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**

**Нечеткая логика**

**Вариант №10**

Выполнил

Студент группы ИВТАСбд-41

Ведин В.А.

Проверил:

Старший преподаватель кафедры «ВТ»

Хайрулин И.Д.

Ульяновск

2025

# Задание

# На языке Python разработайте скрипт, позволяющий выполнить операцию импликации на заданных пользователем нечетких множеств с трапециевидными функциями принадлежности. Входными данными будут параметры функций принадлежности и четкие объекты для каждого из множеств. Выходными – результат импликации данных нечетких множеств. Причем, результат вывести через лингвистические переменные. Импликацию моделировать минимумом.

# Предметная область: Экология

# Уровень загрязнения воздуха: чисто, умеренное загрязнение, загрязнено, сильно загрязнено

# Уровень шума: тихо, средне, шумно, очень шумно

# Ход работы

# Задание параметров задачи

На данном этапе были заданы исходные параметры задачи и определены нечёткие множества для трёх лингвистических переменных: уровня загрязнения воздуха, уровня шума и экологического риска. Все переменные измеряются в диапазоне от 0 до 100, что позволяет удобно моделировать степень проявления каждого показателя в процентах.

Для описания принадлежности значений к определённым лингвистическим термам использовались трапециевидные функции принадлежности. Такой выбор обусловлен тем, что трапециевидная форма позволяет наиболее наглядно и плавно задавать переходы между уровнями переменной, что соответствует реальной нечёткости границ экологических состояний.

Для уровня загрязнения воздуха были определены следующие термы: чисто — [0, 0, 15, 35], умеренное загрязнение — [25, 40, 55, 70], загрязнено — [60, 75, 85, 95], сильно загрязнено — [85, 90, 100, 100]. Аналогичные интервалы были использованы для уровня шума, где термам соответствуют значения: тихо, средне, шумно и очень шумно. Для оценки экологического риска также определены четыре степени: низкий, средний, высокий и очень высокий, которые задаются теми же диапазонами параметров.

На языке Python задание параметров нечётких множеств реализуется с помощью библиотеки scikit-fuzzy, которая позволяет определять функции принадлежности стандартных типов. Ниже, в листинге 1, соответствующий фрагмент

Листинг 1. Параметры задачи

|  |
| --- |
| import numpy as np  import skfuzzy as fuzz  import matplotlib  matplotlib.use('Agg')  import matplotlib.pyplot as plt  pollution = np.arange(0, 101, 1)  noise = np.arange(0, 101, 1)  eco\_risk = np.arange(0, 101, 1)  pollution\_low = fuzz.trapmf(pollution, [0, 0, 15, 35])  pollution\_medium = fuzz.trapmf(pollution, [25, 40, 55, 70])  pollution\_high = fuzz.trapmf(pollution, [60, 75, 85, 95])  pollution\_very\_high = fuzz.trapmf(pollution, [85, 90, 100, 100])  noise\_low = fuzz.trapmf(pollution, [0, 0, 15, 35])  noise\_medium = fuzz.trapmf(pollution, [25, 40, 55, 70])  noise\_high = fuzz.trapmf(pollution, [60, 75, 85, 95])  noise\_very\_high = fuzz.trapmf(pollution, [85, 90, 100, 100])  eco\_low = fuzz.trapmf(pollution, [0, 0, 15, 35])  eco\_medium = fuzz.trapmf(pollution, [25, 40, 55, 70])  eco\_high = fuzz.trapmf(pollution, [60, 75, 85, 95])  eco\_very\_high = fuzz.trapmf(pollution, [85, 90, 100, 100]) |

# Функция импликации

# На следующем этапе была реализована функция импликации, которая моделирует взаимосвязь между уровнем загрязнения воздуха и уровнем шума в контексте экологического риска. В данном случае операция импликации выполняется с использованием метода минимума, что соответствует логическому правилу нечёткой импликации: степень истинности высказывания «если A, то B» определяется как минимум между значениями функций принадлежности A и B.

# В функции implication задаются два входных параметра — конкретные значения загрязнения (pollution\_val) и шума (noise\_val). Сначала для каждого из них вычисляются степени принадлежности к соответствующим лингвистическим термам: низкий, средний, высокий и очень высокий. Для этого используется функция fuzz.interp\_membership, которая возвращает числовое значение принадлежности выбранного чёткого значения к каждой функции.

# Далее создаётся набор правил нечёткой импликации, связывающих уровни загрязнения и шума с итоговой оценкой экологического риска. Например, если загрязнение низкое, а уровень шума низкий — экологический риск также низкий; если загрязнение среднее, а шум высокий — риск повышается; если же оба показателя очень высокие, то экологический риск достигает максимального значения.

