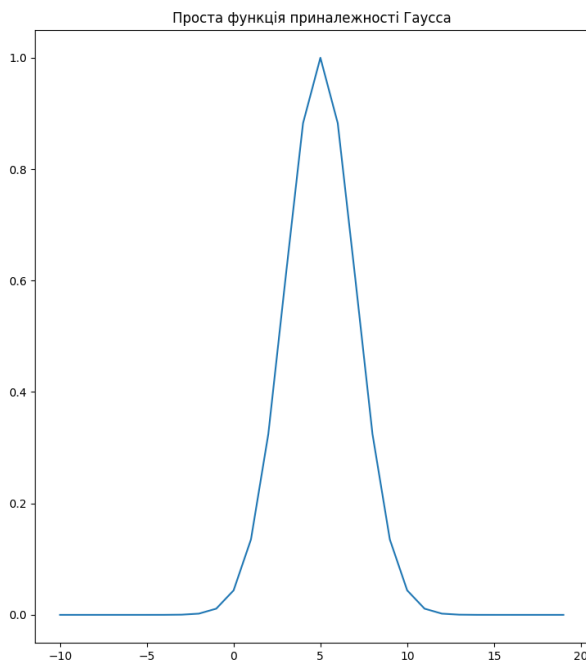
x = np.arange(-5, 5, 1)

# Трикутна функція приналежності
y = fuzzy.trimf(x, [-3, 1, 3])
plt.subplot(121)
plt.plot(x, y)
plt.title('Трикутна функція приналежності')

# Трапецієподібна функція приналежності
y = fuzzy.trapmf(x, [-4, -1, 2, 3])
plt.subplot(122)
plt.plot(x, y)
plt.title('Трапецієподібна функція приналежності')

plt.show()
```

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



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

```
x = np.arange(-10, 20, 1)
```

```
# Проста функція приналежності Гаусса
```

```
y = fuzzy.gaussmf(x, 5, 2)
```

```
plt.subplot(121)
```

```
plt.plot(x, y)
```

```
plt.title('Проста функція приналежності Гаусса')
```

```
# Двостороння функція приналежності Гаусса
```

```
y = fuzzy.gauss2mf(x, 1, 2, 3, 5)
```

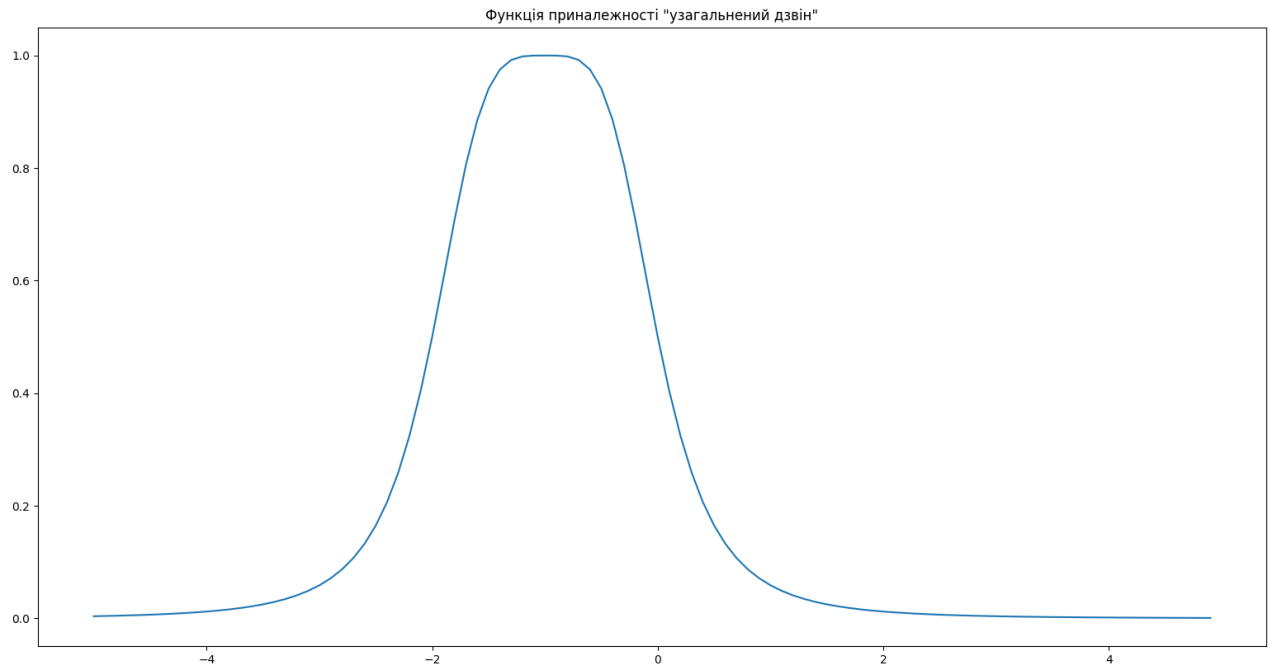
```
plt.subplot(122)
```

