И.Л. КОРОБОВА, Г.В. АРТЕМОВ

ПРИНЯТИЕ РЕШЕНИЙ В СИСТЕМАХ, ОСНОВАННЫХ НА ЗНАНИЯХ

ИЗДАТЕЛЬСТВО ТГТУ

Учебное издание

КОРОБОВА Ирина Львовна
АРТЕМОВ Геннадий Владимирович

ПРИНЯТИЕ РЕШЕНИЙ
В СИСТЕМАХ, ОСНОВАННЫХ
НА ЗНАНИЯХ

Учебное пособие

Редактор З.Г. Чернова
Компьютерное макетирование М.А. Филатовой

Подписано в печать 20.01.05
Формат 60 × 84 / 16. Бумага офсетная. Печать офсетная.
Гарнитура Times New Roman. Объем: 4,65 усл. печ. л.; 4,75 уч.-изд. л.
Тираж 100 экз. С. 15

Издательско-полиграфический центр
Тамбовского государственного технического университета,
392000, Тамбов, Советская, 106, к. 14

Министерство образования и науки Российской Федерации

Тамбовский государственный технический университет

И.Л. КОРОБОВА, Г.В. АРТЕМОВ

ПРИНЯТИЕ РЕШЕНИЙ В СИСТЕМАХ, ОСНОВАННЫХ НА ЗНАНИЯХ

Утверждено Ученым советом университета
в качестве учебного пособия

Тамбов
Издательство ТГТУ
2005

УДК 004.891(07)
ББК □813я73
К68

Рецензент
Доцент, кандидат технических наук
М.Н. Краснянский

Коробова И.Л.
К68 Методы представления знаний: Учеб. пособие / И.Л. Коробова, Г.В. Артемов. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2005. 80 с.

Рассматриваются общие сведения по структуре и составу экспертных систем. Даны рекомендации по различным способам представления знаний в экспертных системах. Рассматриваются различные методы принятия решения в экспертных системах.

Учебное пособие по дисциплине "Интеллектуальные подсистемы в САПР" предназначено для студентов 5 курса дневного отделения специальности 2203.

УДК 004.891(07)
ББК □813я73

ISBN 5-8265-0342-4

ВВЕДЕНИЕ

Система искусственного интеллекта (ИИ) – это программная система, имитирующая на компьютере мышление человека. Для создания такой системы необходимо изучить процесс мышления человека, решающего определенные задачи или принимающего решение в конкретной области, выделить основные шаги этого процесса и разработать программные средства, воспроизводящие их на компьютере.

Среди систем искусственного интеллекта, широко внедряемых в область автоматизации проектирования, следует выделить, так называемые, экспертные системы (ЭС), в основе которых находится обширный запас знаний и экспертных оценок о конкретной предметной области.

Знания, которыми обладает специалист в какой-либо области можно разделить на формализованные и неформализованные. Формализованные знания формулируются в книгах, руководствах, документах в виде общих и строгих суждений (законов, формул, моделей, алгоритмов и т.п.).

Неформализованные знания обычно не попадают в книги и руководства в связи с их конкретностью, субъективностью и приблизительностью. Знания этого рода являются результатом обобщения многолетнего опыта работы и интуиции специалиста. Они обычно представляют собой множество эмпирических приемов и правил. Как правило, неформализованные задачи обладают неполнотой, ошибочностью, неоднозначностью и противоречивостью знаний.

Традиционное программирование в качестве основы для разработки программ использует алгоритм, т.е. формализованное знание.

Экспертные системы не отвергают и не заменяют традиционного подхода к программированию. Они отличаются от традиционных программ тем, что ориентированы на решение неформализованных задач.

Знания в экспертных системах могут быть представлены в различном виде. Предлагаемое пособие, в рамках учебной дисциплины "Интеллектуальные подсистемы САПР" для студентов специальности 22.03, предназначено для изучения методов представления знаний в экспертных системах и дальнейшего их использования при принятии решений в процессе автоматизированном проектировании.

1 ОБЩИЕ СВЕДЕНИЯ ОБ ЭКСПЕРТНЫХ СИСТЕМАХ

Любая система искусственного интеллекта опирается на знания о процессе человеческого мышления [1 – 3].

