

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 27.01.2024 12:02:46  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

## МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра биомедицинской инженерии

Утверждаю  
Проректор по учебной работе  
О.Г. Локтионова  
« 25 » 09 2023 г.



### ИНТЕЛЛЕКТУАЛЬНАЯ ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ В БИОТЕХНИЧЕСКИХ СИСТЕМАХ

Методические рекомендации по выполнению самостоятельных работ  
для студентов направления подготовки 12.04.04 – «Биотехнические  
системы и технологии» (магистр)

Курск 2023

УДК 621.(076.1)

Составители: А.А.Кузьмин

Рецензент:

Кандидат технических наук, доцент *Т.Н. Конаныхина*

Интеллектуальная поддержка принятия решений в биотехнических системах: методические рекомендации по выполнению самостоятельных работ для студентов направления подготовки 12.04.04 – «Биотехнические системы и технологии» (магистр) / Юго-Зап. гос. ун-т; сост.: А.А.Кузьмин. - Курск, 2023. - 50 с.

Содержат методические рекомендации к проведению самостоятельных работ по дисциплине «Интеллектуальная поддержка принятия решений в биотехнических системах». Методические указания по структуре, содержанию и стилю изложения материала соответствуют методическим и научным требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для студентов направления подготовки 12.04.04 – «Биотехнические системы и технологии» (магистр)

Текст печатается в авторской редакции

Подписано в печать 25.09.23 Формат 60x84 1/16

Усо.печ.л. 2,9. Уч.-изд.л. 2,6. Тираж 30 экз. Заказ: 1088. Бесплатно.

Юго-Западный государственный университет.

305040. г. Курск, ул. 50 лет Октября, 94.

## Работа 1

### Управляющие системы на основе нечеткой логики

**Цель работы:** знакомство с использованием пакета Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab и приобретение навыков по построению нечетких систем управления.

#### Краткие теоретические сведения

В последнее время нечеткая технология завоевывает все больше сторонников среди разработчиков систем управления. Термин «fuzzy» так вошел в жизнь, что на многих языках он даже не переводится. В России в качестве примера можно вспомнить рекламу стиральных машин и микроволновых печей фирмы Samsung, обладающих искусственным интеллектом на основе нечеткой логики.

Тем не менее, столь масштабный скачок в развитии нечетких систем управления не случаен. Простота и дешевизна их разработки заставляет проектировщиков все чаще прибегать к этой технологии.

Нечеткая логика основана на использовании таких оборотов естественного языка, как «далеко», «близко», «холодно», «горячо». Диапазон ее применения очень широк - от бытовых приборов до управления сложными промышленными процессами. Многие современные задачи управления просто не могут быть решены классическими методами из-за очень большой сложности математических моделей, их описывающих. Вместе с тем, чтобы использовать теорию нечеткости на цифровых компьютерах, необходимы математические преобразования, позволяющие перейти от лингвистических переменных к их числовым аналогам в ЭВМ.

Классические методы управления хорошо работают при полностью детерминированном объекте управления и детерминированной среде, а для систем с неполной информацией и высокой сложностью объекта управления оптимальными являются нечеткие методы управления.

Логика известна нам как строгая и сугубо теоретическая наука, и большинство ученых (кроме разработчиков последнего поколения компьютеров) продолжают придерживаться этого мнения. Вместе с тем классическая или булева логика имеет один существенный недостаток - с ее помощью невозможно описать ассоциативное мышление человека. Классическая логика оперирует только двумя понятиями: ИСТИНА и ЛОЖЬ,

исключая любые промежуточные значения. Аналогично этому булева логика не признает ничего кроме единиц и нулей. Все это хорошо для вычислительных машин, но недостаточно эффективно при проектировании и эксплуатации систем управления БТС.

Решить эту проблему и призвана нечеткая логика. С термином «лингвистическая переменная» можно связать любую физическую величину, для которой нужно иметь больше значений, чем только ДА и НЕТ. В этом случае вы определяете необходимое число термов и каждому из них ставите в соответствие некоторое значение описываемой физической величины. Для этого значения степень принадлежности физической величины к терму будет равна единице, а для всех остальных значений - в зависимости от выбранной функции принадлежности.

Получившие наибольшее развитие из всех разработок искусственного интеллекта, экспертные системы завоевали устойчивое признание в качестве систем поддержки принятия решений. Одним из основных методов представления знаний в экспертных системах являются продукционные правила, позволяющие приблизиться к стилю мышления человека. Любое правило продукций состоит из посылок и заключения. Главным же недостатком продукционных систем остается то, что для их функционирования требуется наличие полной информации о системе.

Нечеткие системы тоже основаны на правилах продукционного типа, однако в качестве посылки и заключения в правиле используются лингвистические переменные, что позволяет избежать ограничений, присущих классическим продукционным правилам.

Целевая установка процесса управления связывается с выходной переменной нечеткой системы управления, но результат нечеткого логического вывода является нечетким, а физическое исполнительное устройство не способно воспринять такую команду. Необходимы специальные математические методы, позволяющие переходить от нечетких значений величин к вполне определенным. В целом весь процесс нечеткого управления можно разбить на несколько шагов: фаззификация, разработка нечетких правил и дефаззификация.

### ***Фаззификация (переход к нечеткости)***

Точные значения входных переменных преобразуются в значения лингвистических переменных посредством применения некоторых положений теории нечетких множеств, а именно - при помощи определенных функций принадлежности.

### ***Лингвистические переменные***

В нечеткой логике значения любой величины представляются не числами, а словами естественного языка и называются ТЕРМАМИ. Так, значением лингвистической переменной ДИСТАНЦИЯ являются термы ДАЛЕКО, БЛИЗКО и т. д.

### ***Функции принадлежности***

Как уже говорилось, принадлежность каждого точного значения к одному из термов лингвистической переменной определяется посредством функции принадлежности. Ее вид может быть абсолютно произвольным. Стандартные функции принадлежности представлены на рисунке 2.



Рисунок 2. – типовые функции принадлежности

Стандартные функции принадлежности легко применимы к решению большинства задач.

### ***Разработка нечетких правил***

На этом этапе определяются продукционные правила, связывающие лингвистические переменные. Совокупность таких правил описывает стратегию управления, применяемую в данной задаче. Большинство нечетких систем используют продукционные правила для описания зависимостей между лингвистическими переменными. Типичное продукционное правило состоит из антецедента (часть ЕСЛИ ...) и консеквента (часть ТО ...). Антецедент может содержать более одной посылки. В этом случае они объединяются посредством логических связок И или ИЛИ. Процесс вычисления нечеткого правила называется нечетким логическим выводом и подразделяется на два этапа: обобщение и заключение.

### ***Дефаззификация (устранение нечеткости)***

На этом этапе осуществляется переход от нечетких значений величин к определенным физическим параметрам, которые могут служить командами исполнительному устройству. Результат нечеткого вывода, конечно же, будет нечетким.

Для устранения нечеткости окончательного результата существует несколько методов. Рассмотрим некоторые из них.

### **Метод центра максимума (CoM)**

Так как результатом нечеткого логического вывода может быть несколько термов выходной переменной, то правило дефаззификации должно определить, какой из термов выбрать.

### **Метод наибольшего значения (MoM)**

При использовании этого метода правило дефаззификации выбирает максимальное из полученных значений выходной переменной.

### **Метод центроида (CoA)**

В этом случае окончательное значение определяется как проекция центра тяжести фигуры, ограниченной функциями принадлежности выходной переменной с допустимыми значениями.

## **Основные шаги разработки нечеткой системы управления с использованием САД-системы fuzzy ТЕСН 3.0**

Процесс разработки проекта нечеткой системы управления на fuzzy ТЕСН разбивается, как уже говорилось, на четыре основных этапа. Все они схематично показаны на рисунке 3.

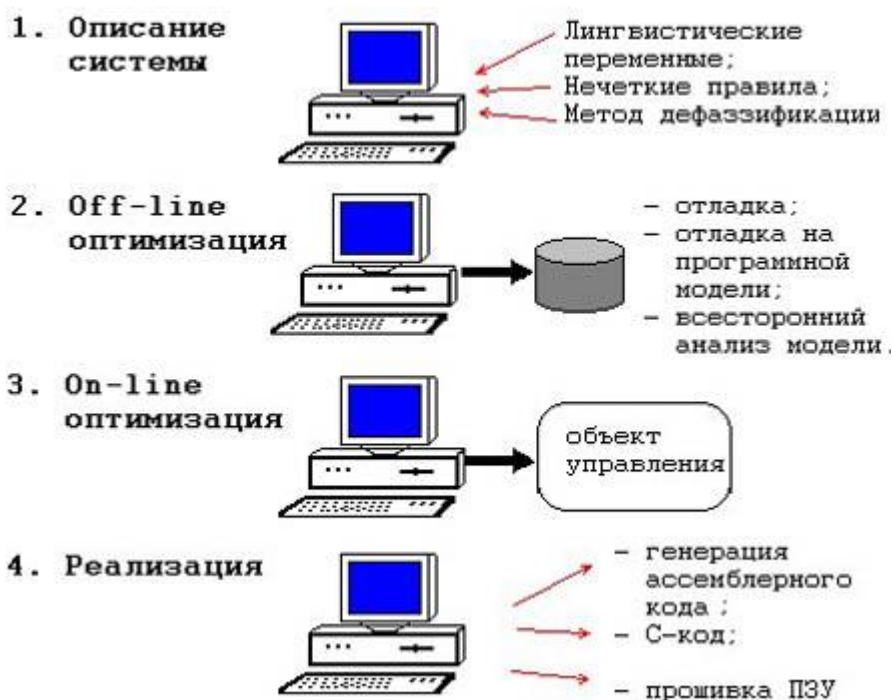


Рисунок 3 – основные этапы разработки нечеткой системы управления.

## Описание системы

На этом этапе при помощи средств, доступных в fuzzy ТЕСН, задача формализуется. Здесь описываются: лингвистические переменные, их функции принадлежности, описать стратегия управления посредством нечетких правил. В целом CASE-технология, на основе которой построен пакет, позволяет все эти действия выполнить только посредством общения с экраном ЭВМ, не заглядывая в программный код.

### Off-line-оптимизация

На этом этапе следует проверить работоспособность созданной системы посредством всех средств fuzzy ТЕСН. Отметим, что можно использовать заранее созданный программный симулятор вашего объекта управления, подобно модели контейнерного крана. Для связи системы управления с моделью используется специально разработанный протокол связи fTlink, в основу которого положена концепция обмена сообщениями Windows. Все необходимые средства для установления связи с вашей моделью находятся в исходных текстах программ связи, поставляемых с пакетом.

### On-line-оптимизация

На этом шаге разрабатываемая система управления и реальный объект управления соединяются физической линией связи (см. рисунок 4)



Рисунок 4 – система управления при On-line-оптимизации

Такой вид отладки позволяет наблюдать поведение системы в реальных условиях и при необходимости вносить изменения в систему управления.

## Функции по работе с пакетом Fuzzy Logic Toolbox

### Редактор функций принадлежности

Редактор функций принадлежности-(**Membership Function Editor**) редактор предназначен для задания следующей информации о термножествах входных и выходных переменных: количество термов; наименования термов; тип и параметры функций принадлежности, которые необходимы для представления лингвистических термов в виде нечетких множеств.

Редактор функций принадлежности может быть вызван из любого GUI-модуля (графического модуля среды Matlab), используемого с системами нечеткого логического вывода, командой **Membership Functions...** меню **Edit** или нажатием клавиш **Ctrl+2**.

