МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРІЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки Кафедра інформаційних систем та технологій

**Звіт з практикуму дисципліни**

**“Методи та технології штучного інтелекту”**

до лабораторної роботи №1

|  |  |
| --- | --- |
| Перевірив:  Шимкович В.М. | Виконав: студент групи IП-01 |

Галько М. В.

Київ-2022

# ЛАБОРАТОРНА РОБОТА №1

## Дослідження способів формування нечітких множин і операцій над ними

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

**Індивідуальні завдання**

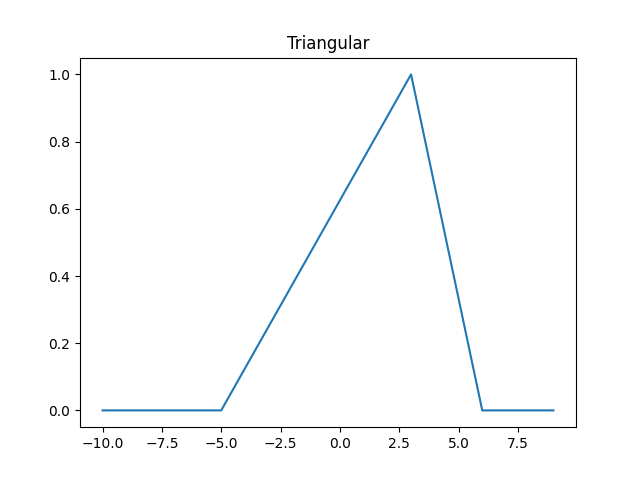
За допомогою пакетів моделювання або мови програмування високого рівня:

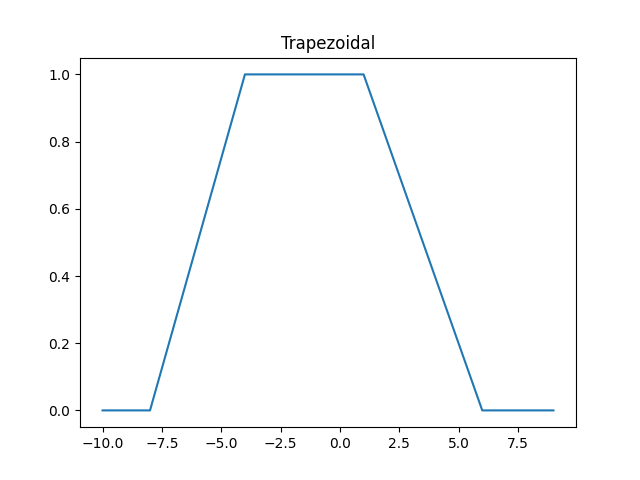
1. Побудувати трикутну і трапецієподібну функцію приналежності.
2. Побудувати просту і двосторонню функцію приналежності Гаусса, утворену за допомогою різних функцій розподілу.
3. Побудувати функцію приналежності "узагальнений дзвін", яка дозволяє представляти нечіткі суб'єктивні переваги.
4. Побудувати набір сігмоїдних функцій: основну односторонню, яка відкрита зліва чи справа; додаткову двосторонню; додаткову несиметричну.
5. Побудувати набір поліноміальних функцій приналежності (Z-, PI- і Sфункцій).
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.

