

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. 321703

Головач В.Д.

Проверил:

В. П. Ивашенко

Минск 2025

ХОД ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ Вариант 3

Тема: Программирование операций обработки и преобразования формул прикладных и неклассических логик.

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

Задание: Реализовать прямой нечеткий логический вывод, используя треугольную норму граничного произведения и нечеткую импликацию Лукасевича.

Описание:

Задача заключается в написании алгоритма прямого нечеткого логического вывода, используя импликацию Лукасевича. Входом программы является файл, содержащий множество нечётких правил и фактов. Для реализации программы использовался язык программирования Python.

Были использованы следующие структуры данных:

- Список
- Множество
- Словарь
- Кортеж

Теоретические сведения:

Правило – формула, которой соответствует вычисляемая операция над конечными последовательностями формул формального языка, которая позволяет от некоторого утверждения, называемого посылкой, перейти к новому утверждению, называемому заключением.

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.

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

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

Формат базы знаний

<база знаний>::= <список фактов>|<список правил><новая строка><список правил>
<список фактов>::= <факт>|<факт><новая строка><список фактов>
<список правил>::= <правило>|<правило><новая строка><список правил>
<факт>::= <имя нечёткого предиката><равенство><нечёткое множество>
<правило>::= <имя нечёткого предиката><нечеткая импликация><имя нечёткого предиката>.
<нечёткое множество>::= <открывающаяся фигурная скобка><список пар нечёткой принадлежности><закрывающаяся фигурная скобка>
<список пар нечёткой принадлежности>::= <пара нечёткой принадлежности>|<пара нечёткой принадлежности><запятая><список пар нечёткой принадлежности>
<пара нечёткой принадлежности>::= <открывающаяся угловая скобка><элемент><запятая><степень принадлежности><закрывающаяся угловая скобка>

<имя нечеткого предиката>::= <имя><открывающаяся полукруглая
 скобка><имя><закрывающаяся полукруглая скобка>
 <элемент>::=<имя>
 <имя>::=<буква>|<буква><символы>
 <символы>::=<символ>|<символ><символы>
 <степень принадлежности>::=<действительное число с 0 по 1>
 <символы>::=<буква>|<цифра>
 <множество>::=<ориентированное множество> |
 <неориентированное множество>
 <неориентированное множество>::={ <список элементов> }
 <ориентированное множество>::=(<элемент>,<список элементов>)
 <список элементов>::=<элемент> ,<элемент> ,<элемент>
 <имя нечёткого множества>::=<имя>
 <действительное число с 0 по 1>::=<единица>|<единица><точка><нули>|<действительное
 число с 0 до 1>
 <действительное число с 0 до 1>::=<ноль>|<ноль><точка><цифры>
 <нули>::=<ноль>|<ноль><нули>
 <цифра>::=<цифра>|<цифра><цифры>
 <символ>::=<буква>|<цифра>
 <цифра>::=0|...|9
 <буква>::=A|...|z
 <единица>::= 1
 <ноль>::= 0
 <точка>::= .
 <запятая>::= ,
 <равенство>::= =
 <нечеткая импликация>::= ~>
 <открывающаяся угловая скобка>::= <
 <закрывающаяся угловая скобка>::= >
 <открывающаяся фигурная скобка>::= {
 <закрывающаяся фигурная скобка>::= }
 <открывающаяся полукруглая скобка>::= (
 <закрывающаяся полукруглая скобка>::=)

Схемы функций программы:

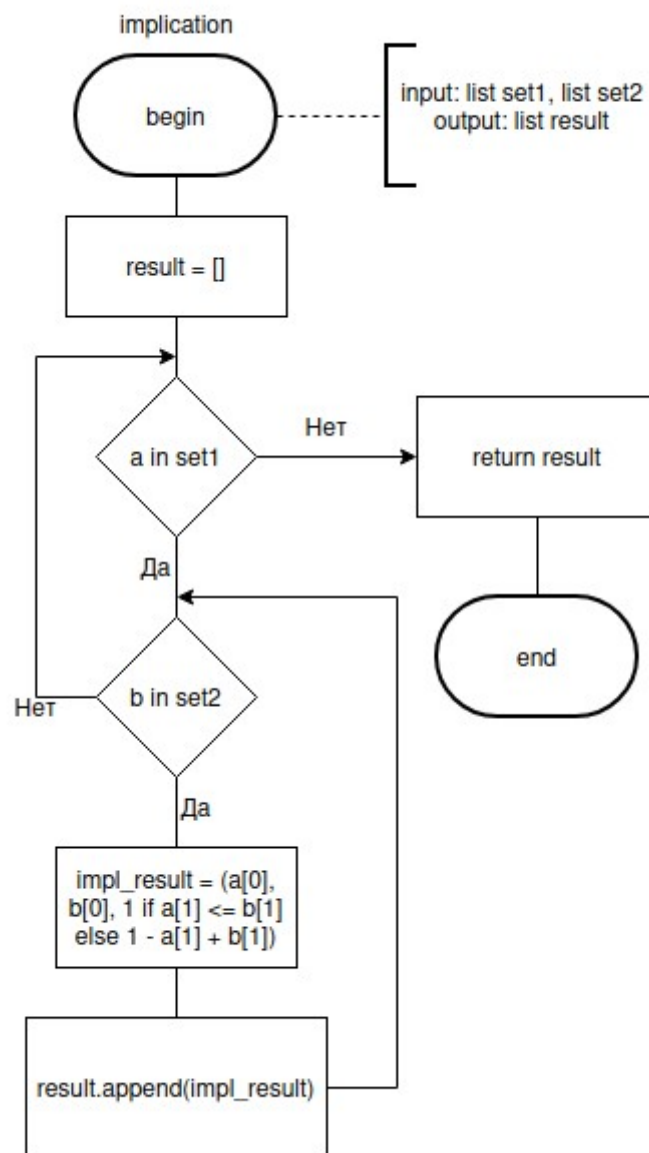


Рис. 1 Функция implication()

