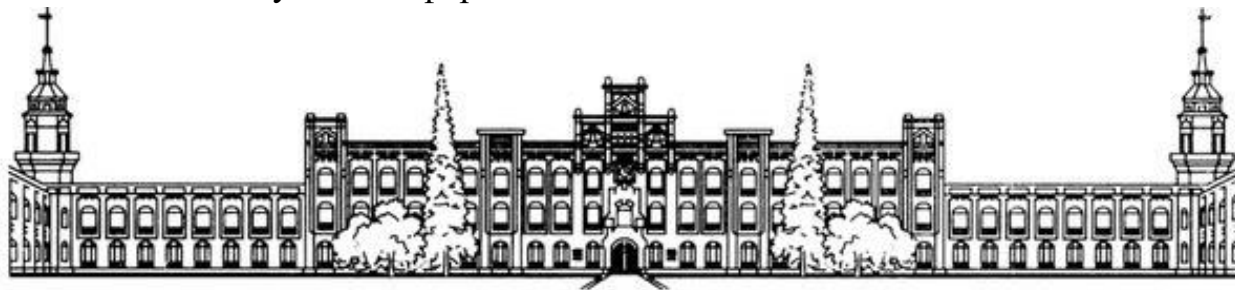


Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки



Кафедра інформаційних систем та технологій

Лабораторна робота №1

з дисципліни «Методи та технології штучного інтелекту»

на тему

«Дослідження способів формування нечітких множин і
операцій над ними»

Виконала:
студентка групи ІС-12
Павлова Софія

Викладач:
Шимкович В. М.

1. Постановка задачі

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

Завдання:

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.

2. Виконання

Трикутна функція приналежності

Лістинг:

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

# Діапазон значень x
x = np.linspace(0, 10, 1000)

# Трикутна функція приналежності
trimf_values = fuzz.trimf(x, [2, 5, 6])
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, trimf_values, label='(2, 5, 6)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік трикутної функції приналежності')
plt.show()
```

Результат:

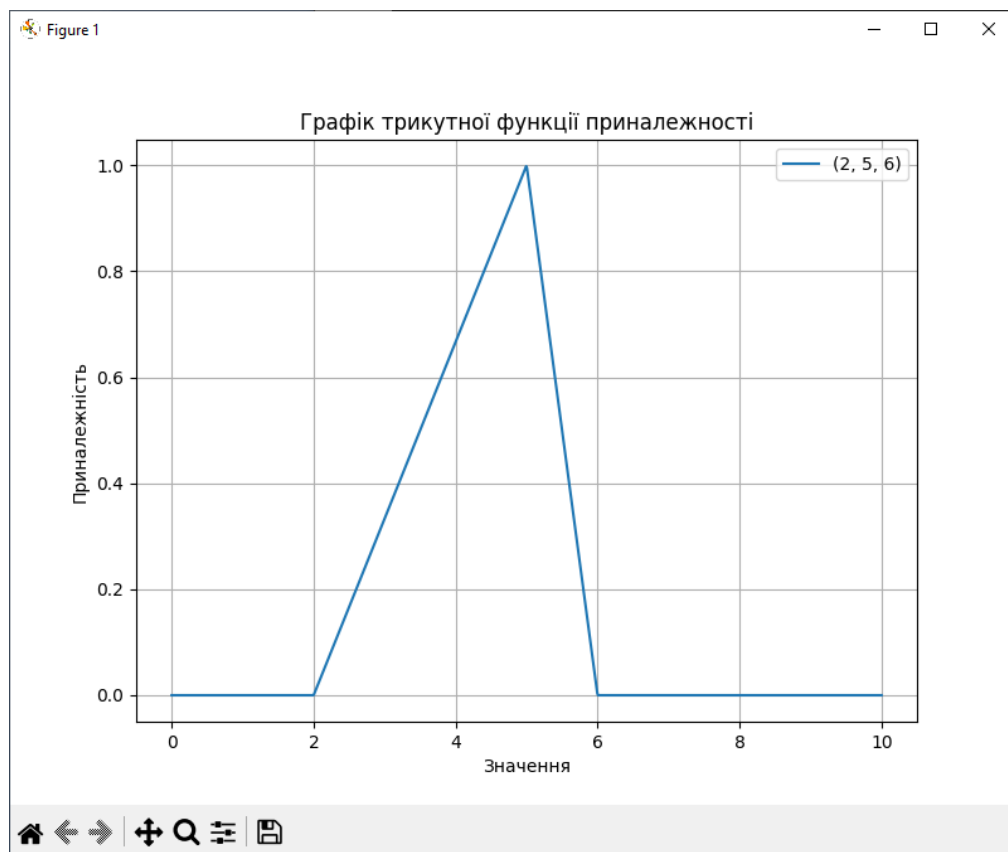


Рисунок 1 – Графік трикутної функції приналежності

Трапецієподібна функція приналежності

Лістинг:

```
# Трапецієподібна функція приналежності
trapmf_values = fuzz.trapmf(x, [3, 4, 7, 9])
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, trapmf_values, label='(3, 4, 7, 9)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік трапецієподібної функції приналежності')
plt.show()
```

Результат:

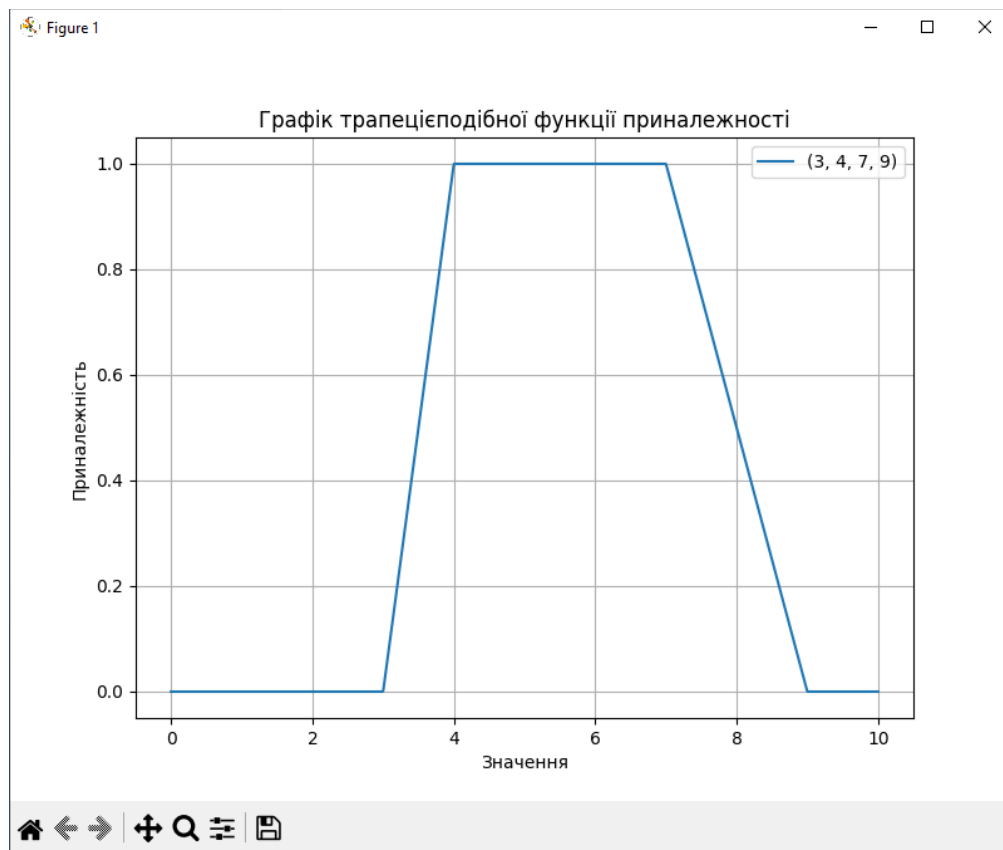


Рисунок 2 – Графік трапецієподібної функції приналежності

Проста функція приналежності Гаусса

Лістинг:

```
# Проста функція приналежності Гаусса
gaussmf_values = fuzz.gaussmf(x, 6, 1)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, gaussmf_values, label='(6, 1)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік простої Гауссівської ФП')
plt.show()
```

Результат:

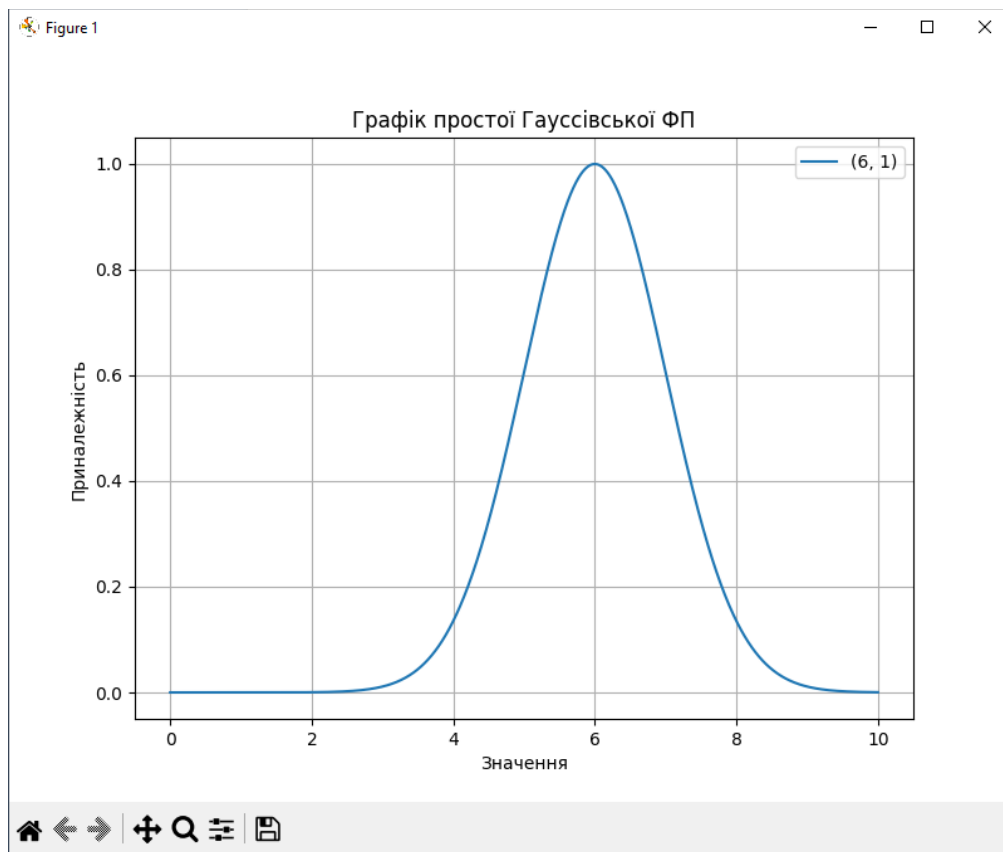


Рисунок 3 – Графік простої Гауссівської ФП

Двостороння функція приналежності Гаусса

Лістинг:

