МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)

Дисциплина  
**«Системы искусственного интеллекта»**

Отчет к лабораторной работе № 3

Вариант 1

Студенты: Гижевская В.Д. и Борисова А.В.  
Группы: 6313-020302D и 6314-020302D

Преподаватель: Додонова Н.Л.

Самара 2020

**Основные понятия**

**Система нечеткого вывода** – это процесс получения нечетких заключений о требуемом управлении объектом на основе нечетких условий или предпосылок, представляющих собой информацию о текущем состоянии объекта.

Этот процесс соединяет в себе все основные концепции теории нечетких множеств: функции принадлежности, лингвистические переменные, методы нечеткой импликации и т.п. Разработка и применение систем нечеткого вывода включает в себя ряд этапов, реализация которых выполняется на основе рассмотренных ранее положений нечеткой логики

Типовую структуру системы нечеткого вывода можно представить следующим образом

1

0

*Х*=

***V1***

***V2***

***Vn***

**Фаззификатор**

**Функции принадлежности**

**Машина нечеткого вывода**

**База знаний**

**Дефаззификатор**

**Степень принадлежности**

**Основные этапы нечеткого логического вывода**

1 Фаззификация входных и выходных переменных

2 Построение базы знаний

3 Агрегирование подусловий в нечетком правиле и активизация (композиция) заключений

4 Аккумулирование заключений нечетких правил

5 Дефаззификация

**База правил систем нечеткого вывода** предназначена для формального представления эмпирических знаний экспертов в той или иной предметной области в форме нечетких продукционных правил. Таким образом, база нечетких продукционных правил системы нечеткого вывода – это система нечетких продукционных правил, отражающая знания экспертов о методах управления объектом в различных ситуациях, характере его функционирования в различных условиях и т.п., т.е. содержащая формализованные человеческие знания.

**Фаззификация (введение нечеткости)** – это установка соответствия между численным значением входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей терма лингвистической переменной. На этапе фаззификации значениям всех входным переменным системы нечеткого вывода, полученным внешним по отношению к системе нечеткого вывода способом, например, при помощи датчиков, ставятся в соответствие конкретные значения функций принадлежности соответствующих лингвистических термов, которые используются в условиях (антецедентах) ядер нечетких продукционных правил, составляющих базу нечетких продукционных правил системы нечеткого вывода.

**Агрегирование** – это процедура определения степени истинности условий по каждому из правил системы нечеткого вывода. При этом используются полученные на этапе фаззификации значения функций принадлежности термов лингвистических переменных, составляющих вышеупомянутые условия (антецеденты) ядер нечетких продукционных правил.

**Активизация в системах нечеткого** вывода – это процедура или процесс нахождения степени истинности каждого из элементарных логических высказываний (подзаключений), составляющих консеквенты ядер всех нечетких продукционных правил. Поскольку заключения делаются относительно выходных лингвистических переменных, то степеням истинности элементарных подзаключений при активизации ставятся в соответствие элементарные функции принадлежности.

**Аккумуляция (или аккумулирование) в системах нечеткого вывода** – это процесс нахождения функции принадлежности для каждой из выходных лингвистических переменных. Цель аккумуляции состоит в объединении всех степеней истинности подзаключений для получения функции принадлежности каждой из выходных переменных.

**Дефаззификация в системах нечеткого вывода** – это процесс перехода от функции принадлежности выходной лингвистической переменной к её четкому (числовому) значению. Цель дефаззификации состоит в том, чтобы, используя результаты аккумуляции всех выходных лингвистических переменных, получить количественные значения для каждой выходной переменной, которое используется внешними по отношению к системе нечеткого вывода устройствами (исполнительными механизмами интеллектуальной САУ).

**Задание:**

Разработать систему нечеткого вывода.

Терм-множество каждой входной и выходной переменной должно содержать не менее 3-5 элементов. База правил нечеткого вывода — не менее 5-7 правил.