1.1 ОСНОВНЫЕ СОСТАВЛЯЮЩИЕ ПРОЦЕССА МЫШЛЕНИЯ

Цели. В основе человеческой деятельности лежит мышление. Когда утром звонит будильник, мозг человека дает команду руке выключить звонок. Это не автоматическая реакция – решение конкретной задачи требует определенного ответа мозга. *Целью* называется конечный результат, на который направлены мыслительные процессы человека. Как только цель (выключить звонок) достигнута, перед человеком сразу встают новые цели, например, почистить зубы, одеться, позавтракать, выйти на автобусную остановку. Осуществление всех этих целей приводит к достижению главной цели – вовремя попасть на работу. Мысли, ведущие к конечному результату, не случайны, а строго обоснованы. Каждый шаг на пути к главной цели имеет свою локальную цель. Мозг всегда сосредоточен на цели, независимо от того, выполняет ли человек простую физическую работу или решает сложную интеллектуальную задачу.

При проектировании экспертной системы всегда следует помнить о цели, для достижения которой она предназначена [2].

Факты и правила. Человек хранит большое количество знаний. В общем случае, интеллект можно представить как совокупность фактов и правил их использования. Отчасти цели достигаются с помощью правил использования всех известных фактов.

П р и м е р 1.1:

Факт 1: Температура кипения воды 100 °С.

Правило 1: Если температура процесса меньше 100 °С, то для нагрева можно использовать воду.

Правило 2: Если температура процесса больше 100 °С, то для нагрева используют электронагреватель.

Заметим, что в приведенном примере все правила выражены условным отношением вида: *если…то…*, т.е. если выполняется некоторое условие, то следует некоторое действие [2].

Упрощение. Когда человеческий мозг приступает к решению даже самой простой задачи, для выбора нужных действий в его распоряжении имеется огромный объем информации. Например, переходя улицу, человек анализирует скорость и объем движения, расстояние до противоположного тротуара, сигнал светофора. Одновременно мозг обрабатывает впечатления, не имеющие прямого отношения к переходу улицы (например, цвет проезжающих машин, вид деревьев и окружающих зданий, одежду проходящих мимо людей и пр.). Если бы человек, прежде чем шагнуть на проезжую часть, анализировал все факты, имеющие хоть какое-нибудь отношение к цели, он простоял бы на тротуаре несколько лет. Но мышление человека включает сложную систему, руководящую выбором правильной реакции на конкретную ситуацию. Такой выбор называется упрощением. Механизм упрощения блокирует факты и правила, не имеющие прямого отношения к решаемой в данный момент задаче [2]. Схематично работа механизма упрощения представлена на рис. 1.

Механизм вывода. Достигая цели, человек не только приходит к решению поставленной перед ним задачи, но и одновременно приобретает новые знания. Часть интеллекта, которая помогает извлекать новые факты, называется механизмом вывода [2].

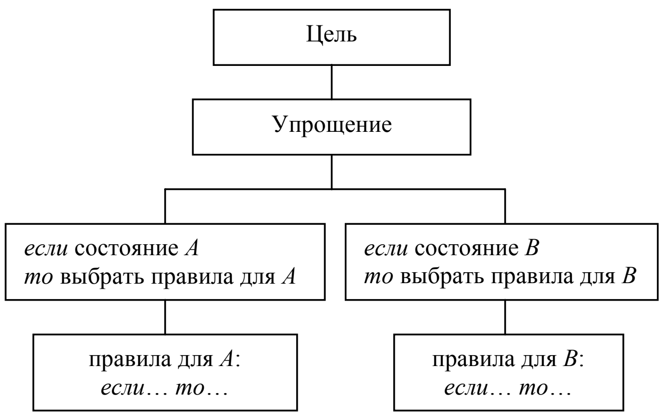


Рис. 1

Пример 1.2.

Факт 1: Мария и Петр – родители Вани.

Факт 2: Мария и Петр – родители Тани.

Правило 1: Если у мальчика и девочки одни и те же родители, то дети – брат и сестра.

Цель: Определить степень родства Тани и Вани.

С помощью имеющихся фактов и правил цель может быть достигнута сразу. Кроме того, в процессе достижения цели получен новый факт:

Новый факт: Ваня и Таня – брат и сестра.

