Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и

Радиоэлектроники

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

по курсу “Логические основы интеллектуальных систем”

Вариант 1

Выполнил:

Студент гр. 221701 Абушкевич А. А

Проверил: Ивашенко В. П.

Минск

2024

**Тема:**

Программирование операций обработки и преобразования формул прикладных неклассических логик.

**Цель:**

Приобрести навыки программирования алгоритмов обработки данных в неклассических логических моделях решения задач.

**Задание:**

Реализовать прямой нечеткий логический вывод, используя треугольную норму min({xi}U{yi}) и нечеткую импликацию Гёделя.

**Описание:**

Задача заключается в написании алгоритма прямого нечеткого логического вывода, используя импликацию Гёделя. Входом программы является файл, содержащий множество нечетких правил и факторов.

Для реализации программы использовался язык программирования С#, а также библиотека ANTLR.

Были использованы следующие структуры данных:

1. Список;
2. Кортеж;
3. Класс;
4. Словарь.

Структура приложения выглядит следующим образом:

1. Compares.cs – реализация классов, отвечающих за сравнение нечетких множеств по носителю и по содержанию;
2. Inference.cs – реализация классов, отвечающих за осуществление нечеткого логического вывода, а также операции нахождения треугольной нормы и импликации нечетких множеств;
3. Директория Input – содержит тестовые вход для программы;
4. Predicate.cs – реализация класса, отвечающего за представление нечеткого множества;
5. KB.cs – реализация класса, отвечающего за представление базы знаний с фактами и правилами;
6. Rule.cs – реализация класса, отвечающего за представление правила нечёткого логического вывода;
7. ExceptionListener.cs – реализация класса, отвечающего за обработку исключений, возникших в ходе анализа входных данных;
8. KBVisitor.cs - реализация класса, отвечающего за обработку лексем записи базы знаний и преобразование их в объекты классов системы. Реализует паттерн проектирования "посетитель";
9. ParserFacade.cs – реализация класса, отвечающего за инкапсуляцию деталей анализатора входных параметров. Реализует шаблон проектирования «фасад»;
10. FuzzyLogic.g4 – представляет файл описания грамматики нечеткой логики;
11. Program.cs – главный файл программы, отвечающий за считывание данных из файла и запуск нечеткого логического вывода;

**Теоретические сведения:**

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 (посылки), а второй бинарный (импликация фактов по заданному правилу);

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

Нечеткий предикат – это нечеткое множество, значение которого интерпретируется как значение истинности;

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

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><импликация><имя нечеткого 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 нечеткого предиката> :: = <имя><открывающая полукруглая скобка><имя><закрывающая полукруглая скобка>

<элемент> :: = <имя>

<имя> :: = <буква>|<буква><символы>

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

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

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

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

<нули> :: = <ноль>|<ноль><нули>

<цифры> :: = <цифра>|<цифра><цифры>

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

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

<имя нечеткого множества> :: = <имя>

<имя> :: = <символ> <символ>

<символ> :: = <буква> | <цифра>

<цифра> :: = 0| … |9

<буква> :: = A| … |z

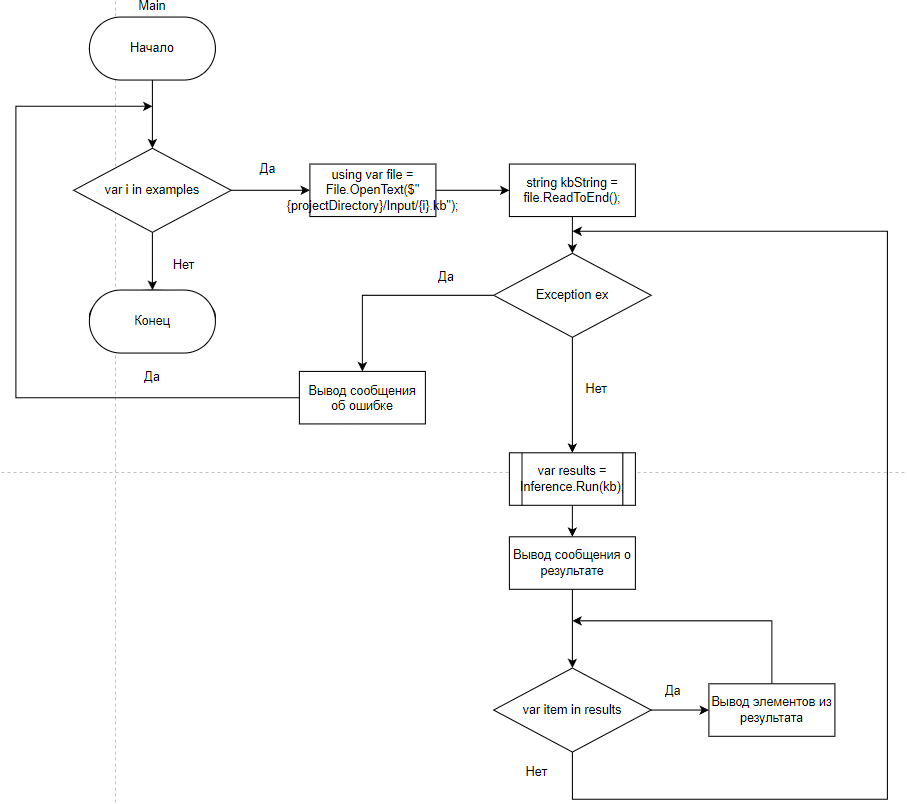
<степень принадлежности> :: = <действительное число с 0 по 1>

