

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.ДВ.09.01 Экспертные системы

**Направление подготовки (специальность) 27.03.04 Управление в технических
системах**

**Профиль подготовки (специализация) “Интеллектуальные системы обработки
информации и управления”**

Квалификация выпускника бакалавр

Форма обучения очная

СОДЕРЖАНИЕ

1. Конспект лекций.....	3
1.1 Лекция №1, 2 Основные понятия экспертных систем. Обобщенная структура ЭС.....	3
1.2 Лекция №3, 4 Типы экспертных систем. Статические и динамические ЭС.	4
1.3 Лекция №5, 6 Состав и организация данных и знаний в ЭС. Способы реализации логического вывода в ЭС с классическими моделями представления знаний.	7
1.4 Лекция №7, 8, 9 Нечеткие знания и способы их обработки. Экспертные системы, основанные на нечеткой логике.....	10
2. Методические указания по проведению практических занятий	8
2.1 Практическое занятие №ПЗ-1, 2 Основные понятия экспертных систем. Обобщенная структура ЭС.....	8
2.2 Практическое занятие № ПЗ-3, 4 Типы экспертных систем. Статические и динамические ЭС.	8
2.3 Практическое занятие № ПЗ-5, 6 Состав и организация данных и знаний в ЭС. Способы реализации логического вывода в ЭС с классическими моделями представления знаний.	92
2.4 Практическое занятие № ПЗ-7 Разработка базы знаний статической ЭС на языке Пролог. Ошибка! Закладка не определена.	3
2.5 Практическое занятие № ПЗ-8, 9 Разработка модуля логического вывода статической ЭС на языке Пролог..	33
2.6 Практическое занятие № ПЗ-10, 11 Нечеткие знания и способы их обработки. Экспертные системы, основанные на нечеткой логике.....	34
2.7 Практическое занятие № ПЗ-12 Разработка эвристических правил для ЭС нечеткого вывода.....	34
2.8 Практическое занятие № ПЗ-13 Обоснование функций принадлежности.....	34
2.9 Практическое занятие № ПЗ-14, 15 Модификация нечетких множеств ЭС...	34
2.10 Практическое занятие № ПЗ-16, 17 Разработка ANFIS-адаптивной системы нейро-нечеткого вывода.....	34

1. КОНСПЕКТ ЛЕКЦИЙ

1.1 Лекция № 1, 2 (4 часа).

Тема: «Основные понятия экспертных систем. Обобщенная структура ЭС»

1.1.1 Краткое содержание вопросов:

Экспертная система – это система, которая использует знания специалистов, представленные в виде деревьев вывода, деревьев целей, нечетких множеств, семантических сетей, фреймов и т.д. Поэтому технологию построения экспертных систем называют инженерией знаний. Инженер по знаниям извлекает их из специалистов и встраивает в экспертную систему. Экспертные системы являются одним из результатов исследований в области искусственного интеллекта и рассматриваются в качестве программного средства, которое позволяет представить знания специалистов высокой квалификации о предметной области. Экспертные системы воспроизводят осознанные мыслительные усилия человека.

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Исследователи в области ЭС для названия своей дисциплины часто используют также термин "инженерия знаний", введенный Е.Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

Программные средства (ПС), базирующиеся на технологии экспертных систем, или инженерии знаний (в дальнейшем будем использовать их как синонимы), получили значительное распространение в мире.

Неформализованные задачи обычно обладают следующими особенностями:

- ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;
- ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;
- большой размерностью пространства решения, т.е. перебор при поиске решения весьма велик;
- динамически изменяющимися данными и знаниями.

Следует подчеркнуть, что неформализованные задачи представляют большой и очень важный класс задач. Многие специалисты считают, что эти задачи являются наиболее массовым классом задач, решаемых ЭВМ.

Экспертные системы и системы искусственного интеллекта отличаются от систем обработки данных тем, что в них в основном используются символьный (а не числовой) способ представления, символьный вывод и эвристический поиск решения (а не исполнение известного алгоритма).

Экспертные системы применяются для решения только трудных практических (не игрушечных) задач. По качеству и эффективности решения экспертные системы не уступают решениям эксперта-человека. Решения экспертных систем обладают "прозрачностью", т.е. могут быть объяснены пользователю на качественном уровне. Это качество экспертных систем обеспечивается их способностью рассуждать о своих знаниях и умозаключениях. Экспертные системы способны пополнять свои знания в ходе взаимодействия с экспертом. Необходимо отметить, что в настоящее время технология

экспертных систем используется для решения различных типов задач (интерпретация, предсказание, диагностика, планирование, конструирование, контроль, отладка, инструктаж, управление) в самых разнообразных проблемных областях, таких, как финансы, нефтяная и газовая промышленность, энергетика, транспорт, фармацевтическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

Следует обратить внимание на то, что некоторые специалисты (как правило, специалисты в программировании, а не в ИИ) продолжают утверждать, что ЭС и СИИ не оправдали возлагавшихся на них ожиданий и умерли. Причины таких заблуждений состоят в том, что эти авторы рассматривали ЭС как альтернативу традиционному программированию, т.е. они исходили из того, что ЭС в одиночестве (в изоляции от других программных средств) полностью решают задачи, стоящие перед заказчиком. Надо отметить, что на заре появления ЭС специфика используемых в них языков, технологии разработки приложений и используемого оборудования (например, Lisp-машины) давала основания предполагать, что интеграция ЭС с традиционными, программными системами является сложной и, возможно, невыполнимой задачей при ограничениях, накладываемых реальными приложениями. Однако в настоящее время коммерческие инструментальные средства (ИС) для создания ЭС разрабатываются в полном соответствии с современными технологическими тенденциями традиционного программирования, что снимает проблемы, возникающие при создании интегрированных приложений.

Экспертная система работает в двух режимах: режиме приобретения знаний и в режиме решения задачи (называемом также режимом консультации или режимом использования ЭС).

