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 Робилко Т. М.

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

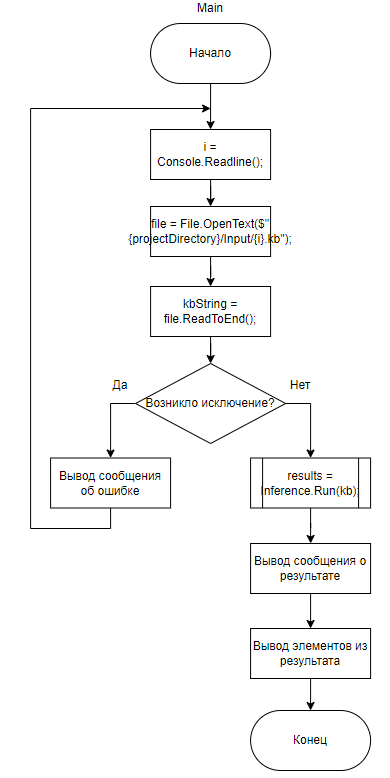
<степень принадлежности> :: = <действительное число с 0 по 1>

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

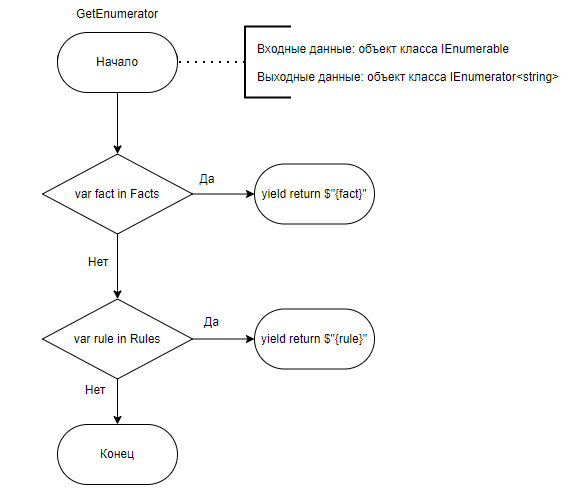
<действительное число с 0 по 1> :: = 0 | <цифра>

**Блок-схемы функций программы:**

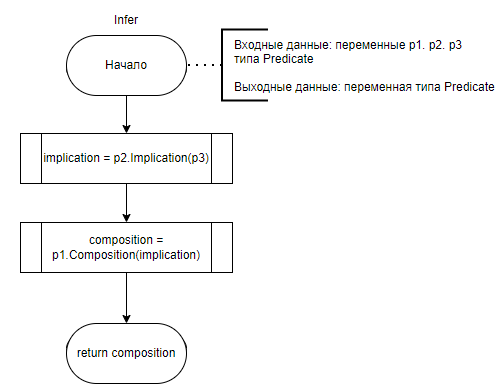
1. Функция Main()



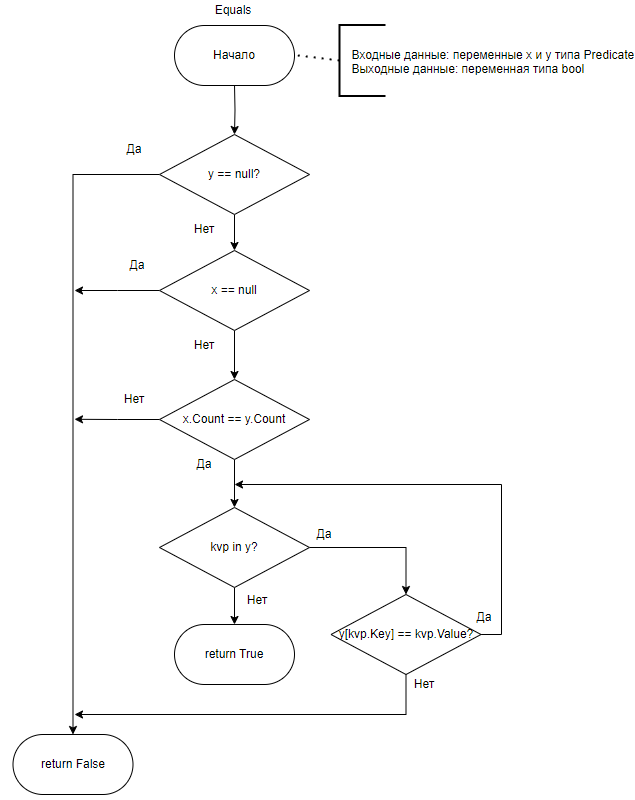
1. Функция GetEnumerator()