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

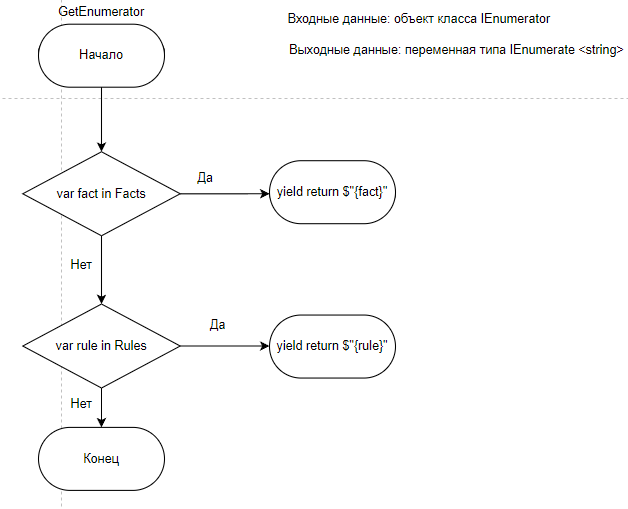
<действительное число с 0 по 1> :: = 0 | <цифра>

**Блок-схемы функций программы:**

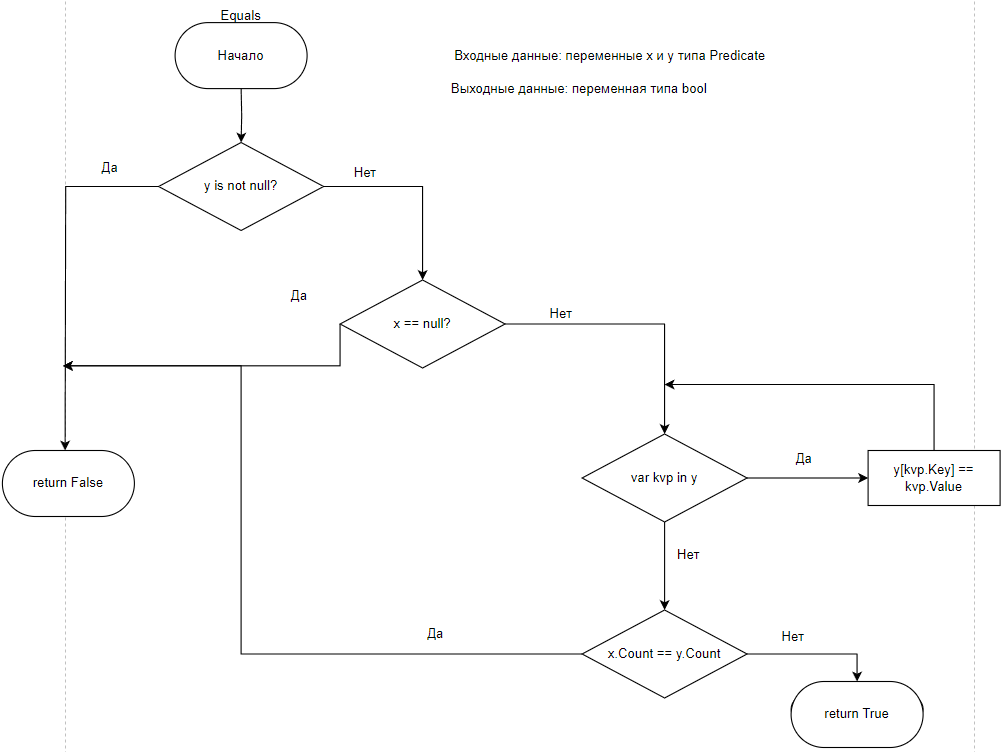
1. Функция Main()