В режиме приобретения знаний общение с ЭС осуществляется (через посредничество инженера по знаниям) экспертом. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной области. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования с данными, характерные для рассматриваемой области.

В режиме консультации общение с ЭС осуществляется конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам), или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к ЭС с целью либо ускорить процесс получения результата, либо возложить на ЭС рутинную работу). В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операции, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения:

"Почему система задает тот или иной вопрос?", "как ответ, собираемый системой, получен?".

Применение ЭС может быть оправдано одним из следующих факторов:

- использование человека-эксперта невозможно либо из-за недостаточного количества экспертов, либо из-за необходимости выполнять экспертизу одновременно в различных местах;
- использование ЭС целесообразно в тех случаях, когда при передаче информации эксперту происходит недопустимая потеря времени или информации;

– использование ЭС целесообразно при необходимости решать задачу в окружении, враждебном для человека.

Приложение соответствует методам ЭС, если решаемая задача обладает совокупностью следующих характеристик:

1) задача может быть естественным образом решена посредством манипуляции с символами (т.е. с помощью символических рассуждений), а не манипуляций с числами, как принято в математических методах и в традиционном программировании;

2) задача должна иметь эвристическую, а не алгоритмическую природу, т.е. ее решение должно требовать применения эвристических правил. Задачи, которые могут быть гарантированно решены (с соблюдением заданных ограничений) с помощью некоторых формальных процедур, не подходят для применения ЭС;

3) задача должна быть достаточно сложна, чтобы оправдать затраты на разработку ЭС. Однако она не должна быть чрезмерно сложной (решение занимает у эксперта часы, а не недели), чтобы ЭС могла ее решать;

4) задача должна быть достаточно узкой, чтобы решаться методами ЭС, и практически значимой.

1.2 Лекция № 3, 4 (2 часа).

Тема: «Типы экспертных систем. Статические и динамические ЭС»

1.2.1 Краткое содержание вопросов:

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Решения экспертных систем обладают "прозрачностью", т.е. могут быть объяснены пользователю на качественном уровне. Это качество экспертных систем обеспечивается их способностью рассуждать о своих знаниях и умозаключениях. ЭС способны пополнять свои знания в ходе взаимодействия с экспертом.

ЭС подразделяются на статические и динамические. Статические ЭС используются в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Статическая система состоит из базы данных, предназначенной для хранения исходных и промежуточных данных решаемой в текущий момент задачи, базы знаний, которая хранит долгосрочные данные, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области, решателя, формирующего такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи, компонента приобретения знаний, автоматизирующего процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом, Объяснительного компонента, который объясняет, как система получила решение и Диалогового компонента, ориентированного на организацию дружественного общения с пользователем как в ходе решения задач. В динамической ЭС учитываются изменения окружающей среды, происходящие за время решения задачи. В архитектуру этой системы по сравнению со статической ЭС вводятся два компонента: подсистема моделирования внешнего мира и подсистема связи с внешним окружением. Последняя осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

В настоящее время технология экспертных систем используется для решения различных типов задач (интерпретация, предсказание, диагностика, планирование,

конструирование, контроль, отладка, инструктаж, управление) в самых разнообразных проблемных областях, таких, как финансы, нефтяная и газовая промышленность, энергетика, транспорт, фармацевтическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

1.3 Лекция № 5, 6 (4 часа).

Тема: «Состав и организация данных и знаний в ЭС. Способы реализации логического вывода в ЭС с классическими моделями представления знаний»

1.3.1 Краткое содержание вопросов:

Типичная статическая ЭС состоит из следующих основных компонентов:

- блока логического вывода и решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базы знаний (БЗ);
- блока приобретения знаний;
- блока объяснений;
- интерфейса.

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС)

системах управления базами данных (СУБД) для обозначения всех данных (в первую очередь долгосрочных), хранимых в системе.

База знаний. Центральным элементом экспертной системы является база знаний, которая отражает знания специалиста-эксперта в соответствии с какой-либо моделью (деревья вывода, деревья целей, семантические сети и т.д.). (БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

Блок логического вывода (решатель), используя исходные данные из базы данных (рабочей памяти) и знаний из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

Блок приобретения знаний автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом.

Блок объяснений объясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Интерфейс ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

Блок логического вывода, необходимый, для обработки базы знаний с целью получения новых знаний.

1.4 Лекция № 7, 8, 9 (6 часов).

Тема: «Нечеткие знания и способы их обработки. Экспертные системы, основанные на нечеткой логике»

1.4.1 Краткое содержание вопросов:

Обширной областью эффективного применения интеллектуальных систем как средства построения информационных систем нового поколения является область нечетких знаний. Это связано с тем, что во всех предметных областях существенное место занимают некорректные, нечетко формулируемые задачи и реальный человеческий способ рассуждения (опирающийся на естественный язык) не может быть описан в рамках традиционных математических формализмов, предполагающих однозначность интерпретации. Другими словами, знания чаще всего нечетки. «Для того, чтобы ИС вышли за рамки простых символных выводов и приблизились к мышлению человека, необходимы методы представления нечетких знаний и механизмы выводов, работающие в их среде.» Исии. Возникла необходимость создания теории, позволяющей формально описывать нестрогие, нечеткие понятия и моделировать рассуждения, содержащие такие понятия.

Все недоопределённости в знаниях можно классифицировать следующим образом:

- 1) недетерминированность выводов,
- 2) многозначность,
- 3) ненадежность,
- 4) неполнота,
- 5) нечеткость или неточность. Именно этому классу уделяется внимание ниже.

Нечеткими называются такие знания, которые допускают суждения об относительной степени истинности или ложности.

Типы, источники, причины нечеткости знаний:

- присутствие неопределенности в фактическом знании,
- неточность языка представления знаний,
- знания, основанные на неполной информации,
- неопределенность, появляющаяся при агрегации (объединении в одну систему) знаний, полученных из разных источников и пр.