Для обеих входных лингвистических переменных исследовать зависимость принимаемого решения от одной входной лингвистической переменной при фиксированном значении другой.

|  |  |
| --- | --- |
| ***Входные лингвистические переменные*** | ***Выходная лингвистическая переменная*** |
| Влажность воздуха,  Температура воздуха | Погода |

**Решение задачи:**

Терм-множества:

* Температура = {холодно, тепло, жарко}
* Влажность = {низкая, оптимальная, высокая}
* Погода = {неприятная, комфортная, приятная}

Обозначим границы значений выходных и выходных переменных:

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | 1 | 2 | 3 |
| Температура | [0;10) | [10;25] | (25;40] |
| Влажность | [40;50) | [50;70] | (70;90] |
| Погода | [0;4) | [4;9] | (9;12] |

Построим графики для множеств:

1. Температура

* Холодно

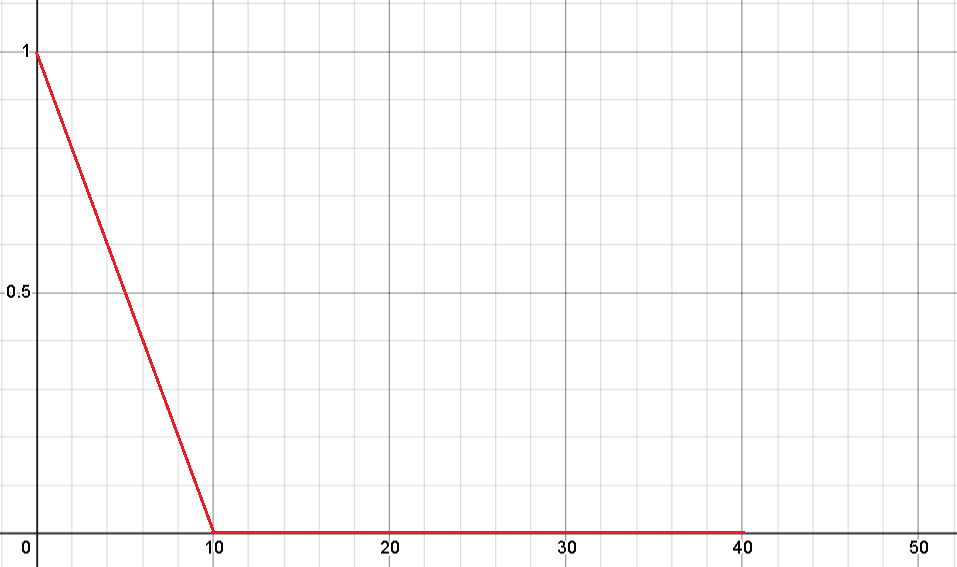


Рисунок 1- График холодной температуры

* Тепло

Рисунок 2- График тёплой температуры

* Жарко

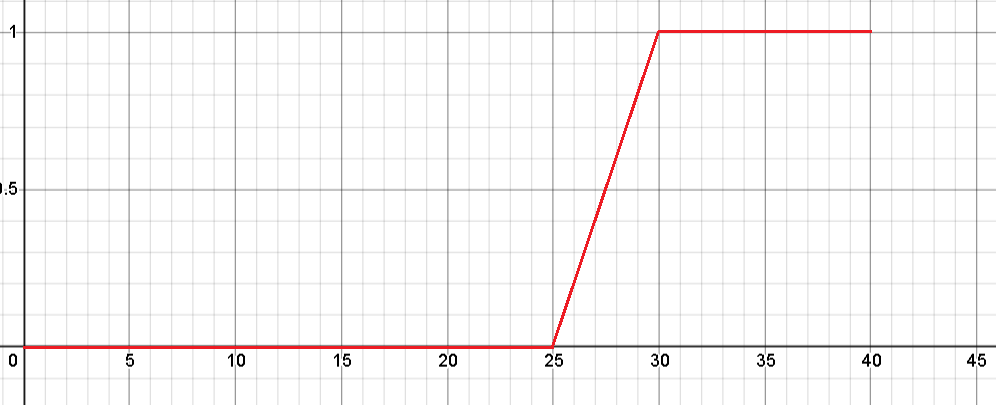


Рисунок 3- График жаркой температуры

1. Влажность

* Низкая

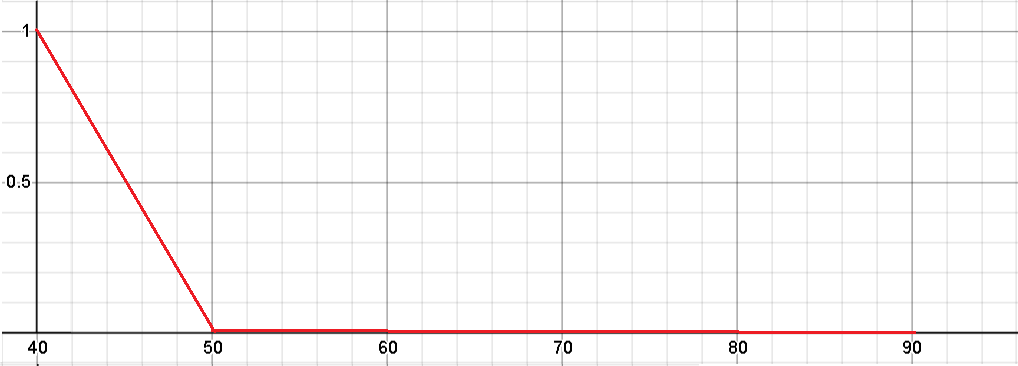


Рисунок 4- График низкой влажности

* Оптимальная

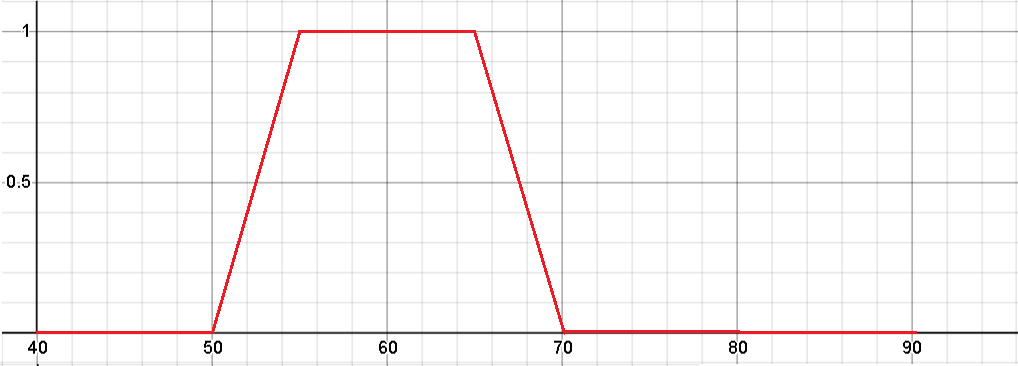


Рисунок 5- График оптимальной влажности

* Высокая

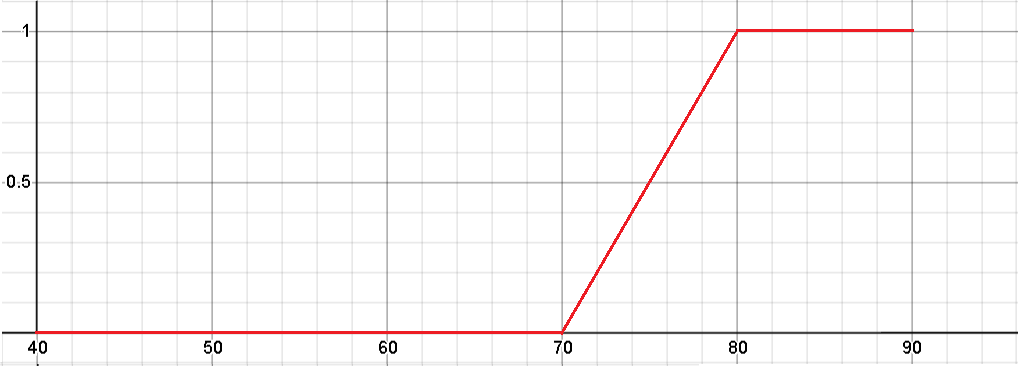
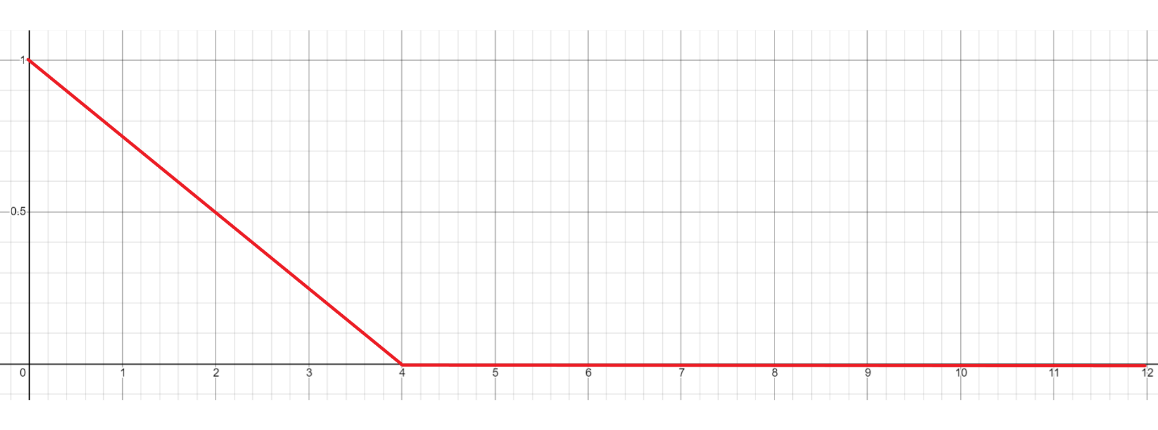


