Министерство образования Республики Беларусь

Учреждение образования “Белорусский государственный университет

информатики и радиоэлектроники” Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

# ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине «Логические основы интеллектуальных систем» на тему

«Прямой нечеткий логический вывод»

# Вариант 3

Выполнил: студент гр. 121702 Шершень К.А.

Проверил: Ивашенко В.П.

Минск 2023

**Цель:** ознакомиться и получить навыки реализации методов прямого нечёткого логического вывода.

**Задача:** реализовать систему прямого нечёткого логического вывода. Исходные данные и результаты должны быть сохранены в файле. Для расчёта меры возможности использовать граничное произведение (импликация Лукасевича). Предусмотреть преодоление зацикливания. Дополнительно: реализовать вычисление мер необходимости

# Состав бригады:

Промчук Даниил Вадимович Шершень Кирилл Андреевич Шакин Иван Валерьевич

# Личный вклад в коллективную работу:

Тестирование программы, оформление отчета, добавление регулярных выражений, определение алгоритма получения коньюнкционной матрицы.

# Описание лабораторной работы:

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

Для реализации был выбран язык программирования Python.

# Ход выполнения работы:

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

# Теоретические сведения:

*Прямой нечеткий логический вывод* – процесс, при котором из нечетких посылок получают некоторые следствия, возможно, тоже нечеткие[1].

*Правило* – утверждение или формулировка, которая определяет, какие выводы можно сделать на основе предпосылок или условий[1].

*Нечеткий предикат* – это нечеткое множество, значения которого интерпретируется как значения истинности[1].

*Импликация Лукасевича* – операция над нечёткими множествами, используемая для определения степени истинности высказывания на основе степени истинности условия и следствия[1].

*Импликация* – бинарная логическая связка, по своему применению приближенная к союзам «*если… , то...* »[1].

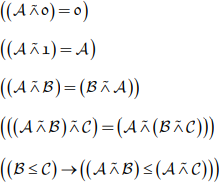
*Нечеткая импликация нечетких высказываний –* бинарная логическая операция, результат которой является нечетким высказыванием

«из А следует B», «если А, то B»[1].

*Нечеткое высказывание* – утверждение, для которого степень истинности не ограничивается только бинарными значениями "истина" или "ложь", а может принимать любое значение в интервале от 0 до 1[1].

*t-норма –* нечеткая логическая операция, удовлетворяющая свойствам нечеткой конъюнкции[1].

*Свойства t-нормы:*

**

# Описание программы и алгоритма:

Программа включает в себя 5 файлов:

1. main.py – файл предназначен для запуска программы и взаимодействия пользователями с основными функциями.
2. main\_functions.py – файл,содержащий основные функции для раоты программы.
3. operations.txt – файл предназначен для получения списка правил, указываемых пользователем.
4. input.txt – файл, из которого происходит чтение исходных данных о предикатах.
5. parcel.txt – файл, из которого происходит чтение и запись полученных в ходе работы программы посылок.