В последнее десятилетие всё больше внимания уделяется подходу, основанному на теории нечетких множеств. Эту теорию предложил ам. учёный Лофти Заде в 1965 году. Главная идея подхода Заде заключается в использовании для моделирования рассуждений нечеткой логики. Заде ввел одно из главных понятий в нечеткой логике – понятие лингвистической переменной.

Использование этого подхода позволяет построить «нечёткие» аналоги основных математических понятий и создать формальный аппарат для моделирования человеческого способа решения задач.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

2.1 Практическое занятие №1, 2 (4 часа).

Тема: «Основные понятия экспертных систем. Обобщенная структура ЭС»

2.1.1 Краткое описание проводимого занятия:

В ходе работ по созданию ЭС сложилась определенная технология их разработки, включающая шесть следующих этапов: идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию.

На этапе идентификации определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей.

На этапе концептуализации проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

На этапе формализации выбираются ИС и определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями.

На этапе выполнения осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и наиболее трудоемким этапом разработки ЭС. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

2.2 Практическое занятие №3, 4 (4 часа).

Тема: «Типы экспертных систем. Статические и динамические ЭС»

2.2.1 Краткое описание проводимого занятия:

Все экспертные системы можно разделить на

- статические;
- динамические.

Если задачи, решаемые экспертной системой не учитывают фактор времени и не изменяют в процессе решения данные, то такие системы называют статической. Если в процессе решения задач такой фактор учитывается, то система называется динамической.

В экономике экспертные системы используются для консультаций по выработке инвестиционных решений, выбору стратегии маркетинга, кредитованию юридических лиц и т. д. С их помощью осуществляется мониторинг производственных, логистических, маркетинговых и других процессов. При необходимости они запускают механизм логического вывода для устранения критических ситуаций с одновременным информированием управленческого персонала.

Статические ЭС используются в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Первые ЭС, получившие практическое использование, были статическими.

2.3 Практическое занятие №5, 6 (4 часа).

Тема: «Состав и организация данных и знаний в ЭС. Способы реализации логического вывода в ЭС с классическими моделями представления знаний»

2.3.1 Краткое описание проводимого занятия:

В разработке ЭС участвуют представители следующих специальностей:

- эксперт в проблемной области, задачи которой будет решать ЭС;
- инженер по знаниям - специалист по разработке ЭС (используемые им технологию, методы называют технологией (методами) инженерии знаний);
- программист по разработке инструментальных средств (ИС), предназначенных для ускорения разработки ЭС.

Необходимо отметить, что отсутствие среди участников разработки инженеров по знаниям (т. е. их замена программистами) либо приводит к неудаче процесса создания ЭС, либо значительно удлиняет его.

Эксперт определяет знания (данные и правила), характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭС знаний.

Инженер по знаниям помогает эксперту выявить и структурировать знания, необходимые для работы ЭС; осуществляет выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС; выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области), которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает ИС (если ИС разрабатывается заново), содержащее в пределе все основные компоненты ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

2.4 Практическое занятие №7 (2 часа).

Тема: «Разработка базы знаний статической ЭС на языке Пролог»

2.4.1 Краткое описание проводимого занятия:

Переменная – последовательность букв, цифр и знака «подчеркивание», обязательно начинаящаяся с прописной (большой) буквы или знака «подчеркивание». Среди переменных выделена переменная "_" (один знак подчеркивания), называемая анонимной. Она используется в ситуациях, когда нас не интересует её значение.

Область действия переменных

Областью действия переменной является утверждение. В пределах утверждения одно и то же имя принадлежит одной и той же переменной. Два утверждения могут использовать одно имя переменной совершенно различным образом. Единственным исключением из правила определения области действия переменных является анонимная переменная, например, "_" в цели любит(X,_). Каждая анонимная переменная есть отдельная сущность.

Она применяется тогда, когда конкретное значение переменной несущественно для данного утверждения. Таким образом, каждая анонимная переменная четко отличается от всех других анонимных переменных в утверждении. Переменные, отличные от анонимных, называются именованными, а неконкретизированные (переменные, которым не было присвоено значение) называются свободными.

Структура представляется на языке Пролог с помощью указания её функтора и компонент в следующем виде:

Имя структуры = функтор(компонент-1, ..., компонента-N),

где в качестве функтора должен выступать атом, а компонентой может быть любой терм (в том числе и структура).

Если объекты структуры относятся к одному и тому же типу доменов, то такая структура называется однодоменной. Если объекты структуры относятся к разным типам доменов – многодоменная.

Объекты могут быть константами, переменными или структурами. Для того, чтобы объединить объекты в структуры, надо выбрать функтор. Каждый функтор определяется двумя параметрами:

- именем, синтаксис которого совпадает с именем предиката;
- арностью – т. е. числом аргументов.

Рассмотрим в структуру date:

date (1, september, 2002),

где date – функтор, все аргументы данной многодоменной структуры являются константами.

date (Day, september, 2002)

В этом примере date представляет функтор многодоменной структуры, аргументы которой являются переменные и константы.

likes ('Katja', fruits(banana, apples, oranges)),

А в этом примере likes – главный функтор, аргументы данной многодоменной структуры – константа и структура.

Рассмотренную структуру в программе можно описать следующим образом:

domains

personal_loking = fruits(type1,type2, type3)

type1, type2, type3 = symbol

predicates

likes(symbol, personal_loking)

2.5 Практическое занятие №8, 9 (4 часа).

Тема: «Разработка модуля логического вывода статической ЭС на языке Пролог»

2.5.1 Краткое описание проводимого занятия:

В программе на Прологе важен порядок *предложений* внутри процедуры, а также порядок хвостовых *целей* в теле *предложений*. От порядка *предложений* зависит порядок поиска решений и порядок, в котором будут находиться ответы на *вопросы*. Порядок *целей* влияет на количество проверок, выполняемых программой при решении.

При описании правил часто возникает необходимость использовать логические связки И и ИЛИ. В качестве связки И используется запятая, а в качестве связки ИЛИ – точка с запятой.