1.2 СТРУКТУРА И СОСТАВ ЭКСПЕРТНОЙ СИСТЕМЫ

Любая система искусственного интеллекта должна содержать все элементы, составляющие процесс принятия решения человеком: цели, знания, механизмы вывода и упрощения [1 – 4].

Порядок работы таких систем (т.е. систем, основанных на правилах) представлен на рис. 2.

Итак, экспертная система – это система ИИ, созданная для решения задач в конкретной предметной области. Источником знаний для наполнения ЭС служат литература, отчеты по НИР и знания экспертов, которые являются наиболее ценными.

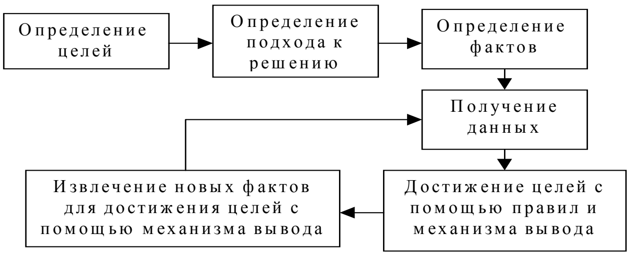


Рис. 2

Типовая ЭС включает в себя следующие основные компоненты:

1) базу знаний;
2) механизм вывода;
3) модуль извлечения знаний;
4) систему объяснений.

На рис. 3 представлена структура ЭС.

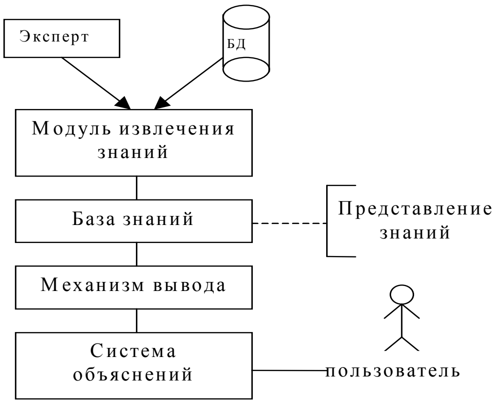


Рис. 3

1.3 БАЗА ЗНАНИЙ

Важнейшей составляющей экспертной системы является база знаний (БЗ), содержащая факты и правила, по которым в зависимости от входной информации принимается то или иное решение.

Факты представляют собой краткосрочную информацию, которая может изменяться в процессе решения задачи. Правила представляют более долговременную информацию о том, как порождать новые факты и гипотезы из имеющихся данных.

Основное отличие БЗ от обычной методики использования базы данных (БД) состоит в больших творческих возможностях. Факты в БД обычно пассивны: они либо там есть, либо их там нет. БЗ, со своей стороны, активно пытается пополнить недостающую информацию.

Правила в формате *если ... то ...* являются распространенным, но не единственным способом представления знаний. Для этой цели в ЭС используются семантические сети, фреймы, нейронные сети и другие способы. Они будут рассмотрены далее. На некотором глубоком уровне все типы представления данных должны быть, очевидно, эквивалентны между собой. Выбор того или иного способа определяется видом задачи и спецификой предметной области.

Многие правила в ЭС являются эвристическими, основанными на опыте экспертов в данной предметной области. Если алгоритмический метод гарантирует корректное или оптимальное решение, то эвристический метод дает в большинстве случаев лишь приемлемое решение.

База знаний является входным потоком данных для механизма логического вывода.

1.4 МЕХАНИЗМ ЛОГИЧЕСКОГО ВЫВОДА

Механизмом логического вывода называются общие знания о процессе нахождения решения. Он выполняет две основные функции:

1) дополнение, изменение БЗ на основе анализа БЗ и исходной информации;
2) управление порядком обработки правил в БЗ.

Если база знаний содержит высококачественные знания о предметной области, то механизм логического вывода содержит информацию о том, как эти знания эффективно использовать.

Если в процессе создания ЭС удается достаточно просто сформулировать базу знаний, то выбор стратегии логического вывода представляет собой достаточно сложную задачу. Это связано и с отсутствием простого и общего метода организации логического вывода, и зависит от специфики предметной области, и от того, как в БЗ структурированы и организованы знания о предметной области.