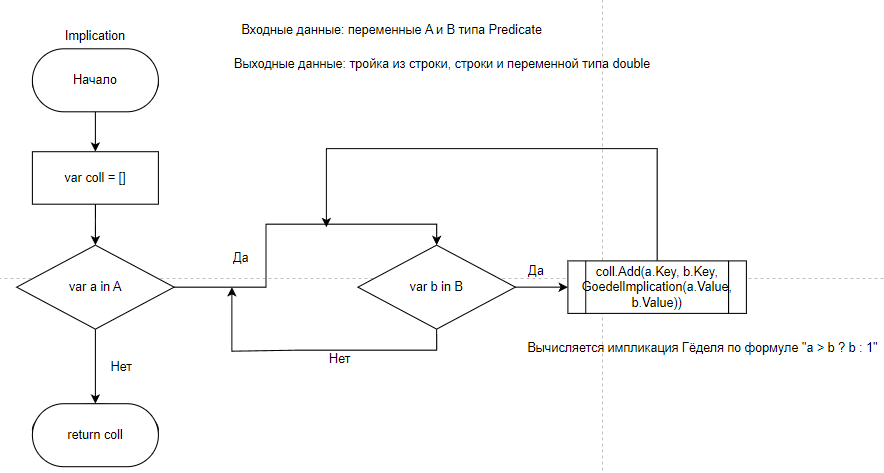
1. Функция GetEnumerator()



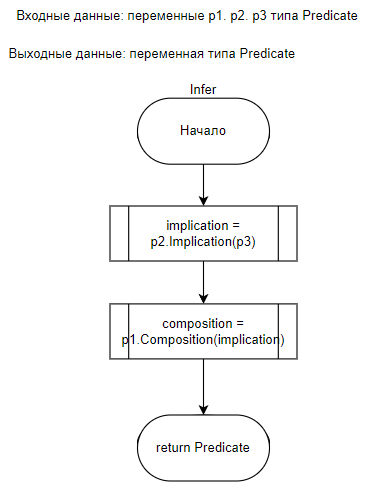
1. Функция Equals()



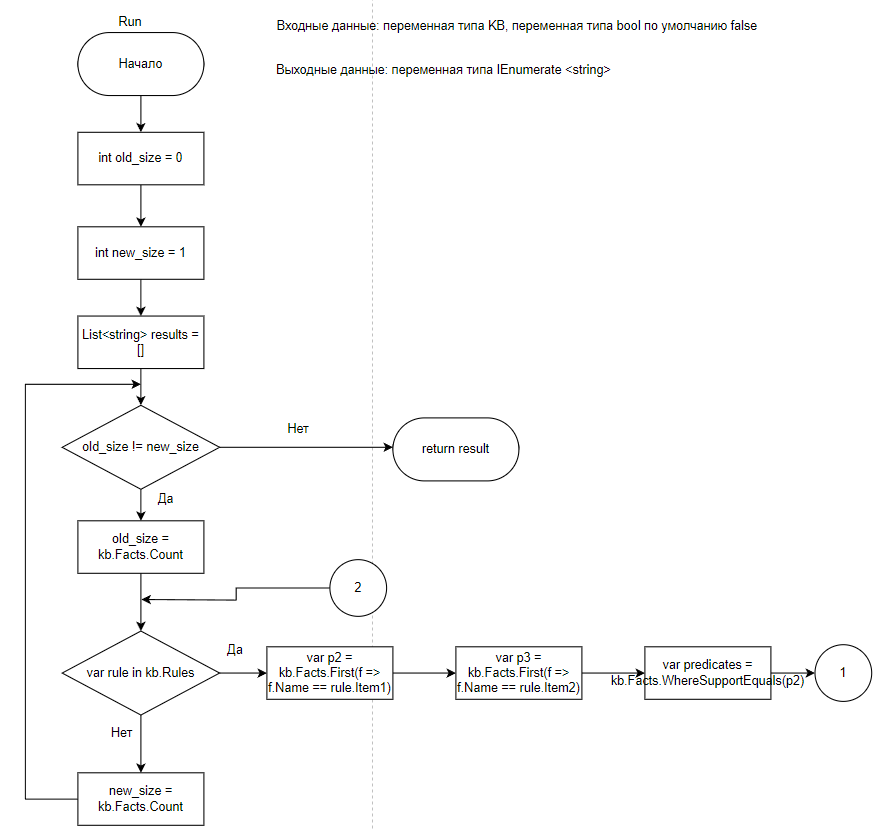
1. Функция Implication()