Например:

gigant(X) :- rost(X,Y), Y>200.

star_or_mlad(X) :- X>70; X<10.

ПРОЛОГ имеет большое количество встроенных предикатов, т.е. предикаты, определяемые автоматически. Например, встроенный предикат nl вызывает перевод строки, а встроенный предикат write применяется для вывода информации на экран. Встроенные предикаты используются так же, как и определяемые пользователем предикаты, но встроенный предикат не может являться головой правила или появляться в факте. Часто используемыми встроенными предикатами являются = (унификация) и логическое отрицание not.

Например:

```
student(X) :- X="Петров", X="Иванов".  
xor_student(X) :- not(X="Петров"), not(X="Иванов").  
planeta(X) :- not(X="солнце").  
Утверждение not(X = Y) эквивалентно X<>Y.
```

Иногда бывает полезно использовать предикаты, про которые заранее известно, истинны они или ложны. Для этих целей используют предикаты true и fail. Предикат true всегда истинен, в то время как fail всегда ложен. Последний предикат используется для управления процессом решения задачи на ПРОЛОГе.

ПРОЛОГ-программа может использовать комментарии, которые не влияют на выполнение программы, но могут оказать помощь человеку, читающему программу. ПРОЛОГ игнорирует произвольное число строк, заключенное между символами /* и */. Все, что находится между % и концом строки, также рассматривается как комментарий.

2.6 Практическое занятие №10, 11 (4 часа).

Тема: «Нечеткие знания и способы их обработки. Экспертные системы, основанные на нечеткой логике»

2.6.1 Краткое описание проводимого занятия:

Недетерминированность выводов. Это характерная черта большинства интеллектуальных информационных систем. Недетерминированность означает, что заранее путь решения конкретной задачи в пространстве ее состояний определить невозможно. Поэтому в большинстве случаев методом проб и ошибок выбирается некоторая цепочка логических заключений, согласующихся с имеющимися знаниями, а в случае если она не приводит к успеху, организуется перебор с возвратом для поиска другой цепочки и т.д.

Например, выезжая на автомобиле, следует учитывать состояние дорог, транспорта, погодные условия и т.д. При нарушении одного из предположений, например, из-за пробки на обычном маршруте, планы меняются и выбирается альтернативный маршрут.

Многозначность. Многозначность интерпретации — обычное явление в задачах распознавания. При понимании естественного языка серьезными проблемами становятся многозначность смысла слов, их подчиненности, порядка слов в предложении и т.п. Проблемы понимания смысла возникают в любой системе, взаимодействующей с пользователем на естественном языке. Распознавание графических образов также связано с решением проблемы многозначной интерпретации.

Неточность и ненадежность знаний и выводов. Как было отмечено выше, количественные данные (знания) могут быть неточными. Неточность в основном связана с объективными причинами: несовершенство измерительных приборов (школьной линейкой нельзя измерять объекты меньше миллиметра или больше километра), несоблюдения условий проведения замеров (повышенная или пониженная температура, влажность и т.п.) и т.д. При этом существуют различные способы оценки такой неточности, разрабатываемые в рамках теории измерений.

Ненадежность знаний в большей степени связана с субъективными причинами: отсутствием формальных процедур получения точных данных, вероятностной природой поступающих данных, недостаточной математической (логической) обоснованностью используемых правил и т.д. Она может относиться как к количественным, так и качественным показателям. Ненадежность означает, что для оценки достоверности знаний

нельзя применить двухбалльную шкалу (1 – абсолютно надежные, 0 – недостоверные). Например, рост человека можно охарактеризовать как «маленький», «средний», «высокий» и т.п., а оценка разных людей этого показателя будет зависеть от их субъективных соображений. Более того, они могут быть не до конца уверены в своей оценке.

В связи с этими особенностями, в интеллектуальных информационных систем применяют вероятностные оценки тех или иных знаний, как в части фактов, так и правил вывода. Так, утверждение $p(\text{высокий(вася)}) = 0.75$ можно интерпретировать как вероятность того, что Вася высокий на три четверти истинна. Утверждение $p(\text{ЕСЛИ настроение_преподователя} = \text{«хорошее» И знание_ответа_на_билет} = \text{«нулевые» ТО оценка_за_экзамен} = \text{«не меньше 3»}) = 0.8$ определяет вероятность истинности правила.

Неполнота знаний и немонотонная логика. Абсолютно полных знаний не бывает, поскольку процесс познания бесконечен. В связи с этим состояние базы знаний должно изменяться с течением времени. В отличие от простого добавления информации, как в базах данных, при добавлении новых знаний возникает опасность получения противоречивых выводов, т.е. выводы, полученные с использованием новых знаний, могут опровергать те, что были получены ранее.

Многие экспертные системы первого поколения были основаны на модели закрытого мира, обусловленной применением аппарата формальной логики для обработки знаний. Модель закрытого мира предполагает жесткий отбор знаний, включаемых в базу, а именно база знаний заполняется исключительно верными понятиями, а все, что ненадежно или неопределенно, заведомо считается ложным. Другими словами, все, что известно базе знаний, является истиной, а остальное – ложью. Такая модель имеет ограниченные возможности представления знаний и таит в себе опасность получения противоречий при добавлении новой информации. Тем не менее, эта модель достаточно распространена. Например, на ней базируется язык Prolog.

В системах, построенных по принципу модели закрытого мира, добавление новых фактов не нарушает справедливость ранее полученных выводов. Это свойство логических выводов называется монотонностью. К сожалению, реальные знания, закладываемые в интеллектуальных информационных системах, крайне редко бывают полными.

2.7 Практическое занятие №12 (2 часа).

Тема: «Разработка эвристических правил для ЭС нечеткого вывода»

2.7.1 Краткое описание проводимого занятия:

Понятие нечеткого вывода занимает центральное место в нечеткой логике и в теории нечеткого управления. Говоря о нечеткой логике в системах управления, можно дать следующее определение системы нечеткого вывода.

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 (рис.1).