Рисунок 6- График высокой влажности

1. Погода

* Неприятная

Рисунок 7- График неприятной погоды

* Комфортная

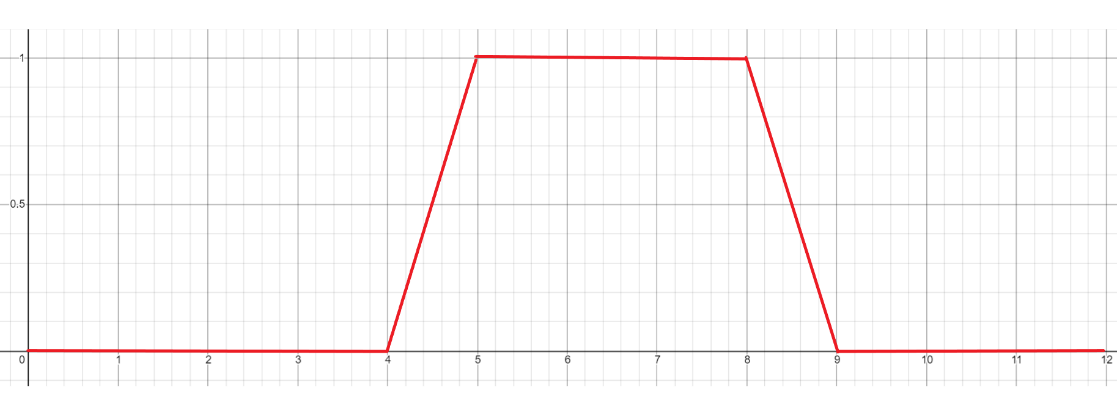


Рисунок 8- График комфортной погоды

* Приятная

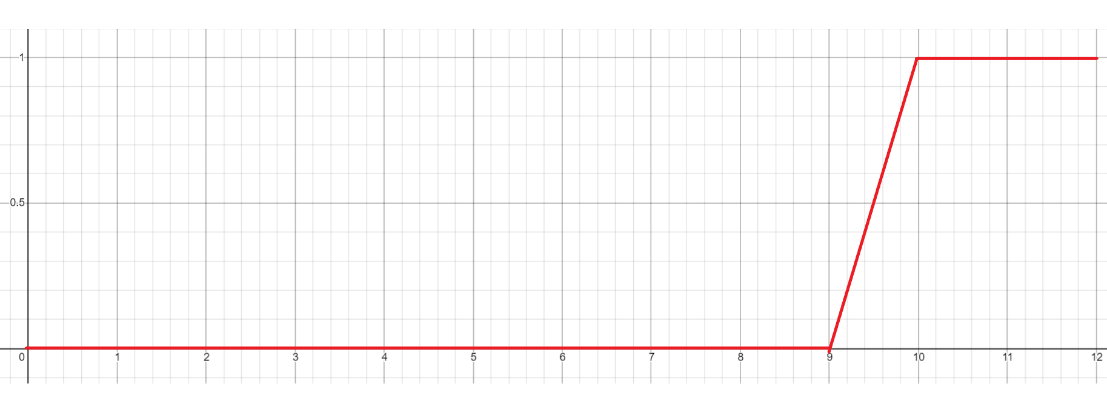


Рисунок 9- График приятной погоды

**Пример работы программы:**

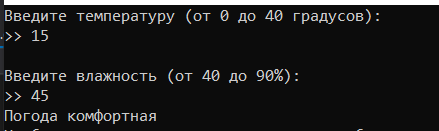


Рисунок 10 - Результат выполнения программы для комфортной погоды

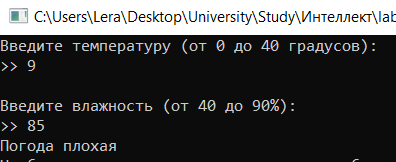


Рисунок 11 - Результат выполнения программы для неприятной погоды

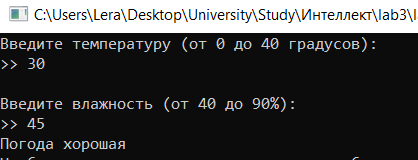


Рисунок 12 - Результат выполнения программы для приятной погоды

**Код программы**

Program.cs

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespace lab3

{

classProgram

{

publicstaticvoid Input(outint x0, outint y0)