# Каждое правило оценивается по принципу минимума (np.fmin), как было сказано выше, что позволяет определить степень активации соответствующего нечёткого множества результата. После вычисления всех возможных комбинаций выполняется агрегирование результатов с помощью функции np.fmax.reduce, которая объединяет все активированные множества в одно результирующее.

# Функция импликации представлена ниже в листинге 2.

# Листинг 2. Функция имликации

|  |
| --- |
| def implication(pollution\_val: int, noise\_val: int) -> np.ndarray:mu\_pollution = {"low": fuzz.interp\_membership(pollution, pollution\_low, pollution\_val),"medium": fuzz.interp\_membership(pollution, pollution\_medium, pollution\_val),"high": fuzz.interp\_membership(pollution, pollution\_high, pollution\_val),"very\_high": fuzz.interp\_membership(pollution, pollution\_very\_high, pollution\_val),}mu\_noise = {"low": fuzz.interp\_membership(noise, noise\_low, noise\_val),"medium": fuzz.interp\_membership(noise, noise\_medium, noise\_val),"high": fuzz.interp\_membership(noise, noise\_high, noise\_val),"very\_high": fuzz.interp\_membership(noise, noise\_very\_high, noise\_val),}eco\_activations = []rules = {('low', 'low'): 'low',('low', 'medium'): 'medium',('low', 'high'): 'medium',('low', 'very\_high'): 'high',('medium', 'low'): 'medium',('medium', 'medium'): 'medium',('medium', 'high'): 'high',('medium', 'very\_high'): 'high',('high', 'low'): 'medium',('high', 'medium'): 'high',('high', 'high'): 'high',('high', 'very\_high'): 'very\_high',('very\_high', 'low'): 'high',('very\_high', 'medium'): 'high',('very\_high', 'high'): 'very\_high',('very\_high', 'very\_high'): 'very\_high',}for (p\_lvl, n\_lvl), eco\_lvl in rules.items():mu\_rule = np.fmin(mu\_pollution[p\_lvl], mu\_noise[n\_lvl])if eco\_lvl == 'low':eco\_activations.append(np.fmin(mu\_rule, eco\_low))elif eco\_lvl == 'medium':eco\_activations.append(np.fmin(mu\_rule, eco\_medium))elif eco\_lvl == 'high':eco\_activations.append(np.fmin(mu\_rule, eco\_high))elif eco\_lvl == 'very\_high':eco\_activations.append(np.fmin(mu\_rule, eco\_very\_high))aggregated = np.fmax.reduce(eco\_activations)return aggregated |

# Определение лингвистической метки

# Для удобства интерпретации результатов и вывода итогового уровня экологического риска был реализован механизм присвоения лингвистической метки конкретному числовому значению.

# Функция fuzzy\_label принимает три параметра: числовое значение, массив значений переменной и словарь с функциями принадлежности, соответствующими различным лингвистическим термам. Для каждого терма вычисляется степень принадлежности с помощью функции fuzz.interp\_membership. После этого выбирается та метка, у которой значение принадлежности максимально, и именно она возвращается в качестве итоговой лингвистической оценки.

# Код упомянутой функции представлен в листинге 3.

# Листинг 3. Определение лингвистической метки

|  |
| --- |
| def fuzzy\_label(value: float,x: np.ndarray,mfs: dict[str, np.ndarray]) -> str:memberships = {name: fuzz.interp\_membership(x, mf, value) for name, mf in mfs.items()}return max(memberships, key=memberships.get) |

# Интерпретация результатов и визуализация

# Заключительный этап работы модели представляет собой полный цикл взаимодействия с пользователем и интерпретацию полученных результатов. В функции main реализован процесс ввода данных, вычисления экологического риска с помощью функции импликации, присвоения лингвистических меток и визуализации результата.

# Сначала пользователь вводит конкретные значения уровня загрязнения воздуха и уровня шума. Ввод проверяется на корректность — значения должны находиться в диапазоне от 0 до 100. Если ввод некорректен, пользователю выводится сообщение об ошибке, и предлагается повторить ввод.

# Далее для введённых значений вызывается функция implication, которая на основе заданных правил и трапециевидных функций принадлежности формирует агрегированное множество экологического риска. Одновременно с этим с помощью функции fuzzy\_label определяется лингвистическая оценка введённых уровней загрязнения и шума, что позволяет представить числовые значения в удобной форме для восприятия.

# Для итоговой оценки экологического риска проводится дефаззификация агрегированного множества методом центроида (centroid), после чего также присваивается соответствующая лингвистическая метка. В результате пользователю выводятся значения: уровень загрязнения, уровень шума и экологическая опасность, как в числовой, так и в лингвистической форме.