Рис.1. Диаграмма процесса нечеткого вывода в нечетких САУ

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.

Нечеткое продукционное правило – это выражение вида:

$$(i): Q; P; A \Rightarrow B; S, F, N,$$

где (i) – имя нечеткой продукции, Q – сфера применения нечеткой продукции, P – условие применимости ядра нечеткой продукции, $A \Rightarrow B$ – ядро нечеткой продукции, в котором A – условие ядра (или антецедент), B – заключение ядра (или консеквент), \Rightarrow – знак логической секвенции или следования, S – метод или способ определения количественного значения степени истинности заключения ядра, F – коэффициент определенности или уверенности нечеткой продукции, N – постусловия продукции.

Сфера применения нечеткой продукции Q описывает явно или неявно предметную область знания, которую представляет отдельная продукция.

Условие применимости ядра продукции P представляет собой логическое выражение, как правило предикат. Если оно присутствует в продукции, то активизация ядра продукции становится возможной только в случае истинности этого условия. Во многих случаях этот элемент продукции может быть опущен или введен в ядро продукции.

Ядро $A \Rightarrow B$ является центральным компонентом нечеткой продукции. Оно может быть представлено в одной из более распространенных форм: «ЕСЛИ A ТО B», «IF A THEN B»; где A и B – некоторые выражения нечеткой логики, которые наиболее часто представляются в форме нечетких высказываний. В качестве выражений могут использоваться и составные логические нечеткие высказывания, т.е. элементарные нечеткие высказывания, соединенные нечеткими логическими связками, такими как нечеткое отрицание, нечеткая конъюнкция, нечеткая дизъюнкция.

S – метод или способ определения количественного значения степени истинности заключения B на основе известного значения степени истинности условия A. Данный

способ определяет схему или алгоритм нечеткого вывода в производственных нечетких системах и называется методом композиции или методом активации.

Коэффициент уверенности F выражает количественную оценку степени истинности или относительный вес нечеткой продукции. Коэффициент уверенности принимает свое значение из интервала $[0;1]$ и часто называется весовым коэффициентом нечеткого правила продукции.

Постуловие нечеткой продукции N описывает действия и процедуры, которые необходимо выполнить в случае реализации ядра продукции, т.е. получения информации об истинности B . Характер этих действий может быть самым различным и отражать вычислительный или иной аспект производственной системы.

Согласованное множество нечетких производственных правил образует нечеткую производственную систему. Таким образом, нечеткая производственная система – это относящийся к определенной предметной области список нечетких производственных правил «IF A THEN B ».

Простейший вариант нечеткого производственного правила:

ПРАВИЛО $<\#>$: ЕСЛИ β_1 «ЕСТЬ α_1 » ТО « β_2 ЕСТЬ α_2 »

RULE $<\#>$: IF « β_1 IS α_1 » THEN « β_2 display:block IS α_2 ».

Антецедент и консеквент ядра нечеткой продукции может быть сложным, состоящим из связок «И», «ИЛИ», «НЕ», например:

ПРАВИЛО $<\#>$: ЕСЛИ « β_1 ЕСТЬ α » И « β_2 ЕСТЬ НЕ α » ТО « β_1 ЕСТЬ НЕ β_2 »

RULE $<\#>$: IF « β_1 IS α » AND « β_2 IS NOT α » THEN « β_1 IS NOT β_2 ».

Наиболее часто база нечетких производственных правил представляется в форме согласованного относительно используемых лингвистических переменных структурированного текста:

ПРАВИЛО_1: ЕСЛИ «Условие_1» ТО «Заключение_1» (F_1 т),

...

ПРАВИЛО_n: ЕСЛИ «Условие_n» ТО «Заключение_n» (F_n),

где $F_i \in [0;1]$ является коэффициентом определенности или весовым коэффициентом соответствующего правила. Согласованность списка означает, что в качестве условий и заключений правил могут использоваться только простые и составные нечеткие высказывания, соединенные бинарными операциями «И», «ИЛИ», при этом в каждом из нечетких высказываний должны быть определены функции принадлежности значений терммножества для каждой лингвистической переменной. Как правило, функции принадлежности отдельных термов представляют треугольными или трапецидальными функциями.

2.8 Практическое занятие №13 (2 часа).

Тема: «Обоснование функций принадлежности»

2.8.1 Краткое описание проводимого занятия:

Существует свыше десятка типовых форм кривых для задания функций принадлежности. Наибольшее распространение получили: треугольная, трапецидальная и гауссова функции принадлежности.

Треугольная функция принадлежности определяется тройкой чисел (a, b, c) , и ее значение в точке x вычисляется согласно выражению:

$$MF(x) = \begin{cases} 1-b-xb-a, & a \leq x \leq b \\ 1-x-bc-b, & b \leq x \leq c \\ 0, & x \notin (a; c) \end{cases} \quad MF(x) = \begin{cases} 1-b-xb-a, & a \leq x \leq b \\ 1-x-bc-b, & b \leq x \leq c \\ 0, & x \notin (a; c) \end{cases}$$

При $(b-a)=(c-b)$ имеем случай симметричной треугольной функции принадлежности, которая может быть однозначно задана двумя параметрами из тройки (a, b, c) .

Аналогично для задания трапецидальной функции принадлежности необходима четверка чисел (a, b, c, d) :

$$MF(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{x-c}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases}$$

$$MF(x) = \begin{cases} 1-b-x & a \leq x \leq b \\ 1 & b \leq x \leq c \\ 1-x & c \leq x \leq d \\ 0 & x > d \end{cases}$$

При $(b-a)=(d-c)$ трапецидальная функция принадлежности принимает симметричный вид.