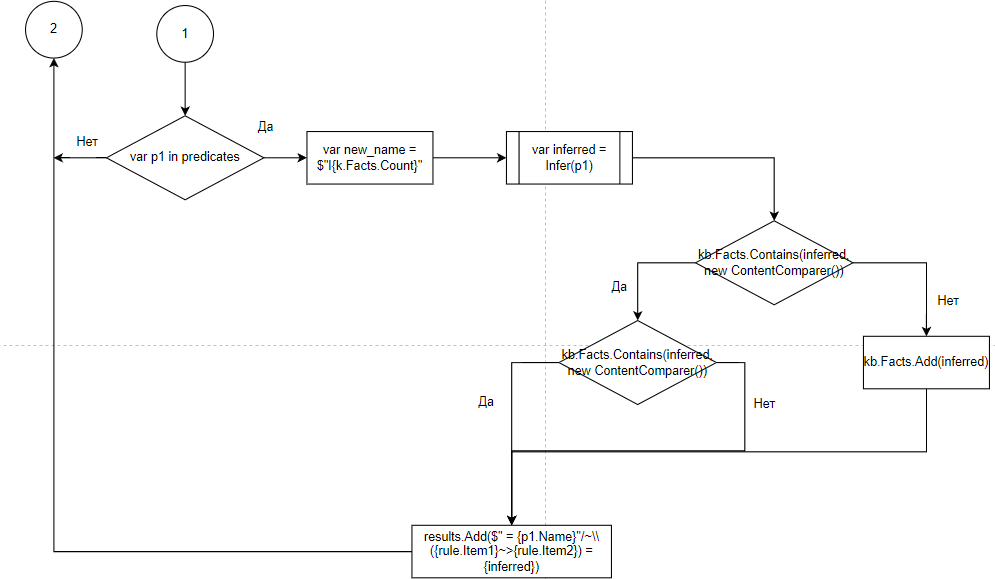


1. Функция Infer()

