Министерство образования Республики Беларусь

Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №1 по дисциплине

# ЛОГИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Студент гр. 121701 В. А. Дичковский

Руководитель В. П. Ивашенко

Минск 2024

# СОДЕРЖАНИЕ

1. [Ход выполнения лабораторной работы](#_bookmark0) . . . . . . . . . . . . . . . 2
   1. [Описание](#_bookmark1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   2. [Теоретические сведения](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . 3
   3. [Формат базы знаний](#_bookmark3) . . . . . . . . . . . . . . . . . . . . . . . 3
   4. [Реализация](#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   5. [Демонстрация результатов работы программы](#_bookmark5) . . . . . . . . 5
      1. [Тест 1](#_bookmark6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
      2. [Тест 2](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      3. [Тест 3](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   6. [Контрольные вопросы](#_bookmark9) . . . . . . . . . . . . . . . . . . . . . . 7
2. [Вывод](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

[Список использованных источников](#_bookmark12) 10

# ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

## Вариант 4

**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

**Задание**: Реализовать систему прямого нечёткого логического выво- да. Исходные данные и результаты должны быть сохранены в файле. используя треугольную норму драстического произведения и нечеткую импликацию Вебера.

## Описание

Задача заключается в написании алгоритма прямого нечёткого логического вывода, используя треугольную норму драстического произведения и нечеткую импликацию Вебера.

Входом программы является файл, содержащий множество нечётких правил и фактов.

Для реализация программы использовался язык программирования C#.

Были использованы следующие структуры данных: список, массив, класс.

Структура приложения выглядит следующим образом:

Inference.cs содержит алгоритм для прямого нечеткого логического вывода;

Program.cs содержит входную точку программы, также в нём происходит чтение исходного файла и запись файла после выполнения прямого нечеткого вывода. Исходные файлы располагаются в папке Input проекта, а записываются файлы в папку Output;

KnowlegeBase.cs содержит список начальных предикатов, список правил и список выведенных предикатов. В этом классе есть метод, который возвращает строку для записи в файл и строку для вывода в консоль выведенных предикатов;

Parser.cs – содержит парсер исходного файла. В парсере есть функция, которая последовательно обрабатывает каждую строку. Также в нём есть функция для анализа предиката и анализ правила;

Predicate.cs -содержит класс для хранения предиката;

InferedPredicate.cs содержит класс для хранения выведенного предиката. Данный касс унаследован от Predicate. В нём дополнительно хранится правило, по которому он был выведен и предикат который использовался для вывода.

Rule.cs -содержит класс для хранения правила

## Теоретические сведения

Нечеткий предикат – это нечеткое множество, значения которого интерпретируются как значения истинности.

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

Правило – импликация, которая выражает зависимость между наблюдаемыми причинами и следствиями.

Прямой нечеткий логический вывод – композиция между двумя нечеткими предикатами, один из которых рассматривается как унарный (посылка), а второй бинарный (импликация фактов по заданному правилу).

Нечеткое высказывание – утверждение, в котором истинность оценивается с использованием степени принадлежности к нечеткому множеству.

Нечеткая импликация нечетких высказываний - это операция, которая определяет отношение между двумя нечеткими высказываниями.

## Формат базы знаний

<база знаний> ::= <список фактов>| <список фактов><новая строка><список правил>

список фактов» ::= <факт>|<факт> <новая строка> <список фактов»

<список правил> ::= <правило> | <правило> <новая строка><список правил>

<факт> ::= <имя нечеткого предиката><равенство><нечеткое множество>

<правило> ::= <имя нечеткого предиката> <нечеткая импликация> <имя нечёткого предиката>

<нечеткое множество>::= <открывающая фигурная скобка> <список пар нечеткой принадлежности> <закрывающая фигурная скобка>

<список пар нечеткой принадлежности> ::= <пара нечеткой принадлежности> |<пара нечеткой принадлежности><запятая>< список пар нечеткой принадлежности>

<пара нечеткой принадлежности> ::= <открывающая угловая скобка><элемент><запятая><степень принадлежности><закрывающая угловая скобка>

<имя нечеткой принадлежности> ::= <имя><открывающая полукруглая скобка><имя><закрывающая полукруглая скобка>

<имя> ::= <буква> |<буква><символы>

<символы> ::= <символ>|<cимвол> <символы>

<степень принадлежности> ::= <действительное число с 0 по 1>

<действительное число с 0 до 1> ::= <единица> |<единица> <точка><нули>|<действительное число с 0 до 1>

<действительное число с 0 до 1> ::= <ноль>|<ноль><точка><цифры>

<цифры> ::= <цифра>|<цифра><цифры>

<символы> ::= <буква>|<цифра>

<буква>A|B|C|D|F|G|H|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<цифра> ::= 0|1|2|3|4|5|6|7|8|9

<единица> ::=1

<ноль>::=0

<точка> ::=.