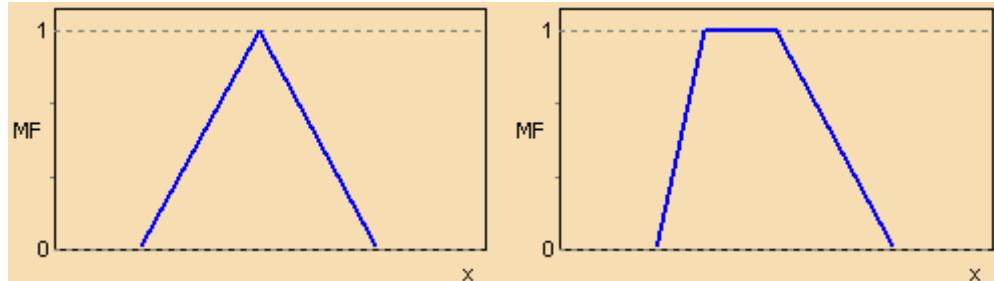


Рисунок 1. Типовые кусочно-линейные функции принадлежности

Функция принадлежности гауссова типа описывается формулой

$$MF(x) = \exp[-(x-c\sigma)^2] \quad MF(x) = \exp[-(x-c\sigma)^2]$$

и оперирует двумя параметрами. Параметр c обозначает центр нечеткого множества, а параметр отвечает за крутизну функции.

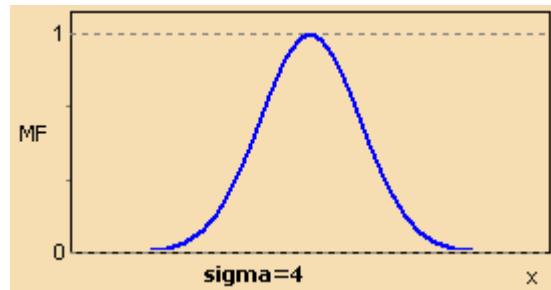


Рисунок 2. Гауссова функция принадлежности

Совокупность функций принадлежности для каждого терма из базового терм-множества T обычно изображаются вместе на одном графике. На рисунке 3 приведен пример описанной выше лингвистической переменной "Цена акции", на рисунке 4 – формализация неточного понятия "Возраст человека". Так, для человека 48 лет степень принадлежности к множеству "Молодой" равна 0, "Средний" – 0,47, "Выше среднего" – 0,20.



Рисунок 3. Описание лингвистической переменной "Цена акции"

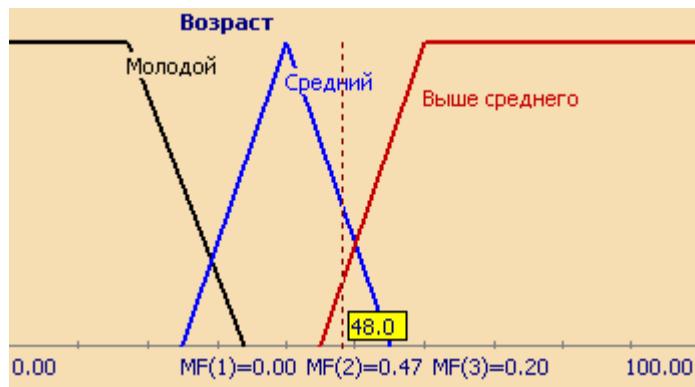


Рисунок 4. Описание лингвистической переменной "Возраст"

Количество термов в лингвистической переменной редко превышает 7.

2.9 Практическое занятие №14, 15 (4 часа).

Тема: «Модификация нечетких множеств ЭС»

2.9.1 Краткое описание проводимого занятия:

Нечеткое множество(fuzzyset) представляет собой совокупность элементов произвольной природы, относительно которых нельзя точно утверждать – обладают ли эти элементы некоторым характеристическим свойством, которое используется для задания нечеткого множества.

Пусть X – универсальное (базовое) множество, x – элемент X , а R – некоторое свойство. Обычное (четкое) подмножество A универсального множества X , элементы которого удовлетворяют свойству R , определяется как множество упорядоченных пар $A = \mu_A x / x$, где $\mu_A x$ – характеристическая функция, принимающая значение 1, если x удовлетворяет свойству R , и 0 – в противном случае.

Нечеткое подмножество отличается от обычного тем, что для элементов x из X нет однозначного ответа «да-нет» относительно свойства R . В связи с этим, нечеткое подмножество A универсального множества X определяется как множество упорядоченных пар $A = \mu_A x / x$, где $\mu_A x$ – характеристическая функция принадлежности (или просто функция принадлежности), принимающая значения в некотором вполне упорядоченном множестве $M = 0 ; 1$. Функция принадлежности указывает степень (или уровень) принадлежности элемента x подмножеству A . Множество M называют множеством принадлежностей. Если $M = 0 ; 1$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество. Степень принадлежности $\mu_A x$ является субъективной мерой того, насколько элемент $x \in X$, соответствует понятию, смысл которого формализуется нечетким множеством A .

Носителем нечеткого подмножества S_A универсального множества X со свойством $\mu_A x > 0$, т.е. $S_A = x \mid x \in X \wedge \mu_A x > 0$. Иными словами, носителем нечеткого множества A является подмножество S_A универсального множества X , для элементов которого функция принадлежности $\mu_A x > 0$ больше нуля. Иногда носитель нечеткого множества обозначают $support A$.

Если носителем нечеткого множества A является дискретное подмножество S_A , то нечеткое подмножество A универсального множества X , состоящего из n элементов, можно представить в виде объединения конечного числа одноточечных множеств $\mu_A x / x$ при помощи символа $\sum : A = \sum_{i=1}^n \mu_A x_i / x_i$. При этом

подразумевается, что элементы x_i упорядочены по возрастанию в соответствии со своими индексами, т.е. $x_1 < x_2 < x_3 < \dots < x_n$.