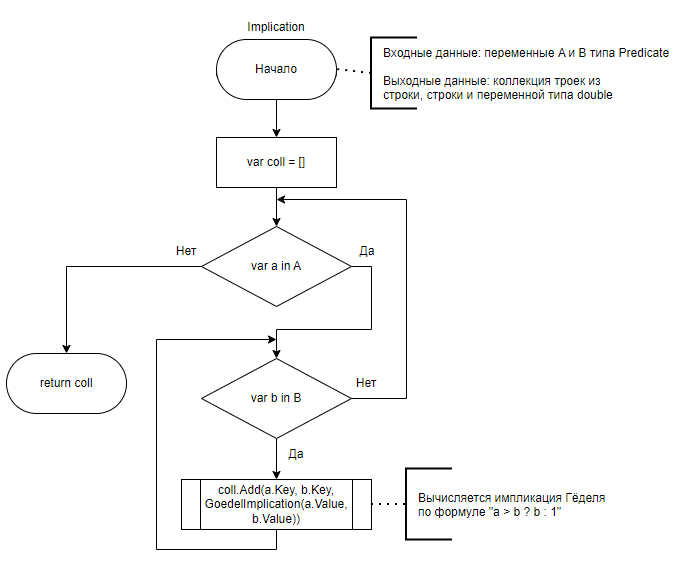
1. Функция Infer()



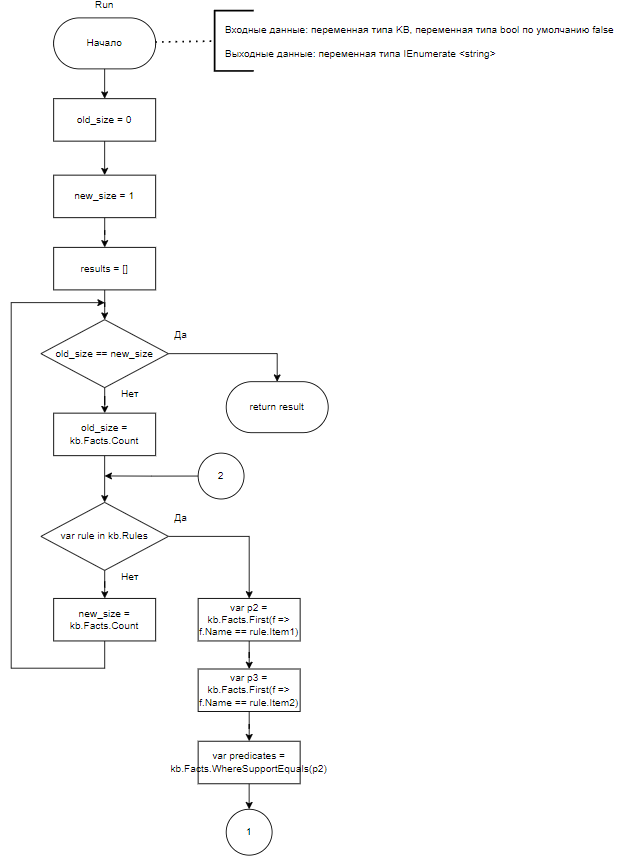
1. Функция Equals()

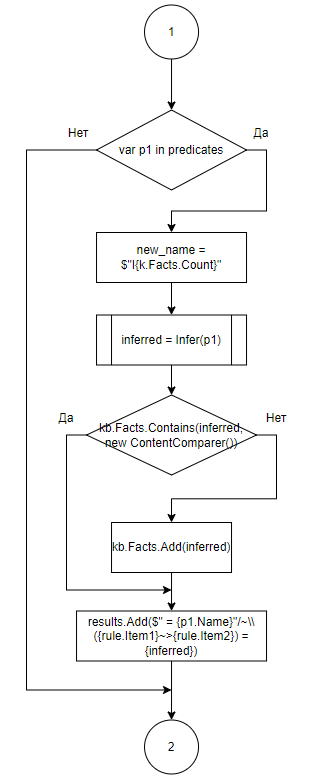


1. Функция Implication()