```
# Двостороння функція приналежності Гаусса
gauss2mf_values = fuzz.gauss2mf(x, 5, 1.5, 6, 1)
gaussmf_1_values = fuzz.gaussmf(x, 5, 1.5)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, gauss2mf_values, label='(5, 1.5, 6, 1)')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.plot(x, gaussmf_1_values, linestyle='dotted', label='(5, 1.5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік двосторонньої Гауссівської ФП')
plt.show()
```

Результат:

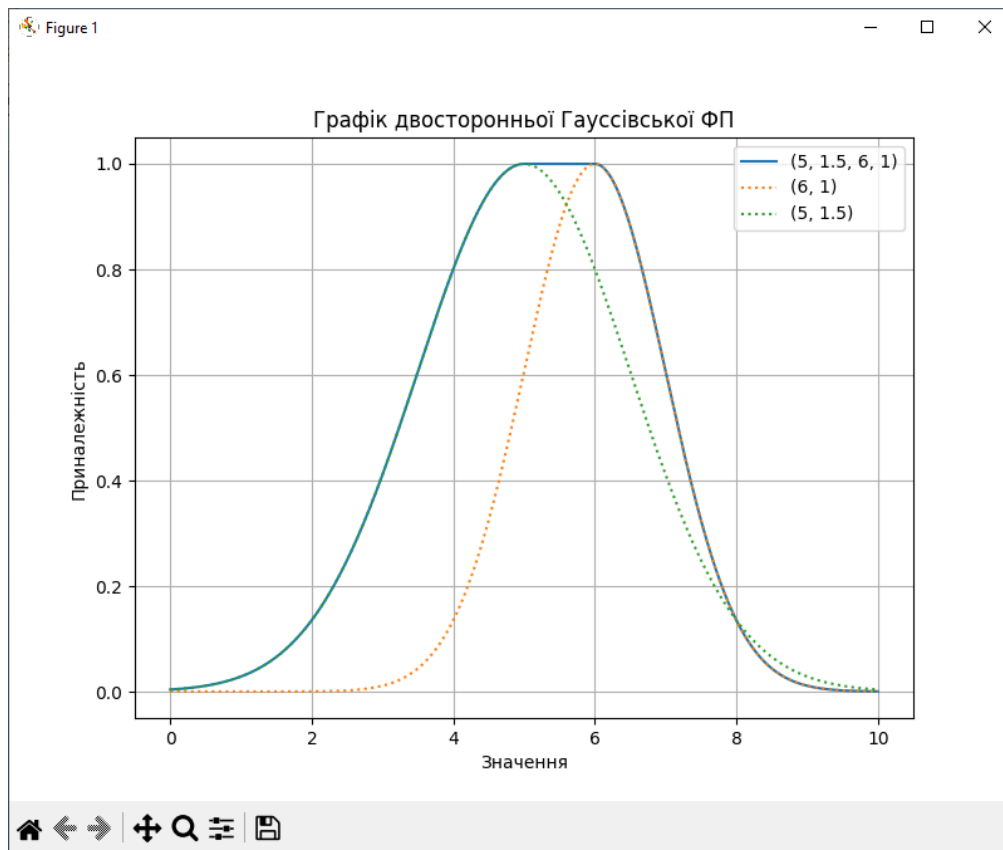


Рисунок 4 – Графік двосторонньої Гауссівської ФП

Функція приналежності "Узагальнений дзвін"

Лістинг:

```
# Функція приналежності "Узагальнений дзвін"
gbellmf_values = fuzz.gbellmf(x, 2, 3, 5)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, gbellmf_values, label='(2, 3, 5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік функції приналежності "Узагальнений дзвін"')
plt.show()
```

Результат:

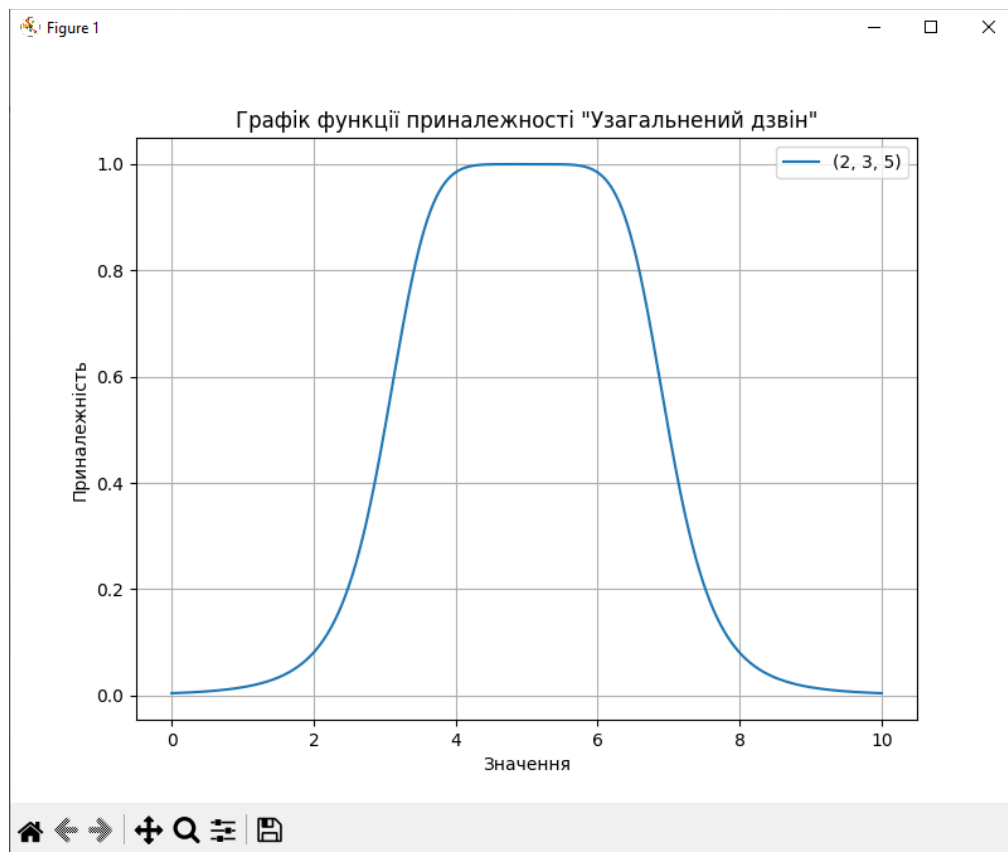


Рисунок 5 – Графік функції приналежності "Узагальнений дзвін"

Сигмоїдні функції приналежності

Лістинг:

```
# Основна одностороння сигмоїдна функція приналежності
sigmf_values = fuzz.sigmf(x, 2, 4)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, sigmf_values, label='(2, 4)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік основної односторонньої сигмоїдної ФП')
plt.show()

# Додаткова двостороння сигмоїдна функція приналежності
dsigmf_values = fuzz.dsigmf(x, 2, 4, 6, 2)
sigmf_1_values = fuzz.sigmf(x, 6, 2)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, dsigmf_values, label='(2, 4, 6, 2)')
plt.plot(x, sigmf_values, linestyle='dotted', label='(2, 4)')
plt.plot(x, sigmf_1_values, linestyle='dotted', label='(6, 2)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік додаткової двосторонньої сигмоїдної ФП')
plt.show()