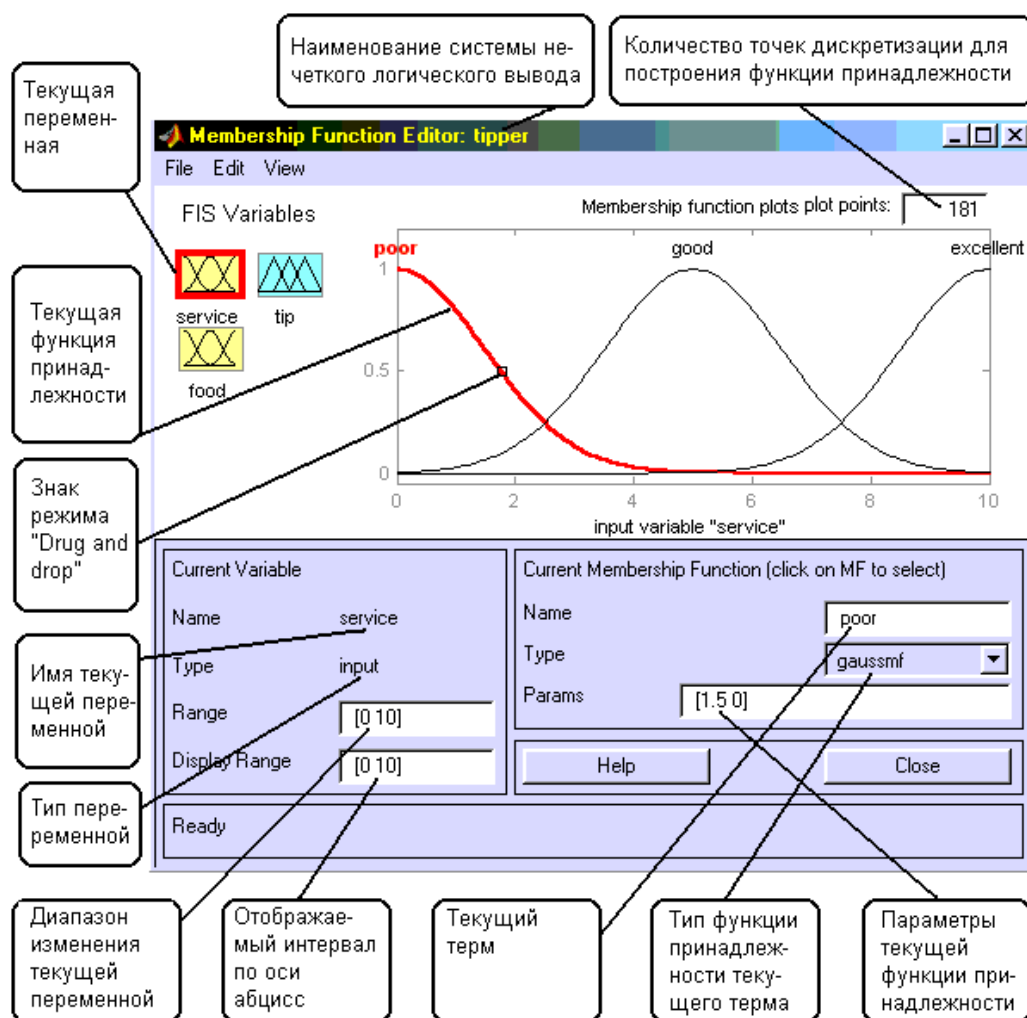


Рисунок 5 – основное диалоговое окно

В FIS-редакторе открыть редактор функций принадлежности можно также двойным щелчком левой кнопкой мыши по полю входной или выходной переменных. Общий вид редактора функций принадлежности с



указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре меню - **File**, **Edit**, **View**, **Type** и четыре окна ввода информации – **Range**, **Display Range**, **Name** и **Params**. Эти четыре окна предназначены для задания диапазона изменения текущей переменной, диапазона вывода функций принадлежности, наименования текущего лингвистического термина и параметров его функции принадлежности, соответственно. Параметры функции принадлежности можно подбирать и в графическом режиме, путем изменения формы функции принадлежности с помощью технологии “Drug and drop”. Для этого необходимо позиционировать курсор мыши на знаке режима “Drug and drop” (см. рис.), нажать на левую кнопку мыши и не отпуская ее изменять форму функции принадлежности. Параметры функции принадлежности будут пересчитываться автоматически.

### Меню **Edit**.

Общий вид меню.



Undo	Ctrl+Z
Add MFs...	
Add Custom MF...	
Remove Selected MF	
Remove All MFs	
FIS Properties...	Ctrl+1
Rules...	Ctrl+3

Рисунок 6 -меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **Add MFs...** позволяет добавить термы в терм-множество, используемое для лингвистической оценки текущей переменной. При выборе этой команды появится диалоговое окно, в котором необходимо выбрать тип функции принадлежности и количество термов. Значения параметров функций принадлежности будут установлены автоматически таким образом, чтобы равномерно покрыть область определения переменной, заданной в окне **Range**. При изменении области определения в окне **Range** параметры функций принадлежности будут промасштабированы.

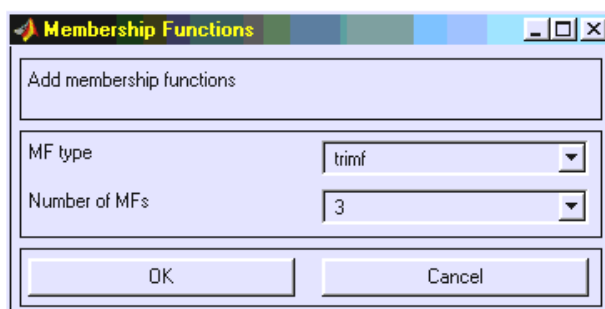


Рисунок 7 - выбор количества термов и типа функций принадлежности.

Команда **Add Custom MF...** позволяет добавить один лингвистический терм, функция принадлежности которого отличается от встроенных. После выбора этой команды появится графическое окно, в котором необходимо напечатать лингвистический терм (поле **MF name**), имя функции принадлежности (поле **M-File function name**) и параметры функции принадлежности (поле **Parameter list**).

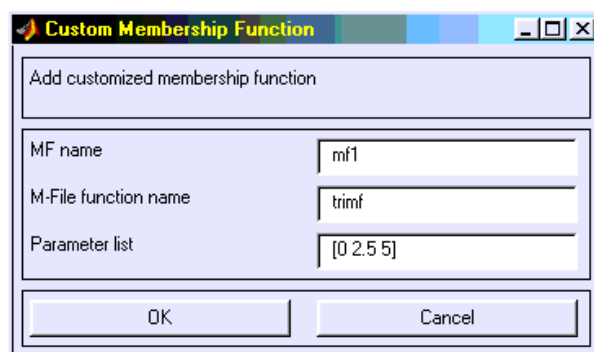


Рисунок 8 - задание лингвистического термина с невстроенной функцией принадлежности.

Команда **Remove Selected MF** удаляет текущий терм из терм-множества текущей переменной. Признаком текущей переменной является красная окантовка ее прямоугольника. Признаком текущего термина является красный цвет его функции принадлежности. Для выбора текущего термина необходимо провести позиционирование курсора мыши на графике функции принадлежности и сделать щелчок левой кнопкой мыши.

Команда **Remove All MFs** удаляет все термы из терм-множества текущей переменной. Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Rules...** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

**Меню Type**

Это меню позволяет установить тип функций принадлежности термов, используемых для лингвистической оценки текущей переменной. На рис. приведено меню **Type**, в котором указаны возможные типы функций принадлежности.



Рисунок 9 - Меню **Type**

### Редактор базы знаний

Редактор базы знаний (**Rule Editor**) предназначен для формирования и модификации нечетких правил. Редактор базы знаний может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **Rules...** меню **Edit** или нажатием клавиш **Ctrl+3**. В FIS-редакторе открыть редактор базы знаний можно также двойным щелчком левой кнопкой мыши по прямоугольнику с названием системы нечеткого логического вывода, расположенного в центре графического окна.

Общий вид редактора базы знаний с указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре системных меню **File**, **Edit**, **View**, **Options**, меню выбора термов входных и выходных переменных, поля установки логических операций И, ИЛИ, НЕ и весов правил, а также кнопки редактирования и просмотра правил.

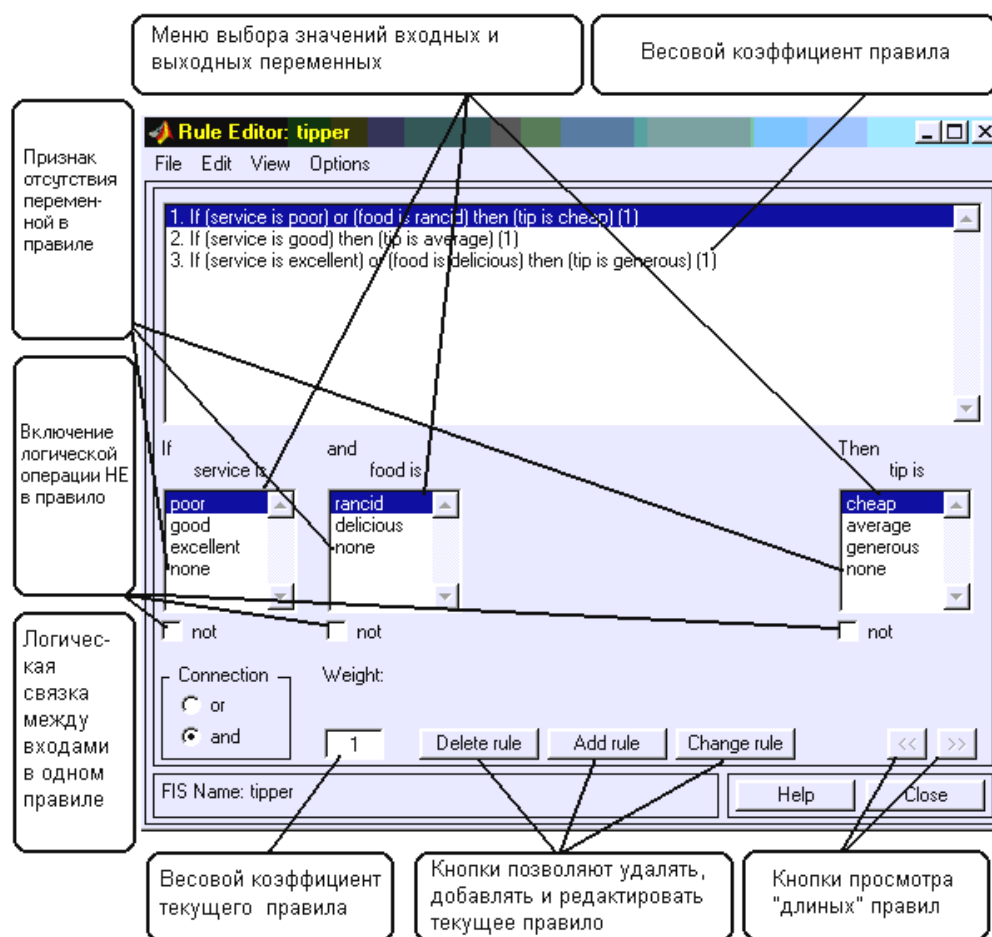


Рисунок 10 – основное диалоговое окно редактора база знаний

Для ввода нового правила в базу знаний необходимо с помощью мыши выбрать соответствующую комбинацию лингвистических термов входных и выходных переменных, установить тип логической связки (И или ИЛИ) между переменными внутри правила, установить наличие или отсутствие логической операции НЕ для каждой лингвистической переменной, ввести значение весового коэффициента правила и нажать кнопку **Add Rule**. По умолчанию установлены следующие параметры: логическая связка переменных внутри правила – И; логическая операция НЕ – отсутствует; значение весового коэффициента правила – 1.

Возможны случаи, когда истинность правила не изменяется при произвольной значении некоторой входной переменной, т.е. это переменная не влияет на результат нечеткого логического вывода в данной области факторного пространства. Тогда в качестве лингвистического значения этой переменной необходимо установить **none**.

Для удаления правила из базы знаний необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу и нажать кнопку **Delete Rule**.

Для модификации правила необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу, затем установить необходимые параметры правила и нажать кнопку **Edit Rule**.

### Меню Edit

Общий вид меню приведен на рисунке 11.



Рисунок 11 - меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Function...** открывает редактор функций принадлежности. Эта команда может быть также выполнена нажатием **Ctrl+2**.

### Меню Options

Это меню позволяет установить язык и формат правил базы знаний.

При выборе команды **Language** появится список языков **English** (Английский), **Deutsch** (Немецкий), **Francais** (Французский), из которого необходимо выбрать один. При выборе команды **Format** появится список возможных форматов правил базы знаний: **Verbose** - лингвистический; **Symbolic** – логический; **Indexed** – индексированный.

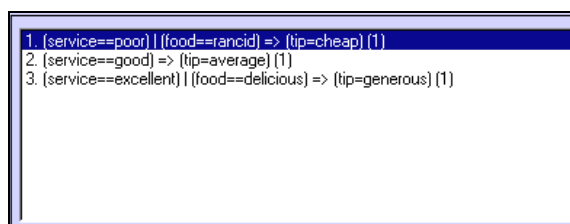


Рисунок 12 - база знаний в формате **Symbolic**.



### **Визуализация нечеткого логического вывода**

Визуализация нечеткого логического вывода осуществляется с помощью GUI-модуля **Rule Viewer**. Этот модуль позволяет проиллюстрировать ход логического вывода по каждому правилу, получение результирующего нечеткого множества и выполнение процедуры дефаззификации. **Rule Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View rules ...** меню **View** или нажатием клавиш **Ctrl+4**. Вид **Rule Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 14.

**Rule Viewer** содержит четыре меню - **File**, **Edit**, **View**, **Options**, два поля ввода информации – **Input** и **Plot points** и кнопки прокрутки изображения влево-вправо (**left-right**), вверх-вниз (**up-down**). В нижней части графического окна расположены также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Каждое правило базы знаний представляется в виде последовательности горизонтально расположенных прямоугольников. При этом первые два прямоугольника отображают функции принадлежности термов посылки правила (ЕСЛИ-часть правила), а последний третий прямоугольник соответствует функции принадлежности термина-следствия выходной переменной (ТО-часть правила). Пустой прямоугольник в визуализации второго правила означает, что в этом правиле посылка по переменной *food* отсутствует (*food is none*). Желтая заливка графиков функций принадлежности входных переменных указывает на несколько значений входов, соответствующих термам данного правила. Для вывода правила в формате **Rule Editor** необходимо сделать однократный щелчок левой кнопки мыши по номеру соответствующего правила. В этом случае указанное правило будет выведено в нижней части графического окна.