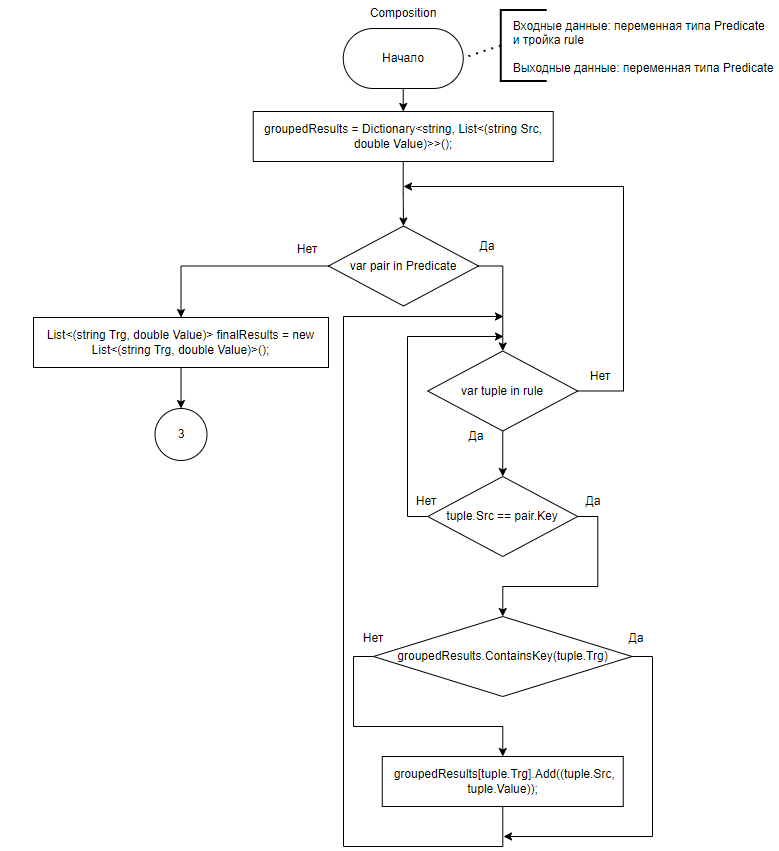


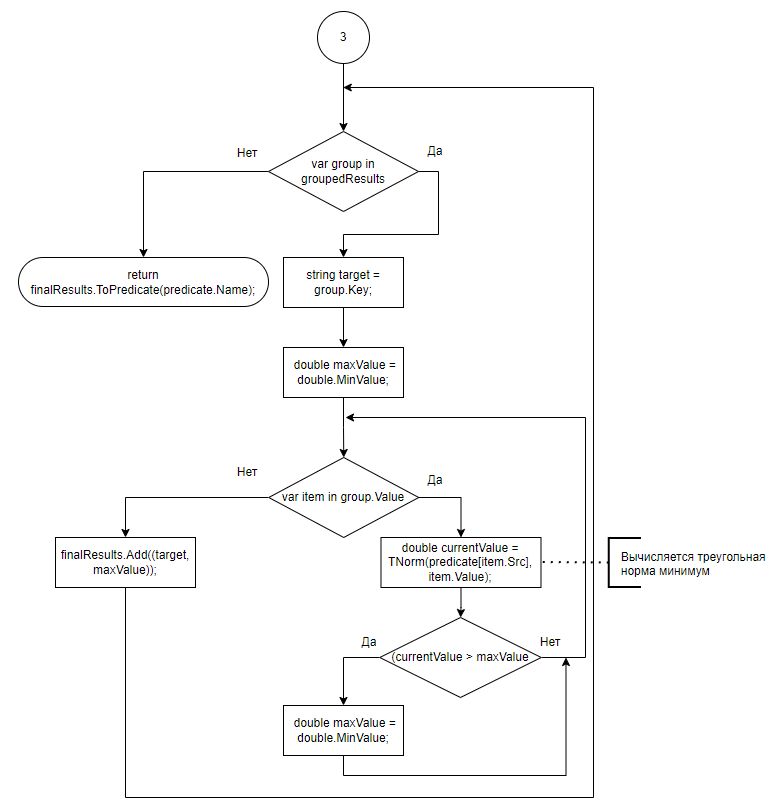
1. Функция Run()





1. Функция Composition()





**Примеры работы программы:**

1. Входные данные:  
   A = {<a, 0.3>, <b, 0.95>}  
   B = {<a, 0.4>, <b, 0.1>}  
   C = {<a, 0.2>, <b, 0.65>}  
   D = {<a, 0.1>, <b, 0.9>, <c,1.0>}  
     
