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»**

Выполнил:

студент группы ИВТАСбд-41

Шайдуллин А. И.

Проверил работу:

Хайруллин И. Д.

Ульяновск 2025

**Общее задание**

**Вариант 6**

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.

**Предметная область:**

8. Управление освещением

Освещенность помещения: темно, приглушенно, ярко, слишком ярко

Время суток: утро, день, вечер, ночь

**Описание предметной области и постановка задачи**

Для практической реализации была выбрана задача автоматического управления мощностью комнатного освещения. Система должна анализировать текущую естественную освещенность и время суток, чтобы поддерживать комфортный уровень света и экономить электроэнергию.

**Шаг 1: Определение лингвистических переменных**

Были определены две входные (Antecedent) и одна выходная (Consequent) переменные с их универсумами рассуждений:

* **Входная переменная Освещенность (illumination)**:
  + **Диапазон:** от 0 до 1000 люкс.
  + **Терм-множества:** темно, приглушенно, ярко, слишком ярко.
* **Входная переменная Время суток (time\_of\_day)**:
  + **Диапазон:** от 0 до 24 часов.
  + **Терм-множества:** ночь, утро, день, вечер.
* **Выходная переменная Мощность лампы (lamp\_power)**:
  + **Диапазон:** от 0% до 100%.
  + **Терм-множества:** выключить, слабая, сильная.

**Шаг 2: Задание трапециевидных функций принадлежности**

Для каждого терма были заданы параметры трапециевидных функций принадлежности. Параметры подбирались эмпирически для создания логичной и плавной системы.

# Функции принадлежности для Освещенности  
illumination['темно'] = fuzz.trapmf(illumination.universe, [0, 0, 50, 150])  
illumination['приглушенно'] = fuzz.trapmf(illumination.universe, [100, 200, 300, 400])  
illumination['ярко'] = fuzz.trapmf(illumination.universe, [350, 500, 700, 800])  
illumination['слишком ярко'] = fuzz.trapmf(illumination.universe, [750, 850, 1000, 1000])  
  
# Функции принадлежности для Времени суток  
time\_of\_day['ночь'] = fuzz.trapmf(time\_of\_day.universe, [0, 0, 4, 6])  
time\_of\_day['утро'] = fuzz.trapmf(time\_of\_day.universe, [5, 7, 9, 11])  
time\_of\_day['день'] = fuzz.trapmf(time\_of\_day.universe, [10, 12, 18, 20])  
time\_of\_day['вечер'] = fuzz.trapmf(time\_of\_day.universe, [19, 21, 23, 24])  
  
# Функции принадлежности для Мощности лампы  
lamp\_power['выключить'] = fuzz.trapmf(lamp\_power.universe, [0, 0, 10, 25])  
lamp\_power['слабая'] = fuzz.trapmf(lamp\_power.universe, [15, 30, 50, 65])  
lamp\_power['сильная'] = fuzz.trapmf(lamp\_power.universe, [60, 80, 100, 100])

Перекрытие диапазонов функций принадлежности (например, между "темно" и "приглушенно") является намеренным и необходимым для обеспечения плавных переходов между состояниями.

**Шаг 3: Формулировка базы правил с использованием операции объединения**

Центральной частью работы была реализация правил на основе операции **объединения («ИЛИ»)**. Были созданы три правила, покрывающие основные сценарии работы системы.