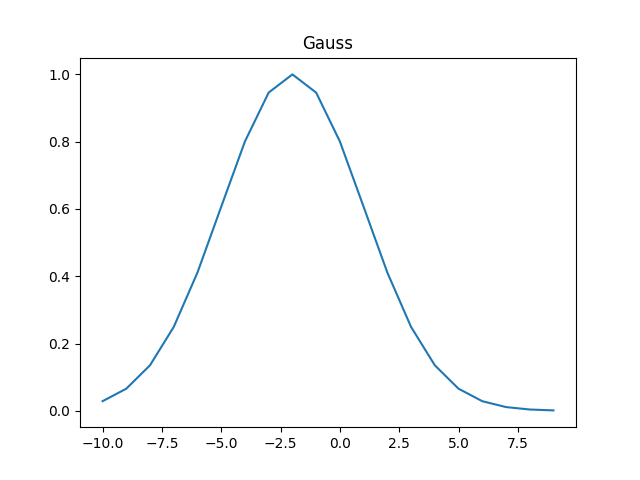
**Реалізація на мові Python:**

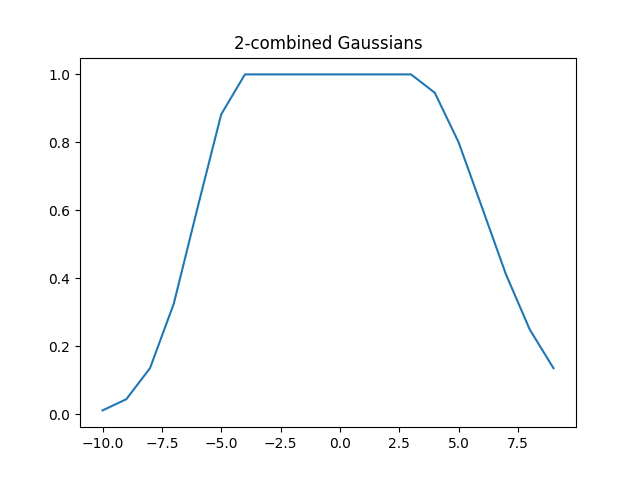
import matplotlib.pyplot as plt  
import skfuzzy as sk  
import numpy as np  
  
x = np.arange(-10, 10)  
  
  
def print\_plot(t, y = np.array([])):  
 if y.size != 0:  
 plt.plot(x, y)  
 plt.title(t)  
 plt.show()  
  
  
# (1) ----------------------------------  
def standard():  
 print\_plot("Triangular", sk.trimf(x, [-5, 3, 6]))  
 print\_plot("Trapezoidal", sk.trapmf(x, [-8, -4, 1, 6]))  
  
  
# (2) ----------------------------------  
def gauss():  
 print\_plot("Gauss", sk.gaussmf(x, -2, 3))  
 print\_plot("2-combined Gaussians", sk.gauss2mf(x, -4, 2, 3, 3))  
  
  
# (3) ----------------------------------  
def bell(): # a\_width b\_slope c\_center : y(x) = 1 / (1 + abs([x - c] / a) \*\* [2 \* b])  
 print\_plot("Generalized Bell", sk.gbellmf(x, 5, 2, 0))  
  
  
# (4) ----------------------------------  
def sigmas(): # y = 1 / (1. + exp[- c \* (x - b)]) c\_width(+-> l->0 && r->1) (if c==0 -> line)  
 print\_plot("Sigmoid unilaterial: [offset 2, slope 3]", sk.sigmf(x, 2, -1))  
 print\_plot("Sigmoids' product", sk.psigmf(x, -3, 1, 3, 1))  
 print\_plot("Sigmoids' difference", sk.dsigmf(x, -4, 2, 5, 0.5))  
  
  
# (5) ----------------------------------  
def polynomial():  
 print\_plot("Z-shaped", sk.zmf(x, -3, 4))  
 print\_plot("S-shaped", sk.smf(x, -7, 7.4))  
 print\_plot("PI-shaped", sk.pimf(x, -9.2, -5, -3, 7))  
  
  
# (6) ----------------------------------  
y1 = sk.gaussmf(x, 5, 3)  
y2 = sk.gaussmf(x, 7, 3)  
z1, z2 = sk.fuzzy\_and(x, y1, x, y2)  
z3, z4 = sk.fuzzy\_or(x, y1, x, y2)  
  
  
def min\_max():  
 plt.plot(z1, z2, color='red')  
 plt.plot(z3, z4, color='black', linestyle='--')  
 print\_plot("Min")  
  
 plt.plot(z3, z4, color='red')  
 plt.plot(z1, z2, color='black', linestyle='--')  
 print\_plot("Max")  
  
  
# (7) ----------------------------------  
def set\_con\_and\_dis(y, t):  
 plt.plot(x, y, color='red')  
 plt.plot(x, y1, linestyle='--', color='black')  
 plt.plot(x, y2, linestyle='--', color='grey')  
 print\_plot(t)  
  
  
def con\_and\_dis():  
 set\_con\_and\_dis(y1 \* y2, "Min interpretation")  
 set\_con\_and\_dis(y1 + y2 - (y1 \* y2), "Max interpretation")  
  
  
# (8) ----------------------------------  
def negation():  
 y = sk.fuzzy\_not(y2)  
 plt.plot(x, y, color='red')  
 plt.plot(x, y2, color='black')  
 print\_plot("Not")