   A~>D  
   C~>B  
   D~>A  
     
   Результат работы:  
   I4 = A/~\(A~>D) = {<a, 0.1>, <b, 0.9>, <c, 0.95>}  
   I5 = B/~\(A~>D) = {<a, 0.1>, <b, 0.4>, <c, 0.4>}  
   I6 = C/~\(A~>D) = {<a, 0.1>, <b, 0.65>, <c, 0.65>}  
   I7 = I5/~\(D~>A) = {<a, 0.3>, <b, 0.4>}  
   I8 = I6/~\(D~>A) = {<a, 0.3>, <b, 0.65>}
2. Входные данные:  
   A = {<a, 0.3>, <b, 0.4>}  
   B = {<a, 0.4>, <b, 0.1>}  
   C = {<a, 0.2>, <b, 0.65>}  
   D = {<a, 0.11>, <b, 0.9>}  
   E = {<a, 0.43>, <b, 0.34>}  
   F = {<a, 0.75>, <b, 0.11>}  
   G = {<a, 0.5>, <b, 0.75>}  
   H = {<a, 0.7>, <b, 0.46>}  
   I = {<a, 0.2>, <b, 0.86>}  
   J = {<a, 0.4>, <b, 0.4>}  
   K = {<a, 0.3>, <b, 0.97>}  
   L = {<a, 0.2>, <b, 0.79>}  
   M = {<a, 0.1>, <b, 0.0>}  
     
   A~>D  
   C~>B  
   D~>A  
   M~>L  
   I~>J  
     
   Результат работы:  
     
   I13 = A/~\(A~>D) = {<a, 0.11>, <b, 0.4>}  
   I14 = C/~\(A~>D) = {<a, 0.11>, <b, 0.65>}  
   I15 = E/~\(A~>D) = {<a, 0.11>, <b, 0.43>}  
   I16 = F/~\(A~>D) = {<a, 0.11>, <b, 0.75>}  
   I17 = H/~\(A~>D) = {<a, 0.11>, <b, 0.7>}  
   I18 = I/~\(A~>D) = {<a, 0.11>, <b, 0.86>}  
   I19 = K/~\(A~>D) = {<a, 0.11>, <b, 0.97>}  
   I20 = L/~\(A~>D) = {<a, 0.11>, <b, 0.79>}  
   I21 = M/~\(A~>D) = {<a, 0.1>, <b, 0.1>}  
   I22 = E/~\(C~>B) = {<a, 0.43>, <b, 0.1>}  
   I23 = F/~\(C~>B) = {<a, 0.75>, <b, 0.1>}  
   I24 = G/~\(C~>B) = {<a, 0.5>, <b, 0.1>}  
   I25 = H/~\(C~>B) = {<a, 0.7>, <b, 0.1>}  
   I26 = E/~\(D~>A) = {<a, 0.43>, <b, 0.43>}  
   I27 = F/~\(D~>A) = {<a, 0.75>, <b, 0.75>}  
   I28 = G/~\(D~>A) = {<a, 0.5>, <b, 0.5>}  
   I29 = H/~\(D~>A) = {<a, 0.7>, <b, 0.7>}  
   I30 = C/~\(M~>L) = {<a, 0.65>, <b, 0.65>}  
   I31 = D/~\(M~>L) = {<a, 0.9>, <b, 0.9>}  
   I32 = I/~\(M~>L) = {<a, 0.86>, <b, 0.86>}  
   I33 = K/~\(M~>L) = {<a, 0.97>, <b, 0.97>}  
   I34 = L/~\(M~>L) = {<a, 0.79>, <b, 0.79>}  
   I35 = I24/~\(A~>D) = {<a, 0.11>, <b, 0.5>}  
   I36 = I30/~\(C~>B) = {<a, 0.65>, <b, 0.1>}  
   I37 = I31/~\(C~>B) = {<a, 0.9>, <b, 0.1>}  
   I38 = I32/~\(C~>B) = {<a, 0.86>, <b, 0.1>}  
   I39 = I33/~\(C~>B) = {<a, 0.97>, <b, 0.1>}  
   I40 = I34/~\(C~>B) = {<a, 0.79>, <b, 0.1>}
3. Входные данные:  
   A = {<x1, 0.0>, <x2, 0.2>, <x3, 0.7>, <x4, 1.0>}  
   B = {<x1, 0.7>, <x2, 0.4>, <x3, 1.0>, <x4, 0.1>}  
     
   A ~> B  
     
   Результат работы:  
   I2 = B/~\(A~>B) = {<x1, 1>, <x2, 0.7>, <x3, 1>, <x4, 0.7>}  
   I3 = I2/~\(A~>B) = {<x1, 1>, <x2, 1>, <x3, 1>, <x4, 1>}

**Контрольные вопросы:**

1. При каких A’(x) можно получить субнормальное нечёткое множество B’(y), когда посылка и заключение нечёткой импликации являются нормальными нечёткими множествами?  
   Пример входных данных программы:  
   A = {<x, 1.0>, <y, 0.1>, <z, 0.3>, <w, 0.6>}  
   B = {<x, 1.0>, <y, 0.8>, <z, 0.2>, <w, 0.2>, <v, 0.0>}  
   C = {<x, 0.3>, <y, 0.0>, <z, 0.3>, <w, 0.0>}  
     
   A~>B  
     
   Вывод программы для данного случая:  
   I3 = C/~\(A~>B) = {<x, 0.3>, <y, 0.3>, <z, 0.2>, <w, 0.2>, <v, 0>}  
     
   Предикаты A, B являются нормальными, в то время как предикат C и результат I3 являются субнормальным.

