Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

**ОТЧЁТ**

по дисциплине «Логические основы интеллектуальных систем»

Лабораторная работа №2

Вариант 1

Выполнил: Самута Д. В.

гр. 221703

Проверил: Ивашенко В. П.

Минск 2024

**Тема**: программирование логического вывода прикладных и неклассических логик.

**Цель**: приобрести навыки разработки и прикладных систем логического вывода , включая системы неклассического логического.

**Постановка задачи**: запрограммировать обратный нечеткий логический вывод, на основе операции нечеткой композиции (max({min({xi}U{yi})|i}).

**Описание лабораторной работы:**

На вход программе подается файл, содержащий нечеткий предикат и нечеткий бинарный предикат – правило и заключение.

Для решения поставленной задачи требуется сделать следующее:

1. Проверить валидность введенных предикатов
2. Сформировать удобные для работы с предикатами структуры данных
3. Произвести обратный нечеткий вывод на основе операции нечеткой композиции (max({min({xi}U{yi})|i}).

**Дополнительные теоретические сведения**:

1. Нечеткий предикат – это нечеткое множество, значения которого интерпретируются как значения истинности.
2. Правило – импликация, которая выражает зависимость между наблюдаемыми причинами и следствиями.
3. Прямой нечеткий логический вывод – композиция между двумя нечеткими предикатами, один из которых рассматривается как унарный (посыл- ка), а второй бинарный (импликация фактов по заданному правилу).
4. Обратный нечеткий логический вывод – обратное по отношению к прямому нечеткому логическому выводу действие.
5. Нечеткое высказывание – утверждение, в котором истинность оценивается с использованием степени принадлежности к нечеткому множеству.
6. Импликация – бинарная логическая связка, по своему применению приближенная к союзам «если..., то...».
7. Нечеткая импликация нечетких высказываний - это операция, которая определяет отношение между двумя нечеткими высказываниями.

Использованные структуры данных:

HashMap в Java – одна из реализаций ассоциативного массива в Java.

ArrayList – это усовершенствованный массив, в котором можно изменять кол-во элементов и с легкостью выполнять различные операции.

Set – коллекция уникальных элементов.

**Формат базы знаний:**

<база знаний>::= <список фактов> <список правил>

<список фактов>::= <факт> <факт>

<список правил>::= <правило> <правило>

<факт>::= <имя нечёткого множества>=<нечёткое множество>

<правило>::= <имя нечёткого множества>~><имя нечёткого множества>

<нечёткое множество>::={ <список пар нечёткой принадлежности> }

<список пар нечёткой принадлежности>::=

<пара нечёткой принадлежности>, <пара нечёткой принадлежности>

<пара нечёткой принадлежности>::=(<элемент>,<степень принадлежности>)

<элемент>::=<имя> | <множество>

<множество>::=<ориентированное множество> |

<неориентированное множество>

<неориентированное множество>::={ <список элементов> }

<ориентированное множество>::=(<элемент>,<список элементов>)