```
plt.plot(x, y)
```

```
plt.title('Двостороння функція приналежності Гаусса')
```

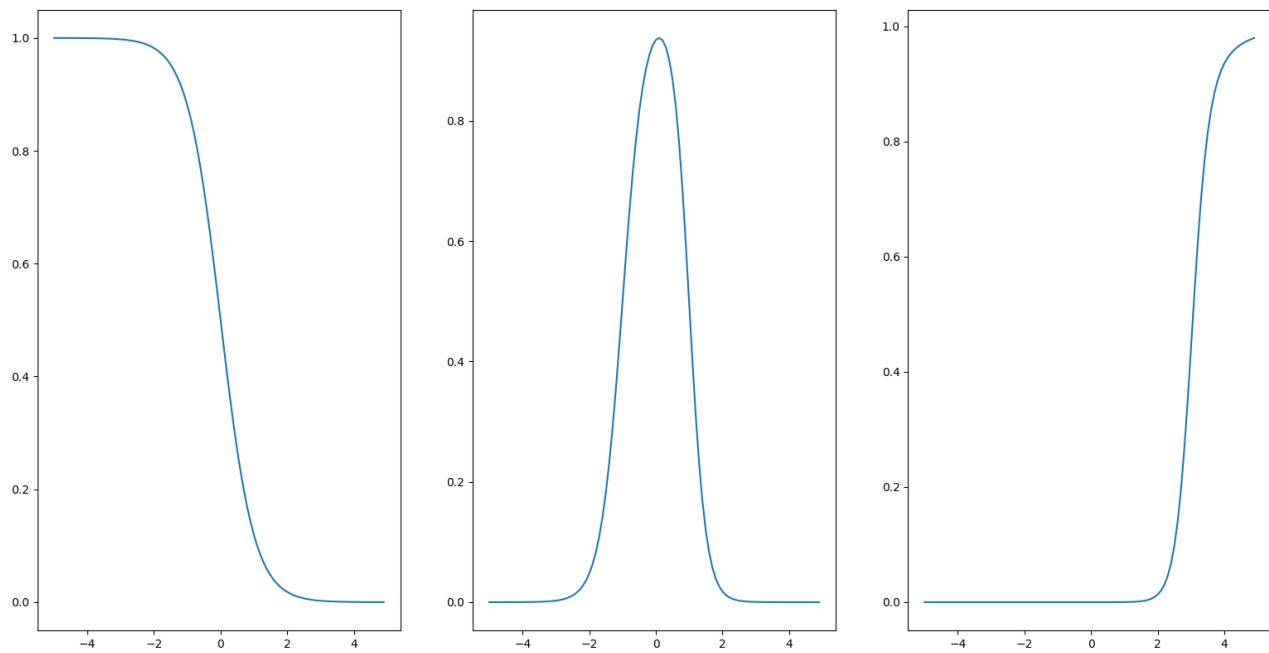
```
plt.show()
```

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