# Дополнительно реализована визуализация результата с помощью графика, на котором показаны функции принадлежности всех уровней экологического риска и область, соответствующая вычисленной импликации.

# Функция, упомянутая выше, представлена в листинге 4.

# Листинг 4. Интерпретация результатов и визуализация

|  |
| --- |
| def main() -> None:while True:pollution\_value = float(input("Введите уровень загрязнения воздуха (0-100): "))if pollution\_value < 0 or pollution\_value > 100:print("Введены неккоректные значения уровня загрязнения. Попробуйте ещё раз.")continuenoise\_value = float(input("Введите уровень шума (0-100): "))if noise\_value < 0 or noise\_value > 100:print("Введены неккоректные значения уровня шума. Попробуйте ещё раз.")continuebreakresult = implication(pollution\_value, noise\_value)pollution\_label = fuzzy\_label(pollution\_value, pollution, {"Чисто": pollution\_low,"Умеренное загрязнение": pollution\_medium,"Загрязнено": pollution\_high,"Сильно загрязнено": pollution\_very\_high})noise\_label = fuzzy\_label(noise\_value, noise, {"Тихо": noise\_low,"Средне": noise\_medium,"Шумно": noise\_high,"Очень шумно": noise\_very\_high})eco\_value = fuzz.defuzz(eco\_risk, result, 'centroid')eco\_label = fuzzy\_label(eco\_value, eco\_risk, {"Низкая": eco\_low,"Средняя": eco\_medium,"Высокая": eco\_high,"Очень высокая": eco\_very\_high,})print(f"\nЗагрязнение ({pollution\_value:.1f}): {pollution\_label}")print(f"Шум ({noise\_value:.1f}): {noise\_label}")print(f"Экологическая опасность ({eco\_value:.1f}): {eco\_label}")plt.figure(figsize=(8, 4))plt.plot(eco\_risk, eco\_low, 'b', linestyle='--', label='eco\_low')plt.plot(eco\_risk, eco\_medium, 'g', linestyle='--', label='eco\_medium')plt.plot(eco\_risk, eco\_high, 'y', linestyle='--', label='eco\_high')plt.plot(eco\_risk, eco\_very\_high, 'r', linestyle='--', label='eco\_very\_high')plt.fill\_between(eco\_risk, 0, result, color='orange', alpha=0.6, label='Импликация')plt.title(f'Результат импликации: экологическая опаность\n(Загрязнение {pollution\_value}, Шум: {noise\_value})')plt.xlabel('Экологическая опасность')plt.ylabel('Степень принадлежности')plt.legend()plt.savefig(f"eco\_result.png")if \_\_name\_\_ == '\_\_main\_\_':main() |

# Тестирование

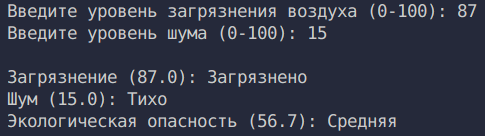


Рис. 1. Вывод результатов

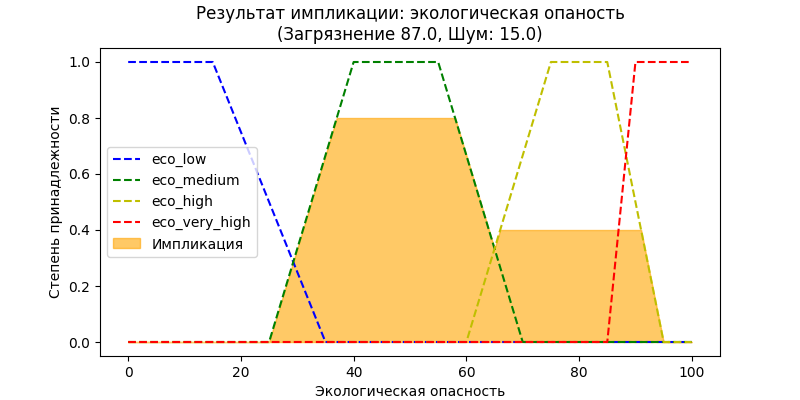


Рис. 2. График, отображающий результаты импликации

# Вывод

В лабораторной работе была создана модель нечёткого вывода для оценки экологического риска на основе уровней загрязнения воздуха и шума. Реализованы трапециевидные функции принадлежности, функция импликации с минимумом и присвоение лингвистических меток.

# Приложение.

# Ссылка на репозиторий с кодом лабораторной работы: <https://github.com/Tsaranchik/Artifical_Intelligence_Systems/tree/master/lab2>