Если носителем нечеткого множества A является непрерывное подмножество S_A , то нечеткое подмножество A универсального множества X , рассматривая символ \int как непрерывный аналог введенного выше символа объединения для дискретных нечетких множеств \sum , можно представить в виде объединения бесконечного числа одноточечных множеств $\mu_A(x)/x$:

$$A = \int X \mu_A(x)/x.$$

Пример. Пусть универсальное множество X соответствует множеству возможных значений толщин изделия от 10 мм до 40 мм с дискретным шагом 1мм. Нечеткое множество A , соответствующее нечеткому понятию «малая толщина изделия», может быть представлено в следующем виде:

$$A = 1 / 10 ; 0,9 / 11 ; 0,8 / 12 ; 0,7 / 13 ; 0,5 / 14 ; 0,3 / 15 ; 0,1 / 16 ; 0 / 17 ; \dots ; 0 / 40 ,$$

$$A = 1 / 10 + 0,9 / 11 + 0,8 / 12 + 0,7 / 13 + 0,5 / 14 + 0,3 / 15 + 0,1 / 16 + 0 / 17 + \dots + 0 / 40 ,$$

где знак суммирования обозначает не операцию арифметического сложения, а объединения элементов в одно множество. Носителем нечеткого множества A будет конечное подмножество (дискретный носитель):

$$S_A = 10 ; 11 ; 12 ; 13 ; 14 ; 15 ; 16 .$$

Если же универсальное множество X является множеством действительных чисел от 10 до 40, т.е. толщина изделия может принимать все возможные значения в этих пределах, то носителем нечеткого множества A является отрезок $S_A = 10 ; 16$.

Нечеткое множество с дискретным носителем может быть представлено в виде отдельных точек на плоскости, нечеткое множество с непрерывным носителем может быть представлено в виде кривой, что соответствует дискретной и непрерывной функциям принадлежности $\mu_A(x)$, заданным на универсальном множестве X (рис.1).

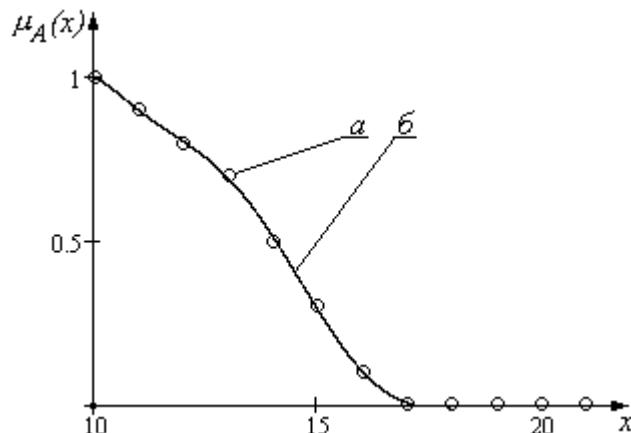


Рис.1. Функции принадлежности нечетких множеств с (а)-дискретным и (б)-непрерывным носителями

Пример. Пусть $X = 0 ; 1 ; 2 ; \dots$ – множество целых неотрицательных чисел. Нечеткое множество A малый можно определить как $\mu_A(x) = x/2 - 1$.

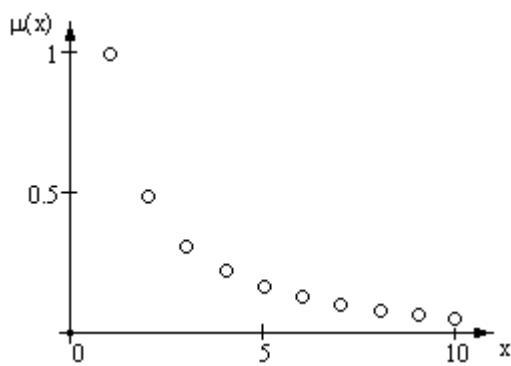


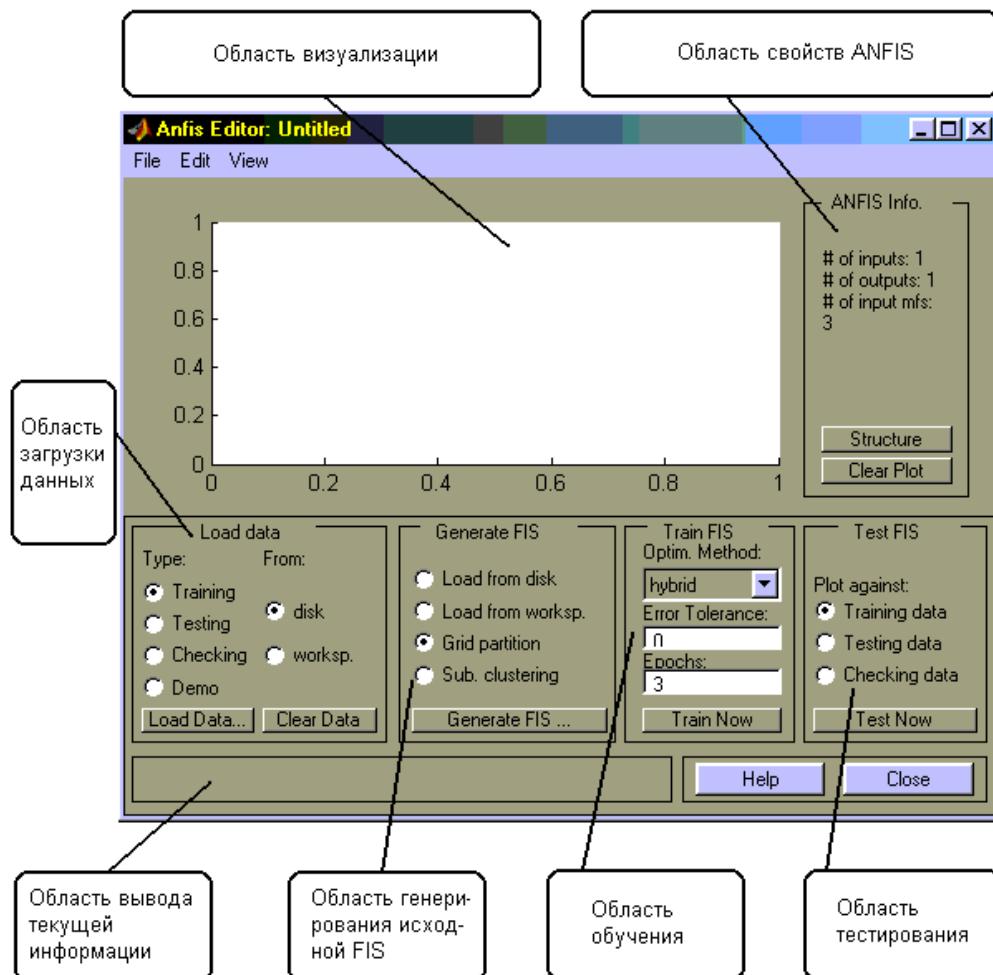
Рис.2. Графическое представление нечеткого множества малый

2.10 Практическое занятие №16, 17 (4 часа).