# Додаткова несиметрична сигмоїдна функція приналежності
psigmf_values = fuzz.psigmf(x, 2, 4, 6, 2)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, psigmf_values, label='(2, 4, 6, 2)')
plt.plot(x, sigmf_values, linestyle='dotted', label='(2, 4)')
plt.plot(x, sigmf_1_values, linestyle='dotted', label='(6, 2)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік додаткової несиметричної сигмоїдної ФП')
plt.show()
```


Результат:

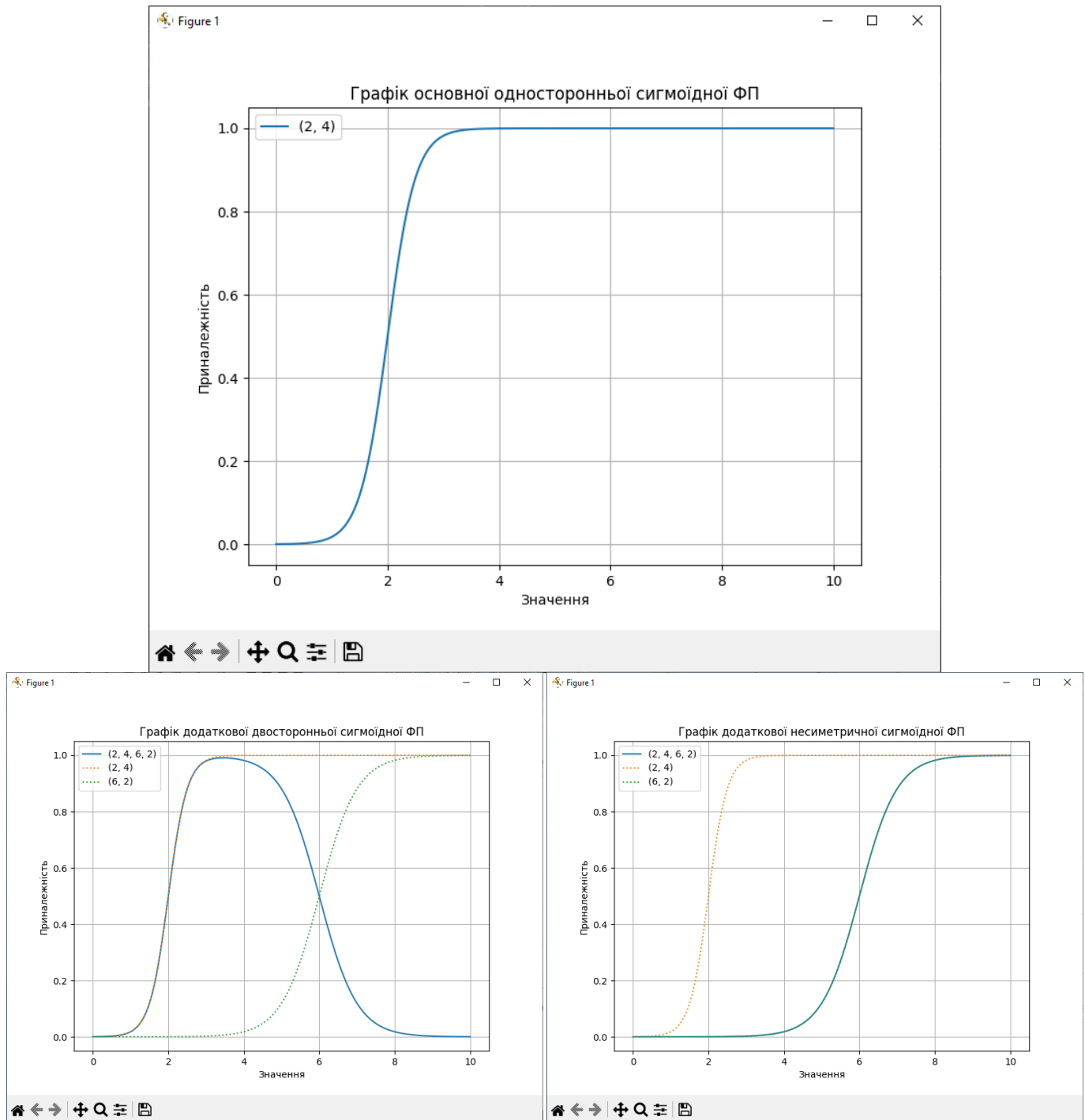


Рисунок 6 – Графіки основної односторонньої, додаткової двосторонньої та додаткової несиметричної сигмоїдної ФП

Поліноміальні функції приналежності

Лістинг:

```
# Поліноміальна Z-функція приналежності
zmf_values = fuzz.zmf(x, 2, 4)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, zmf_values, label='(2, 4)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної Z-функції приналежності')
plt.show()

# Поліноміальна S-функція приналежності
smf_values = fuzz.smf(x, 5, 8)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, smf_values, label='(5, 8)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної S-функції приналежності')
plt.show()