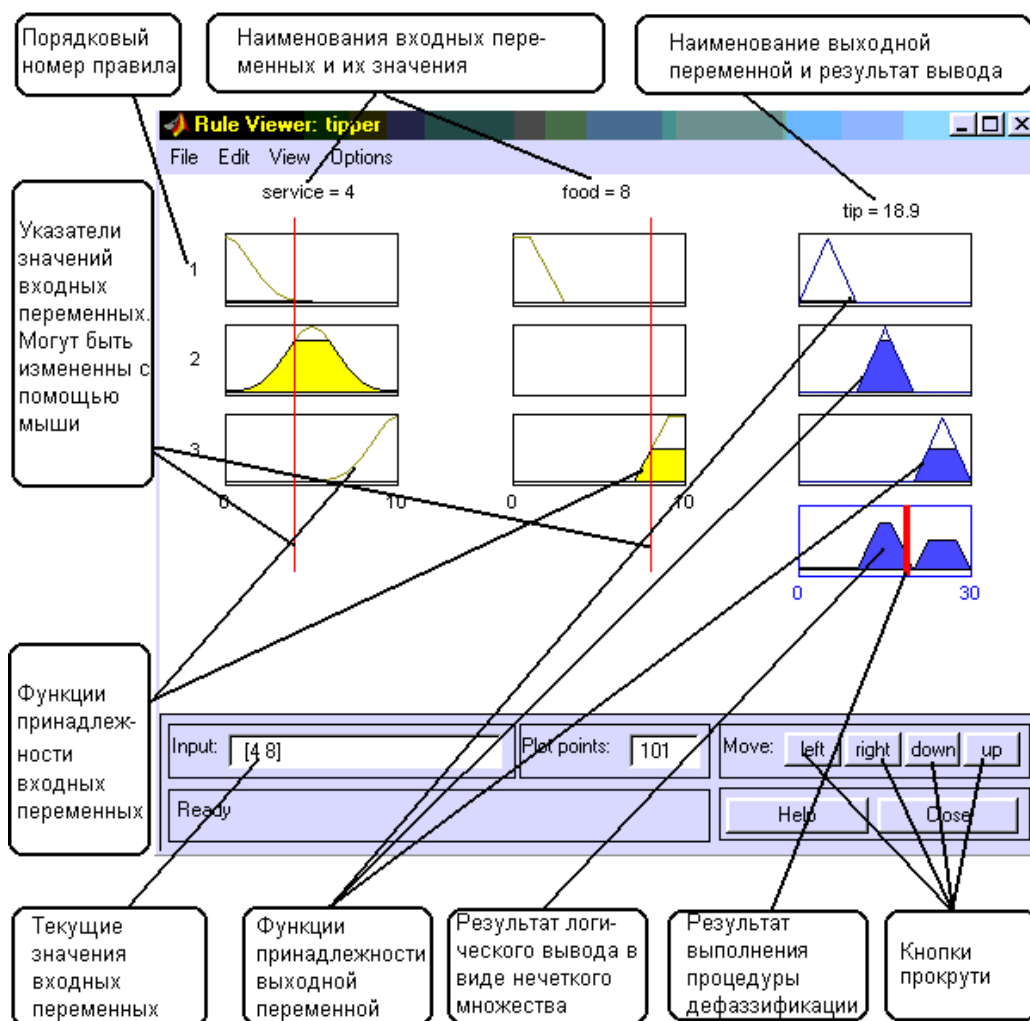


Рисунок 14 – основное меню визуализации логического вывода с помощью **Rule Viewer**

Голубая заливка графика функции принадлежности выходной переменной представляет собой результат логического вывода в виде нечеткого множества по данному правилу. Результирующее нечеткое множество, соответствующее логическому выводу по всем правилам показано в нижнем прямоугольнике последнего столбца графического окна. В этом же прямоугольнике красная вертикальная линия соответствует четкому значению логического вывода, полученного в результате дефаззификации.

Ввод значений входных переменных может осуществляться двумя способами:

- путем ввода численных значений в поле **Input**;
- с помощью мыши, путем перемещения линий-указателей красного цвета.

В последнем случае необходимо позиционировать курсор мыши на красной вертикальной линии, нажать на левую кнопку мыши и не отпустить.

кая ее переместить указатель на нужную позицию. Новое численное значения соответствующей входной переменной будет пересчитано автоматически и выведено в окно **Input**.

В поле **Plot points** задается количество точек дискретизации для построения графиков функций принадлежности.

### Меню Edit

Общий вид меню приведен на рисунке 15.



Undo	Ctrl+Z
FIS Properties...	Ctrl+1
Membership Functions...	Ctrl+2
Rules...	Ctrl+3

Рисунок 15 – общий вид меню.

Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Functions...** открывает редактор функций принадлежности. Эта команда может быть также выполнена нажатием **Ctrl+2**. Команда **Rules...** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

### Меню Options

Меню **Options** содержит только одну команду **Format**, которая позволяет установить один из следующих форматов вывода выбранного правила в нижней части графического окна: **Verbose** - лингвистический; **Symbolic** – логический; **Indexed** – индексированный.

### Визуализация поверхности «входы-выход»

Визуализация поверхности «входы-выход» осуществляется с помощью GUI-модуля **Surface Viewer**. Этот модуль позволяет вывести графическое изображение зависимости значения любой выходной переменной от произвольных двух (или одной) входных переменных. **Surface Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View surface ...** меню **View** или нажатием клавиш **Ctrl+4**. Общий вид модуля **Surface Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 16.



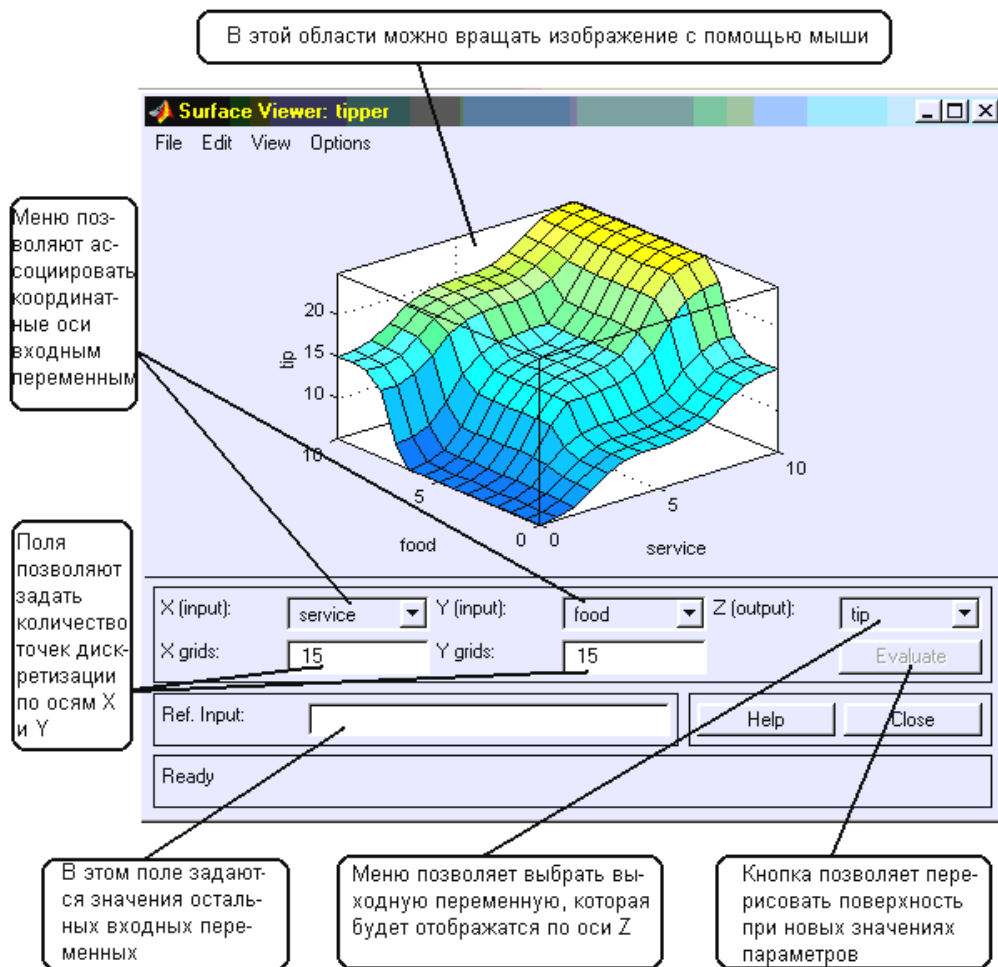
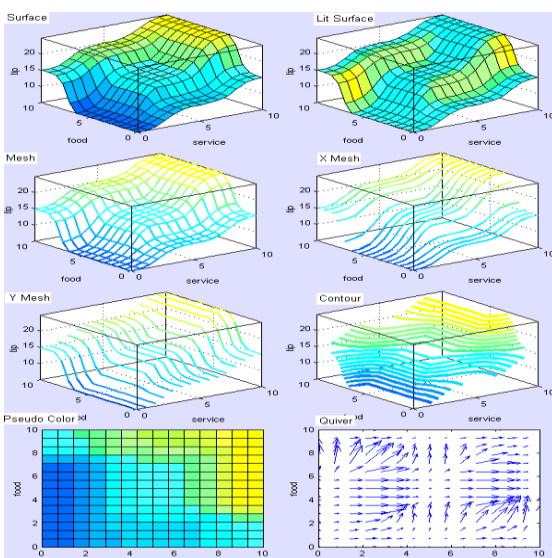


Рисунок 16 – основное меню визуализация поверхности «входы-выход» с помощью Surface Viewer.

**Surface Viewer** содержит верхних системных меню - **File**, **Edit**, **View**, **Options**, три меню выбора координатных осей - **X (input)**, **Y (input)**, **Z (output)**, три поля ввода информации – **X grids**, **Y grids**, **Ref. Input** и кнопку **Evaluate** для построения поверхности при новых параметрах. В нижней части графического окна расположены также кнопки

**Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

**Surface Viewer** позволяет вращать поверхность «входы-выход» с помощью мыши. Для этого необходимо позиционировать курсор мыши на поверхности «входы-выход», нажать на левую кнопку мыши и не отпуская ее повернуть графическое изображение на требуемый угол.



Поля **X girds** и **Y girds** предназначены для задания количества точек дискретизации по осям **X** и **Y**, для построения поверхности «входы-выход». По умолчанию количество дискрет по каждой оси равно 15. Для изменения этого значения необходимо установить маркер на поле **X girds** (**Y girds**) и ввести новое значение.

Поле **Ref. Input** предназначено для задания значений входных переменных, кроме тех, которые ассоциированы с координатными осями. По умолчанию это значения середины интервалов изменения переменных. Для изменения этого значения необходимо установить маркер на поле **Ref. Input** и ввести новые значение.

### Меню координатных осей

Меню **X (input)**, **Y (input)**, **Z (output)** позволяют поставить в соответствие осям координат входные и выходные переменные. При этом входные переменные могут отображаться только по осям **X** и **Y**, а выходные переменные только по оси **Z**. В **Surface Viewer** предусмотрена возможность построения однофакторных зависимостей “вход-выход”. Для этого в меню второй координатной оси (**X (input)** или **Y (input)**) необходимо выбрать **none**.

### Меню Options

Меню **Options** изображено на рисунке 17. Оно содержит команды **Plot**, **Color Map** и **Always evaluate**.

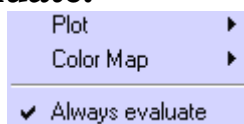


Рисунок 17 – общий вид меню.

Команда **Plot** позволяет управлять форматом вывода поверхности «входы-выход». При выборе этой команды появляется меню (рисунок 18), в котором необходимо выбрать формат вывода поверхности.

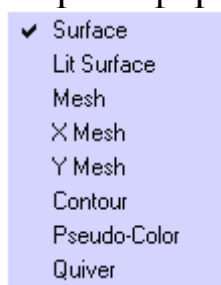


Рисунок 18 – меню команды Plot.

Команда **Color Map** позволяет управлять палитрой цветов при выводе поверхности «входы-выход». При выборе этой команды появляется меню, в котором необходимо выбрать одну из палитр: **default** – использо-

вать палитру, установленную по умолчанию; **blue** – холодная синеголубая палитра; **hot** – теплая палитра, состоящая из черного, красного, желтого и белого цветов; **HSV** – палитра насыщенных цветов: красный, желтый, зеленый, циан, голубой, мажента, красный.