<список элементов>::=<элемент> ,<элемент> ,<элемент>

<имя нечёткого множества>::=<имя>

<имя>::=<символ> <символ>

<символ>::=<буква>|<цифра>

<цифра>::=0|...|9

<буква>::=A|...|z

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

<действительное число с 0 по 1>::=<единица>|

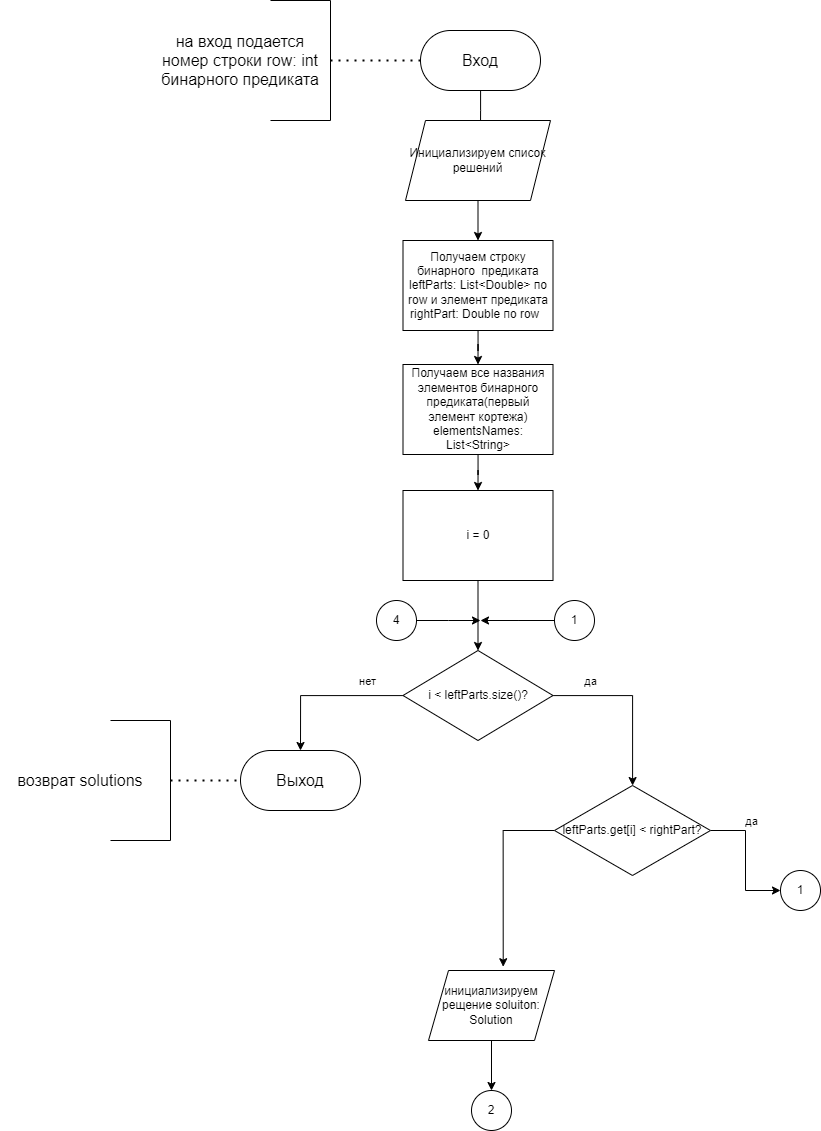
<действительное число с 0 до 1>

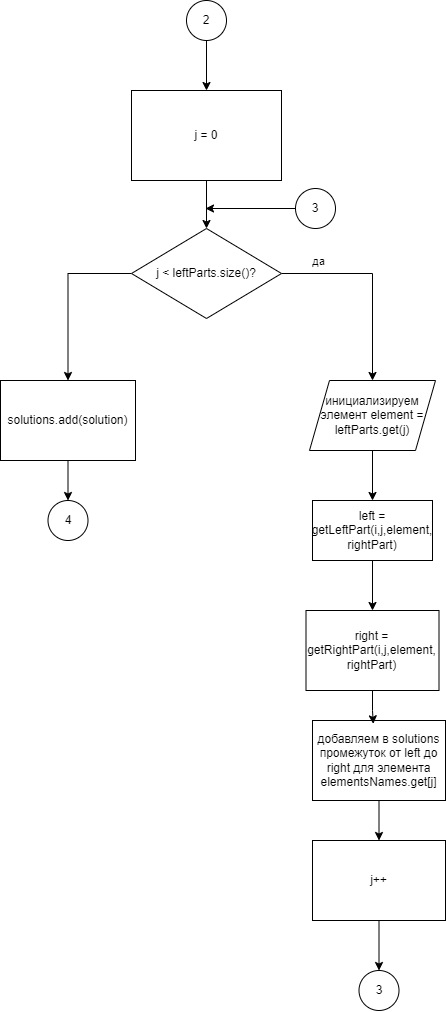
<действительное число с 0 до 1>::=0 . <цифра>