{

Console.Clear();

Console.Write("Введите температуру (от 0 до 40 градусов): \n>> ");

x0 = Convert.ToInt32(Console.ReadLine());

Console.Write("\nВведите влажность (от 40 до 90%): \n>> ");

y0 = Convert.ToInt32(Console.ReadLine());

}

publicstaticvoid Task()

{

//Console.WriteLine(");

while (true)

{

try

{

int x0, y0;

Input(out x0, out y0);

if (x0 < 0 || x0 > 40 || y0 < 40 || y0 > 90)

{

Console.WriteLine("Ошибка. Температура должна быть в диапазонe от 0 до 40. Влажность от 40 до 90");

Console.WriteLine("\nЧтобы ввести новые значения, нажмите любую клавишу.");

Console.ReadKey();

Input(out x0, out y0);

}

function fi = new function(x0, y0);

fi.Result();

Console.WriteLine("Чтобыввестиновыезначения, нажмителюбуюклавишу.");

Console.ReadLine();

Console.WriteLine("");

}

catch (Exception e)

{

Console.WriteLine("Нужновводитьчисла!");

}

}

}

staticvoid Main(string[] args)

{

Task();

}

}

}

Function.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3

{

classfunction

{

int x0, y0;

publicfunction(int x0, int y0)

{

this.x0 = x0;

this.y0 = y0;

}

//температура

publicdouble Cold()//холодно

{

if (x0 >= 0 & x0 <= 10) return (-0.1) \* x0 + 1;

if (x0 > 10) return 0;

elsereturn 0;

}

publicdouble Warm()//тепло

{

if (x0 < 10 || x0 > 25) return 0;

if (x0 >= 15 && x0 <= 20) return 1;

if (x0 > 10 && x0 < 15) return 0.2 \* x0 - 2;

if (x0 > 20 && x0 <= 25) return (-0.2) \* x0 + 5;

return 0;

}

publicdouble Hot()//жарко

{

if (x0 < 25) return 0;

if (x0 > 25 && x0 <= 40) return 0.2 \* x0 - 5;

if (x0 > 30) return 1;

return 0;

}

//влажность

publicdouble Low()//низкаявлажность

{

if (y0 >= 40 && y0 <= 50) return (-0.1) \* y0 + 5;

if (y0 > 50) return 0;

return 0;

}

publicdouble Middle()//средняя/оптимальнаявлажность

{

if (y0 < 50 || y0 > 70) return 0;

if (y0 <= 65 && y0 >= 55) return 1;

if (y0 > 50 && y0 < 55) return 0.2 \* y0 - 10;

if (y0 > 65 && y0 <= 70) return (-0.2) \* y0 + 14;

return 0;

}

publicdouble High()//высокаявлажность

{

if (y0 < 70) return 0;

if (y0 >= 80 & y0 <= 90) return 1;

if (y0 > 70 & y0 < 80) return (0.1) \* y0 - 7;

return 0;

}

privateint Weather\_Comfort()

{

if(!(Hot() == 0) && !(Low() == 0)) return 1;

if (!(Cold() == 0) && !(Middle() == 0)) return 1;

if (!(Warm() == 0) && !(High() == 0)) return 1;

elsereturn 0;

}

privateint Weather\_Good()

{

if (!(Warm() == 0) && !(Middle() == 0)) return 1;

if (!(Warm() == 0) && !(Low() == 0)) return 1;

if (!(Hot() == 0) && !(Middle() == 0)) return 1;

elsereturn 0;

}

privateint Weather\_Bad()

{

if (!(Cold() == 0) && !(High() == 0)) return 1;

if (!(Cold() == 0) && !(Low() == 0)) return 1;

if (!(Hot() == 0) && !(High() == 0)) return 1;

elsereturn 0;

}

publicvoid Result()

{

if (Weather\_Bad() == 1) Console.WriteLine("Погодаплохая");

if (Weather\_Good() == 1) Console.WriteLine("Погодакомфортная");

if (Weather\_Comfort() == 1) Console.WriteLine("Погодахорошая");

}

}

}