```
# 3. Побудувати функцію приналежності "узагальнений дзвін",  
# яка дозволяє представляти нечіткі суб'єктивні переваги  
  
x = np.arange(-5, 5, 0.1)  
  
# Функція приналежності "узагальнений дзвін"  
y = fuzzy.gbellmf(x, 1, 2, -1)  
plt.plot(x, y)  
plt.title('Функція приналежності "узагальнений дзвін"')  
plt.show()
```

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



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

```
x = np.arange(-5, 5, 0.1)
```

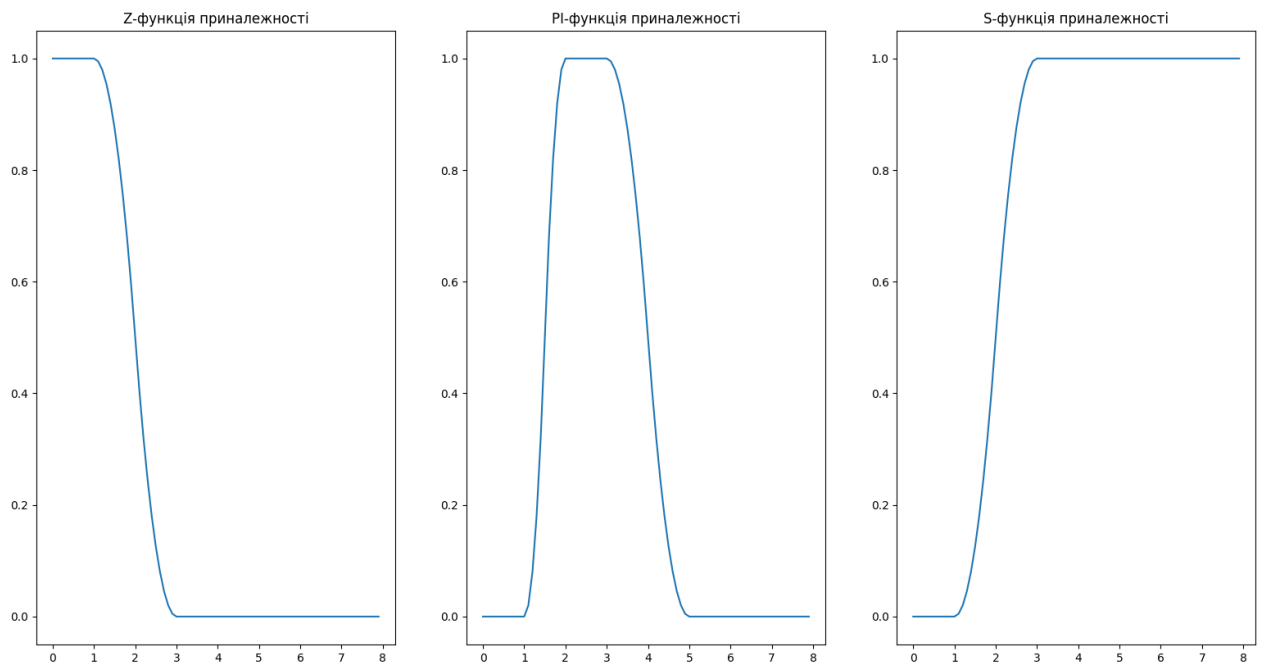
```
y = fuzzy.sigmf(x, 0, -2)
plt.subplot(131)
plt.plot(x, y)
```