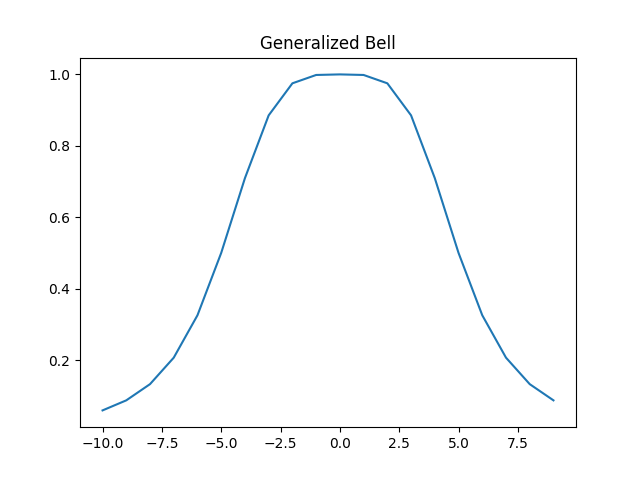
**Результати виконання функцій:**

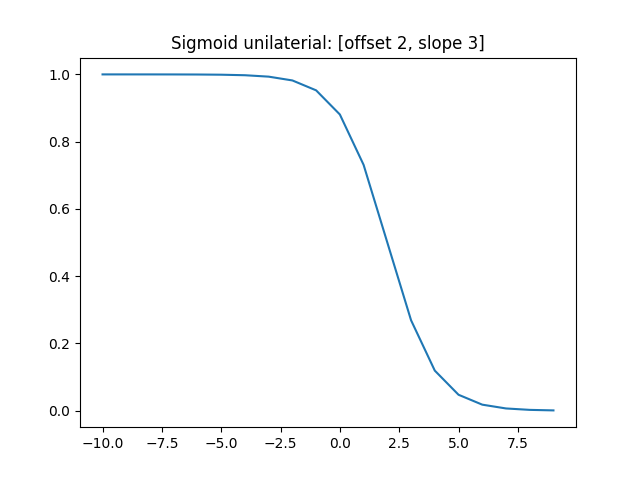


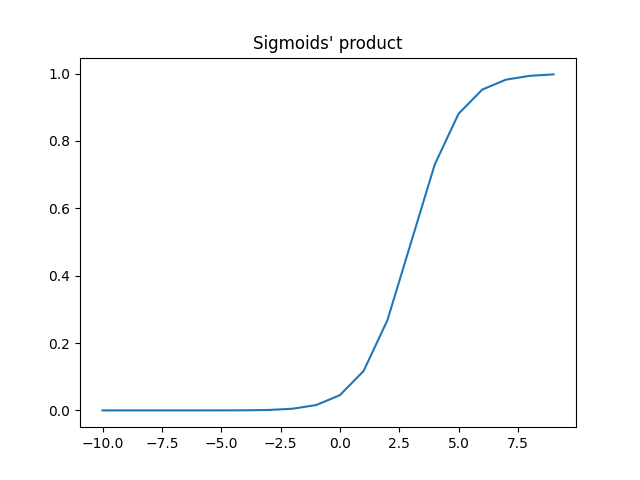