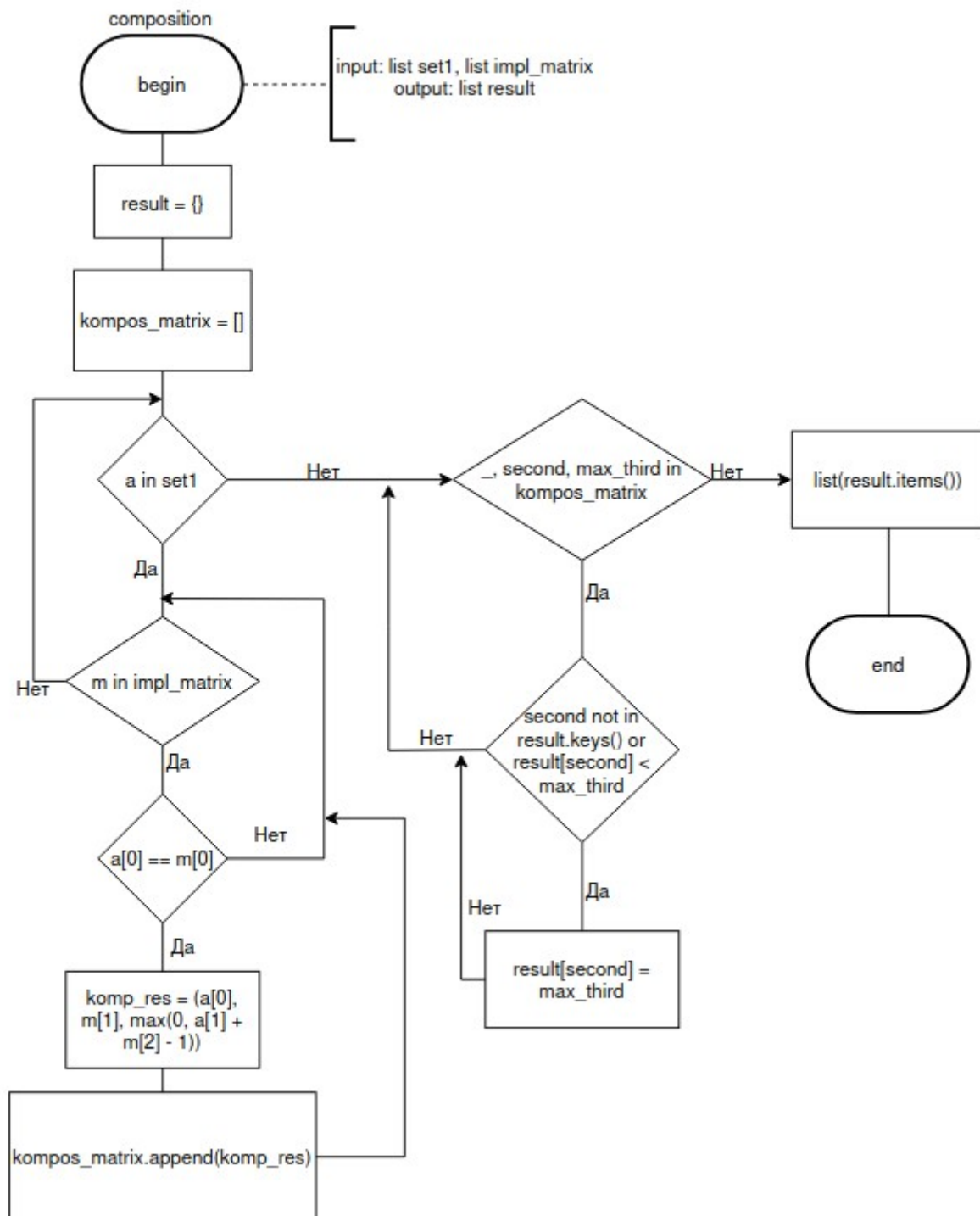


Рис. 2 Функция composition()