Команда **Always evaluate** позволяет установить или отменить режим автоматического, т.е. без нажатия кнопки **Evaluate**, перерисовывания поверхности «входы-выход» при любом изменении параметров.

## Демонстрационный пример

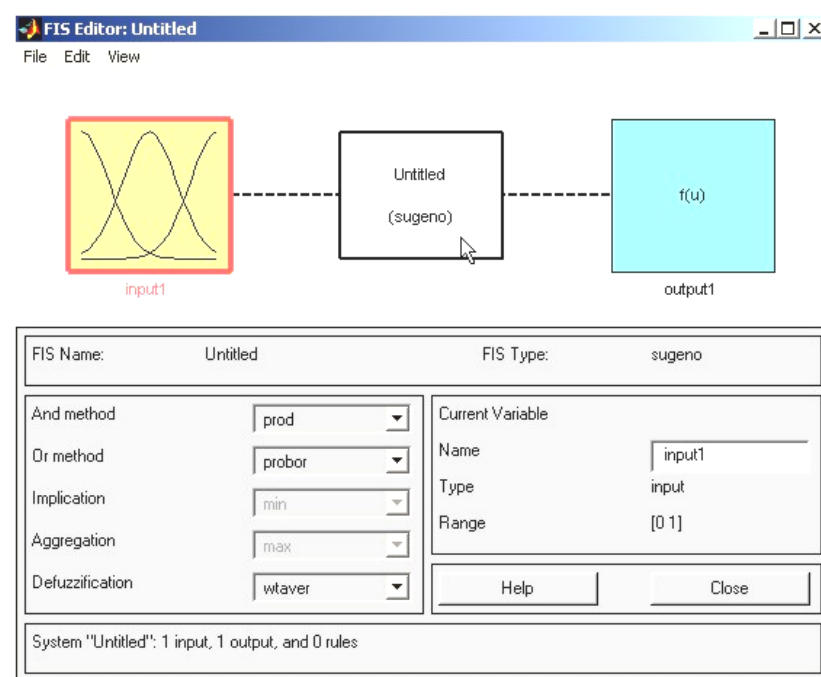
Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными  $x$  и  $y$ . Зависимость  $y = f(x)$  задана пятью точками, приведенными в таблице.

	-2	-1	0	1	2
<b>y</b>	-8	-1	0	1	8

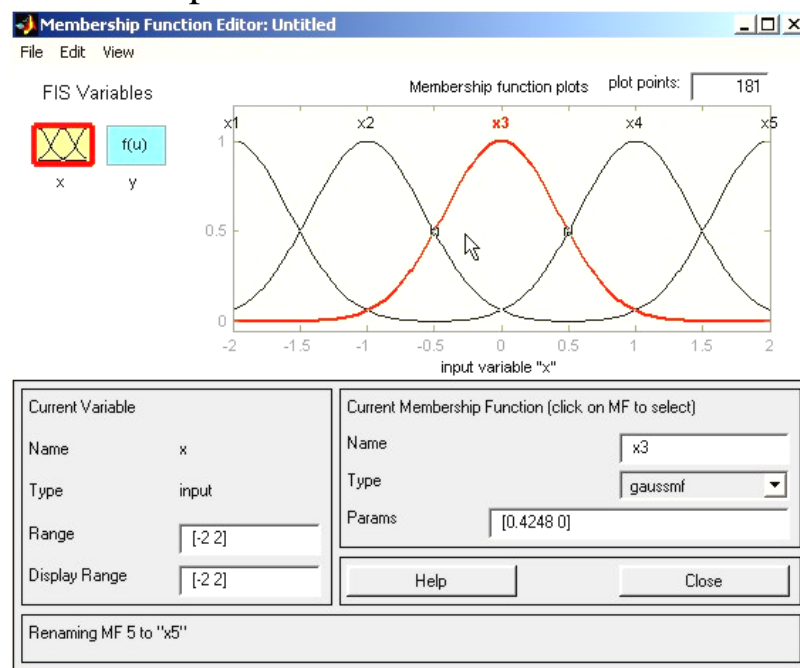
Опишем действия необходимые для построения интерполяции данной функции. Для решения задания будем использовать пакет Fuzzy Logic Toolbox (пакет нечеткой логики), входящий в состав MatLab. Этот пакет позволяет конструировать нечеткие экспертные или управляющие системы.

Откроем пункт меню Start/ Toolboxes/Fuzzy Logic/FIS Editor.

Создаем новую нечеткую систему вывода типа Sugeno: File/ New FIS /Sugeno.



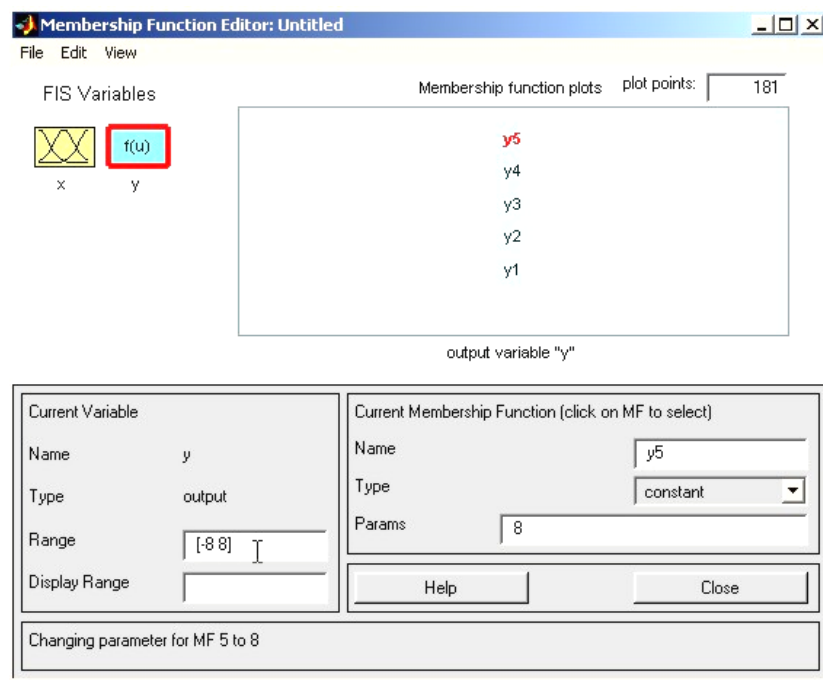
Далее в редакторе функций принадлежности добавим пять новых гауссовых функций принадлежности (для входного сигнала) при помощи Edit/ Add variables.../Input .



В поле Диапазон (Range) задаем диапазон изменения переменной  $x$  от -2 до 2, соответственно таблице.

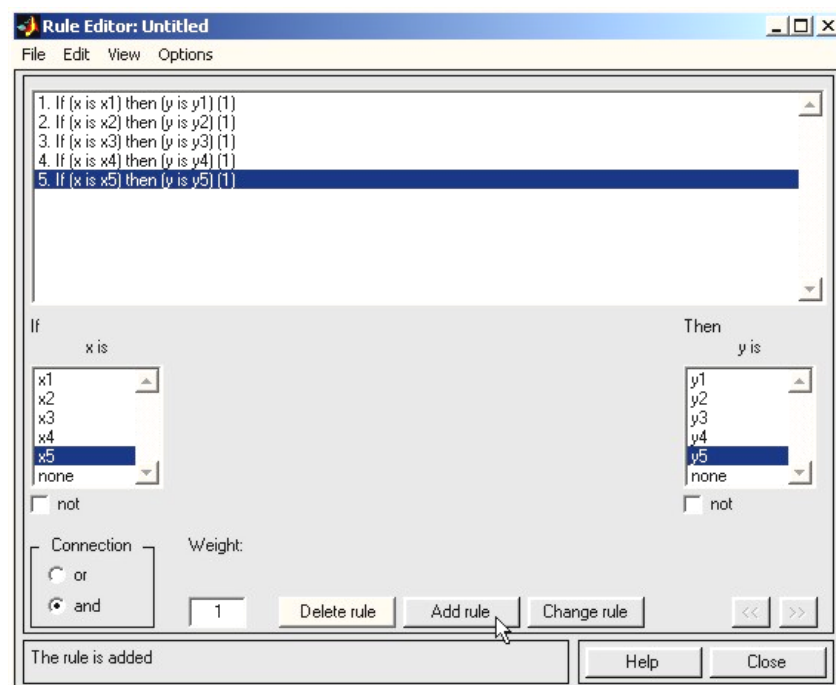
Для успешного решения поставленной задачи необходимо, чтобы ординаты максимумов функций принадлежности совпадали с заданными значениями аргумента  $x$ . Смещая графики функций принадлежности по горизонтали, добиваемся выполнения этого условия.

Зададим теперь функции принадлежности для выходной переменной  $y$ . Добавим 5 постоянных функций принадлежности (5 – число различных значений выходной переменной  $y$ , постоянные функции соответствуют алгоритму Sugeno).

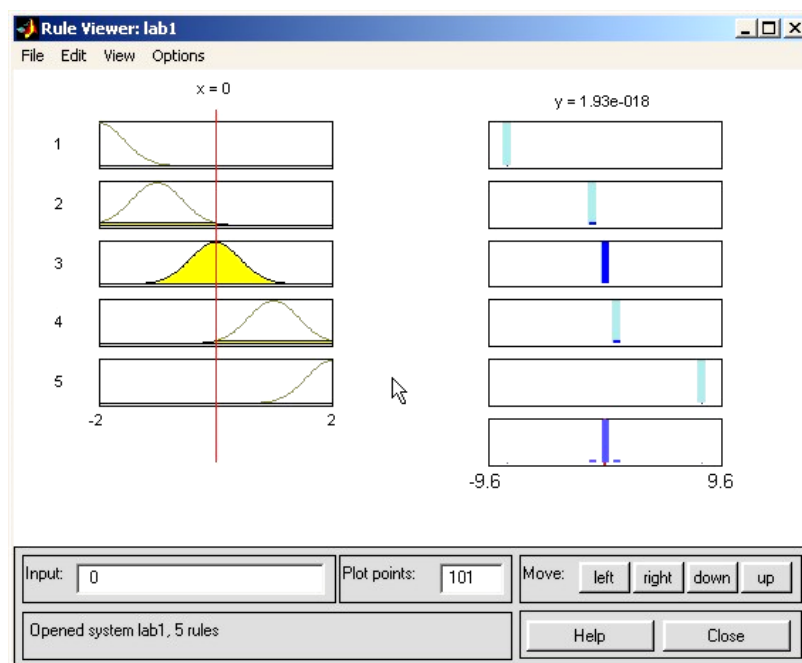


Зададим диапазон изменения выходной переменной от -8 до 8 и для каждой переменной зададим значения из таблицы.

В редакторе правил необходимо сконструировать соответствующие правила. При вводе каждого правила устанавливаем соответствие между функцией принадлежности аргумента  $x$  и числовым значением  $y$ .



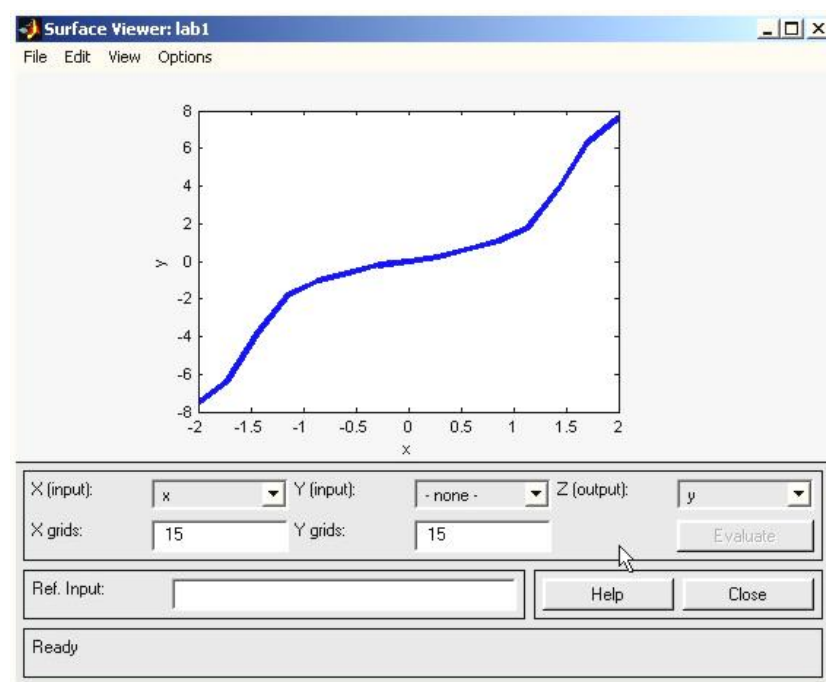
Выберем в пунктах меню Просмотр правил и изучим добавленные зависимости:



В правой части окна в графической форме представлены функции принадлежности аргумента  $x$ , а в левой – переменной выхода  $y$  с пояснением механизма принятия решений.