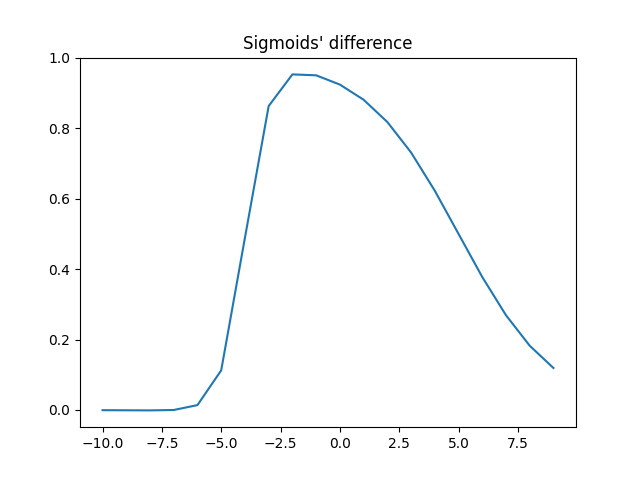


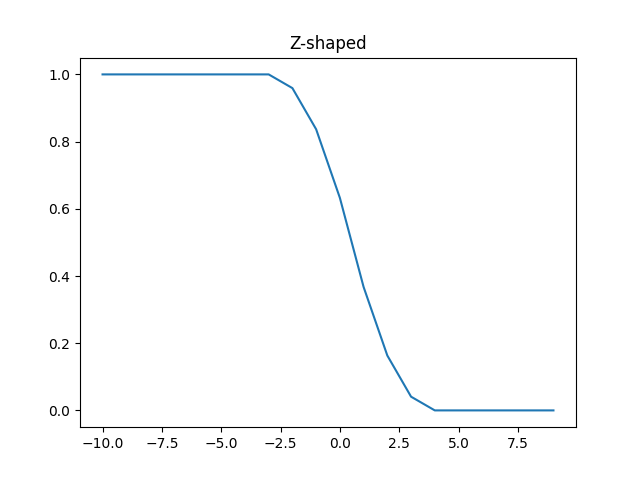


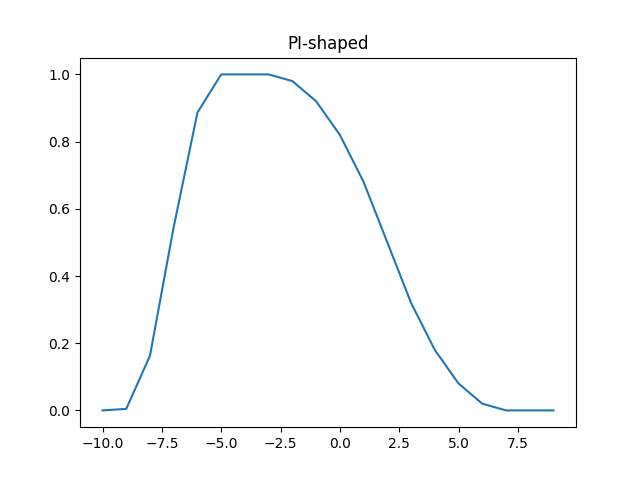
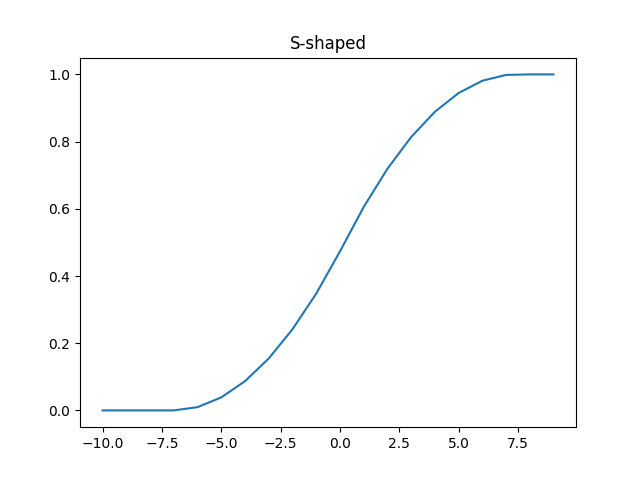