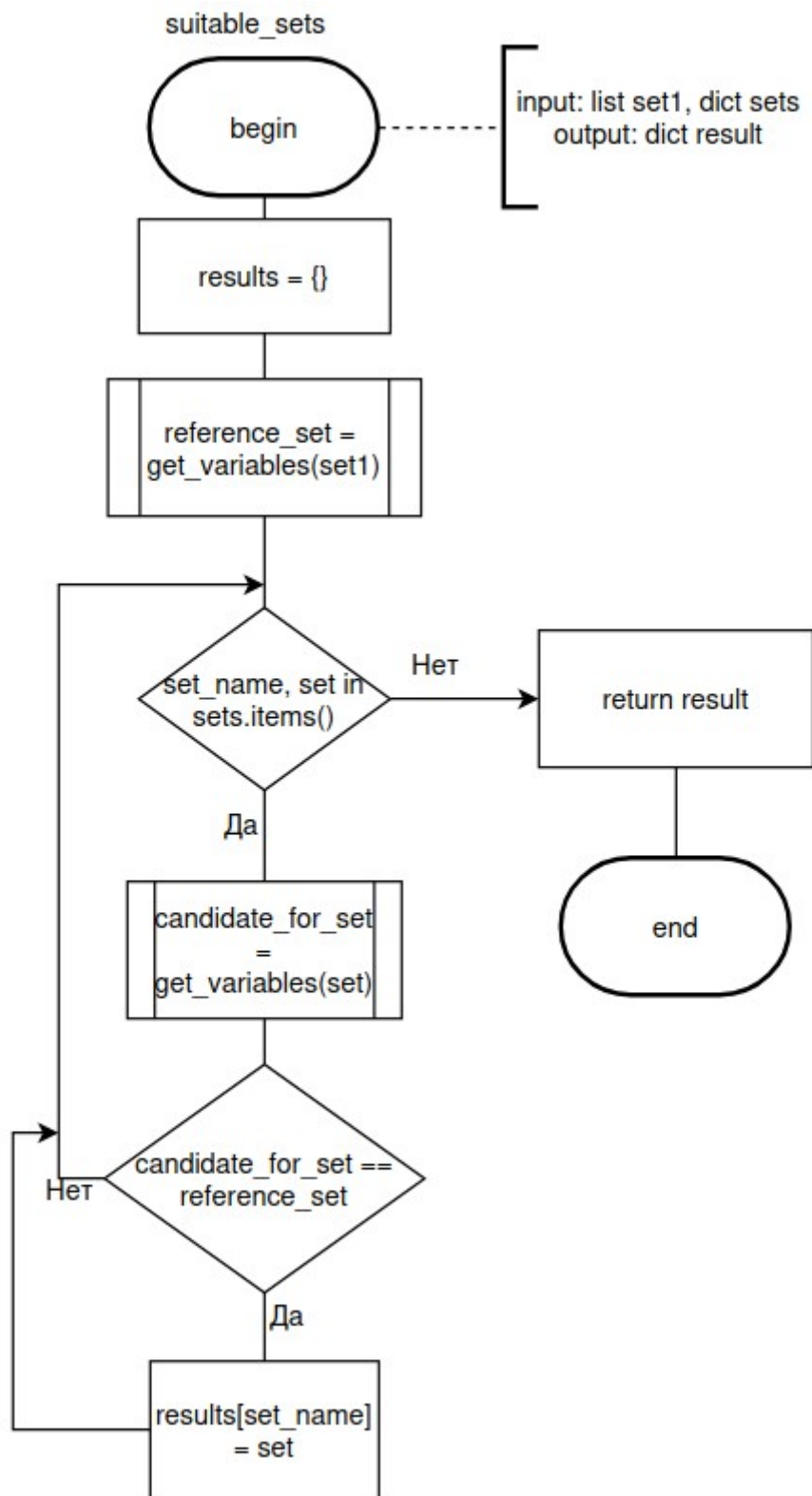


Рис. 3 Функция suitable_sets()