<единица>::=1 . 0

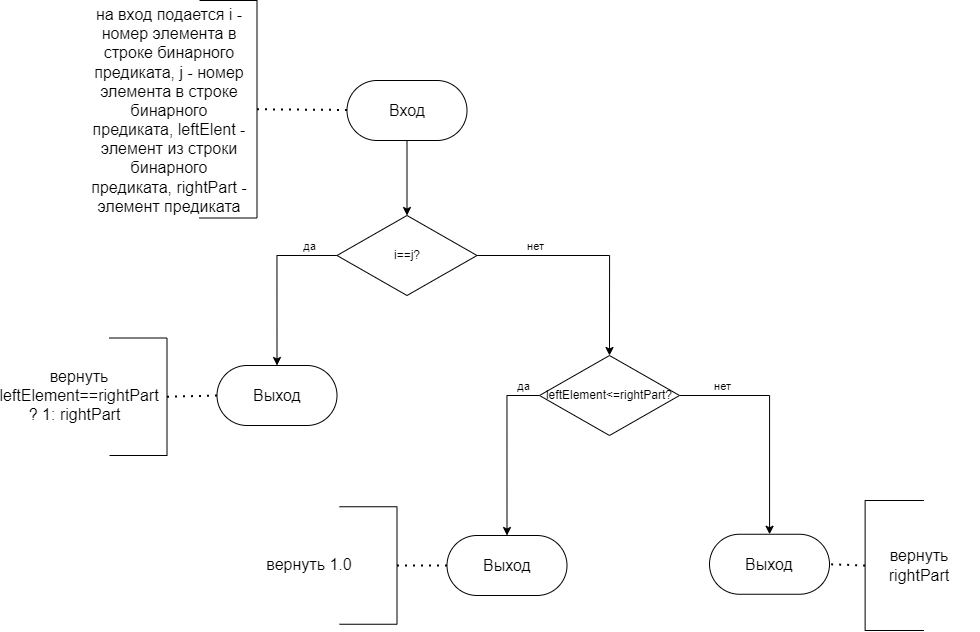
**Описание реализации алгоритмов:**

1. Метод, вычисляющий решение для одной строки бинарного предиката. Название метода: computeImplication(int row).