```
y = fuzzy.dsigmf(x, -1, 3, 1, 4)
plt.subplot(132)
plt.plot(x, y)
```

```
y = fuzzy.psigmf(x, 1, 1, 3, 4)
plt.subplot(133)
plt.plot(x, y)
```

```
plt.suptitle('Сигмоїдальні функції приналежності')
plt.show()
```

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



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

x = np.arange(0, 8, 0.1)

# Z-функція приналежності
y = fuzzy.zmf(x, 1, 3)
plt.subplot(131)
plt.plot(x, y)
plt.title("Z-функція приналежності")

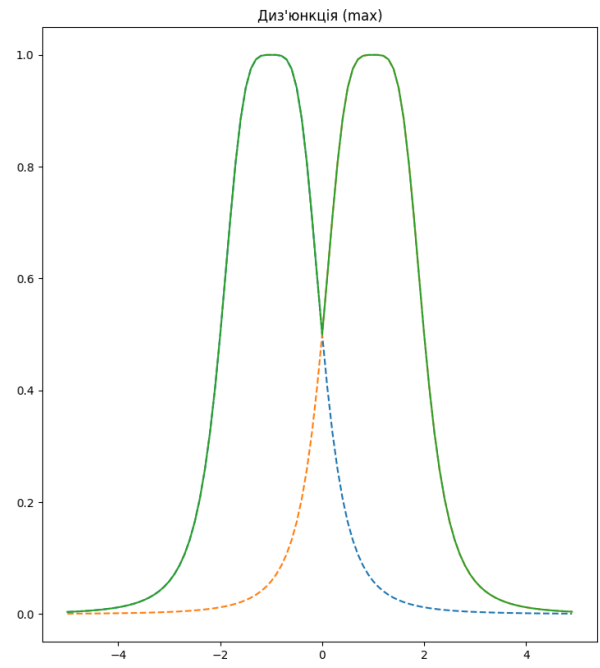
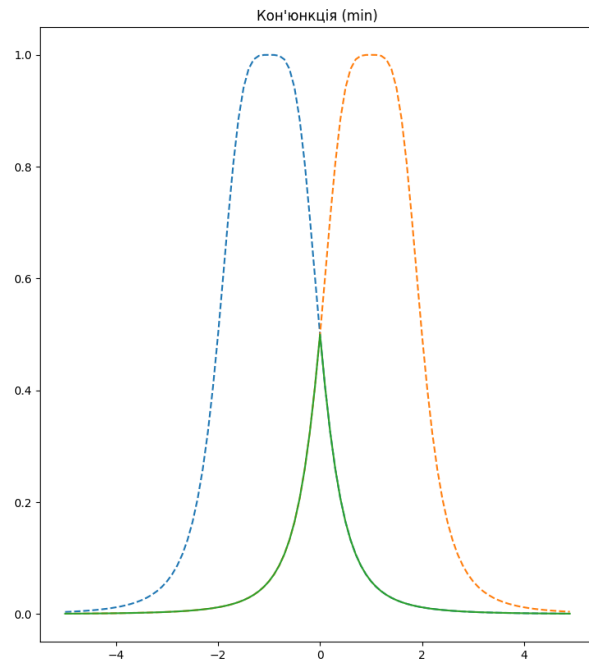
# PI-функція приналежності
y = fuzzy.pimf(x, 1, 2, 3, 5)
plt.subplot(132)
plt.plot(x, y)
plt.title('PI-функція приналежності')

# S-функція приналежності
y = fuzzy.smf(x, 1, 3)
plt.subplot(133)
plt.plot(x, y)
plt.title('S-функція приналежності')

plt.show()
```