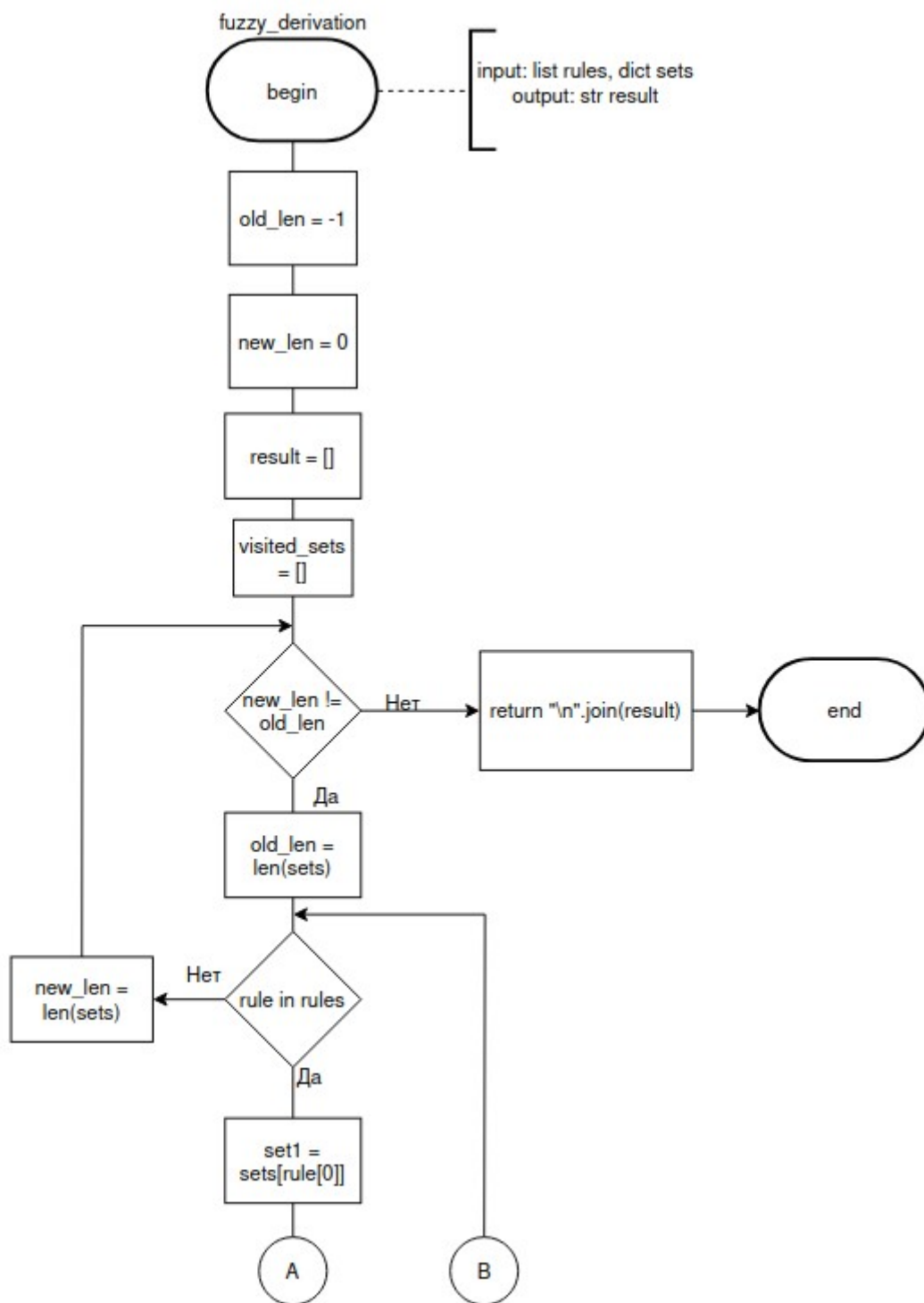


Рис. 4.1 Функция fuzzy_derivation()