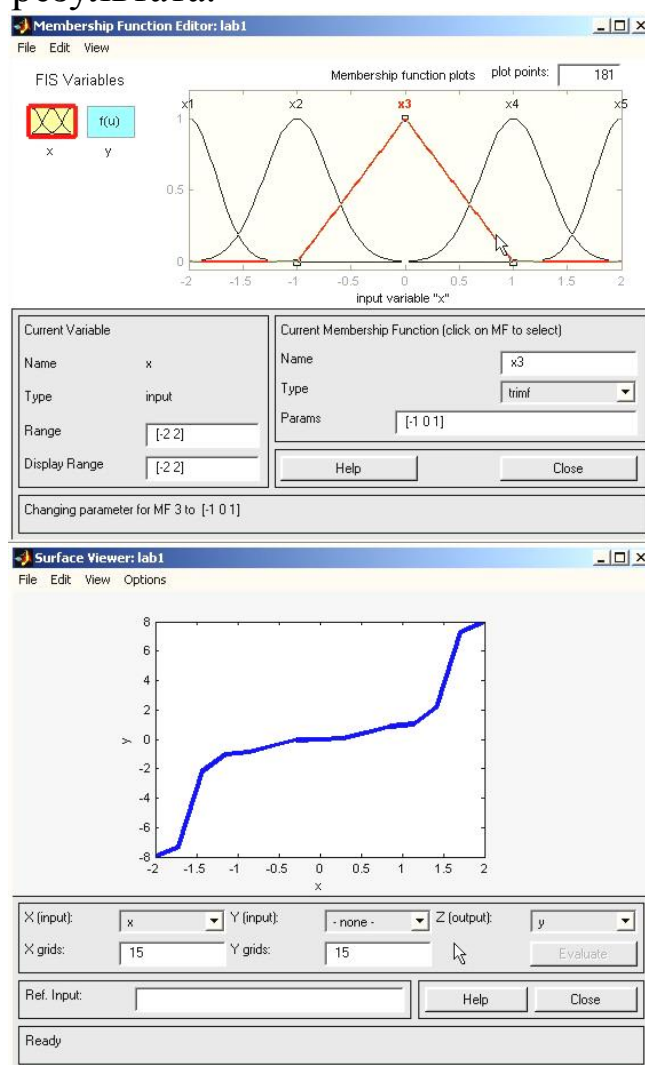
Перемещая красную черту или изменяя значения аргумента  $x$  в правой части окна, получаем значения переменной  $y$  в левой части окна. Таким образом, получено решение задачи интерполяции функции, заданной координатами пяти точек.

В окне просмотра поверхности отклика (выхода) можно визуальнo изучить полученный график функции  $y = f(x)$ .



Как видно из приведенного графика функция получилась недостаточно четкой. Это обусловлено небольшим количеством исходных точек.

Но изменяя параметры функций принадлежности можно добиться более оптимального результата.



## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выпишите задание согласно варианта из таблицы. Рассчитайте значения двенадцати последовательных значений для заданной функции, составьте таблицу значений функции и аргумента.
2. Составить нечеткую экспертную систему, которая бы аппроксимировала функцию по таблицам значений функции и аргумента, составленной в пункте 1.
3. Смоделируйте, как изменится выходное значение при изменении выходов.
4. Рассчитайте ответ нечеткой системы «вручную» и проверьте его при помощи созданной НЭС.

## Индивидуальные задания.

№ варианта	Функция f(x)
1	$\frac{x^2}{2} + 5 \cos(x)$
2	$2e^x - \frac{5}{2}x^2$
3	$e^{-x} + x^2$
4	$\frac{x^2}{(x-1)^2} + x^4$
5	$\sqrt{x+x^2} + \frac{1}{1+x^2}$
6	$\arctg(\sin(x) - \cos(x))$
7	$\frac{x^2 - 6x + 4}{3x - 2} + e^{2x}$
8	$\frac{2x^2 - 1}{\sqrt{x^2 - 2}} + x^4$
9	$sh^2(x) + e^x$
10	$\frac{1}{\ln(x)} + x^2$
11	$\frac{1}{x} + ch(x)$
12	$\frac{1}{x^2} + tg(x)$
13	$\sin(x)^2 + \sin^2(x)$
14	$\frac{1}{sh(x)} + x^4$
15	$\frac{e^{x-3}}{x-3} + \ln(x)$



## **Контрольные вопросы:**

1. Когда рекомендуется применять нечеткие системы управления?
2. Какие наиболее часто употребляемые функции принадлежности?
3. Как реализуется механизм фузификации?
4. Как реализуется механизм дефузификации?
5. Каким образом задаются тип и параметры функций принадлежности в среде MATLAB?
6. В чем заключается метод центра максимума ?
7. В чем заключается метод наибольшего значения ?
8. В чем заключается метод центроида?
9. Каким образом осуществляется переход от нечеткого множества к четному?
10. В чем заключаются основные этапы синтеза нечеткой системы управления?

## **Литература:**

1. Березин С.Я. Основы кибернетики и управление в биологических и медицинских системах: учебное пособие / С.Я. Березин. – Старый Оскол: ТНТ, 2013. – 244 с.
2. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2007.-284 с.
3. Дьяконов В.П., Круглов В.В. MATLAB 6/5 SP1/7/7 SP 5/6. Инструменты искусственного интеллекта и биоинформатика. Серия «Библиотека профессионала».-М.: СОЛОН-ПРЕСС, 2006. -456 с.
4. Емельянов, С. Г. Автоматизированные нечетко-логические системы управления [Текст]: монография / С. Г. Емельянов, В. С. Титов, М. В. Бобырь. - М.: ИНФРА-М, 2011. - 176 с. - (Научная мысль).
5. Методы робастного, нейро-нечеткого и адаптивного управления [Текст]: учебник для студентов вузов / ред. Н. Д. Егупов. - М. : МГТУ, 2001. - 743 с.

## Работа 2

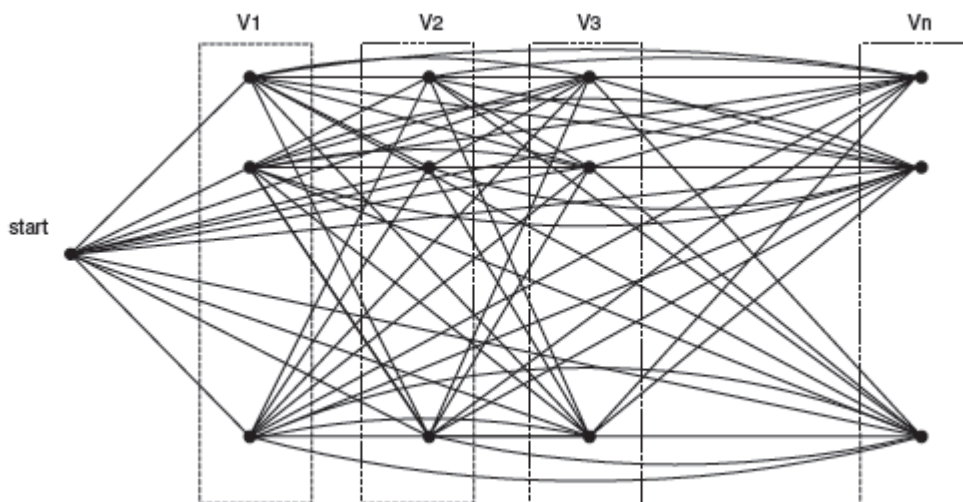
### Классификация с использованием муравьиного алгоритма

Впервые применение алгоритма муравьиной колонии для классификации было представлено R. Pagrinelli в 2002 году. Метод извлечения классификационных правил на основе муравьиного алгоритма получил название AntMiner.

Цель метода — получить простые правила вида *если условие то следствие*.

Предполагается, что все атрибуты категориальны. Т.е. термы представлены в виде Атрибут = Значение, например Пол = Мужской. AntMiner последовательно формирует упорядоченный список правил. Вычисление начинается с пустого множества правил и после формирования первого, все тестовые единицы данных, покрытые этим правилом, удаляются из тестового набора.

Алгоритм использует при работе направленный граф, где каждому атрибуту сопоставляется столько вершин, сколько возможных значений он принимает в тестовом наборе. Соответственно, предполагается, что перед началом работы алгоритма из тестового набора выделены множества атрибутов и их возможных значений, а также множество возможных классов.



Псевдокод алгоритма выглядит следующим образом:

```
testSet = Весь тестовый набор
ПОКА (|testSet| > макс_число_непокрытых_случаев)
```

$i = 0$ ;

ВЫПОЛНЯТЬ

$i = i + 1$ ;

$i$ -й муравей последовательно строит классификационное правило;

Обновление феромонов на пути, представленным правилом  $i$ -го муравья;

ПОКА ( $i < \text{Число\_муравьёв}$ )

Выбрать лучшее из всех правил;

testSet = случаи, не покрытые лучшим правилом;

КОНЕЦ ЦИКЛА

Рассмотрим последовательно шаги алгоритма.

Инициализация значений феромона

Все ребра графа инициализируются начальным значением феромона по следующей формуле:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^a b_i}$$

, где  $a$  — общее число атрибутов,  $b_i$  — количество возможных значений  $i$ -го атрибута.

Построение правила

Каждое правило в алгоритме AntMiner содержит антецедент (набор термов Атрибут = Значение) и представляемый им класс. В оригинальном алгоритме исходные данные содержат только категориальные атрибуты. Предположим, что терм $_{ij} \approx A_i = V_{ij}$ , где  $A_i$  это  $i$ -й атрибут, а  $V_{ij}$  это  $j$ -ое значение  $A_i$ .

Вероятность, что этот терм будет добавлен в текущее частичное правило, составляемое муравьём определяется следующей формулой:

$$P_{ij}(t) = \frac{\tau_{ij}(t) \cdot \eta_{ij}}{\sum_i \sum_j \tau_{ij}(t) \cdot \eta_{ij}, \forall i \in I}$$

, где  $\eta_{ij}$  это значение эвристики (будет определена далее),  $\tau_{ij}$  значение фе-

ромона на этом ребре графа, а I множество атрибутов, еще не использованных текущим муравьём.

### Эвристика

В традиционном алгоритме муравьиной колонии для решения транспортной задачи веса ребер используются совместно со значением феромона для принятия решения о выборе следующей вершины. В алгоритме AntMiner эвристикой является количество информации — термин, использующийся в теории информации. Качеством здесь измеряется с помощью энтропии, для предпочтения одних правил над другими:

$$\eta_{ij} = \frac{\log_2(k) - InfoT_{ij}}{\sum_i^a \sum_j^{b_i} \log_2(k) - InfoT_{ij}}$$

$$InfoT_{ij} = -\sum_{w=1}^k \left[ \frac{freqT_{ij}^w}{|T_{ij}|} \right] * \log_2 \left[ \frac{freqT_{ij}^w}{|T_{ij}|} \right]$$

где  $k$  — число классов,  $T_{ij}$  — подмножество, содержащее все единицы данных, где атрибут  $A_i$  имеет значение  $V_{ij}$ ,  $|T_{ij}|$  число элементов в  $T_{ij}$ ,  $freq(T_{ij}^w)$  — число единиц данных, имеющих класс  $w$ ,  $a$  — общее число атрибутов,  $b_i$  — число возможных значений  $i$ -го атрибута.

Чем выше значение  $Info T_{ij}$ , тем меньше шансов что муравей выберет данный терм.

### Обновление значений феромонов

После того как каждый муравей завершит конструирование правила, выполняется обновление феромонов по следующей формуле:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) \cdot Q \quad \forall \text{терм } ij \in \text{текущее правило}$$

где  $Q$  — качество правила, которое вычисляется по следующей формуле:

$$Q = \left( \frac{TruePos}{TruePos + FalseNeg} \right) \times \left( \frac{TrueNeg}{FalsePos + TrueNeg} \right)$$

Здесь  $TruePos$  — количество единиц данных, которые покрываются правилом и класс которых совпадает с классом, представляемым правилом,  $FalsePos$  — количество единиц данных, которые покрываются правилом, но класс которых отличается от класса, представляемым правилом,  $TrueNeg$  — количество единиц данных, которые не покрываются правилом и класс которых не совпадает с классом, представляемым правилом,

FalseNeg — количество единиц данных, которые не покрываются правилом, но класс которых совпадает с классом, представляемым правилом. Также требуется выполнять пересчёт феромонов, имитирующий его испарение. Это можно сделать простой нормализацией значений с учётом обновленного значения феромонов.

### Улучшенное обновление феромонов

Для более гибкой настройки выполнения алгоритма используется следующая формула обновления вероятностей ребер принадлежащих текущему правилу:

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t-1) + (1 - \frac{1}{1+Q}) \cdot \tau_{ij}(t-1)$$

где  $\rho$  — коэффициент испарения (обычно устанавливается значение  $\sim 0,1$ ),  $Q$  качество правила, описанное выше. Ребра, не покрытые текущим правилом просто нормализуются. Этот способ обеспечивает снижение значения феромона при низком  $Q$  и повышение при высоком (в отличие от оригинального способа).