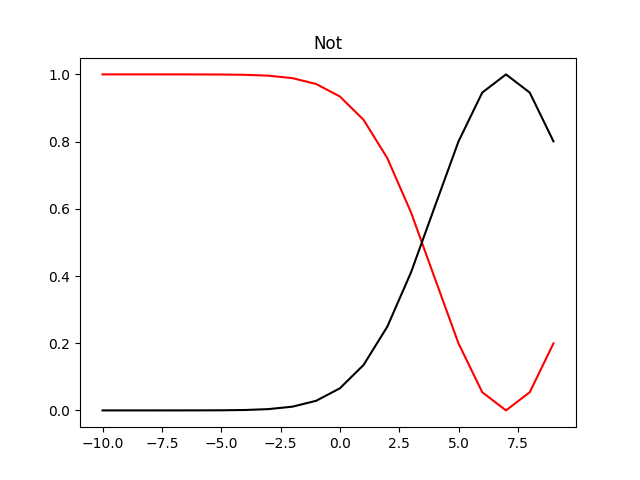
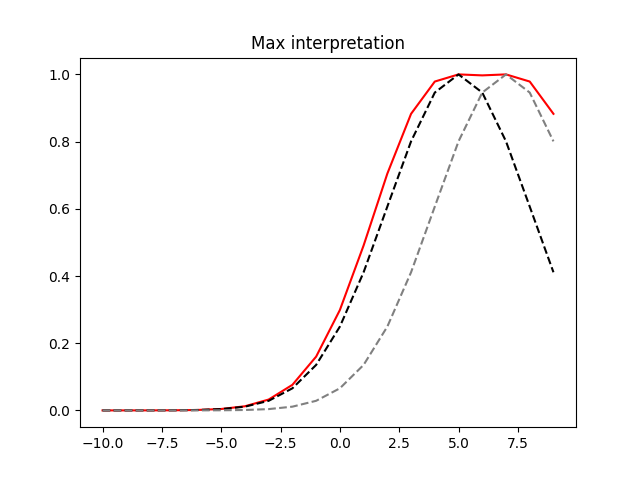
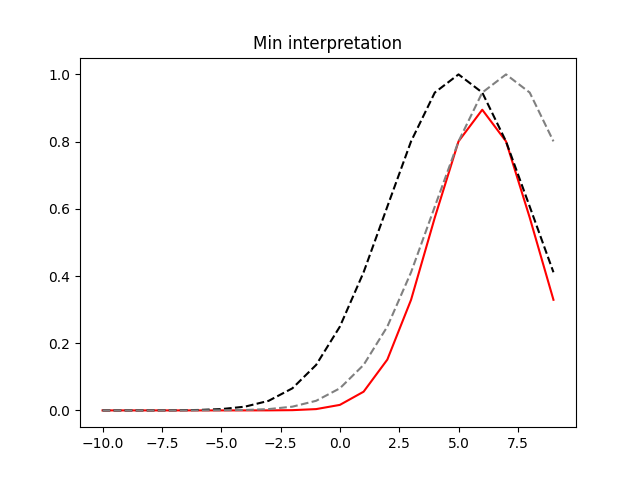
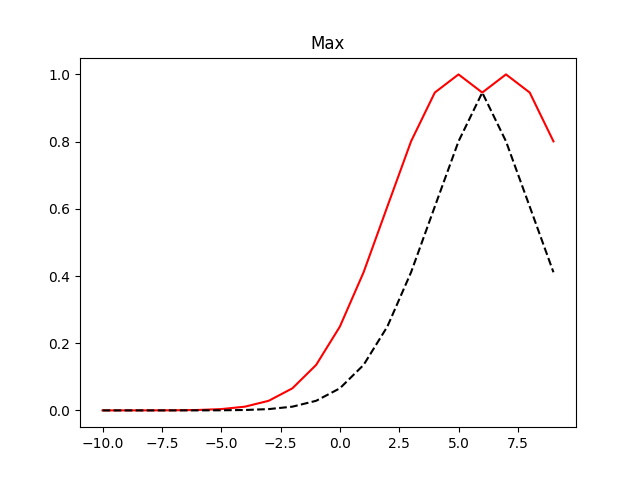
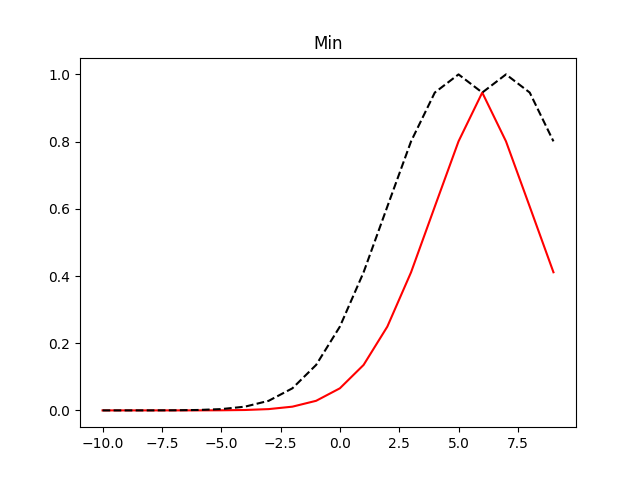












**Висновок**

В ході виконання лабораторної роботи були досліджені різні функції приналежності для побудови нечітких множин. В результаті чого, була створена реалізація на мові Python з використанням можливостей бібліотеки skfuzzy, у якості алгоритмів нечіткої логіки, та matplotlib.pyplot для відображення результатів у вигляді графіків.

Також крім побудови самих множин були реалізовані найпоширеніші операції над ними.