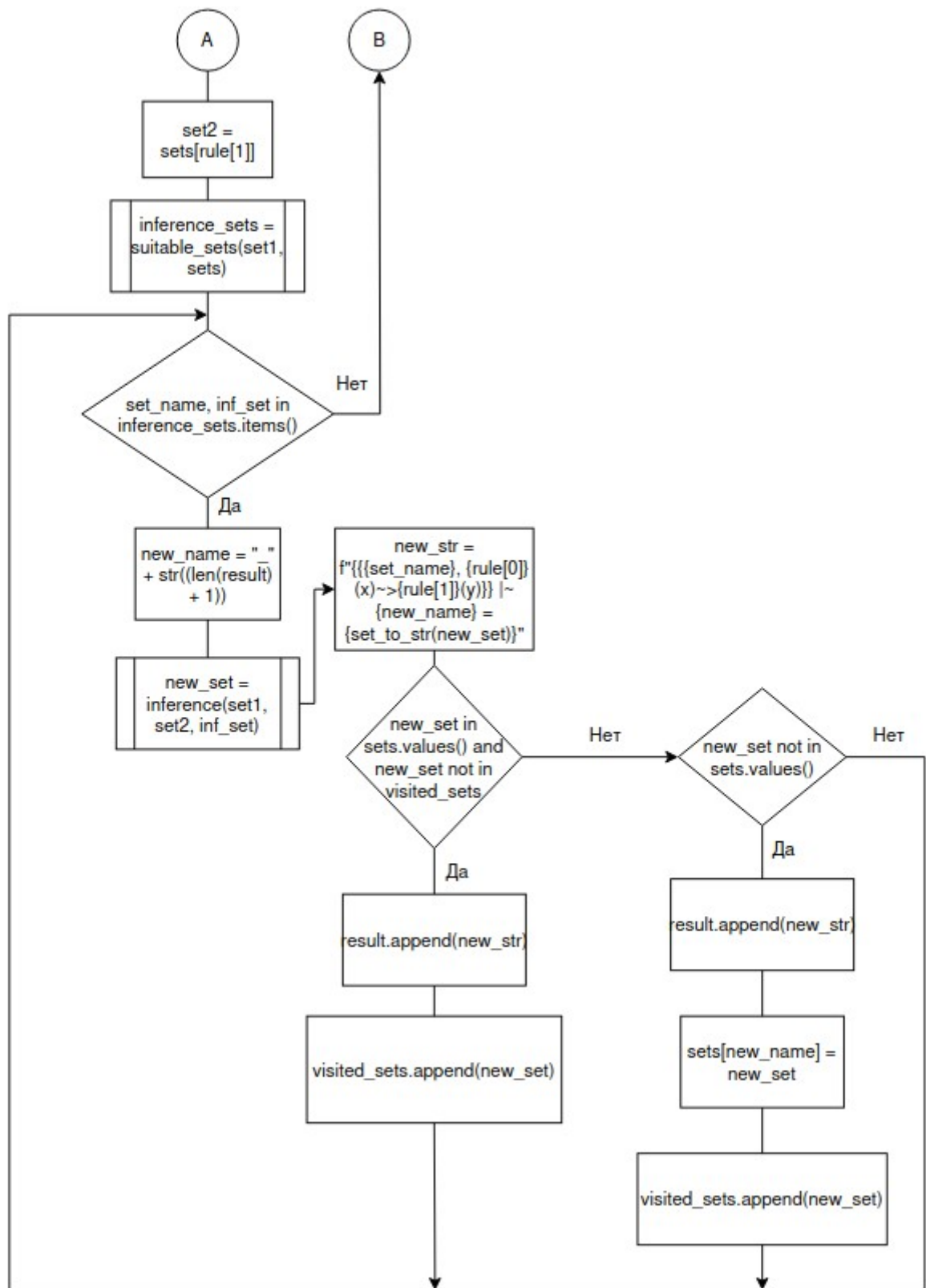


Рис. 4.2 Функция `fuzzy_derivation()`

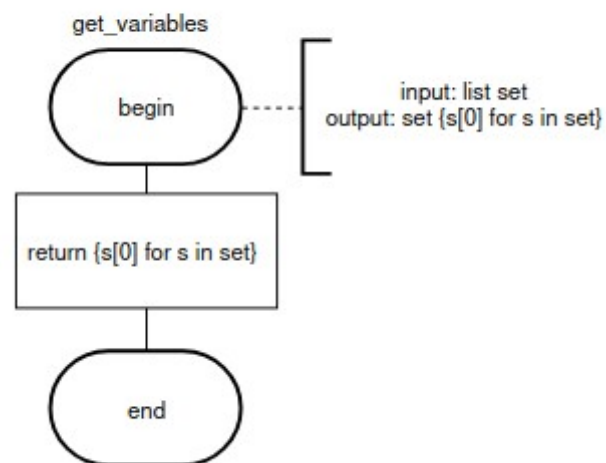


Рис. 5 Функция get_variables()

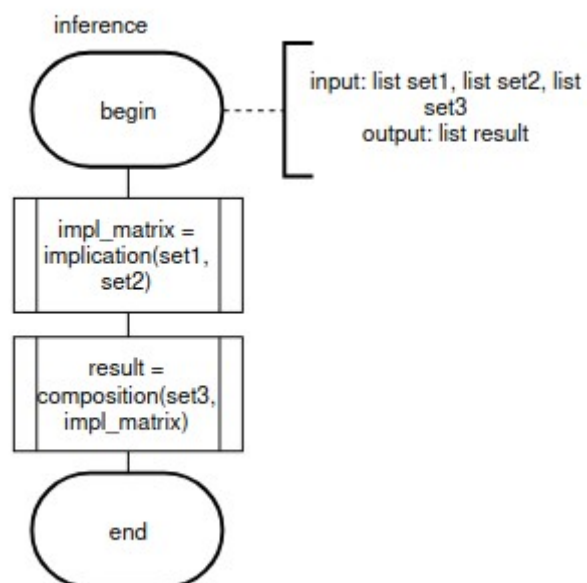


Рис. 6 Функция inference()