### Элитные муравьи

Также можно улучшить сходимость алгоритма, вводя некоторое число так называемых “элитных” муравьёв, которые выбирают терм, для которого  $P = \max P_{ij}$ .

В оригинальном алгоритме выбор терм выбирается по следующему алгоритму:

ДЛЯ ВСЕХ  $(i,j) \in J_i$   
 ЕСЛИ  $q \leq \sum P_{ij}$  ТО Выбрать терм  $ij$

Где  $q$  случайная величина  $[0..1]$  с равномерным распределением. Т.е. используется плотность вероятностей термов.

Для введения элитных муравьёв алгоритм выбора терма изменяется следующим образом:

ЕСЛИ  $q_1 \leq \phi$  ТО  
 ДЛЯ ВСЕХ  $(i,j) \in J_i$

ЕСЛИ  $q_2 \leq \sum P_{ij}$  ТО Выбрать терм  $i_j$   
ИНАЧЕ  
Выбрать терм с  $P = \max P_{ij}$

Здесь  $q_1$  и  $q_2$  случайные величины  $[0..1]$  с равномерным распределением.

Некоторые наблюдения

Аналогом AntMiner является известный алгоритм Naive Bayes. Согласно моим экспериментам по их сравнению AntMiner либо незначительно превосходит Naive Bayes, либо и вовсе уступает ему на различных наборах данных. Для сравнения использовалась реализация AntMiner представленная [здесь](#) и реализация Naive Bayes из библиотеки [WEKA](#). Кроме того, AntMiner значительно уступает и по времени вычисления. Возможно, есть ситуации или некоторые предусловия, при которых AntMiner оказывается действительно лучшим выбором.

## Работа 3

### Проектирование, эксплуатация и жизненные стадии систем поддержки принятия решений

**Цель работы:** изучение принципов работы, проектирования и овладение навыками эксплуатации экспертной системы диагностики заболеваний.

#### **Краткие теоретические сведения.**

Успехи врачевания в первую очередь определяются качеством распознавания болезней. В связи с этим правомочен афоризм: «Правильная диагностика – хорошее лечение».

Непрерывный научно-технический прогресс во всех научных областях, в том числе и в медицине, дает для расширения диапазона диагностических возможностей врача множество современных методов обследования больного. Современные методы исследования болезни позволяют более глубоко и детально изучить их, способствуют открытию новых патологических состояний, дают возможность уточнить этиопатогенетические механизмы их возникновения. Однако качество диагноза определяется не только открытием и внедрением инструментально-технических и других методов распознавания болезней в клиническую медицину, но и умением практических врачей мыслить. С увеличением информации о болезнях усложняется процесс мыслительной деятельности врача, повышаются требования к логической обоснованности, своевременности, завершенности и точности установления диагноза. Проблема удовлетворения современным требованиям, предъявляемым к диагнозу, является самой главной, определяющей задачей врача, которая реализуется с помощью методической части клинической медицины – диагностики - науки о распознавании болезней.

• **Диагностика** – раздел медицинской науки изучающей методы распознавания болезней в процессе исследования, наблюдения больного и мышления врача с целью установления диагноза

Решая задачу распознавания болезни, врач имеет несколько возможностей, использование которых обеспечивает высокое качество диагностики:

- основательный уровень владения методами и техникой врачебного исследования;
- внедрение в клиническую медицину новых современных методов исследования;
- совершенствование и углубление знаний о признаках болезни и их диагностической ценности;
- совершенствование мыслительных операций с выявленными

признаками болезни научно обоснованными способами диагностики;

- совершенствование и разработка методов диагностических решений (создание диагностического алгоритма, схем эвристического поиска, диагностических тестов и др.).

Анализ клинической картины начинается с процесса обследования больного. Важно понимать, что обследование – отнюдь не механическое, а творческое занятие. Его творческий компонент неразрывно связан с глубиной характеристики симптома. Группировка и классификация симптомов являются основными моментами анализа клинической картины. Процедура систематизации подразумевает группировку симптомов по системам органов. В этом процессе проверяется полнота обследования больного и вырабатывается привычка к строгому, последовательному обследованию и изложению обнаруженных фактов.

Компоненты синдрома – симптомы, полученные при сборе анамнеза и физикальных исследований. Синдром синтезируется из симптомов, с целью локализовать зону поражения и выявить патологию. Например, установленное при обследовании системы дыхания локализованное притупление перкуторного звука, бронхиальное дыхание и усиление бронхофонии свидетельствуют об уплотнении легочной ткани. Важнейшая предпосылка успеха диагноза – глубокий патогенетический анализ клинической ситуации.

Дифференциальная диагностика – это сравнение клинической картины заболевания у данного конкретного больного со всеми заболеваниями, имеющими сходную клиническую картину, для исключения всех болезней, кроме одной.

Это сравнение должно проводиться не хаотично, не стихийно, а по строгим правилам и в определенном порядке. Иными словами, мыслительный процесс при дифференциальной диагностике должен подчиняться такой же строгой схеме, как, например, проведение физикального исследования или анализ электрокардиограммы. Практически все авторы, занимавшиеся этой проблемой сходятся, с небольшими отличиями, в том мнении, что проведение дифференциальной диагностики состоит из следующих пяти этапов:

- группировка симптомов для выделения синдромов;
- выделение ведущего синдрома;
- генерация диагностической гипотезы;
- построение дифференциально-диагностического ряда;
- исключение синдромно-сходных заболеваний и формулировку окончательного диагноза.



Рассмотрим последовательность действий на каждом этапе проведения дифференциальной диагностики.

Первый этап – группировка симптомов для выделения синдромов, был достаточно описан выше, но хотелось бы подчеркнуть, что при выделении синдрома следует оценить его особенности. Подробная патогенетическая характеристика синдромов абсолютно необходима для плодотворной деятельности на следующих этапах.

На втором этапе ведущий синдром выделяется по тем же критериям, что и ведущий симптом. Главное здесь – плодотворность генерации диагностической гипотезы.

На третьем этапе формируется диагностическая гипотеза. Диагностическая гипотеза – это, в сущности, рабочий диагноз, который создается методом аналогии на основе синдромного анализа.

Характерными чертами диагностической гипотезы являются:

- неполнота аргументации;
- уникальность диагностической гипотезы.

Неполнота аргументации означает что рабочий диагноз формулируется у постели больного на базе данных анамнеза и физикального обследования. Следует иметь в виду, что как анамнестическое, так и физикальное обследование, несмотря на стереотипизм их реализации, не предусматривают некоторых вопросов, приемов, методов, которые имеют существенное значение для диагностики относительно редких заболеваний. Например, проведение провокационных проб для диагностики кровотоковости петехиально-пятнистого типа, оценка ширины пространства между относительной и абсолютной тупостью сердца для верификации гидроперикарда и т.д.

Под уникальностью диагностической гипотезы в данном случае, имеется в виду то, что диагностическая гипотеза генерируется в процессе работы с каждым больным. Она может звучать одинаково, при работе с довольно большим количеством больных, например «гипертоническая болезнь», но аргументация в пользу этого предположения каждый раз создается заново.

Считается, что в процессе генерации диагностической гипотезы возможен ряд ошибок логического порядка:

- ошибка поспешного обобщения;
- ошибка в умозаключении по аналогии;
- ошибка ложной последовательности.

Ошибкой поспешного обобщения является учет диагностической гипотезой не всех, а лишь некоторых клинических данных. Ошибка в умоза-

ключении по аналогии происходит, когда обобщаются процессы, имеющие разный патогенез. Ошибка ложной последовательности возможна, если умозаключение движется по пути «после этого, значит вследствие этого», тогда как далеко не всегда последовательное возникновение симптомов означает наличие между ними причинно – следственной связи.

На четвертом этапе строится дифференциально - диагностический ряд, что требует от врача широкой эрудиции с одной стороны и навыков системного мышления с другой. Следует обратить особое внимание, что дифференциально - диагностический ряд - не простое перечисление заболеваний, имеющих один отличительный признак (выделенный в качестве ведущего синдрома), на большие или меньшие группы. Разделение это осуществляется на патогенетической основе.

Такого рода внутрисиндромные классификации чрезвычайно полезны, т.к. позволяют в дальнейшем проводить дифференциальную диагностику, исключая заболевания не по одному, а целыми группами. Работа по составлению дифференциально-диагностического ряда сопровождается его написанием. Это служит гарантией того, что ни одно из заболеваний не будет забыто в процессе дифференциальной диагностики.

На пятом этапе осуществляется процесс исключения синдромно - сходных заболеваний в два этапа. На первом этапе производятся межгрупповые сопоставления и исключение всех групп заболеваний, кроме одной. На второй фазе происходит сопоставление заболеваний внутри оставшейся группы и исключение всех болезней, кроме одной.

Определяются пять основных принципов дифференциальной диагностики.

Первый принцип – принцип существенного различия.

Принцип существенного различия означает, что наблюдаемый случай болезни не принадлежит к сравниваемому с ним виду заболеваний, т.к. в клинической картине у данного больного отсутствует синдром, обязательно имеющий место при сравниваемой болезни, или группе болезней. Принцип существенного различия имеет и другую трактовку: заболевание не относится к сравниваемому с ним виду, т.к. у больного имеется синдром (симптом), никогда не встречающийся при сравниваемом заболевании.

Второй принцип – исключения через противоположность. Этот принцип можно сформулировать так: у больного имеется симптом, прямо противоположный тому, который закономерно выявляется при сравниваемой болезни.

Третий принцип – принцип несовпадения признаков – важнейший и наиболее часто используемый. В данном случае речь идет о существенном сходстве симптома и в то же время – о его отличительных характеристиках при различных заболеваниях.

Четвертый принцип – индуктивного дифференциального диагноза -предусматривает постоянную работу в библиотеке. По прочтении соответствующей предполагаемому заболеванию литературы неизбежно возникает потребность повторить обследование больного с фиксацией внимания на специфических проявлениях болезни.

Пятый принцип – проверки диагноза. В зависимости от случая, проверка диагноза может быть либо очень короткой (в самых простых случаях она не нужна), либо продолжительной, трудной для врача и больного. Иногда приходится двигаться от простых методов к все более и более сложным, если простые и необременительные способы диагностики на момент их использования не дают определенных результатов.

Применение методики дифференциальной диагностики по одному ведущему синдрому плодотворна, но иногда, особенно при сложных, полисиндромных клинических картинах моно-синдромный дифференциальный диагноз не приводит к успеху. Диагностическая гипотеза (рабочий диагноз) не оказывается ни подтвержденной, ни отвергнутой. В этой ситуации все приходится начинать сначала, выделяя другой синдром в качестве основного. При безуспешности этой попытки следует перейти к дифференциальной диагностике по третьему синдрому и т.д.

Вообще при полисиндромных процессах иногда целесообразно взглянуть на вещи более широко: все многообразие клиники может быть обусловлено не одним заболеванием, а их сочетанием.

После сбора анамнеза, проведения физикального исследования с целью проверки диагностических версий, врач использует дополнительные специальные диагностические тесты, применяя лабораторные и инструментальные методы исследования.. Диагностический тест необходим для подтверждения или исключения болезни .

В зависимости от цели выделяют специальные методы исследования – инвазивные и неинвазивные.. Выделяют также следующие методы исследования :

- функциональные методы, оценивающие ту или иную функцию органа или системы;
- индикаторные, указывающие на наличие какого-либо патологического процесса или состояния (воспаление, цитолиз, иммунные нарушения и т. д.);

- визуализирующие методы, позволяющие оценить макроструктуру органа;
- морфологические — прижизненно оценивающие микроструктуру органа.

При дополнительном обследовании больного необходимо учитывать чувствительность, специфичность и информативность используемых методов. Информативность исследования – неколичественная характеристика метода, отражающая объем полезной информации, получаемой при применении данного метода в конкретной патологии и, следовательно, позволяющей судить, насколько данный метод необходим в конкретном случае.

Наибольшую ценность в диагностике имеют диагностические тесты высокой чувствительности и специфичности, позволяющие обнаруживать высокоспецифические и патогномоничные диагностические стигматы.

Оценка надежности различных диагностических методов (тестов) проводится в специальных научных исследованиях, где изучаемый метод сравнивают с эталонным. Однако, прежде чем определить критерии надежности результатов диагностического теста, необходимо оценить априорную вероятность наличия болезни.

Она особенно важна в тех случаях, когда чувствительность и специфичность применяемого метода низкая, а эталонный тест либо отсутствует, либо его трудно осуществить. Оценка априорной вероятности основывается на данных эпидемиологических характеристик болезни: степени ее распространенности, особенности ее течения в той или иной местности, факторах риска и степени соответствия клинической картины данного больного классическому описанию болезни. В практической деятельности врач, оценивая априорную вероятность болезни ограничивается приблизительным количественным ориентиром. Эта ориентировочная, приблизительная, количественная оценка априорной вероятности выводится на основании знаний литературных источников, подсказанной интуиции, клинических наблюдений и накопленного опыта.