# Поліноміальна PI-функція приналежності
pimf_values = fuzz.pimf(x, 2, 4, 5, 8)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, pimf_values, label='(2, 4, 5, 8)')
plt.plot(x, zmf_values, linestyle='dotted', label='(2, 4)')
plt.plot(x, smf_values, linestyle='dotted', label='(5, 8)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік поліноміальної PI-функції приналежності')
plt.show()
```

Результат:

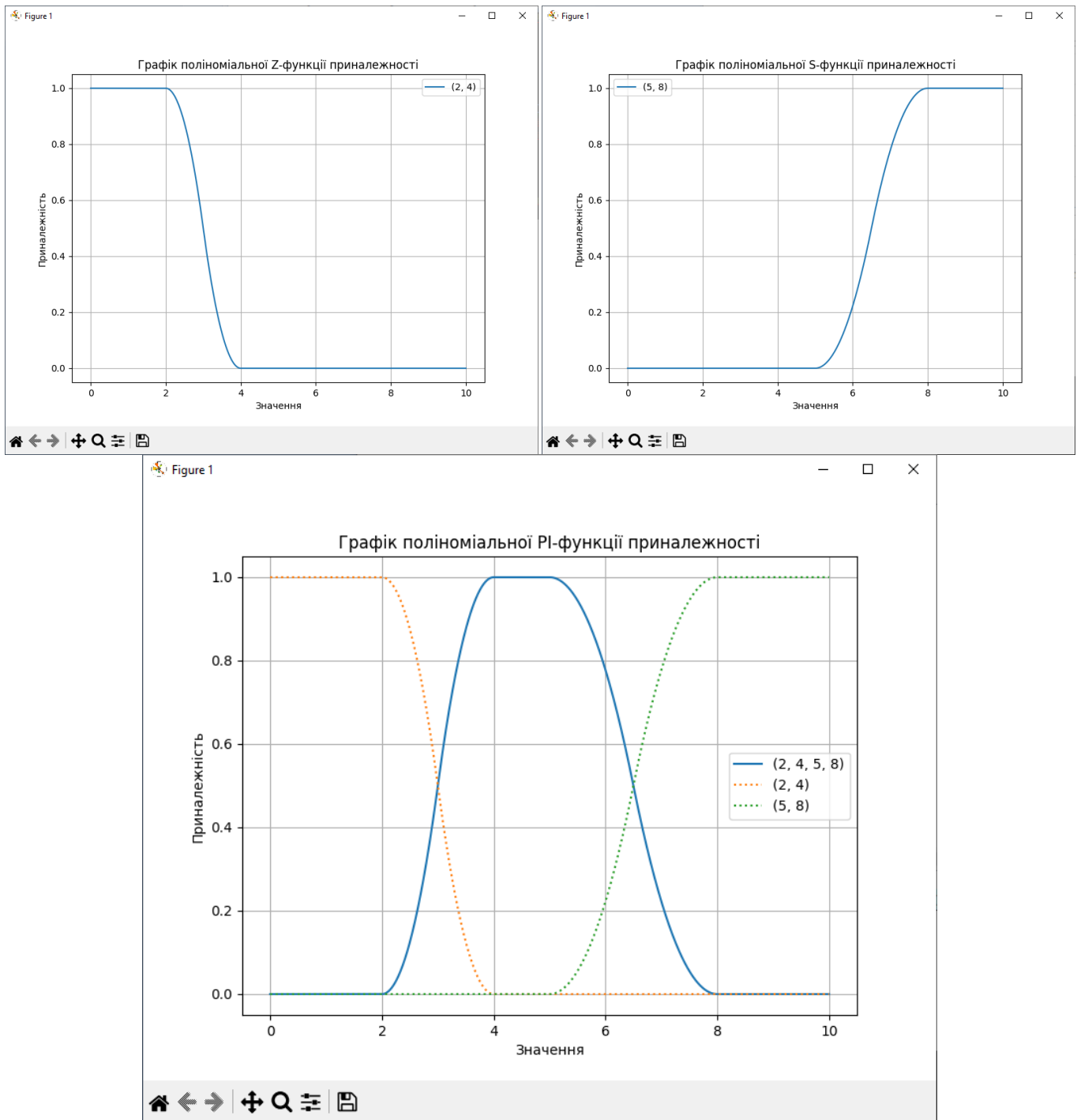


Рисунок 7 – Графіки поліноміальної S-, Z- та PI- ФП

Мінімаксна інтерпретація

Лістинг:

```
# Мінімаксна інтерпретація логічного оператора AND (кон'юнкція)
min_values = np.fmin(gaussmf_values, gaussmf_1_values)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, min_values, label='Мінімум')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.plot(x, gaussmf_1_values, linestyle='dotted', label='(5, 1.5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік мінімаксної інтерпретації логічного оператора AND')
plt.show()