Методы, реализуемые программой, по файлам:–

# main\_functions.py:

* 1. *input\_transformation()* – функция, осуществляющая проверку корректного ввода прочитанной из файла input.txt информации об исходных посылках, возвращает преобразованные с помощью регулярного выражения вида <[a-z]\d+?,\d.\d> данные для последующих функций
  2. *operation\_reader()* – функция, считывающая операции, полученные из файла operations.txt, возвращает список операций
  3. parcel\_transformation(level) – функция, получающая в качестве аргумента число, определяющее уровень считывания посылки. Осуществляет проверку полученных данных из файла parcel.txt и преобразование ихс помощью регулярного выражения <[a-z]\d+?,\d.\d> для последующего использования в программе
  4. get\_spaces\_count() – вспомогательная функция для parcel\_transformation(), помогает отформатировать вывод для корректного считывания
  5. lukasiewicz\_matrix\_formation(input\_information\_dict, array\_of\_operations) – функция, осуществляющая импликацию элементов в матрице, в качестве аргументов получает вывод функций обработки файлов, возвращает импликативную матрицу .Формула для импликации min(1, 1 - external\_element + internal\_element).
  6. t\_norm(input\_information\_dict,input\_values\_dict,matrix\_dict, parcel\_information\_dict,parcel\_values\_dict,used\_rules, parcel\_to\_rule\_list) – функция, получающая подготовленный вывод из предыдущих функций и матрицу импликации, отвечает за применение граничного произведения к матрице импликации и формирование коньюнкционной матрицы, необходимой для получения прямого вывода,также форматируем вывод с помощью регулярного выражения вида [A-Z]\d+|[A-Z]. Т-норма: max(0,parcel\_element + matrix\_element — 1).
  7. strait\_output(new\_matrix\_dict, input\_information\_dict, input\_values\_dict, parcel\_information\_dict, external\_counter, parcel\_to\_rule\_list) – функция, которая получает коньюнкционную матрицу и формирует логический вывод ,также проверяет его на соотвествие выражению вида [A-Z]\d+|[A-Z]

# main.py:

* 1. *main()* – функция, запускающая выполнение всего кода и задающая корректные стартовые значения для всех переменных, определяет основной цикл с условием для корректного выполнения программы

# Тесты:

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

# Тест 1:

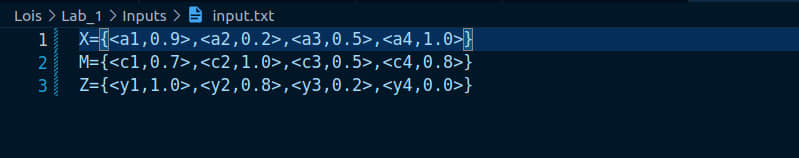


Рис. 1. Входные данные теста 1 (часть 1).

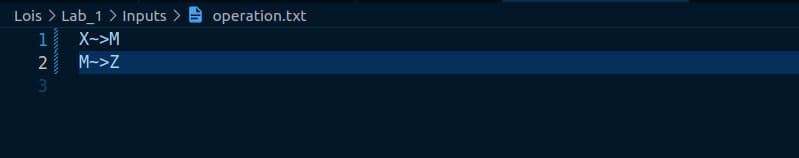


Рис. 2. Входные данные теста 1 (часть 2).

Рис. 3. Входные данные теста 1 (часть 3).

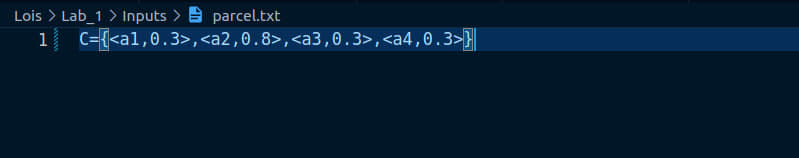
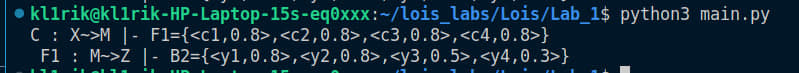


Рис. 4. Выходные данные теста 1 .



# Тест 2:

# 

Рис. 5. Входные данные теста 2 (часть 1).

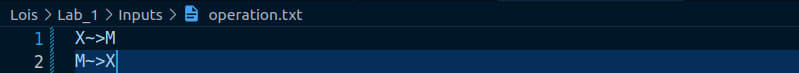


Рис. 6. Входные данные теста 2 (часть 2).

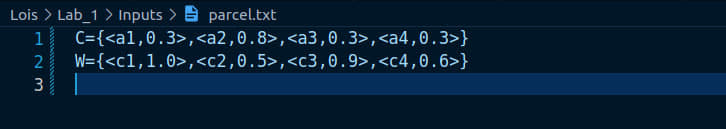
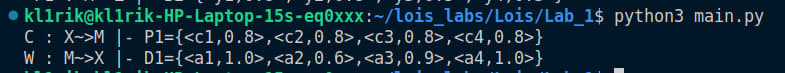


Рис. 7. Выходные данные теста 2 .