<запятая>::=,

<равенство>::= =

<нечеткая импликация> ::=~>

<открывающая угловая скобка> ::= <

<закрывающая угловая скобка> ::=<

<открывающая фигурная скобка> ::= {

<закрывающая фигурная скобка>::=}

<открывающая полукруглая скобка> ::= (

<закрывающая полукруглая скобка> ::=)

## Реализация

Программа включает в себя класс Inference, который отвечает за прямой нечеткий логический вывод. Он включает в себя следующие методы:

private static float[] GetMaxInColumns(float[,] dractricMatrix) – метод, находящий точную верхнюю границу Sup для каждого столбца в матрице – float DrasticMultiplication(float x1, float x2)– реализация драстического произведения;

public static voiProcessInference(List<Predicate>InputPredicates) - метод, находящий все выводимые уникальные предикаты.

## Демонстрация результатов работы программы

* + 1. **Тест 1**

Входной файл input1.kb:

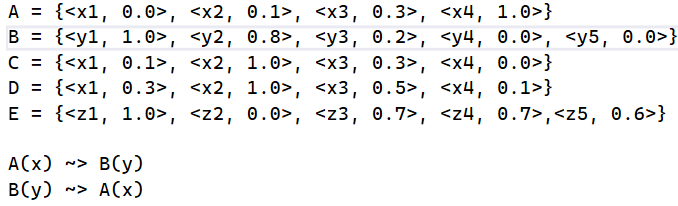


Рисунок 1.1 – Входной файл input1.kb для теста 1

Результат работы программы для файла input1.kb:

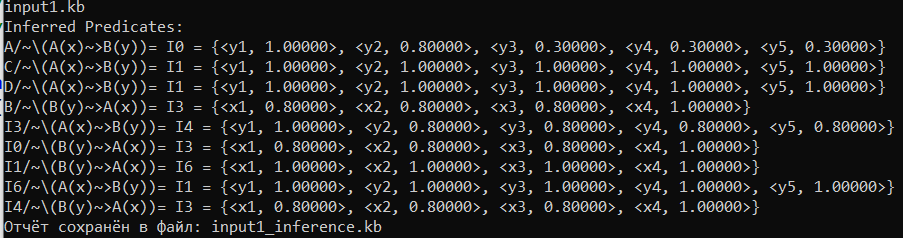


Рисунок 1.2 – Результат работы программы для файла input1.kb

## Тест 2

Входной файл input2.kb:

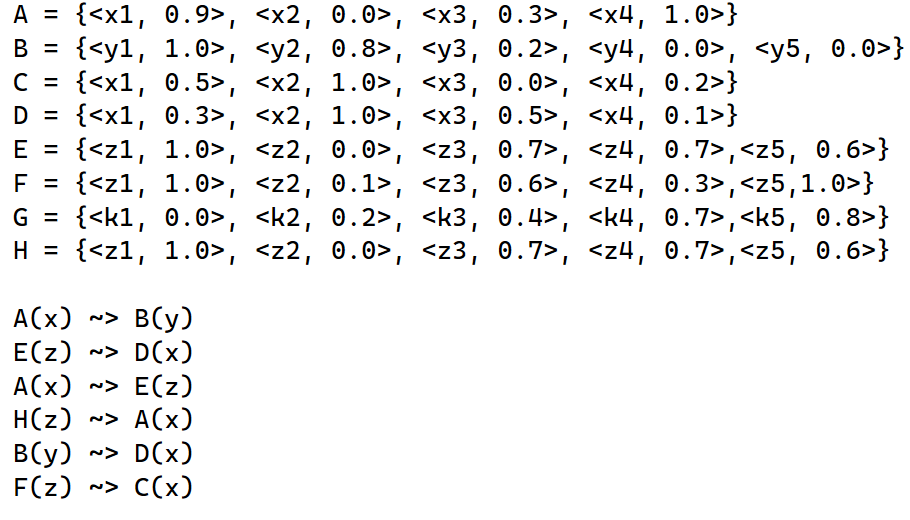


Рисунок 1.3 – Входной файл input2.kb для теста 2

Результат работы программы для файла input2.kb:

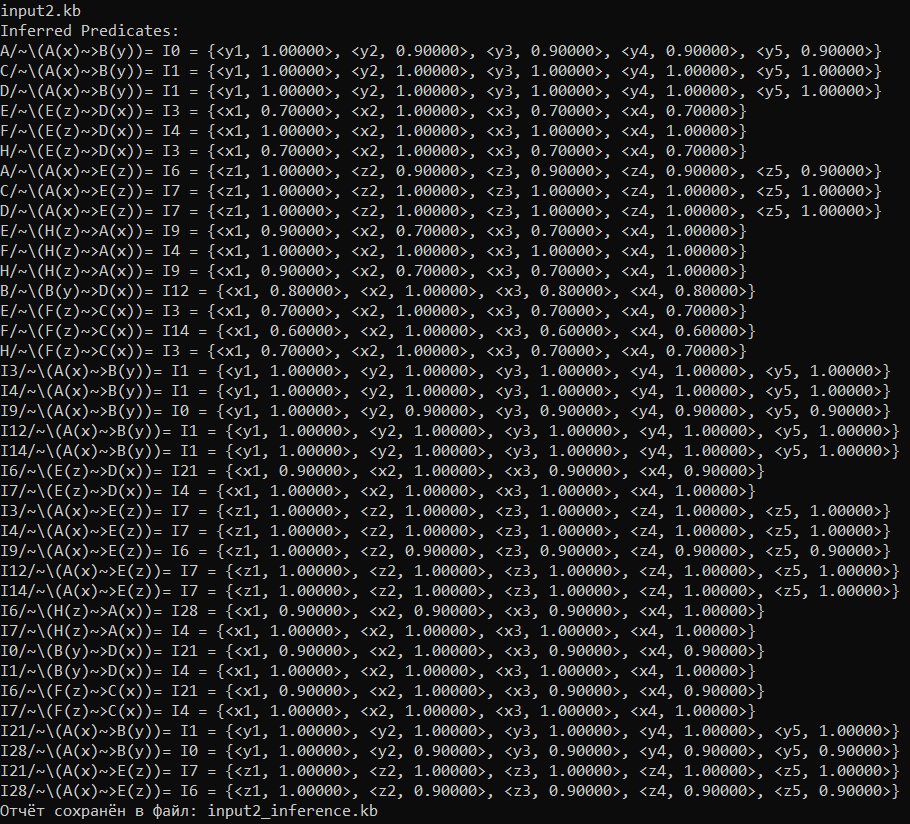


Рисунок 1.4 – Результат работы программы для файла input2.kb

## Тест 3

Входной файл input3.kb:

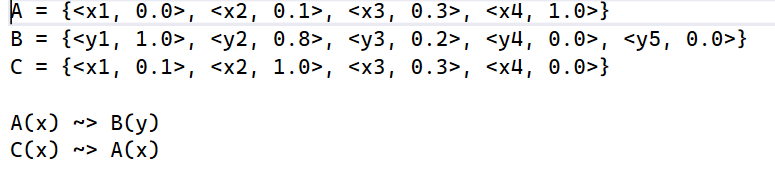


Рисунок 1.5 – Входной файл input3.kb для теста 3

Результат работы программы для файла input3.kb:

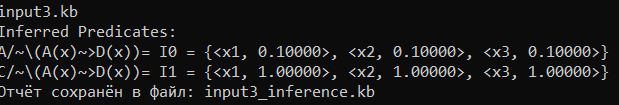


Рисунок 1.6 – Результат работы программы для файла input3.kb

## Контрольные вопросы

## Нечеткое множество называется нормальным, если точная верхняя грань его функции принадлежности равно 1

1. **Если множества *α* и *β* являются нормальными, то возможен ли случай при каких-либо значениях *α′*, когда результат не будет являться нормальным множеством?**

Приведем пример на рисунке 11:

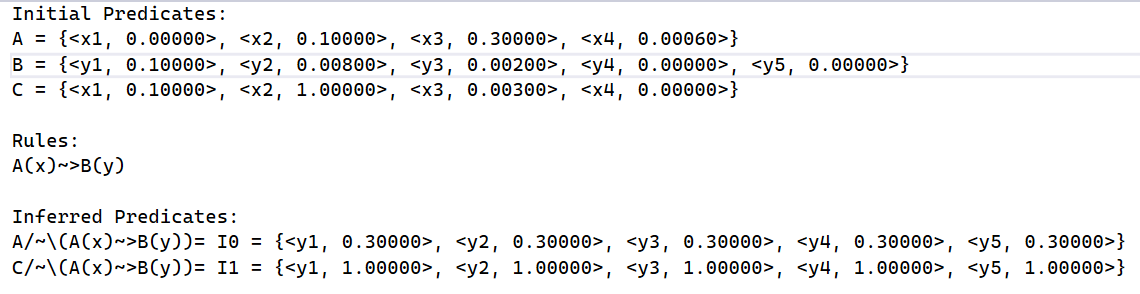


Рисунок 11 Пример вывода нормального предиката из не нормальных *α* и *β*

Предикат A,B являются нормальными, предикат C не является нормальным, результат является не нормальным.

## Если множества *α* и *β* не являются нормальными, то возможен ли случай при каких-либо значениях *α′* , когда результат будет являться нормальным множеством?

Приведем пример на рисунке 12:

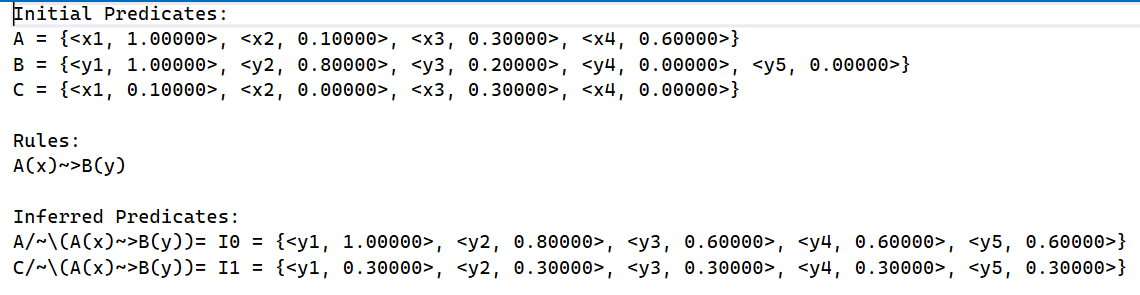


Рисунок 12 Пример вывода не нормального предиката из нормальных *α* и *β*

Предикат *A*, *B* не являются нормальными, а выведенный предикат I1 следовательно, такие случаи возможны.

## Какими значениями *α′* , *α* и *β* можно гарантировать, что результат будет являться нормальным множеством?

– Предикат, используемый при выводе, является нормальным.

При каких A можно получить субнормальное нечеткое множество

При каких A можно получить нормальное нечеткое множество когда посылка и заключение нечёткой импликации являются нормальными начёткими множествами? и заключение начёткой импликации являются субнормальными изчеткими множествами? Можно ли получить нечеткое множество, когда посылка является Можно ли получить нечетков множество В(9) когда посылка является субнормальным нечетким множеством, в заключение- Требования к программной реализации. Можно ли получить нечеткое множество Ву) В(у), когда посылка и заключение нечеткой импликации являются субнормальными нечеткими множествами? когда посылка и заключение нечеткой импликации являются нормальными нечеткими множествами? нормальным нечетким множеством, а заключение субнормальным?

**Личный вклад**

Дичковский Владимир занимался программированием прямого логического вывода.

Максимков Максим отвечал за наполнение и оформление отчёта.

Ответы на вопросы были сделаны совместно.

# ВЫВОД

В процессе выполнения лабораторной работы, были получены навыки реализации нечёткой логики, а именно прямого нечёткого логического вывода при помощи программирования. В рамках данной работы были разработаны модули, отвечающие за анализ исходного текста базы знаний, а также непосредственно алгоритм прямого нечёткого логического вывода. При помощи разработанного программного продукта удалось построить корректные выводы для нескольких случаев, а также дать ответы

на контрольные вопросы, прилагающиеся к лабораторной работе.

# CПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Голенков, В. В. Логические основы интеллектуальных систем. Прак- тикум: учеб.-метод. пособие / В. В. Голенков. — БГУИР, 2011.