Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Системы искусственного интеллекта»

**Лабораторная работа №2**

«Нечетная логика»

Выполнил

студент группы ИВТАПбд-41

Каршибоев Я.Ш.

Проверил:

преподаватель кафедры «ВТ»

Хайруллин И.Д.

Ульяновск

2025

# Постановка задачи

На языке Python разработайте скрипт, позволяющий выполнить операцию объединения заданных пользователем нечетких множеств с трапециевидными функциями принадлежности. Входными данными будут параметры функций принадлежности и четкие объекты для каждого из множеств. Выходными – объединение данных нечетких множеств.

# Ход работы

Вариант содержит предметную область:

Энергопотребление

• Уровень потребления: низкий, средний, высокий, критический

• Энергоэффективность: низкая, умеренная, высокая

1) Определение нечётких множеств

Основные проблемы, решаемые в нечеткой логике, связаны с моделированием интеллектуальных операций приближенных рассуждений человека (эксперта), а также объектов, над которыми эти операции выполняются:

• Объекты – лингвистические переменные. Таким образом, операндами и результатом интеллектуальных операций являются значения особого вида – нечеткие множества.

• Основные интеллектуальные операции строятся с помощью операций нечеткой логики.

• Алгоритмы вычисления нечетких значений предназначены для манипулирования со значениями, представленными нечеткими множествами на основе операций нечеткой логики, поэтому они классифицируются как нечеткие системы логического вывода. Часто используют сокращенную форму обозначенного класса моделей – нечеткие модели или нечеткие системы.

Простыми словами, мы переводим точные значения в общее множество, которое воспринимает человек.

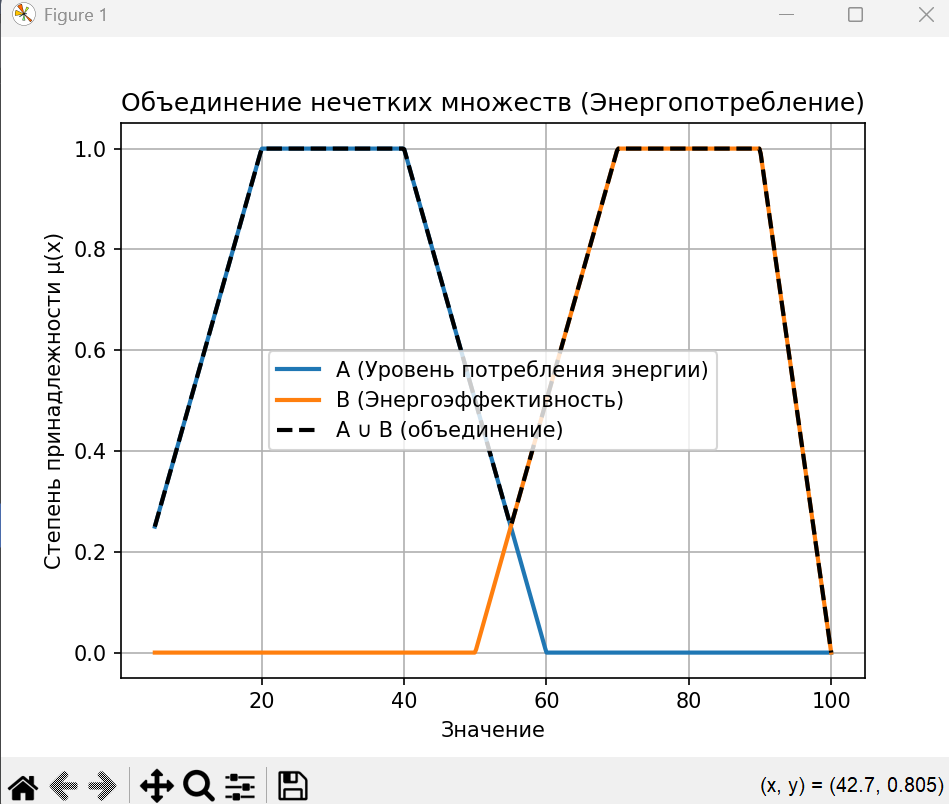


Рис.1 - Объединение трапециевидных нечетких множеств

2) Определение функции принадлежности

Нечеткое множество моделируется параметрической функцией особого класса, называемого классом функций принадлежности.

Трапециевидная функция принадлежности имеет вид:

μ(x) = 0, если x ≤ a или x ≥ d

μ(x) = (x-a)/(b-a), если a < x < b

μ(x) = 1, если b ≤ x ≤ c

μ(x) = (d-x)/(d-c), если c < x < d

Операция объединения: μA∪B(x) = max(μA(x), μB(x))

3) Основной цикл

Основной цикл программы сперва принимает от пользователя параметры трапециевидных функций для двух множеств A и B, затем четкие объекты x.

