Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления Кафедра Интеллектуальных информационных технологий

Отчёт по лабораторной работе №1

по дисциплине

«Логические основы интеллектуальных систем»

Выполнили студенты гр. 121702 И.А. Готин

Проверил: В. П. Ивашенко

Минск 2023

**Тема**: Представление и обработка информации в условиях наличия не-факторов в рамках логических моделей

**Цель**: Приобрести навыки программирования алгоритмов обработки структур и формул в нечёткой логике

**Состав разработчиков программной системы:** Заломов Р.А., Готин И.А., Буланович В.И.

**Задача**: Реализовать прямой нечеткий логический вывод, используя импликацию Гогена.

**Описание лабораторной работы**:

Задача заключается в создании программного модуля, который будет выполнять прямой нечеткий логический вывод на основе импликации Гогена. Данная задача предполагает работу с нечеткими правилами и фактами, а также применение операции импликации Гогена для получения нечеткого вывода.

Ход выполнения работы:

1. Изучение нечеткой логики и импликации Гогена:
   * Изучить основы нечеткой логики, включая нечеткие множества, функции принадлежности и операции нечеткого вывода.
   * Ознакомиться с операцией импликации Гогена, которая используется для выполнения нечеткого логического вывода.
2. Разработка программного модуля:
   * Реализовать функции, решающие поставленную задачу.
3. Тестирование и анализ результатов
   * Написать несколько тестовых случаев, чтобы проверить работу программного модуля. В тестах учесть различные комбинации правил и фактов
   * Провести тесты и проанализировать их результаты

**Теоретические сведения**:

*Прямой нечеткий логический вывод* - процесс, при котором из нечетких посылок получают некоторые следствия, возможно, тоже нечеткие.

*Правило (в рамках данной работы)* - импликация, которая выражает зависимость между наблюдаемыми причинами и следствиями.

*Нечёткое множество* – множество, содержащее пары, состоящие из элемента и числа из отрезка [0;1], называемого степенью принадлежности.

*Нечеткий предикат* – нечёткое множество, степени принадлежности элементов которого интерпретируются как значения истинности.

*Импликация Гогена* – импликация, основанная на алгебраическом произведении.

*Импликация* - бинарная логическая связка, по своему применению приближенная к союзам «если. . . , то. . . ».

*Нечеткая импликация нечетких высказываний* - это бинарная логическая операция, результат которой является нечетким высказыванием «из А следует B», «если А, то B».

*Нечеткое высказывание* - это законченная мысль, об истинности или ложности которой можно судить только с некоторой степенью истинности, принимающей значения в отрезке [0;1].

**Описание алгоритма**:

В программе были реализованы такие функции как:

* def \_\_gauguin\_norm\_delta(f\_belonging\_degree: int | float, s\_belonging\_degree: int | float) -> float

- Вычисление степени принадлежности пары, принадлежащей нечёткому отношению, которое является результатом нечёткой импликации Гогена.

* def \_\_fuzzy\_implication(cls, fuzzy\_set\_1: dict, fuzzy\_set\_2: dict) -> pd.DataFrame

- Вычисление результата операции нечёткой импликации над двумя нечёткими предикатами.

* def \_\_fuzzy\_conclusion(cls, fact: dict, implication\_matrix: pd.DataFrame) -> dict | None

- Вычисление результата прямого нечёткого логического вывода.

* def \_\_solve\_implications(cls, parse\_result: dict) -> dict | None

- Вычисление всевозможных нечётких импликаций (на основе данных правил).

* def solve(cls, parse\_result: dict) -> list[NamedFuzzyConclusion] | None

- Метод, собирающий все функции решателя. Принимает результат парсинга файла с описанием правил и фактов.

* def \_\_get\_all\_conclusions(cls, program\_file: str) -> list

- Метод класса, объединяющего решатель и парсер. Получает готовые для вывода на экран результаты прямых нечётких логических выводов.

* def print\_conclusions\_results(cls, program\_file: str)

- Метод класса, объединяющего решатель и парсер. Выводит на экран результаты прямых нечётких логических выводов.

* def fuzzy\_set\_dict\_to\_str(conclusion\_result: dict | None) -> str | None

- Превращение читаемого для системы формата нечёткого предиката в читаемую для пользователя системы строку.

* def \_\_parse\_fuzzy\_set(cls, raw\_line: str) -> dict | None

- Парсинг строки с нечётким предикатом в удобный для системы формат.

* def \_\_parse\_program\_file(cls, file\_dir: str) -> dict | None

- Певрвоначальный парсинг текстового файла, содержащего описание фактов и правил.