# Тест 3:

# 

Рис. 8. Входные данные теста 3 (часть 1).

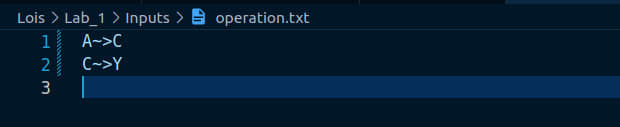


Рис. 9. Входные данные теста 3 (часть 2).

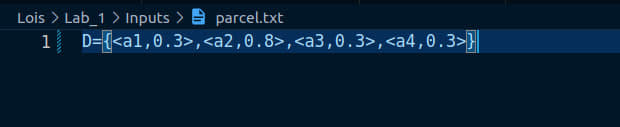
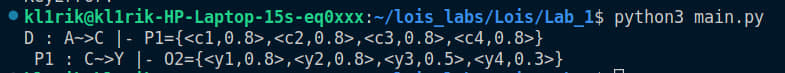


Рис. 10. Выходные данные теста 3 .



# Тест 4:

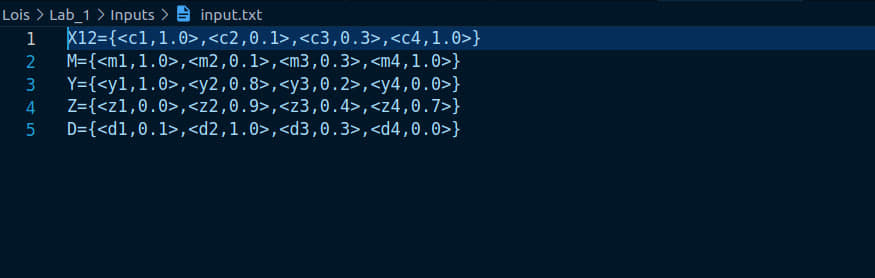


Рис. 11. Входные данные теста 4 (часть 1).

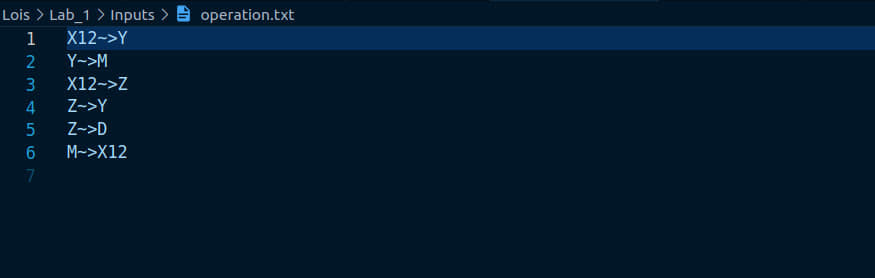


Рис. 12. Входные данные теста 4 (часть 2).

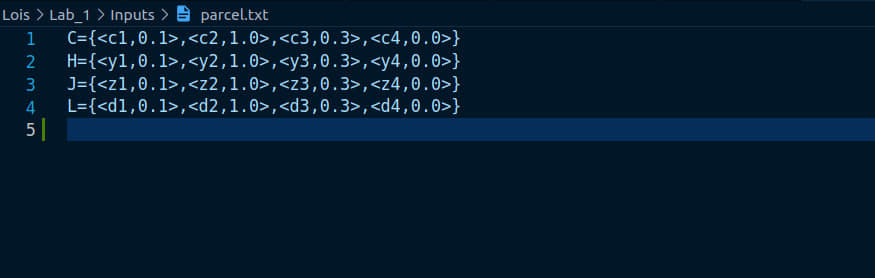
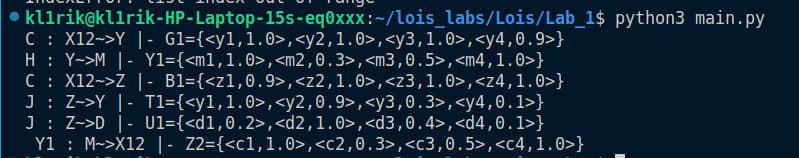
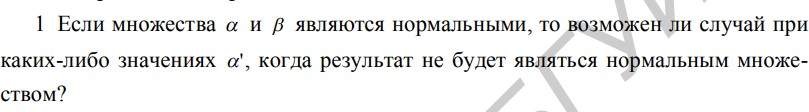