Вероятность болезни, определяемая по результатам проведения диагностического теста, называется апостериорной вероятностью.

Результат теста может быть либо положительным (патология), либо отрицательным (норма), а заболевание может либо присутствовать, либо отсутствовать. Возможны четыре варианта толкования результатов теста – два истинных и два ложных. Правильный ответ – это положительный результат, если заболевание есть, или отрицательный, если заболевание отсутствует. Однако результаты могут быть и ложноотрицательными (бо-

лезнь есть, но тест ее исключает), и ложноположительными (болезни нет, но тест ее подтверждает) (таблицы 1,2). Оценка точности теста основана на сравнении его результата с критерием, заслуживающим доверия истины, - эталоном. С этой целью определяют чувствительность и специфичность теста, предсказательную ценность положительного и отрицательного теста, а также диагностическую точность метода.

Таблица 1 - Варианты оценки результатов теста

Тесты	Болезнь есть	Болезни нет
Положительный	A - истинноположительный	B - ложноположительный
Отрицательный	C - ложноотрицательный	D - истинноотрицательный

Таблица 2 - Оценка точности результатов теста

Чувствительность теста	$A / (a + c)$ - доля истинно положительных тестов среди больных
Специфичность теста	$D / (b + d)$ - доля истинно отрицательных тестов среди здоровых
Предсказанная ценность положительного теста	$A / (a + b)$ = доля истинно положительных тестов среди всех положительных тестов
Предсказанная ценность отрицательного теста	$D / (c + d)$ = доля истинно отрицательных тестов среди всех отрицательных тестов
Диагностическая точность метода	$(a + d) / (a + b + c + d)$ = доля истинных результатов среди всех результатов теста

Поскольку точные способы установления диагноза почти всегда трудоемкие, дорогие и сопряжены с риском исследования, оправдано назначение более простых и доступных тестов, особенно на ранних этапах исследования больного. Вероятность положительного результата диагностического теста в присутствии болезни называется чувствительностью метода, а вероятность отрицательного результата в отсутствие болезни – его специфичностью. При выборе диагностического теста врач должен принимать во внимание как чувствительность, так и его специфичность.

Чувствительность метода для выявления конкретного признака определяется соотношением количества больных с наличием признака по данным применяемого метода исследования и количества обследованных больных с действительным наличием этого признака, выраженным в процентном отношении. Если метод или тест обладает высокой чувствительностью, то он выявляет патологию у большинства больных с данным заболеванием (истинно положительные результаты) и лишь у незначительного количества больных дает ложноотрицательные результаты. Чувствительность признака при конкретном заболевании – показатель, отражающий частоту встречаемости признака при данной патологии. Специфичность метода для выявления конкретного признака – соотношение количества обследованных здоровых пациентов, у которых признак отсутствует, и общего количества здоровых пациентов, выраженное в процентном отношении. Таким образом, специфичность отражает частоту «ошибок» метода исследования у здоровых людей. Специфичность признака при данном заболевании – показатель, отражающий насколько частый признак бывает у людей, не имеющих этого заболевания. Чувствительный тест следует убрать, если есть риск пропустить опасную, но излечимую болезнь. Чувствительные тесты также рациональны на ранних стадиях диагностического поиска с целью сужения круга вероятных заболеваний].

Диагностический метод безупречен, если и чувствительность и специфичность равны 100 %. Однако таких совершенных методов исследования мало или они труднодоступны. Если проводится эталонный тест, при котором чувствительность и специфичность равны 100 %, определение априорной вероятности подозреваемой болезни почти не играет роли. И наоборот, чем меньше чувствительность и специфичность, тем значительнее предварительная информация о больном.

Таким образом, в одних случаях решающим фактором для интерпретации результатов диагностического теста будет как можно более точная оценка априорной вероятности, т. е. вероятности еще до проведения диагностического теста, а в других определяющее значение в верификации болезни будет иметь эталонный тест или тест с высокой чувствительностью и его специфичностью.

При положительном результате диагностического теста апостериорная вероятность болезни идентифицирует предсказательную ценность

Из данных таблицы 3 видно, что апостериорная вероятность болезни значительно зависит от оценки априорной вероятности. Следовательно, при оценке эффективности положительных результатов для подтверждения диагноза и отрицательных – для его исключения необходимо учиты-

вать вероятность подозреваемой болезни у обследуемого больного еще до проведения диагностического теста, особенно, если диагностические тесты несовершенны (низкой чувствительности и специфичности).

Таблица 3 - Пример диагностического теста

Априорная вероятность	2%		30%		70%	
	Есть	Нет	Есть	Нет	Есть	Нет
Болезнь	Есть	Нет	Есть	Нет	Есть	Нет
Чувствительность и специфичность	ДЧ	ДС	ДЧ	ДС	ДЧ	ДС
Положительный тест	18	196	270	140	630	90
Отрицательный тест	2	784	30	560	70	210
Всего	20	980	300	700	700	300
Предсказанная ценность положительного теста	8,4% 18/(18+196)		65,85% 270/(270+140)		87,5% 630/(630+90)	

В таблице ДЧ и ДС – диагностические чувствительность и специфичность, соответственно.

Считается, что вероятность болезни тем выше, чем больше свидетельств в ее пользу. Поэтому, если результат диагностического теста сомнительный, необходимо полагаться на сумму доказательств. Но эти данные повышают диагностические возможности только в том случае, если назначаемые диагностические тесты дают независимые друг от друга информативные доказательства.

Экспертные системы дифференциальной диагностики предназначены для решения следующих основных задач: интерпретация, прогноз, диагностика, проектирование, планирование, обучение, наблюдение, управление.

Экспертные системы имеют в своем составе обширную базу данных – факты выбранной предметной области, а также базу знаний, в которой отражены профессиональные навыки и умения специалистов высокого уровня в данной области.

В качестве примера российской разработки можно привести комплекс РОФЭС (Регистратор Оценки Функционально-Эмоционального Состояния). Система состоит из комплекса датчиков и программного обеспечения, с помощью которых производится диагностика и анализ состояния организма. Она может осуществить подбор препаратов для конкретного человека, провести частотную и цветотерапию (с использованием обрат-

ной связи), определить гормональный профиль и риск опасных видов заболеваний. Таким образом, РОФЭС производит не только оценку, но и управление состоянием здоровья..

В качестве примера современных экспертных систем, использующихся в медицинских учреждениях, можно привести следующие:

1 Экспертная система ДИН. С целью повышения качества диагностики и эффективности лечения критических состояний у детей в Московском НИИ педиатрии и детской хирургии создана экспертная система ДИН, ориентированная на решение задачи распознавания неотложного состояния у детей в терминах синдрома или нескольких синдромов (точнее, характеризующих их состояний, которые отражают степень выраженности синдрома) при предъявлении признаков заболеваний, под которыми следует понимать анамнестические, клинические и лабораторные проявления. База знаний ЭС содержит описания 34 синдромов, которые включают 84 состояния. Для системы это список диагностических предположений-гипотез. База экспертных знаний врача-реаниматолога содержит более 1000 диагностических критериев и заключений о динамике развития неотложного состояния. С целью ускорения работы системы, ориентированной на угрожающие жизни состояния, все множество синдромов подразделено на 14 групп в соответствии с преобладающими нарушениями со стороны той или иной системы жизнеобеспечения, и каждая группа имеет свои дифференцирующие симптомы. Знания о синдромах представлены таким образом, что охватывают: условие выдвижения гипотезы о возможности возникновения синдрома, которое является некоторой логической комбинацией симптомов; клиническую картину, то есть симптомы заболевания; необходимые и достаточные условия для подтверждения гипотезы; информацию о дополнительных синдромах, состоящих в некоторых отношениях с рассматриваемым:

1) причинно-следственные связи, предполагающие информацию о синдромах, которые могут быть причиной данного синдрома или, наоборот, являться его следствием;

2) временные связи, позволяющие как прогнозировать состояние ребенка, так и восстанавливать возможный анамнез болезни;

3) ассоциативные связи, дающие возможность учитывать, на фоне каких состояний может развиваться данный синдром, фоном для каких синдромов он может служить и с какими синдромами может быть совместим, то есть какие синдромы могут встречаться одновременно;

- информацию о состояниях, взаимоисключающих друг друга;

- сведения о дифференцируемых синдромах (синдромах-конкурентах).



Все знания о синдромах подразделяются на декларативные - для описания самого синдрома (клиническая картина, дополнительные синдромы) - и процедурные, указывающие на то, как использовать знания в процессе диагностики.

Функциональная схема ДИН соответствует схеме классической ЭС: интерфейс, блок представления знаний, блок метазнаний, блок механизма логического вывода, блок рабочей области, блок пополнения и модернизации знаний.

2. *Информационно-диагностическая система по наследственным болезням у детей «Диаген».* Система «ДИАГЕН» - предназначена для консультативной помощи врачам при диагностике редко встречающихся и трудно распознаваемых заболеваний наследственной природы на долабораторном этапе обследования. База знаний системы включает сведения о 1200 моногенных и хромосомных болезнях, а модуль визуального представления данных - более 1000 фотографий больных с наследственными болезнями. Система включает три основных блока: диагностический, выдающий дифференциально-диагностический ряд; справочник, который предоставляет полную информацию о признаках и синдромах, хранящихся в системе (ориентирован в первую очередь на врача-педиатра широкого профиля); архив, который обеспечивает хранение и повторное использование данных о диагностированных больных.

Система может быть использована при решении вопроса о предварительном (долабораторном) диагнозе при подозрении на моногенную и хромосомную патологию как справочная система, а также в учебном процессе.

Система "ДИАГЕН" реализована для IBM-совместимых персональных компьютеров. Программное обеспечение написано на языке Borland C++ и функционирует под MS DOS.

### 3 *Экспертная система Мутант.*

Система Мутант обеспечивает проведение первичного скрининга заболеваний с целью получения диагностической информации, необходимой для организации более углубленного обследования пациентов. Предлагаемый подход к дифференциальной диагностике основан на интервьюировании пациента по определенной схеме. Вначале пациенту предлагается ответить на первичный (основной) опросник, включающий паспортные и некоторые антропометрические данные (масса тела, пульс, артериальное давление), личный и семейный анамнез, перечень основных жалоб, обычно приводящих пациента к врачу-терапевту. На основании компьютерной обработки ответов пациента делаются заключения двух

типов:- о наличии риска определенных заболеваний (например, ишемической болезни сердца, сахарного диабета, хронического бронхита, рака легких и др.). В этом случае при отсутствии жалоб указывается, какие дополнительные исследования следует провести;- при наличии тех или иных жалоб пациент адресуется к соответствующему дополнительному (специализированному) опроснику, содержащему углубленную проработку каждого симптома по различным характеристикам, например, точная локализация болей, их характер, иррадиация, продолжительность, сопутствующие явления, условия прекращения и курирования. Для этого по каждому разделу предлагается набор альтернативных ответов. При обработке полученных ответов дополнительно используются такие элементы основного опросника как, например, сведения о поле, возрасте, анамнезе, факторах риска и других жалобах. На этом основании формируется диагностическое заключение в форме синдрома (например, сердечная или легочная недостаточность, синдром малоабсорбации и др.) или определенной нозологической единицы с указанием степени его вероятности в соответствии с количеством информационных синдромов, полученных от больного (50-75%, 76-95%, >95%). Врач должен учитывать, что диагностическое заключение носит рекомендательный характер. В дальнейшем врач может провести более углубленное обследование пациента, заведя в системе данные объективного осмотра пациента (осмотр головы, живота, ног и т.д.) и результаты дополнительных методов исследования (анализы крови, мочи, кала, мокроты и т.д.). В результате система уточняет раннее выставленные диагнозы.

Экспертная система Мутант состоит из следующих подсистем:

- подсистема консультаций (ПК);
- подсистема пополнения знаний (ППЗ).

Моделью представления знаний описываемой ЭС является продукционная система.

Фактологическими знаниями (базой фактов) ЭС Мутант являются:

- карта первичного опроса (КПО) и специализированные опросники (СО);
- список диагностируемых заболеваний;
- список анализов, рекомендуемых системой;
- список анализов, которые могут быть сделаны в данном медицинском учреждении;
- список врачей - специалистов, рекомендуемых системой;
- список врачей - специалистов, которыми располагает данное медицинское учреждение.

Процедурные знания (базы знаний) для системы содержат правила двух типов:

- правило 1-го типа - Если (условие), то (медицинское заключение);
- правило 2-го типа - Если (условие), то (список опросников). Здесь условием является список выбранных экспертом вопросов из карты первичного опроса и одного из специализированных опросников.

Медицинское заключение включает в себя:

- список предполагаемых диагнозов;
- список рекомендуемых системой анализов;
- список рекомендуемых системой врачей-специалистов.

Список опросников содержит номера специализированных опросников, которые необходимо предложить пациенту для уточнения медицинского заключения.