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

Мінімаксна інтерпретація логічних операторів




```

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

x = np.arange(-5, 5, 0.1)

# Функції приналежності "узагальнений дзвін"
y1 = fuzzy.gbellmf(x, 1, 2, -1)
y2 = fuzzy.gbellmf(x, 1, 2, 1)

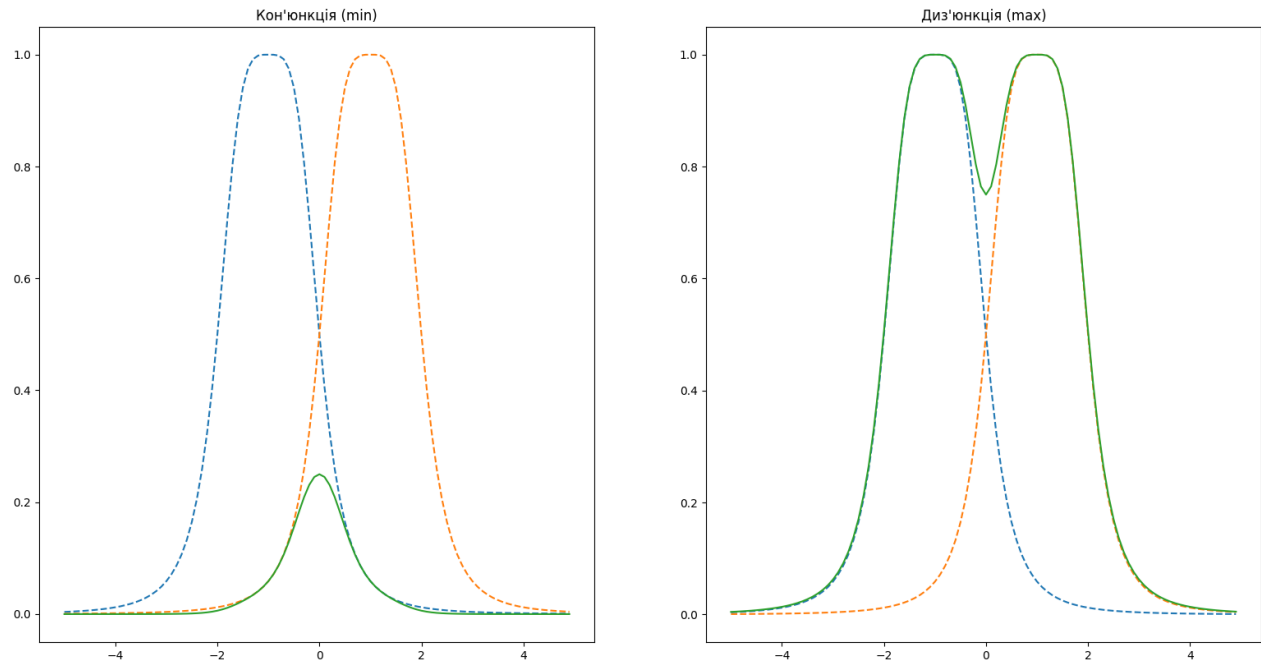
# Кон'юнкція (min)
min_func = np.fmin(y1, y2)
plt.subplot(121)
plt.plot(x, y1, linestyle='--')
plt.plot(x, y2, linestyle='--')
plt.plot(x, min_func)
plt.title('Кон\'юнкція (min)')

# Диз'юнкція (max)
max_func = np.fmax(y1, y2)
plt.subplot(122)
plt.plot(x, y1, linestyle='--')
plt.plot(x, y2, linestyle='--')
plt.plot(x, max_func)
plt.title('Диз\'юнкція (max)')

plt.suptitle("Мінімаксна інтерпретація логічних операторів")
plt.show()

```

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



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

x = np.arange(-5, 5, 0.1)