Демонстрация результатов работы программы

```
≡ fuzzy_input.txt
1  A = { <x1, 1>, <x2, 0.4>, <x3, 1>}
2  B = {<y1, 0.3>, <y2, 0.4>, <y3, 1>}
3  C = {<x1, 0.1>, <x3,0.9>, <x2, 0>}
4
5  A(x) ~> B(y)
6
```

Рис. 7 Пример 1

```
.py"
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.3>,<y2, 0.4>,<y3, 1.0>}
{C, A(x)~>B(y)} |~ _2 = {<y1, 0.2>,<y2, 0.3>,<y3, 0.9>}
(venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-FA507RE:~/Paf
```

Рис. 8 Результат 1

```
≡ fuzzy_input.txt
1  A = { <x1, 1>, <x2, 0.4>, <x3, 1>}
2  B = {<y1, 0.3>, <y2, 0.4>, <y3, 1>}
3  C = {<x1, 0.1>, <x3,0.9>, <x2, 0>}
4  D = {<z1, 0.8>, <z2, 0.7>}
5
6  A(x) ~> B(y)
7  C(x) ~> A(y)
8  D(x) ~> B(y)
```

Рис. 9 Пример 2

```
.py
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.3>,<y2, 0.4>,<y3, 1.0>}
{C, A(x)~>B(y)} |~ _2 = {<y1, 0.2>,<y2, 0.3>,<y3, 0.9>}
{A, C(x)~>A(y)} |~ _3 = {<x1, 1.0>,<x2, 1.0>,<x3, 1.0>}
{C, C(x)~>A(y)} |~ _4 = {<x1, 0.9>,<x2, 0.4>,<x3, 0.9>}
{D, D(x)~>B(y)} |~ _5 = {<y1, 0.3>,<y2, 0.4>,<y3, 0.8>}
{ _3, A(x)~>B(y)} |~ _6 = {<y1, 0.9>,<y2, 1.0>,<y3, 1.0>}
{ _4, A(x)~>B(y)} |~ _7 = {<y1, 0.3>,<y2, 0.4>,<y3, 0.9>}
{ _4, C(x)~>A(y)} |~ _8 = {<x1, 0.9>,<x2, 0.9>,<x3, 0.9>}
{ _8, A(x)~>B(y)} |~ _9 = {<y1, 0.8>,<y2, 0.9>,<y3, 0.9>}
(venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-FA507RE:~
```

Рис. 10 Результат 2

```
≡ fuzzy_input.txt
1  A = { <x1, 1>, <x1, 0.2>, <x3, 1>}
2  B = {<y1, 0.3>, <y2, 0.4>, <y3, 1>}
3  C = {<x1, 0.1>, <x3,0.9>, <x2, 0>}
4
5  A(x) ~> B(y)
6  C(x) ~> A(y)
7
8
```

Рис. 11 Пример 3

```

.py
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.3>, <y2, 0.4>, <y3, 1.0>}
{C, C(x)~>A(y)} |~ _2 = {<x1, 0.2>, <x3, 0.9>}
{ _2, A(x)~>B(y)} |~ _3 = {<y1, 0.2>, <y2, 0.3>, <y3, 0.9>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-

```

Рис. 12 Результат 3

Ответы на вопросы:

1. при каких $A'(x)$ можно получить субнормальное нечеткое множество $B'(y)$ когда посылка и заключение нечеткой импликации являются нормальными нечеткими множествами.