База знаний состоит из двух частей, в каждой из которых хранятся правила только одного типа. Каждое правило представлено одним термом. Все термы, соответствующие правилам, относящимся к одному опроснику, и группируются в одну цепь. Каждое правило при занесении в БЗ автоматически получает уникальный номер, который не может быть модифицирован, и удаляется из БЗ только при удалении этого правила.

Механизм вывода ЭС Мутант заключается в сопоставлении ответов пациента на вопросы карты первичного опроса и специализированных опросников с условной частью правил из БЗ. Поскольку задачей данной системы является выявление всех возможных предварительных заключений, то для ее решения применяется прямой вывод. При этом в зависимости от отношения количества ответов пациента, совпавших с условной частью правила 1-го типа, к общему количеству условий в данном правиле, система различает три степени достоверности диагнозов:

- диагноз маловероятен (50-75%);
- диагноз вероятен (76-95%);
- диагноз весьма вероятен (более 95%).

Для работы со знаниями врач-эксперт использует две подсистемы ведения БЗ, каждая из которых предназначена для работы со знаниями 1-го или, соответственно, 2-го типа.

Каждая из подсистем предлагает эксперту следующие виды работы:

- ввод нового правила в БЗ;
- коррекция правила в БЗ;
- удаление правила из БЗ;
- просмотр БЗ.

Применением базы знаний системы Мутант было ее совместное использование с базой данных медицинских карт. Данный подход позволяет не только хранить результаты сеанса работы системы с пациентом, но и дает возможность врачу занести в БД свои собственные наблюдения, результаты анализов и т.п.

В этих системах реализован интерактивный подход в использовании экспертной системы, когда в результате ответов пациента непосредственно формируется заключение. Однако авторы сочли возможным использование экспертной системы Мутант и в скрытом режиме в качестве контролирующей системы [11, 29].

#### 4. *Экспертная система FIRSTConsult.*

Ресурс FIRSTConsult (прежнее название - PDxMD) является частью MD Consult Clinical Knowledge System. Эта система научно обоснованной электронной клинической информации о первичной медицинской помощи предназначена для улучшения процесса принятия решений специалистами посредством предоставления специализированных диагностических средств и непрерывно обновляемых данных о последних тенденциях в области диагностики, вариантах ведения пациентов и результатах лечения пациентов. Она обеспечивает эффективное принятие решений, сочетая подход на основе передовой практики с гибкостью, необходимой для учета индивидуальных различий между пациентами и предпочтений врача; ускоряет клинические исследования с целью постановки диагноза и оптимизирует результаты, предлагая планирование научно обоснованного лечения и повышая эффективность использования времени и ресурсов; снижает риск ошибок и улучшает клиническую документацию, предоставляя всеобъемлющую и постоянно обновляемую информацию. Ресурс FIRSTConsult доступен в трех форматах: он-лайн, для карманных компьютеров и в печатном. Формат FIRSTConsult Online обновляется еженедельно и состоит из инструмента «Интерактивная дифференциальная диагностика», научно обоснованных Файлов по медицинским состояниям, Файлов с информацией о пациентах, Справочных центров и Процедурных файлов. Формат FIRSTConsult Handheld включает в себя инструмент «Интерактивная дифференциальная диагностика» и Файлов по медицинским состояниям, обновляемых ежеквартально.

Файлы по медицинским состояниям охватывают более 450 медицинских состояний, которые чаще всего наблюдаются врачами первичной медицинской помощи. Каждое состояние включает в себя самую необходимую краткую информацию из других разделов; справочную информацию, включая коды ICD 9 и эпидемиологию; диагностику, включая диф-

ференциальную диагностику, признаки и симптомы, предлагаемые вопросы для истории болезни, ссылки на литературу, полезную при принятии решений медицинскими учреждениями, и клинические тесты, которые можно провести; терапию с кратким перечнем терапевтических вариантов, подробными сведениями о лекарствах и проблемах пациентов и медицинских учреждений; результаты, включая эффективность терапии, прогноз и осложнения; профилактику с указанием факторов риска, рекомендаций в отношении образа жизни и сохранения здоровья, и скрининг; ресурсы, включая основную литературу и ссылку на информацию для пациента; а также научные данные, увязанные с библиографическими ссылками и клиническими руководствами.

Файлы по дифференциальной диагностике позволяют пользователям ознакомиться с таблицей диагностических алгоритмов, в которой перечисляются потенциальные диагнозы с разбивкой по возрасту и распространенности для более чем 330 признаков и симптомов. Потенциальные критические состояния показаны красным. Выбор диагноза выводит на экран краткую сводку, включая клиническую картину начала заболевания, соотношение процентов среди мужчин и женщин, особенности у разных этносов, характер, географические особенности, клиническую картину течения заболевания и сопутствующие заболевания со ссылкой на файлы по медицинским состояниям. Файлы с информацией о пациентах содержат раздаточные материалы для пациентов, написанные в формате вопросов и ответов. Эти материалы имеют ссылку на файлы по медицинскому состоянию и могут быть адаптированы под конкретного пациента. Раздаточные материалы доступны различных языках в различных форматах и предназначаются для пациента, у которого было диагностировано данное состояние, либо для общего сведения.

Рассмотрим **обобщенный алгоритм работы** диагностической экспертной системы.

В алгоритме учитывается, что система медицинской диагностики должна позволять на основе неполных данных ставить диагноз с некоторой точностью, и выдавать, насколько точно она определила болезнь.

Входные данные можно разбить на два больших блока. Первый блок - это данные, поступающие из пользовательского интерфейса. Вторым блоком - это содержимое базы знаний, заполненной экспертом. База знаний должна храниться на жестком диске в виде четырех файлов. Структура базы знаний будет описана более подробно в одном из последующих пунктов.

Таким образом, входными данными для разработки алгоритма программы системы медицинской диагностики являются:

- ответы пользователя на вопросы системы;
- база данных с описаниями симптомов;
- база данных с описанием болезней;
- таблица соответствий между болезнями и симптомами;
- таблица «весов» (вероятностей) симптомов для болезней;

Выходными данными экспертной системы является диагноз, построенный на основе наблюдаемых симптомов и базы знаний о болезнях. Этот диагноз появляется на экран дисплея ПВМ.

В процессе работы система предлагает несколько сгенерированных рабочих версий окончательного диагноза, и по требованию пользователя «отсеивет» лишние гипотезы, которые имеют вес, меньший, чем некоторое значение, заранее заданное разработчиком программы или опытным пользователем.

Например, если в процессе работы сформировалось 4 версии с вероятностями от 71 до 93 %: диагноз №1 – 93%; диагноз № 2 – 86%; диагноз № 3 – 79%; диагноз № 4 – 64% и порог уверенности, заданный разработчиком программы – 80%.,то система показывает все версии, вероятности которых больше или равны 80% (например, их 2 - Диагноз №1 и Диагноз №2 из сформированных).

При заполнении базы знаний необходимо привлекать эксперта.

База знаний представляет собой совокупность следующих баз данных:

- база данных с описаниями диагнозов.
- база данных с описаниями симптомов.
- таблица соответствий симптомов и диагнозов.
- таблица весов симптомов для диагнозов.

Вес – это относительная величина значимости каждого симптома для конкретного диагноза.

Кроме того, в экспертной системе предусмотрен порог уверенности – число, заранее определенное пользователем, означающее максимальный вес диагноза в данном случае, ниже которого гипотезы не рассматриваются.

Например, структуры данных представлены в Paradox 7.0 показаны в таблицах 1–2.

Работу с этими таблицами обеспечивают DBE и DBD, поставляемые в комплекте с Borland Delphi для работы с базами данных распространенных форматов.

Таблица 1 – Таблица диагнозов

Имя поля	Наименование поля	Тип	Длина	Краткое описание
	Num	Integer	9	Идентификационный номер диагноза
	Name	Character	14	Наименование диагноза
	FileName	Character	19	Имя файла с рекомендациями

Таблица 2 – Таблица симптомов

Имя поля	Наименование поля	Тип	Длина	Краткое описание
	Num	Integer		Идентификационный номер
	NameS	Character	15	Наименование симптома
	FileName	Character	12	Имя файла с рекомендациями

Все значения полей таблицы соответствий имеют тип `numeric`. Число в пересечении колонок является «указателем» на тот диагноз, на который указывает определенный симптом, либо на тот симптом, на который указывает определенный диагноз. Всем значениям полей таблицы весов назначается тип `numeric`. Таблица имеет размерность  $[n,m]$ , где  $n$  – число диагнозов,  $m$  – число симптомов. Число в пересечении колонок является весом данного симптома для данного диагноза.

Работа программы начинается с функции инициализации, которая ответственна за присвоение данным начального значения. После чего управление получает процедура «Главного меню». Далее, в зависимости от действий пользователя, происходит вызов одной из ниже перечисленных процедур:

- работа с базами знаний;
- непосредственный запуск экспертной системы;
- настройки;
- контекстно-зависимая справочная система.

На каждом шаге алгоритма у системы предусмотрен «откат», т.е. пользователю предоставлена возможность вернуться в предыдущее состояние системы, нажав на программную кнопку «Назад». На вход системы изначально поступает неполная информация в виде данных, введенных пользователем с клавиатуры. Вследствие чего система не может однозначно поставить диагноз.

Для решения проблемы выбора диагноза используется следующий алгоритм:

1. Сбор предварительной информации. При входе в систему пользователю предоставлен список всех симптомов, имеющих в базе. Напротив выборочных пунктов списка симптомов пользователь должен выставлять «галочки» отмечая симптомы, которые он наблюдает. При нажатии пользователем программной кнопки «Далее» система переходит к следующему пункту.

2. Составление первоначального списка диагнозов, к которым подходит данный набор симптомов. После ввода пользователем симптомов, система анализирует, к каким диагнозам принадлежат данные симптомы по следующему алгоритму:

- а) создается список пустой первоначальный список диагнозов
- б) открывается таблица соответствий;
- в) для каждого  $i$ -го симптома,  $i=1..m$ , где  $m$ –максимальное число симптомов по таблице соответствий циклом делаем полный перебор всех эле-



ментов (диагнозов), стоящих в столбце данного симптома и определяем предварительный диагноз;

г) если диагноз уже имеется в списке диагнозов, то переходим к пункту е);

д) если элемент таблицы равен 0, то диагноз в список диагнозов не включается;

е) переходим к проверке следующего симптома.

ж) если все симптомы (i) перебраны, то первоначальный список болезней сформирован;

з) закрываем таблицу соответствий.

Каждый симптом может принадлежать сразу нескольким диагнозам, поэтому в первоначальный список диагнозов следует уточнить, используя таблицу весов. И следующим пунктом обобщенного алгоритма будет уточнение полученных данных.

### 3. Уточнение первоначального списка диагнозов.

Имея начальный список диагнозов, система осуществляет их дифференциацию. Самыми распространенными методами логического вывода являются прямая цепочка рассуждений (прямой вывод) и обратная цепочка рассуждений (обратный вывод). В основном, при решении задач диагностики используется обратный вывод. Можно сказать, что обратный вывод более эффективен, когда пользователь должен выбирать из набора возможных последствий в случае медицинской диагностики. В предлагаемой системе реализуется механизм смешанного вывода, который позволяет и прямой вывод от фактов к заключениям, и обратный – чтобы подтвердить или опровергнуть гипотезу. В процессе уточнения информации система, задавая пользователю вопросы, осуществляет «отсеивание» лишних гипотез, имеющих малый вес. Для просчета веса гипотез система должна открыть таблицу весов. В таблице весов указывается вес данного симптома для данного диагноза на основе априорных знаний эксперта. Порог уверенности заранее задается в настройках приложения и использованием интуитивно понятного интерфейса.

### 4. Принятие окончательного решения.

В процессе предыдущих шагов выявляется несколько версий окончательного результата, которые система должна распределять по порядку возрастания вероятности того или иного диагноза. Вероятности диагнозов считаются по таблице весов. Предлагается следующий алгоритм подсчитывания веса:

а) выбирается диагноз из списка диагнозов, сформированного на предыдущих этапах;

- б) система просматривает, какие симптомы из списка симптомов имеют отношение к данному диагнозу;
- в) суммируются веса, определенные по таблице весов, всех симптомов, имеющих отношение к данному диагнозу;
- г) фиксирование конечного веса диагноза;
- д) после подсчета весов всех диагнозов выбирается диагноз, имеющий максимальный вес;
- е) система выбирает те диагнозы, оценки вероятности которых выше «порога».

Для конечного подсчета вероятности предлагается суммирование всех элементов таблицы весов имеющих отношение к болезни и нормировка всех конечных вероятностей с целью «укладывания» их в промежуток от 1 до 100 с целью получения окончательного ответа в процентах.

#### 5. Представление конечного результата.

Система предоставляет врачу диагнозы, которые были выбраны в предыдущем пункте в порядке процентного убывания на экран монитора ПВМ и предоставлять пользователю возможность корректировать базу знаний на основе новых экспертных знаний и своего опыта.