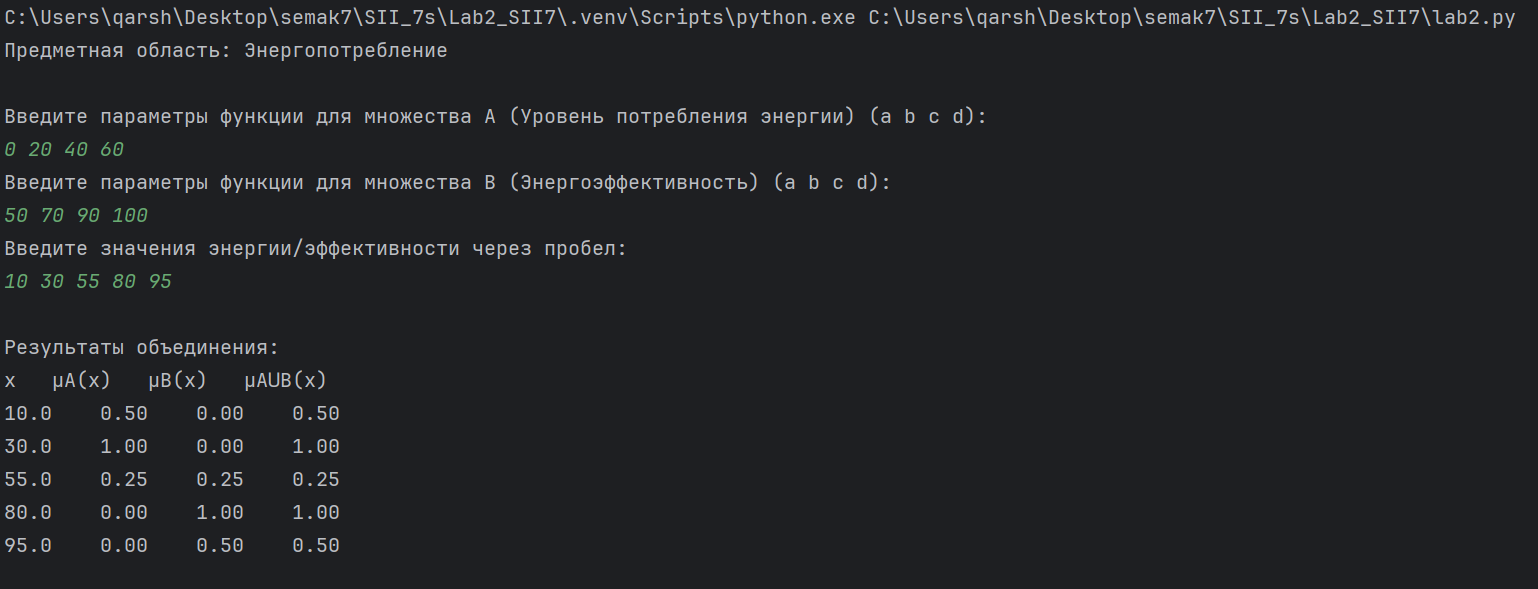


Рис.2 - Объединение трапециевидных множеств

# Вывод

В процессе выполнения лабораторной работы был разработан скрипт на языке Python, который позволяет выполнить операцию объединения заданных пользователем нечетких множеств с трапециевидными функциями принадлежности.

# Листинг

|  |
| --- |
| import numpy as np import matplotlib.pyplot as plt  # Трапециевидная функция принадлежности  def trapezoidal\_mf(x, a, b, c, d):  if x <= a or x >= d:  return 0.0  elif a < x < b:  return (x - a) / (b - a)  elif b <= x <= c:  return 1.0  elif c < x < d:  return (d - x) / (d - c)  return 0.0  # Ввод параметров множества def input\_trapezoid(name):  print(f"Введите параметры функции для множества {name} (a b c d):")  a, b, c, d = map(float, input().split())  return (a, b, c, d)  # Основная программа if \_\_name\_\_ == "\_\_main\_\_":  print("Предметная область: Энергопотребление\n")   # Параметры множеств  A\_params = input\_trapezoid("A (Уровень потребления энергии)")  B\_params = input\_trapezoid("B (Энергоэффективность)")   # Четкие объекты  print("Введите значения энергии/эффективности через пробел:")  x\_values = list(map(float, input().split()))   print("\nРезультаты объединения:")  print("x\tμA(x)\tμB(x)\tμA∪B(x)")  for x in x\_values:  muA = trapezoidal\_mf(x, \*A\_params)  muB = trapezoidal\_mf(x, \*B\_params)  muUnion = max(muA, muB)  print(f"{x:.1f}\t{muA:.2f}\t{muB:.2f}\t{muUnion:.2f}")   # Построение графиков  X = np.linspace(min(x\_values) - 5, max(x\_values) + 5, 500)  muA = [trapezoidal\_mf(x, \*A\_params) for x in X]  muB = [trapezoidal\_mf(x, \*B\_params) for x in X]  muUnion = [max(a, b) for a, b in zip(muA, muB)]   plt.plot(X, muA, label="A (Уровень потребления энергии)", linewidth=2)  plt.plot(X, muB, label="B (Энергоэффективность)", linewidth=2)  plt.plot(X, muUnion, label="A ∪ B (объединение)", linestyle="--", color="black", linewidth=2)   plt.title("Объединение нечетких множеств (Энергопотребление)")  plt.xlabel("Значение")  plt.ylabel("Степень принадлежности μ(x)")  plt.legend()  plt.grid(True)  plt.show() |