Ответ: $B'(y)$ субнормально, когда $A'(x)$ субнормально во всех точках x , где значения принадлежности $A(x)$ и $B(y)$ максимальны (рис.13)

```

39 A = {<x1, 1>, <x2, 1>}
40 B = {<y1, 1>, <y2, 1>}
41
42 X = {<x1, 0.6>, <x2, 0.8>}
43
44 A(x) ~> B(y)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
{A, A(x)~>B(y)} |~ _1 = {<y1, 1.0>, <y2, 1.0>}
{X, A(x)~>B(y)} |~ _2 = {<y1, 0.8>, <y2, 0.8>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-

```

Рис. 13 Ответ на вопрос 1

2. при каких $A'(x)$ можно получить нормальное нечеткое множество $B'(y)$ когда посылка и заключение нечеткой импликации являются субнормальными нечеткими множествами

Ответ: Для получения нормального $B'(y)$ необходимо и достаточно, чтобы высота $A'(x)$ была равна 1 и нашлась такая точка x со степенью принадлежности 1, где значение принадлежности посылки $A(x)$ не превосходит значения заключения $B(y)$ для некоторого y (рис.14)

```

38
39 A = {<x1, 0.3>, <x2, 0.8>}
40 B = {<y1, 0.1>, <y2, 0.9>}
41
42 X = {<x1, 0.6>, <x2, 1>}
43
44 A(x) ~> B(y)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.1>, <y2, 0.8>}
{X, A(x)~>B(y)} |~ _2 = {<y1, 0.4>, <y2, 1.0>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-

```

Рис. 14 Ответ на вопрос 2

3. можно ли получить $B'(y) = B(y)$ когда посылка и заключения нормальные?

Ответ: Такое возможно только в одном частном случае: если $A'(x) = A(x)$ (рис.15)

```
37
38 A = {<x1, 1>, <x2, 0.8>}
39 B = {<y1, 1>, <y2, 0.5>}
40
41 X = {<x1, 1>, <x2, 0.8>}
42
43 A(x) ~> B(y)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
{A, A(x)~>B(y)} |~ _1 = {<y1, 1.0>, <y2, 0.5>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
```

Рис. 15 Ответ на вопрос 3

4. можно ли получить $B'(y) = B(y)$ когда посылка и заключения субнормальные?

Ответ: Да, получить $B'(y) = B(y)$ при субнормальных посылке и заключении можно в частном случае, если $A'(x) = A(x)$ (рис.16)

```
39
40 A = {<x1, 0.1>, <x2, 0.8>}
41 B = {<y1, 0.5>, <y2, 0.5>}
42
43 X = {<x1, 0.1>, <x2, 0.8>}
44
45 A(x) ~> B(y)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.5>, <y2, 0.5>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
```

Рис. 16 Ответ на вопрос 4

5. можно ли получить $B'(y) = B(y)$ когда посылка нормальное нечеткое множество а заключение субнормальное?

Ответ: Да, если $A'(x) = A(x)$ (рис.17)

```
39
40 A = {<x1, 1>, <x2, 0.8>}
41 B = {<y1, 0.5>, <y2, 0.5>}
42
43 X = {<x1, 1>, <x2, 0.8>} |
44
45 A(x) ~> B(y)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
{A, A(x)~>B(y)} |~ _1 = {<y1, 0.5>, <y2, 0.5>}
(.venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
```

Рис. 17 Ответ на вопрос 5

6. можно ли получить $B'(y) = B(y)$ когда посылка субнормальное нечеткое множество а заключение нормальное?

Ответ: Нет, поскольку максимальное значение $B'(y)$ будет ограничено максимальным значением принадлежности $A(x)$, которое меньше 1, а $B(y)$ достигает 1. (рис.18)

```
40 A = {<x1, 0.1>, <x2, 0.8>}
41 B = {<y1, 1>, <y2, 0.5>}
42
43 X = {<x1, 0.1>, <x2, 0.8>}
44
45 A(x) ~> B(y)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
{A, A(x)~>B(y)} |~ 1 = {<y1, 0.8>, <y2, 0.5>}
(venv) artem@artem-ASUS-TUF-Gaming-A15-FA507RE-
```

Рис. 18 Ответ на вопрос 6

ЛИЧНЫЙ ВКЛАД

В ходе выполнения данной лабораторной работы разработал описание реализации со схемами (блок-схемами) алгоритмов

ВЫВОД

В процессе выполнения лабораторной работы, мы получили 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 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. <https://github.com/rastsislaux>