* def \_\_parse\_program\_file\_result(cls, raw\_parse: dict) -> dict | None

- Второй этап парсинга текствого файла, содержащего описание фактов и правил. Результат является интерпретируемым системой.

* def \_\_parse\_fuzzy\_implication(cls, raw\_line: str) -> FuzzyImplication

- Парсинг строки, содержащей правило.

* def parse(cls, file\_dir: str = 'program') -> dict | None

- Метод, объединяющий работу всех методов парсинга парсера системы.

# Тесты:

Для проверки работоспособности программы были проведены тесты, которые указаны ниже:

**Тест 1.**

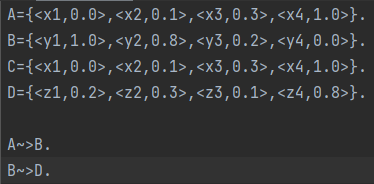
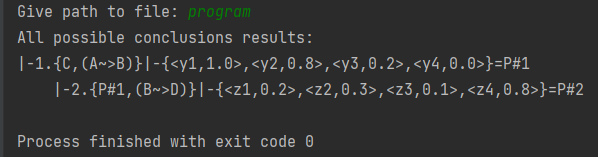
 

Рис. 1. Входные данные теста 1. Рис. 2. Выходные данные теста 1.

**Тест 2.**

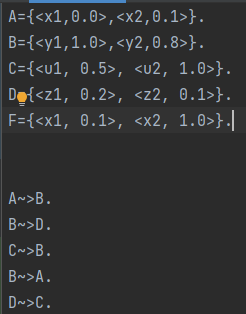
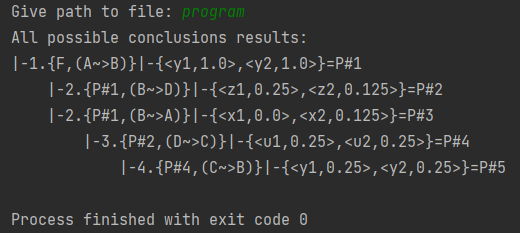
 

Рис. 3. Входные данные теста 2. Рис. 4. Выходные данные теста 2.

**Тест 3.**

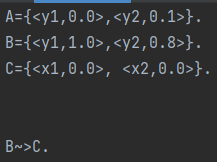
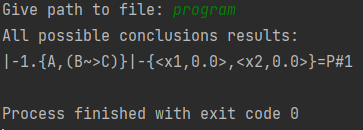
 

Рис. 5. Входные данные теста 3 Рис. 6. Выходные данные теста 3

**Тест 4.**

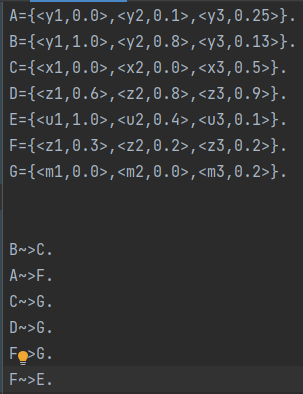
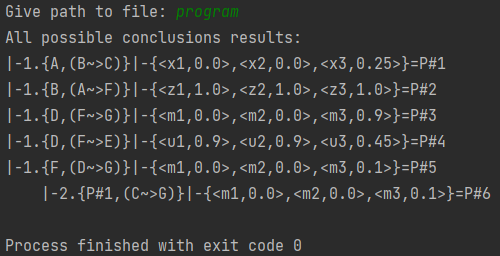
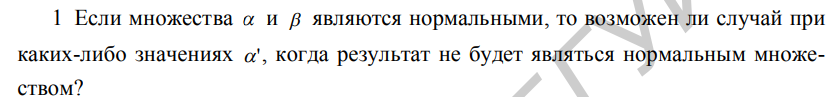
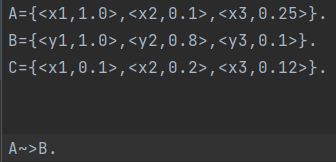
 

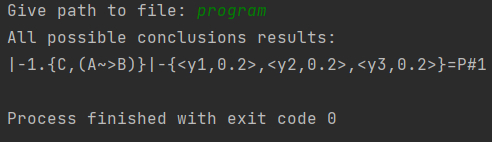
Рис. 7. Входные данные теста 4. Рис. 8. Выходные данные теста 4.

**Контрольные вопросы** (ответы приведены для импликации Гогена, в соответствии с вариантом):

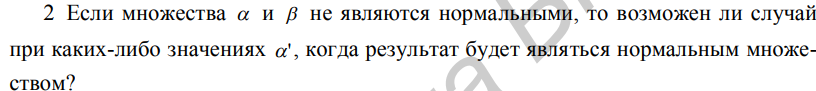


Ответ:

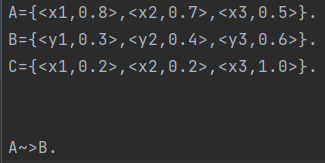
  
Рис 9. Входные данные, иллюстрирующие ответ на контрольный вопрос 1.