Тема: «Разработка ANFIS-адаптивной системы нейро-нечеткого вывода»

2.10.1 Краткое описание проводимого занятия:

ANFIS является аббревиатурой Adaptive Neuro-Fuzzy Inference System – (адаптивная нейро-нечеткая система). ANFIS-редактор позволяет автоматически синтезировать из экспериментальных данных нейро-нечеткие сети. Нейро-нечеткую сеть можно рассматривать как одну из разновидностей систем нечеткого логического вывода типа Сугэно. При этом функции принадлежности синтезированных систем настроены (обучены) так, чтобы минимизировать отклонения между результатами нечеткого моделирования и экспериментальными данными. Загрузка ANFIS-редактора осуществляется по команде **anfisedit**.



ANFIS-редактор содержит 3 верхних меню - **File**, **Edit** и **View**, область визуализации, область свойств ANFIS, область загрузки данных, область генерирования исходной системы нечеткого логического вывода, область обучения, область тестирования, область вывода текущей информации, а также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть **ANFIS**-редактор, соответственно.

Область визуализации

В этой области выводится два типа информации:

- при обучении системы – кривая обучения в виде графика зависимости ошибки обучения от порядкового номера итерации.
- при загрузке данных и тестировании системы – экспериментальные данные и результаты моделирования.

Экспериментальные данные и результаты моделирования выводятся в виде множества точек в двумерном пространстве. При этом по оси абсцисс откладывается порядковый номер строчки данных в выборке (обучающей, тестирующей или контрольной), а по оси ординат – значение выходной переменной для данной строчки выборки. Используются следующие маркеры:

- голубая точка (.) – тестирующая выборка;
- голубая окружность (o) – обучающая выборка;
- голубой плюс (+) – контрольная выборка;
- красная звездочка (*) – результаты моделирования.

Область свойств ANFIS

В области свойств ANFIS (**ANFIS info**) выводится информация о количестве входных и выходных переменных, о количестве функций принадлежностей для каждой входной переменной, а также о количестве строчек в выборках. В этой области расположены две кнопки **Structure** и **Clear Plot**.

Область загрузки данных

В области загрузки данных (**Load data**) расположены:

- меню выбора типа данных (**Type**), содержащее альтернативы:
 - **Traning** - обучающая выборка;
 - **Testing** - тестирующая выборка;
 - **Checking** - контрольная выборка;
 - **Demo** - демонстрационный пример;
- меню выбора источника данных (**From**), содержащее альтернативы:
 - **disk** – диск;
 - **worksp.** - рабочая область MatLab;
- кнопка загрузки данных **Load Data...**, по нажатию которой появляется диалоговое окно выбора файла, если загрузка данных происходит с диска, или окно ввода идентификатора выборки, если загрузка данных происходит из рабочей области;
- кнопка очистки данных **Clear Data**.

Примечание. В течении одной сеанса работы **ANFIS**-редактора можно загружать данные одного формата, т.е. количество входных переменных в выборках должно быть одинаковым.

Область генерирования исходной системы нечеткого логического вывода

В области генерирования (**Generate FIS**) расположены меню выбора способа создания исходной системы нечеткого логического вывода. Меню содержит следующие альтернативы:

- **Load from disk** – загрузка системы с диска;
- **Load from worksp.** – загрузка системы из рабочей области MatLab;
- **Grid partition** - генерирование системы по методу решетки (без кластеризации);

- **Sub. clustering** – генерирование системы по методу субкластеризации.

В области также расположена кнопка **Generate**, по нажатию которой генерируется исходная система нечеткого логического вывода.

При выборе **Load from disk** появляется стандартное диалоговое окно открытия файла.

При выборе **Load from worksp.** появляется стандартное диалоговое окно ввода идентификатора системы нечеткого логического вывода.

Range of influence – уровни влияния входных переменных;

Squash factor – коэффициент подавления;

Accept ratio – коэффициент, устанавливающий во сколько раз потенциал данной точки должен быть выше потенциала центра первого кластера для того, чтобы центром одного из кластеров была назначена рассматриваемая точка;

Reject ratio – коэффициент, устанавливающий во сколько раз потенциал данной точки должен быть ниже потенциала центра первого кластера, чтобы рассматриваемая точка была исключена из возможных центров кластеров.

Область обучения

В области обучения (**Train FIS**) расположены меню выбора метода оптимизации (**Optim. method**), поле задания требуемой точности обучения (**Error tolerance**), поле задания количества итераций обучения (**Epochs**) и кнопка **Train Now**, нажатие которой запускает режим обучение. Промежуточные результаты обучения выводятся в область визуализации и в рабочую область MatLab. В ANFIS-редакторе реализованы два метода обучения:

- **backpropa** – метод обратного распространения ошибки, основанный на идеях метода наискорейшего спуска;
- **hybrid** – гибридный метод, объединяющий метод обратного распространения ошибки с методом наименьших квадратов.

Область тестирования

В области тестирования (**Test FIS**) расположены меню выбора выборки и кнопка **Test Now**, по нажатию по которой происходит тестирование нечеткой системы с выводом результатов в область визуализации.

Область вывода текущей информации

В этой области выводится наиболее существенная текущая информация, например, сообщения об окончании выполнении операций, значение ошибки обучения или тестирования и т.п.