Рис. 13. Выходные данные теста 4 .



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



# Ответ:

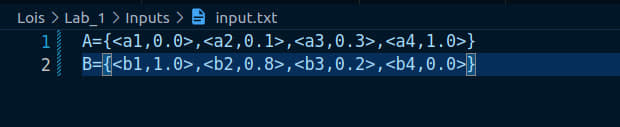


Рис. 14. Входные данные для ответа на контрольный вопрос 1 (часть 1).

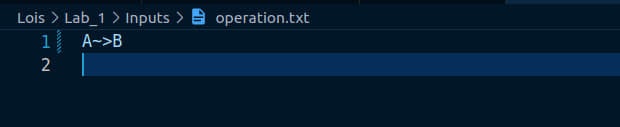


Рис. 15. Входные данные для ответа на контрольный вопрос 1 (часть 2).

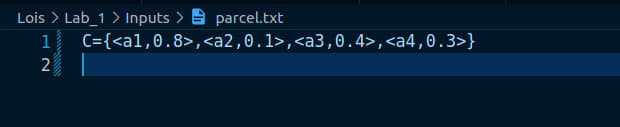


Рис. 16. Выходные данные для ответа на контрольный вопрос 1.



Нечеткое множество A считается нормальным, если его высота равна 1, то есть верхняя граница его функции принадлежности, обозначаемая как sup x ∈ X μ A x, равна 1. В случае, когда sup x ∈ X μ A x < 1, нечеткое множество считается субнормальным [5]. В данном контексте множества А и B считаются нормальными, поскольку в них присутствует степень

принадлежности, равная 1. В отличие от них, нечеткое множество C является субнормальным, так как все его степени принадлежности меньше 1. Результатом является нечеткое множество, которое также считается субнормальным. Поскольку элемент таблицы конъюнкции равен минимальному элементу из соответствующих элементов посылки и таблицы импликации, в случае с субнормальным нечетким множеством все элементы в таблице конъюнкции будут меньше 1. Следовательно, результат, представляющий собой нечеткое множество, степень принадлежности элемента которого является максимальным значением в соответствующем столбце таблицы конъюнкции, также будет субнормальным.

Таким образом, существует возможность, что для некоторого C и нормальных A и B результат не будет являться нормальным множеством.

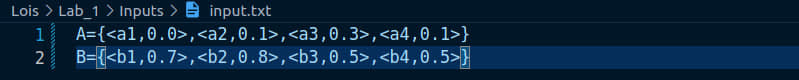
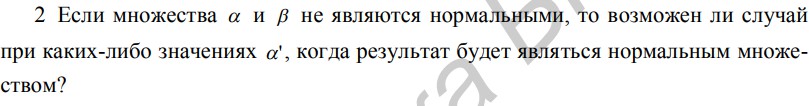


Рис. 17. Входные данные для ответа на контрольный вопрос 2 (часть 1).

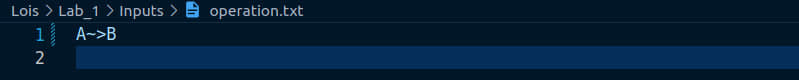


Рис. 18. Входные данные для ответа на контрольный вопрос 2 (часть 2).