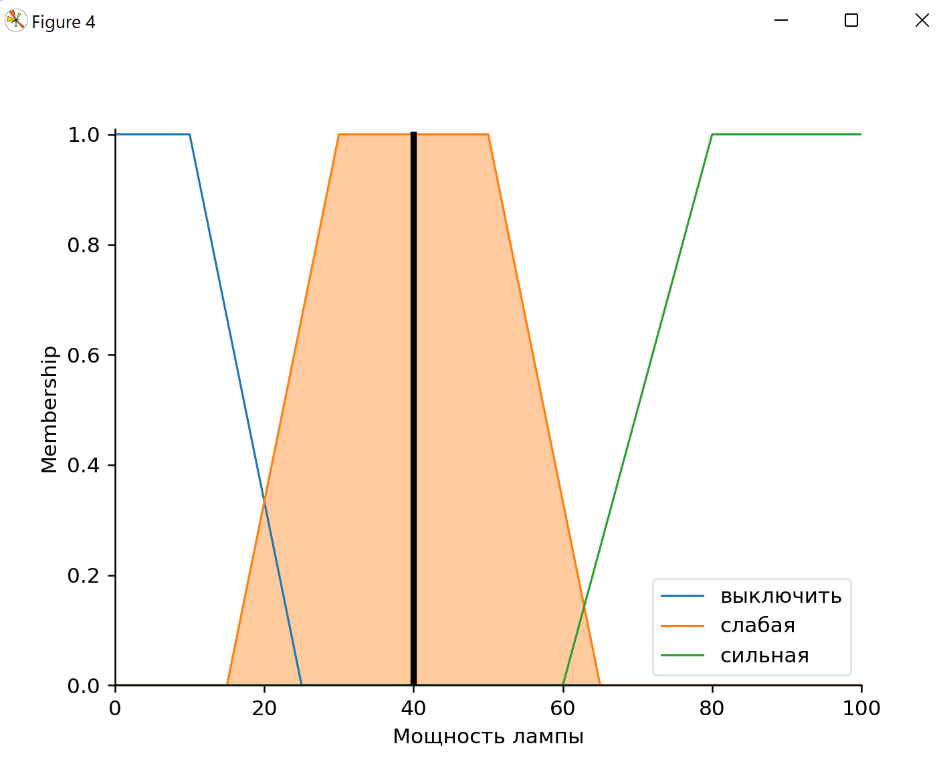
1. **Правило 1 (для сильного света):** ЕСЛИ Время суток «ночь» ИЛИ Освещенность «темно», ТО Мощность лампы «сильная».  
   Это правило активирует максимальную мощность, если наступила ночь, либо если в помещении стало очень темно по любой другой причине (например, из-за грозы днем).
2. **Правило 2 (для слабого света):** ЕСЛИ Время суток «утро» ИЛИ «вечер» ИЛИ Освещенность «приглушенно», ТО Мощность лампы «слабая».  
   Это правило обеспечивает комфортный, неяркий свет в переходные периоды суток или при недостаточной естественной освещенности.
3. **Правило 3 (для выключения света):** ЕСЛИ Время суток «день» ИЛИ Освещенность «ярко» ИЛИ «слишком ярко», ТО Мощность лампы «выключить».  
   Это правило отвечает за энергосбережение, выключая лампу, когда естественного света достаточно.

В коде scikit-fuzzy операция «ИЛИ» реализуется с помощью оператора |.

**4. Тестирование**

**Тестовый случай:** вечер, приглушенный свет

* **Входные данные:** время суток = 20:00, освещенность = 250 люкс.
* **Полученный результат:** рекомендуемая мощность лампы: 40.00%.



**Анализ:**

1. **Фаззификация:**
   * 20:00 имеет степень принадлежности ~0.5 к терму «вечер» и 0 к остальным.
   * 250 люкс имеет степень принадлежности 1.0 к терму «приглушенно» (так как попадает на плато трапеции [100, 200, 300, 400]) и 0 к остальным.
2. **Оценка правил:**
   * **Правило 1 (сильная):** max(μ\_ночь(20), μ\_темно(250)) = max(0, 0) = 0. Правило неактивно.
   * **Правило 2 (слабая):** max(μ\_утро(20), μ\_вечер(20), μ\_приглушенно(250)) = max(0, 0.5, 1.0) = 1.0. Правило активно с максимальной силой.
   * **Правило 3 (выключить):** max(μ\_день(20), μ\_ярко(250), μ\_слишком\_ярко(250)) = max(0, 0, 0) = 0. Правило неактивно.
3. **Активация и аккумуляция:** активируется только выходной терм слабая с силой 1.0. Итоговая нечеткая фигура полностью совпадает с формой функции принадлежности для слабая.
4. **Дефаззификация:** метод центра тяжести для трапеции [15, 30, 50, 65] дает четкое значение **40.0**.

**5. Заключение**

В ходе выполнения данной лабораторной работы была успешно решена задача разработки системы нечеткого управления освещением. Я на практике освоил основные этапы построения подобных систем: от определения лингвистических переменных и функций принадлежности до формулировки базы правил и анализа результатов.