# Мінімаксна інтерпретація логічного оператора OR (диз'юнкція)
max_values = np.fmax(gaussmf_values, gaussmf_1_values)
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, max_values, label='Максимум')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.plot(x, gaussmf_1_values, linestyle='dotted', label='(5, 1.5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік мінімаксної інтерпретації логічного оператора OR')
plt.show()
```

Результат:

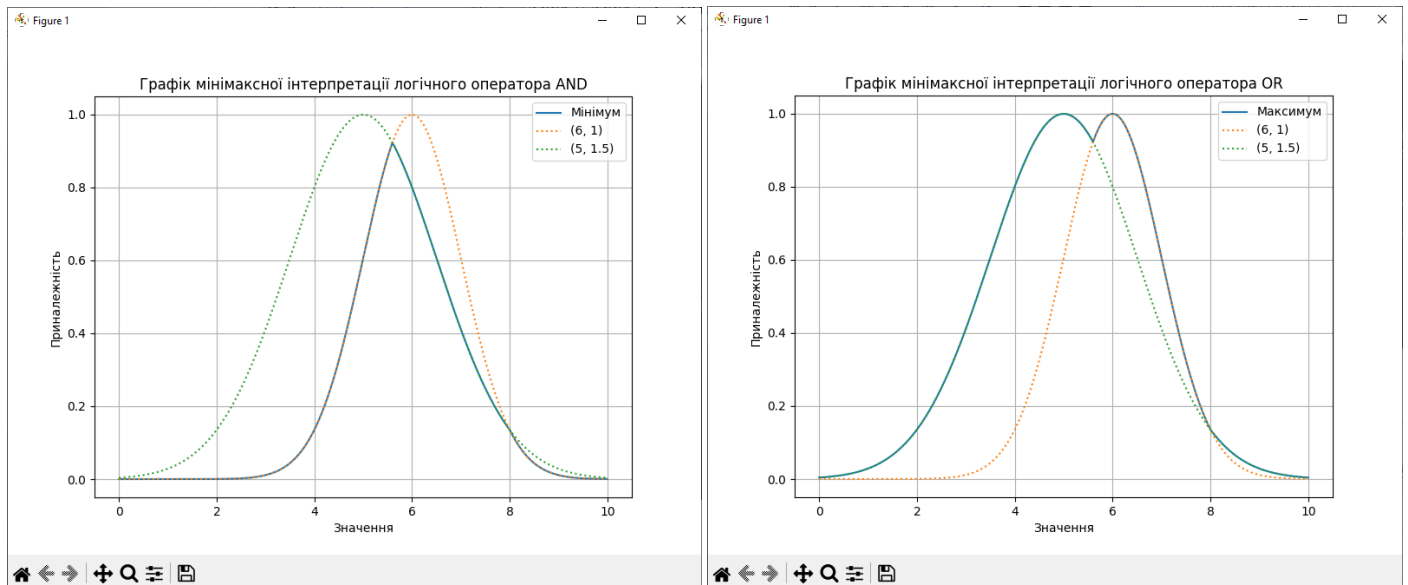


Рисунок 8 – Графіки мінімаксної інтерпретації кон'юнкції та диз'юнкції

Імовірнісна інтерпретація

Лістинг:

```
# Імовірнісна інтерпретація логічного оператора AND (кон'юнкція)
i_min_values = gaussmf_values * gaussmf_1_values
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, i_min_values, label='Мінімум')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.plot(x, gaussmf_1_values, linestyle='dotted', label='(5, 1.5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік імовірнісної інтерпретації логічного оператора AND')
plt.show()

# Імовірнісна інтерпретація логічного оператора OR (диз'юнкція)
i_max_values = gaussmf_values + gaussmf_1_values - i_min_values
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, i_max_values, label='Максимум')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.plot(x, gaussmf_1_values, linestyle='dotted', label='(5, 1.5)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік імовірнісної інтерпретації логічного оператора OR')
plt.show()
```

Результат:

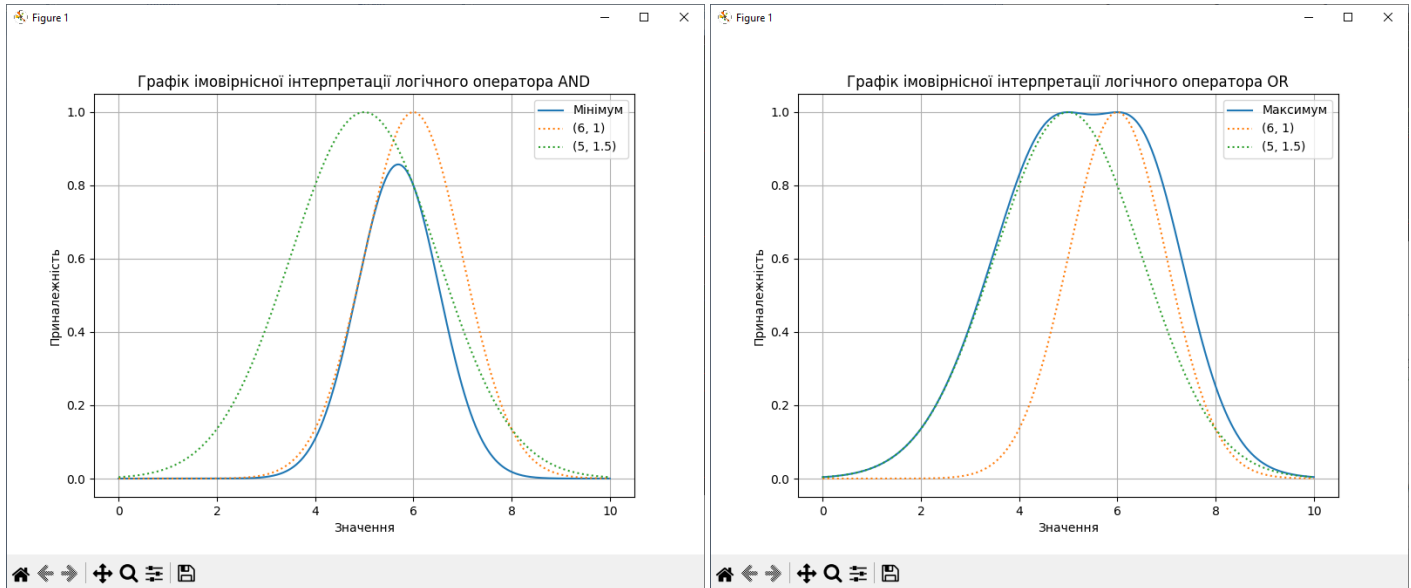


Рисунок 9 – Графіки імовірнісної інтерпретації кон'юнкції та диз'юнкції

Доповнення нечіткої множини

Лістинг:

```
# Доповнення нечіткої множини, інтерпретація логічного оператора NOT (заперечення)
not_values = 1 - gaussmf_values
# Графік
plt.figure(figsize=(8, 6))
plt.plot(x, not_values, label='Доповнення')
plt.plot(x, gaussmf_values, linestyle='dotted', label='(6, 1)')
plt.xlabel('Значення')
plt.ylabel('Приналежність')
plt.legend()
plt.grid(True)
plt.title('Графік інтерпретації логічного оператора NOT')
plt.show()
```

Результат:

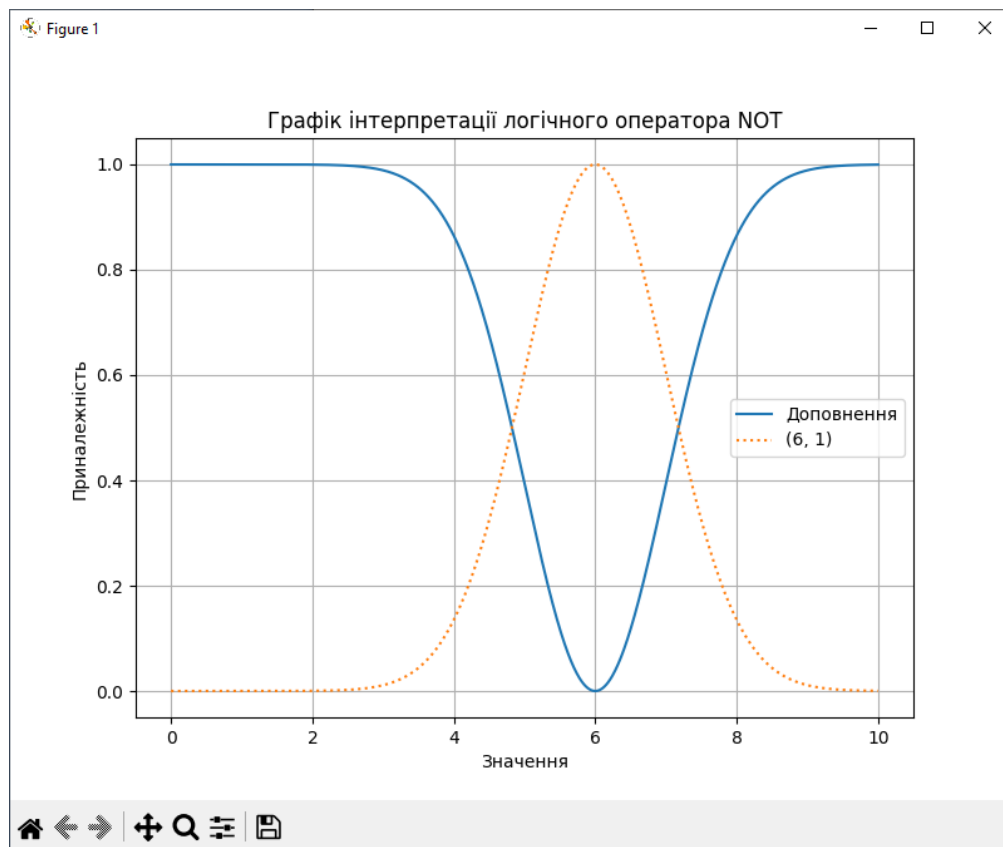


Рисунок 10 – Графік інтерпретації заперечення

Висновок:

У даній лабораторній роботі я ознайомилась з методами побудови нечітких множин за допомогою різних типів функцій приналежності.

Розробила програму на мові програмування python для побудови трикутної, трапецієвидної, Гауссівських, узагальненого дзвону, сигмоїдальних та поліноміальних функцій приналежності.

Відтворила операції кон'юнкції та диз'юнкції в мінімаксній та імовірнісній інтерпретаціях, виконала операцію заперечення для побудованих нечітких множин.