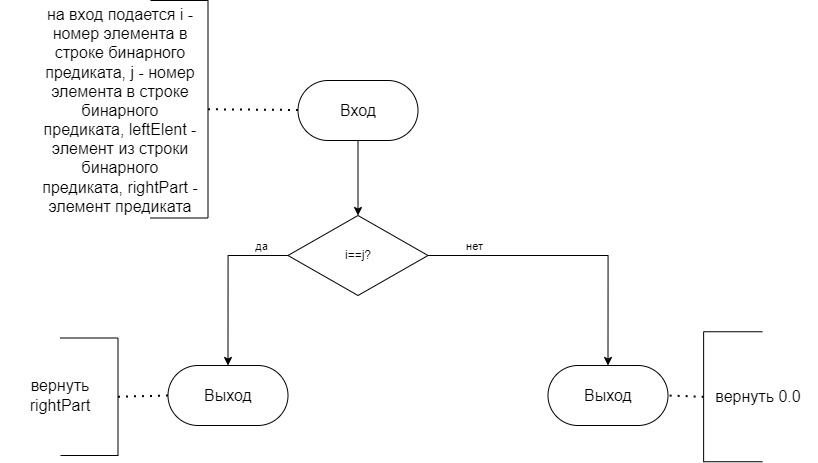




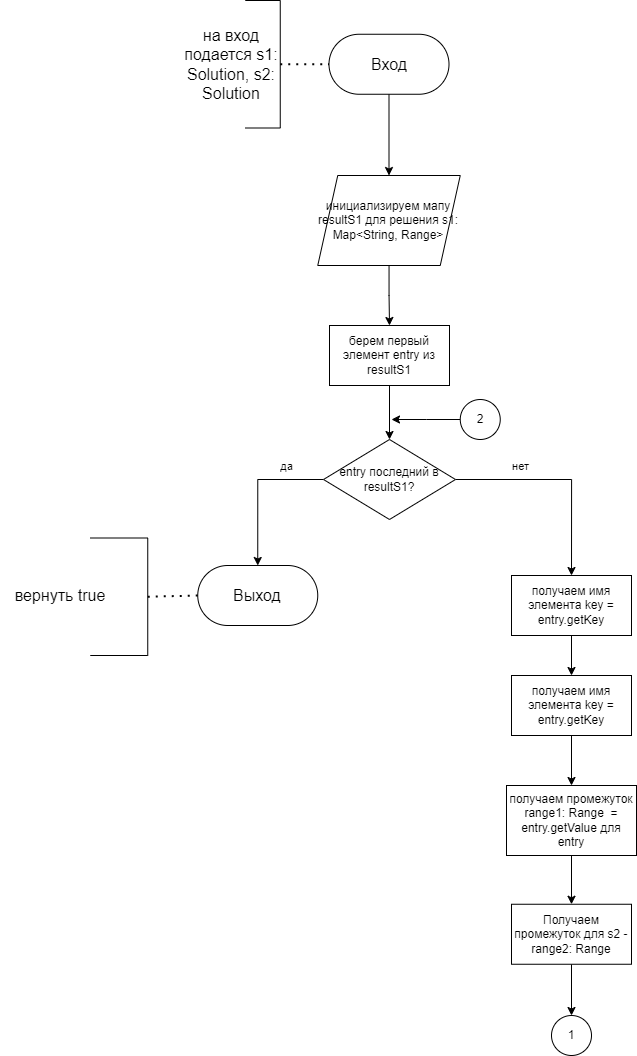
1. Метод, вычисляющий левую часть промежутка. Название метода: getLeftPart(int i, int j, double leftElement, double rightElement).

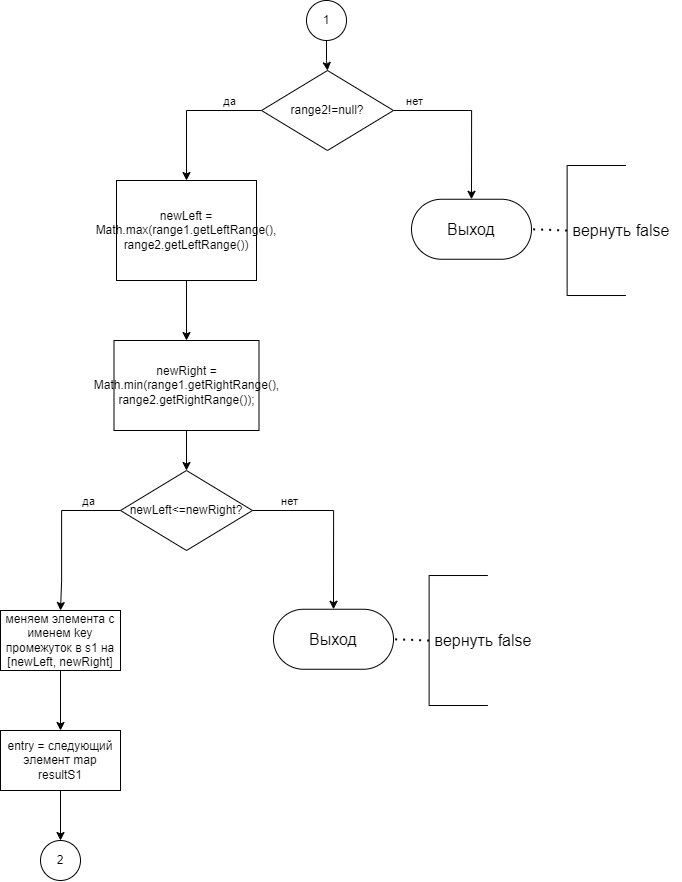


1. Метод, вычисляющий правую часть промежутка. Название метода: getRightPart(int i, int j, double leftElement, double rightElement).