Механизм логического вывода функционирует циклически. В каждом цикле решаются следующие задачи:

- сопоставление – предполагает сравнение условных частей правил с исходными данными и имеющимися фактами в БЗ;
- выбор – в случае наличия множества правил с истинностью условных частей необходимо выбрать одно из них для срабатывания;
- действие – предполагает выполнение какого-либо действия, предусмотренного в случае срабатывания правила, что обычно приводит к выполнению какого-либо физического действия и к модификации базы знаний.

Таким образом, каждый цикл начинается с последовательного просмотра всех правил и сопоставления их условных частей с исходными данными и фактами в БЗ. Если правил, у которых условные части и факты совпадают, несколько, то возникает конфликтное множество правил. На основе каких-либо критериев выбирается одно правило, которое считается сработавшим, и выполняется действие.

Существует две основные стратегии логического вывода [2, 4]:

- прямая цепочка рассуждений – основана на сопоставлении исходных данных с правилами и фактами БЗ с получением результата;
- обратная цепочка рассуждений – предполагается, что выдвигается некоторая гипотеза о предполагаемом решении задачи и путем анализа БЗ ищется подтверждение этой гипотезы путем сравнения результатов с исходными данными. Если гипотеза не подтверждается, то ищется новое решение.

Наиболее ценными являются ЭС, которые реализуют и прямую и обратную цепочки рассуждений.

1.5 МОДУЛЬ ИЗВЛЕЧЕНИЯ ЗНАНИЙ

Важной составной частью ЭС является модуль извлечения знаний. Его основное назначение – предоставление экспертных знаний, их структурирование в виде, пригодном для использования в компьютерной системе. В задачу модуля входит приведение правила к виду, позволяющему применить это правило в процессе работы. В простейшем случае в качестве такого модуля может выступать обычный редактор, который просто заносит правила в файл.

В некоторых системах извлечение знаний осуществляется не одним, а несколькими способами, например, часть знаний извлекается с помощью программных средств, анализирующих грамматику описания знаний (эта грамматика задает форму представления знаний); другие знания могут быть представлены графически и потребуются специальные средства, которые позволят воспринимать графически изображения и проверять их на правильность (например графически могут быть представлены электрические схемы); наконец, возможны знания, которые самой системой не используются, а при необходимости могут вводиться в диалоговом режиме.

Модуль извлечения знаний является наиболее трудоемким и дорогостоящим.

1.6 СИСТЕМА ОБЪЯСНЕНИЙ

Система объяснений предназначена для показа пользователю всего процесса рассуждений, в результате которого было найдено или не найдено решение.

Большинство специалистов – пользователей ЭС – не смогут с доверием относиться к выведенному системой заключению, пока не будут знать, как оно было получено. Если врач установил у вас наличие некоторого заболевания, то вы, конечно, захотите знать, почему он пришел к такому выводу. Вы, вероятно, попросите показать вам рентгеновский снимок, результаты анализов или что-то другое, на основе чего врач сделал свое заключение. К экспертной системе предъявляются те же самые требования, т.е.

необходимо получить не только само решение, но и всю цепочку вывода в форме, понятной пользователю.

2 ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ЭКСПЕРТНЫХ СИСТЕМАХ

Наиболее общими методами представления знаний в ЭС являются:

– правила;
– семантические сети;
– фреймы.

Кроме того, в данном разделе рассматривается представление знаний в виде нечетких правил, а также в виде нейронных сетей.

2.1 ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ВИДЕ ПРАВИЛ

Правила являются наиболее понятным и популярным методом представления знаний [1 – 4]. Они чаще подходят в тех случаях, когда предметные знания возникают из эмпирических ассоциаций, накопленных за годы работы по решению задач в данной области.

В ЭС, основанных на правилах, предметные знания представляются набором правил, которые проверяются на группе фактов и знаний о текущей ситуации (входной информации). Когда часть правила *если* удовлетворяет фактам, то действия, указанные в части *то*, выполняется. Когда это происходит, то говорят, что правило срабатывает. Интерпретатор правил сопоставляет части правил *если* с фактами и выполняет то правило, часть *если* которого сходится с фактами, т.е. интерпретатор правил работает в цикле "сопоставить – выполнить", формируя последовательность действий (рис. 4).