## При каких A’(x) можно получить нормальное нечёткое множество B’(y), когда посылка и заключение нечёткой импликации являются субнормальными нечёткими множествами?

Пример входных данных программы:

A = {<x, 0.0>, <y, 0.1>, <z, 0.3>, <w, 0.6>}  
B = {<x, 0.1>, <y, 0.8>, <z, 0.2>, <w, 0.0>, <v, 0.0>}  
C = {<x, 0.1>, <y, 1.0>, <z, 0.3>, <w, 0.0>}

A~>B

Вывод программы для данного случая:

I3 = A/~\(A~>B) = {<x, 0.1>, <y, 0.6>, <z, 0.2>, <w, 0>, <v, 0>}

I4 = C/~\(A~>B) = {<x, 1>, <y, 1>, <z, 1>, <w, 0.1>, <v, 0.1>}

Предикаты A и B являются субнормальными, предикат C является

нормальным, результат I4 является нормальным.

## Можно ли получить нечёткое множество B’(y)=B(y), когда посылка и заключение нечёткой импликации являются субнормальными нечёткими множествами? Да, можно.

Пример входных данных программы:

A = {<x, 0.2>, <y, 0.4>, <z, 0.6>}  
B = {<x, 0.5>, <y, 0.5>, <z, 0.5>, <w, 0.5>}  
C = {<x, 0.5>, <y, 0.5>, <z, 0.5>}

A~>B

Вывод программы для данного случая:

I3 = A/~\(A~>B) = {<x, 0.5>, <y, 0.5>, <z, 0.5>, <w, 0.5>}

I4 = C/~\(A~>B) = {<x, 0.5>, <y, 0.5>, <z, 0.5>, <w, 0.5>}

## Можно ли получить нечёткое множество B’(y)=B(y), когда посылка и заключение нечёткой импликации являются нормальными нечёткими множествами? Да, можно.

Пример входных данных программы:

A = {<x, 1.0>, <y, 0.7>, <z, 0.5>}  
B = {<x, 1.0>, <y, 0.5>, <z, 0.3>, <w, 0.7>}  
C = {<x, 1.0>, <y, 0.0>, <z, 0.0>}

A~>B

Вывод программы для данного случая:

I3 = A/~\(A~>B) = {<x, 1.0>, <y, 0.5>, <z, 0.3>, <w, 0.7>}

I4 = C/~\(A~>B) = {<x, 1.0>, <y, 0.5>, <z, 0.3>, <w, 0.7>}

## Можно ли получить нечёткое множество B’(y)=B(y), когда посылка является нормальным нечётким множеством, а заключение – субнормальным? Да, можно.

Пример входных данных программы:

A = {<x, 1.0>, <y, 0.7>, <z, 0.5>}  
B = {<x, 0.2>, <y, 0.5>, <z, 0.3>, <w, 0.7>}  
C = {<x, 1.0>, <y, 0.0>, <z, 0.0>}  
A~>B  
  
Вывод программы для данного случая:  
I3 = A/~\(A~>B) = {<x, 0.2>, <y, 0.5>, <z, 0.3>, <w, 0.7>}  
I4 = C/~\(A~>B) = {<x, 0.2>, <y, 0.5>, <z, 0.3>, <w, 0.7>}

## Можно ли получить нечёткое множество B’(y)=B(y), когда посылка является субнормальным нечётким множеством, а заключение – нормальным? Да, можно.

Пример входных данных программы:

A = {<x, 0.2>, <y, 0.4>, <z, 0.5>}  
B = {<x, 1.0>, <y, 1.0>, <z, 1.0>, <w, 1.0>}  
C = {<x, 1.0>, <y, 0.0>, <z, 0.0>}

A~>B

Вывод программы для данного случая:

I3 = A/~\(A~>B) = {<x, 0.5>, <y, 0.5>, <z, 0.5>, <w, 0.5>}

= C/~\(A~>B) = {<x, 1>, <y, 1>, <z, 1>, <w, 1>}

= A/~\(A~>B) = {<x, 0.5>, <y, 0.5>, <z, 0.5>, <w, 0.5>}

= C/~\(A~>B) = {<x, 1>, <y, 1>, <z, 1>, <w, 1>}

**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.