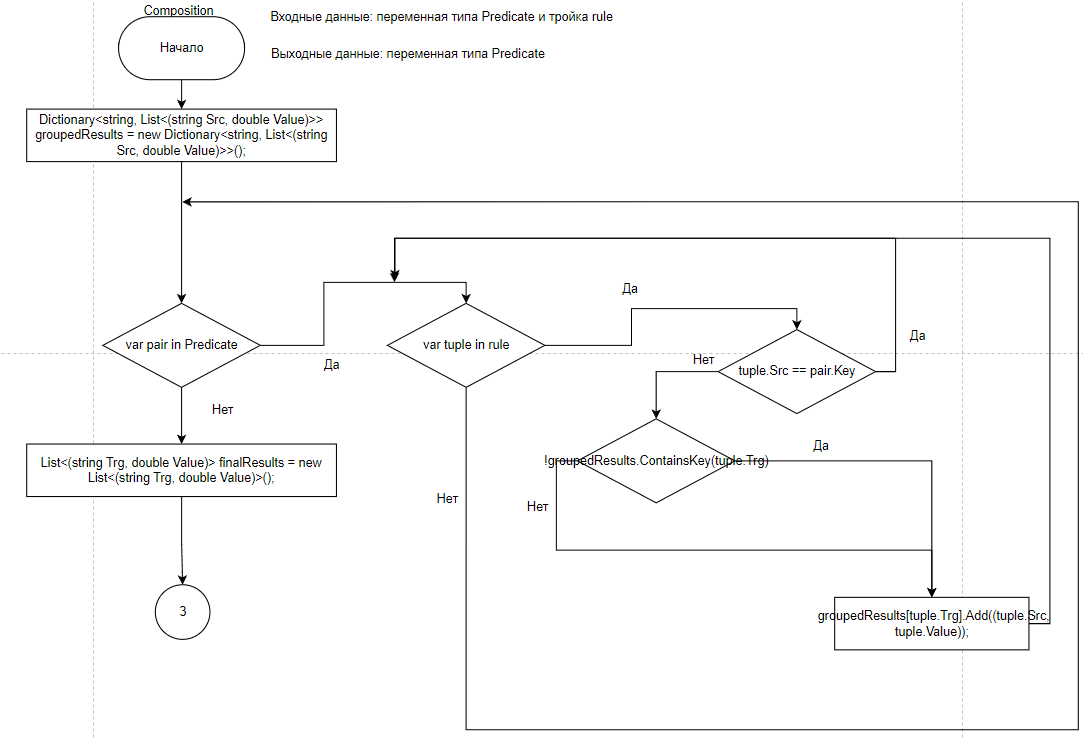


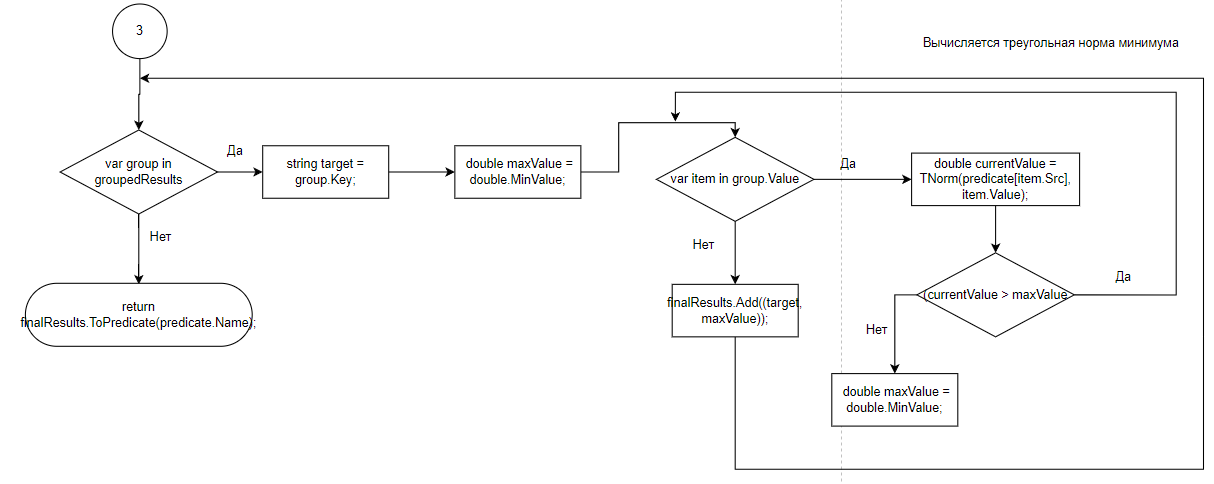
1. Функция Run()





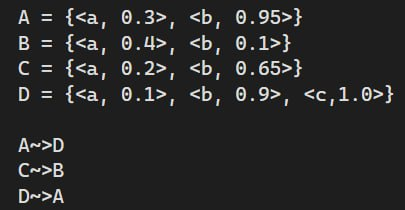
1. Функция Composition()



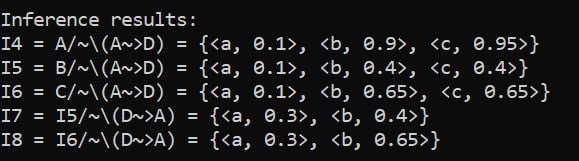


**Пример работы программы:**

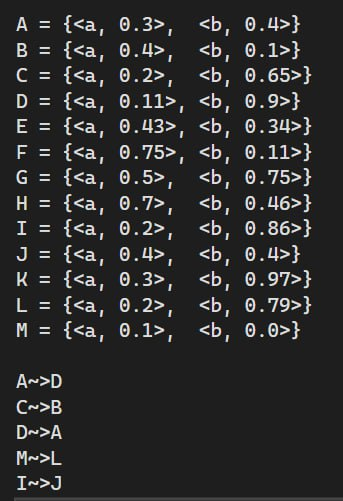
Пример 1:



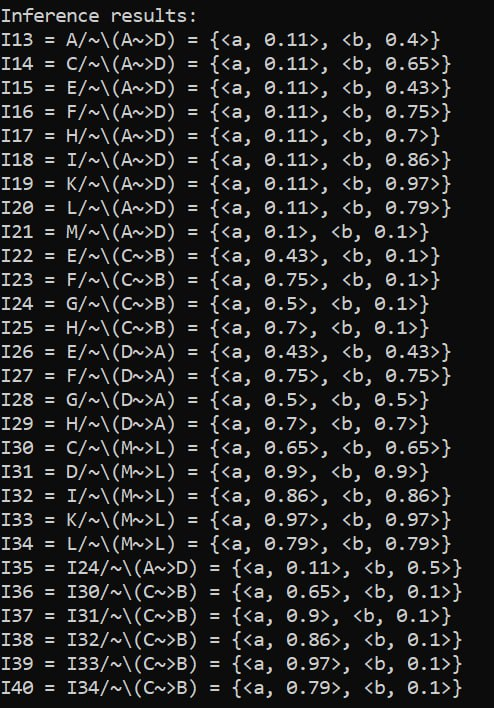
Результат:



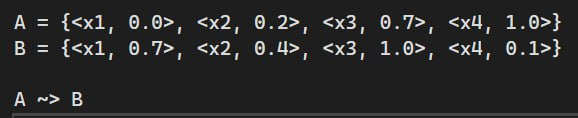
Пример 2:



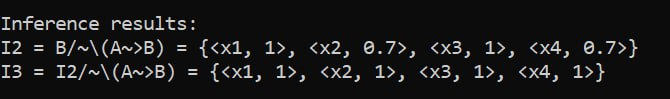
Результат:



Пример 3:



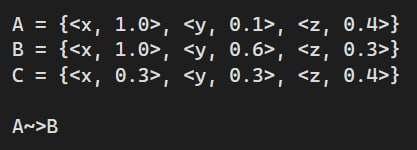
Результат:



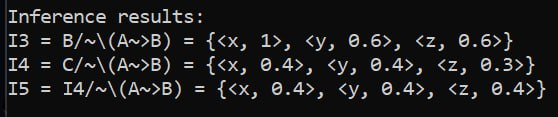
**Контрольные вопросы:**

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

Вход программы:



Вывод программы:

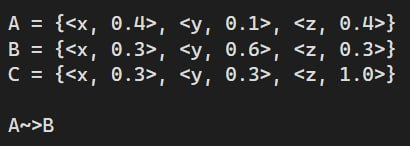


Множества не являются нормальными, следовательно, такие случаи возможны.

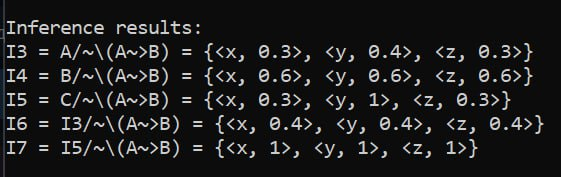
Если предикат не является нормальным, то результат также не будет нормальным, потому что значение степеней принадлежности будут меньше 1, что приведёт к образованию не нормального множества в результате нечеткого вывода.

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

Вход программы:



Вывод программы:



Множества являются нормальными, следовательно, такие случаи возможны.

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

Факт, используемый при выводе, является нормальным множеством.

**Вывод:**

В ходе выполнения данной выполнения лабораторной работы мы получили навыки реализации алгоритмов нечёткой логики, а именно прямого нечёткого логического вывода. 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 нам удалось построить корректные выводы для нескольких случаев, а также дать ответы на контрольные вопросы, прилагающиеся к лабораторной работе.

**Теоретические сведения были взяты из следующих источников:**

1. Логические основы интеллектуальных систем. Практикум: учебно-методическое пособие / В. В. Голенков, В. П. Ивашенко, Д. Г. Колб, К. А. Уваров. – Минск: БГУИР, 2011.
2. Нечеткая логика: алгебраические основы и приложения. Монография / С. Л. Блюмин, И. А. Шуйкова, П. В. Сараев, И. В. Черпаков. – Липецк: ЛЭГИ, 2002. – 111с.