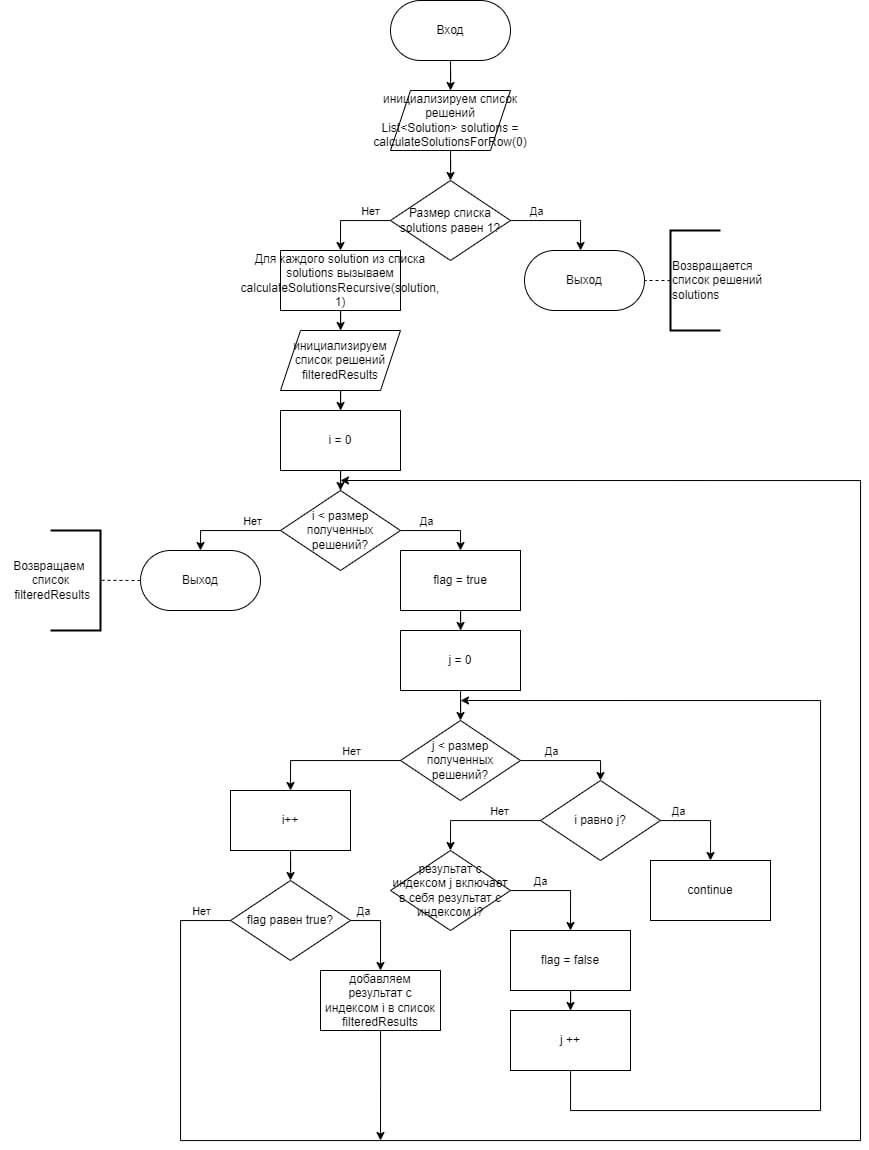


1. Метод вычисляющий пересечение между для двух решений.