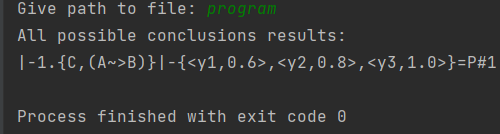
  
Рис 10. Выходные данные, иллюстрирующие ответ на контрольный вопрос 1.

Да. Множества A и B - нормальные, множество C - субнормальное. Учитывая данную треугольную норму, выходит, что если предикат, который используется при выводе (в этом случае C) не является нормальным нечётким множеством, то и результат не будет нормальным нечётким множеством, т.к. при умножении на единицу числа, меньшего единицы, получается число меньшее единицы. Это будет действительным для каждой строки матрицы нечёткого отношения (результат нечёткой импликации A и B), а значит итоговые степени принадлежности в любом случае будут числами меньше 1, а это, в свою очередь, означает, что максимальное число оттуда тоже будет меньше 1, что приведёт к получению субнормального множества как результата нечёткого прямого логического вывода.



Ответ:

  
Рис 11. Входные данные, иллюстрирующие ответ на контрольный вопрос 2.

  
Рис 12. Выходные данные, иллюстрирующие ответ на контрольный вопрос 2.

Да. Множества A и B - субнормальные (все степени принадлежности меньше 1). Результирующее множество может быть нормальным, если для факта, который используется в выводе и для фактов, участвующих в правиле, выполняются следующие условия : существует такой элемент из носителя факта, участвующего в выводе, степень принадлежности которого принимает значение 1 (в множестве C это элемент x3 – рис 11.). В свою очередь, в факте, который является первой импликантой нечёткой импликации (в иллюстрирующем примере это факт A – рис. 11), этот же элемент носителя должен иметь степень принадлежности, меньшую чем степень принадлежности какого-нибудь элемента из второй импликанты (элемент y3 из множества B имеет степень принадлежности 0.6, а элемент x3 из A имеет степень принадлежности 0.5). Результирующее множество получилось нормальным (множество P#1 из рис. 12)



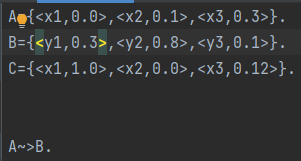


Рис 13. Входные данные, иллюстрирующие ответ на контрольный вопрос 3.

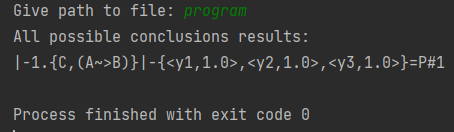


Рис 14. Выходные данные, иллюстрирующие ответ на контрольный вопрос 3.

Некоторые случаи возможности получения требуемого условия:  
1. Входные данные и условия аналогичные ответу на контрольный вопрос 2.  
2. Вторая импликанта нечёткой импликации – нормальное множество, факт, используемый при выводе – нормальное множество.

**Личный вклад**

В рамках реализации данной программной системы данным разработчиком были реализованы следующие этапы разработки:  
- Разработка парсера системы, решателя задач;  
- Подготовка теоретического материала.

# Вывод:

В ходе лабораторной работы были приобретены практические навыки программирования обработки структур и формул нечёткой логики посредством реализации программной системы прямого нечеткого логического вывода с использованием импликации Гогена. Разработанный программный модуль позволяет выполнять нечеткий логический вывод на основе заданных нечетких правил и фактов. В ходе тестирования была проверена работоспособность реализованного программного модуля с субнормальными и нормальными нечеткими множествами. Тестирование показало соответствие результатов ожидаемым значениям вывода.

**Список использованных источников**:

[1] Логические основы интеллектуальных систем. Практикум : учеб.- метод. пособие / В. В. Го- ленков [и др.]. – Минск : БГУИР, 2011. – 70 с. : ил. ISBN 978-985-488-487-5.

# [2] Прямой нечеткий логический вывод [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/111187/> — Дата доступа: 20.10.2023.

# [3] Проведение эксперимента по обоснованию выбора нечёткой импликации, пригодной для решения задач классификации рисков и выработки наилучших рекомендаций по рискам / М.И. Тенетко, О.Ю. Пескова, 2012. – 118 c.

# [4] Нечёткие предикаты [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/2873635/page:16/> — Дата доступа: 20.10.2023.

# [5] Нечёткая логика [Электронный ресурс]. – Режим доступа: <https://inlnk.ru/NDgPPp> — Дата доступа: 20.10.2023.