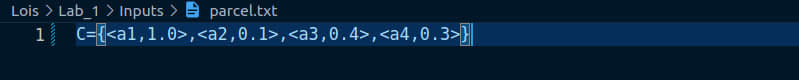
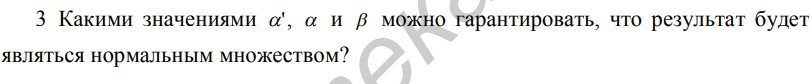


Рис. 19. Выходные данные для ответа на контрольный вопрос 2.



Множества А и B не могут считаться нормальными, поскольку все степени принадлежности в них меньше 1. В отличие от них, нечеткое множество C считается нормальным из-за наличия степени принадлежности, равной 1. Результирующее нечеткое множество также является нормальным. Поскольку элемент в таблице конъюнкции равен минимальному значению из соответствующих элементов посылки и таблицы импликации, если посылка является нормальным множеством и соответствующая строка в таблице импликации (строка, сравниваемая с необходимым значением в посылке) содержит 1, то все элементы этой строки в таблице конъюнкции будут равны 1. Таким образом, результатом является нечеткое множество, и степень принадлежности его элемента равна максимальному значению в соответствующем столбце таблицы конъюнкции, что также делает его нормальным[1].



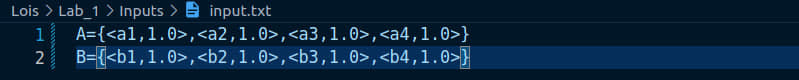


Рис. 20. Входные данные для ответа на контрольный вопрос 3 (часть 1).

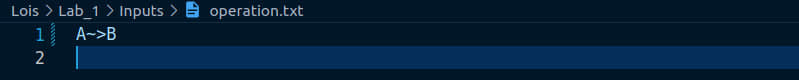


Рис. 21. Входные данные для ответа на контрольный вопрос 3 (часть 2).

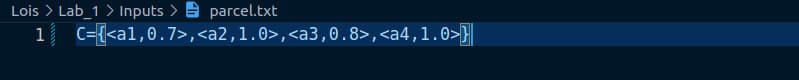


Рис. 22. Выходные данные для ответа на контрольный вопрос 3.



Для того чтобы результат был нормальным нечетким множеством с высотой, равной 1, необходимо, чтобы в таблице конъюнкций присутствовало хотя бы одно значение 1. Это обусловлено тем, что результат формируется путем проверки каждого столбца матрицы конъюнкции и выбора максимального элемента из него. Чтобы матрица конъюнкции содержала значение 1, необходимо, чтобы хотя бы одна единица находилась на соответствующей позиции в посылке и в соответствующей строке матрицы импликации. Для достижения значения 1 в строке матрицы импликации требуется, чтобы степень принадлежности соответствующего элемента множества α была меньше или равна степени принадлежности любого элемента множества β[1].

Как следует из начальных входных данных, данная ситуация возможна, если все степени принадлежности в начальных множествах будут равны 1.

# Вывод:

В процессе выполнения лабораторной работы были освоены навыки применения методов прямого нечеткого логического вывода. Была разработана система прямого нечеткого логического вывода, которая использует (нечеткое) граничное произведение (импликацию Лукасевича) для вычисления меры возможности. Работоспособность программы была проверена и подтверждена результатами тестирования[2][3].

# Список использованных источников:

1. Логические основы интеллектуальных систем. Практикум : учеб.- метод. пособие / В. В. Голенков [и др.]. – Минск : БГУИР, 2011. – 70 с. : ил. ISBN 978-985-488-487-5.
2. Нечеткие высказывания и максиминные операции над ними [Электронный ресурс]. – Режим доступа: https://studfile.net/preview/5562531/page:29/
3. Определение и основные характеристики нечетких множеств [Электронный ресурс]. – Режим доступа: <http://nrsu.bstu.ru/chap21.html>