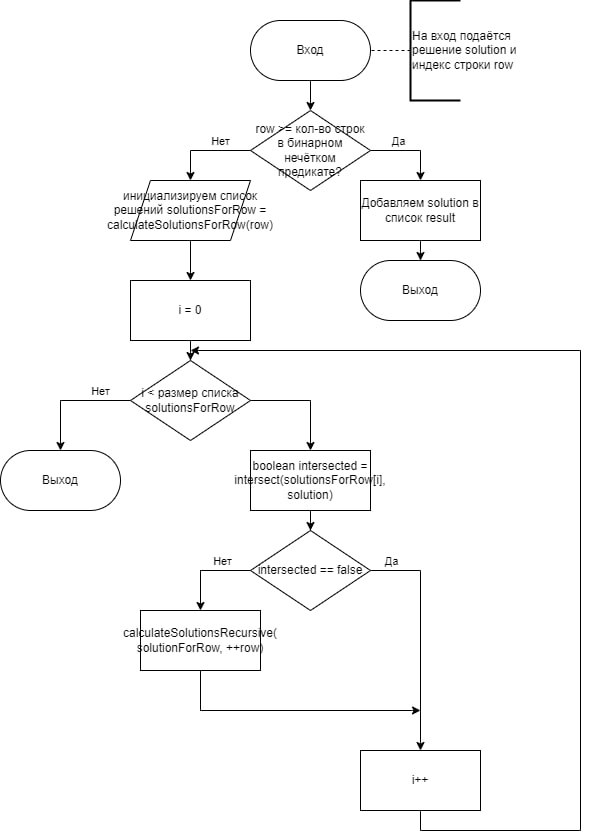




1. Метод, вычисляющий список решений calculateSolutions()



1. Метод, рекурсивно вычисляющий решения calculateSolutionsRecursive(Soluiton solution, int row)



Демонстрация результатов работы:

Тест 1:

Входные данные:

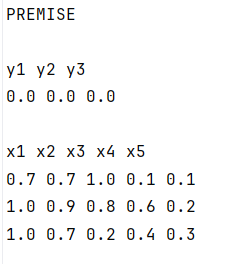


Рис. 1 Входные данные 1-го теста

Выходные данные:



Рис. 2 Выходные данные 2-го теста

Тест 2:

Входные данные:

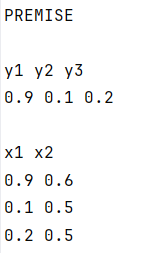


Рис. 3 Входные данные 2-го теста

Выходные данные:



Рис. 4 Выходные данные 2-го теста

Тест 3:

Входные данные:

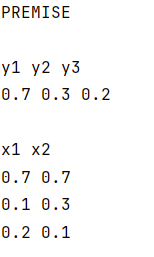


Рис. 5 Входные данные 3-го теста

Выходные данные:



Рис. 6 Выходные данные 2-го теста

**Личный вклад:**

В рамках данной лабораторной работы Самутой Д.В. был реализован алгоритм для поиска всех возможных решений для одной строки системы уравнений обратного логического вывода.

Семеновым Е. Г. был реализован поиск всех возможных решений системы уравнений обратного логического вывода.

Отчет писали вместе.

**Вывод:**

В ходе выполнения лабораторной работы мы освоили навыки реализации нечёткой логики, в частности, обратного нечёткого логического вывода с использованием программирования. В рамках работы были разработаны модули для анализа исходного текста базы знаний, а также алгоритм обратного нечёткого логического вывода. С помощью созданного программного обеспечения удалось получить корректные выводы для нескольких случаев и успешно ответить на контрольные вопросы, приложенные к лабораторной работе.

# **Список использованных источников:**

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

[2] Ито, О. Промышленные применения. // Прикладные нечеткие системы /О. Ито [и др.] ; под ред. Т. Тэрано, К. Асаи, М. Сугэно; пер. с япон. Ю. Н. Чернышова. – М. : Мир, 1993.