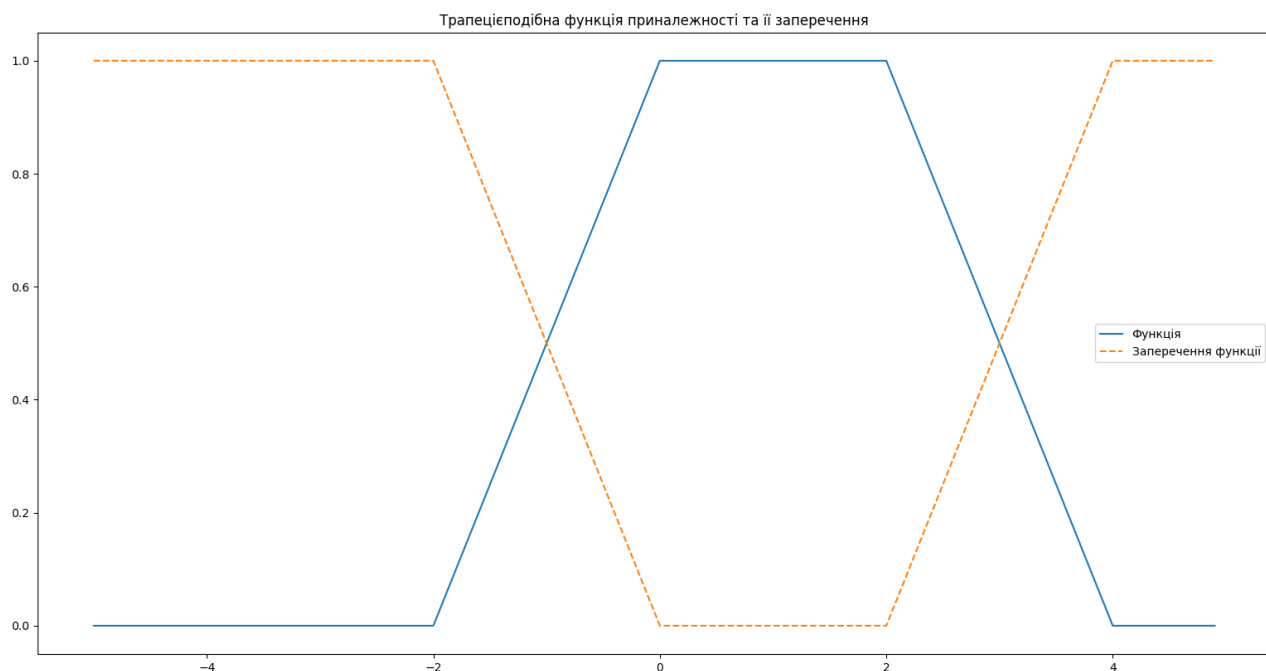
# Функції приналежності "узагальнений дзвін"
y1 = fuzzy.gbellmf(x, 1, 2, -1)
y2 = fuzzy.gbellmf(x, 1, 2, 1)

# Кон'юнкція (min)
min_func = y1 * y2
plt.subplot(121)
plt.plot(x, y1, linestyle='--')
plt.plot(x, y2, linestyle='--')
plt.plot(x, min_func)
plt.title('Кон\'юнкція (min)')

# Диз'юнкція (max)
max_func = y1 + y2 - y1 * y2
plt.subplot(122)
plt.plot(x, y1, linestyle='--')
plt.plot(x, y2, linestyle='--')
plt.plot(x, max_func, label='Max function')
plt.title('Диз\'юнкція (max)')

plt.suptitle("Вірогідна інтерпретація кон'юнктивних і диз'юнктивних операторів")
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. Побудувати доповнення нечіткої множини,  
# яке описує деяке розмите судження і представляє собою  
# математичний опис вербального вираження,  
# який заперечує це нечітка множина  
  
x = np.arange(-5, 5, 0.1)  
  
# Трапецієподібна функція приналежності та її заперечення  
y0 = fuzzy.trapmf(x, [-2, 0, 2, 4])  
y1 = 1 - y0  
plt.plot(x, y0, label='Функція')  
plt.plot(x, y1, linestyle='--', label='Заперечення функції')  
plt.title('Трапецієподібна функція приналежності та її заперечення')  
plt.legend()  
plt.show()
```

Висновок.

При виконанні лабораторної роботи було побудовано нечіткі множини з використанням різних типів функцій приналежності: трикутна, трапецієподібна, Гаусса, «узагальнений дзвін», сигмоїдні, поліноміальні. Також були виконані логічні операції диз'юнкції, кон'юнкції та заперечення над

нечіткими множинами. Поставлені завдання виконувалися засобами мови програмування Python та бібліотек numpy, skfuzzy та matplotlib.