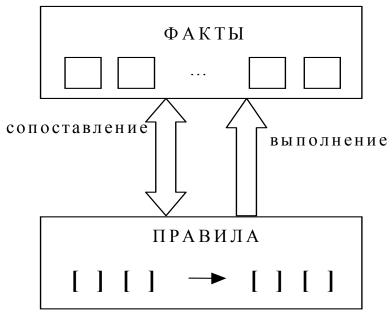


Рис. 4

Действия правил могут состоять:

– в модификации набора фактов в базе знаний, например добавление нового факта, который сам может быть использован для сопоставления с частями *если*;
– во взаимодействии с внешней средой (например "Вызвать пожарную команду").

П р и м е р 2.1: Рассмотрим базу знаний, которая включает три факта и девять правил:

Факты:

1 Кислота = агрессивная жидкость;
2 Азотная кислота = кислота;
3 Сернистая кислота = кислота

Правила:

1 *если* среда = кислота *и* концентрация >70 %
то среда = концентрированная кислота;
2 *если* среда = кислота *и* концентрация <70 %
то среда = разбавленная кислота;
3 *если* среда = концентрированная кислота
то материал ванны = хромоникелевая сталь;
4 *если* среда = разбавленная кислота

то материал ванны = углеродистая сталь;

5 *если* среда = агрессивная жидкость *то* футеровка = есть;

6 *если* футеровка = есть *и* среда = сернистая кислота
то материал футеровки = свинец;

7 *если* футеровка = есть *и* среда = азотная кислота
то материал футеровки = винипласт;

8 *если* температура = меньше 100 °С
то тип обогрева = пароводяная рубашка;

9 *если* температура = больше 100 °С
то тип обогрева = электронагреватель

Процесс сопоставления с фактами частей *если* порождает цепочку выводов. Эта цепочка выводов показывает, как система, используя правила, выводит заключение. Цепочки выводов ЭС могут быть предъявлены пользователю, что помогает понять, как система достигает свои заключения.

Правила, по сравнению с другими способами представления знания имеют следующие преимущества [2, 4]:

1) модульность;

2) единообразие структуры;

3) естественность (вывод заключения в такой системе аналогичен процессу рассуждения эксперта);

4) гибкость иерархии понятий, которая поддерживается только как связи между правилами (изменив правило, вы можете изменить иерархию).

Однако, такие системы не свободны от недостатков:

1) процесс вывода менее эффективен, чем при других способах представления, так как большая часть времени затрачивается на непроизводительную проверку применимости правил;

2) этот процесс трудно поддается управлению;

3) сложно представить иерархию понятий.

Представление знаний в виде правил иногда называют плоским (по аналогии с реляционными базами данных), так как в них отсутствуют средства для установления иерархии правил. Объем базы знаний растет линейно по мере включения в нее новых фрагментов знаний. Большинство существующих коммерческих ЭС основаны на правилах. При этом правила могут быть представлены в одном из двух видов [5].

1 Если в зависимости от возможных четких значений входных параметров делается вывод о значениях выходного параметра, то такая система называется системой $L^{(1)}$ – типа. Данная система представляется в виде

$$L^{(1)} = \begin{cases} L_1^{(1)} :< \text{если } A_1 \text{ то } B_1 > \\ L_2^{(1)} :< \text{если } A_2 \text{ то } B_2 > \\ \dots \\ L_m^{(1)} :< \text{если } A_m \text{ то } B_m >, \end{cases} \qquad (1)$$

где m – число экспертных высказываний; A_j – четкое значение входного параметра; B_j – четкое значение выходного параметра или некоторое конкретное действие процесса проектирования.

2 В случаях, когда в зависимости от возможных значений выходной ситуации (B_j) экспертом делается предположение о возможной входной ситуации (A_j), система экспертных высказываний называется системой
$L^{(2)}$ – типа и представляется в виде:

$$L^{(2)} = \begin{cases} L_1^{(2)} :< \text{если } B_1 \text{ то } A_1 > \\ L_2^{(2)} :< \text{если } B_2 \text{ то } A_2 > \\ \dots \\ L_m^{(2)} :< \text{если } B_m \text{ то } A_m >. \end{cases} \qquad (2)$$

2.2 ПРЕДСТАВЛЕНИЕ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМОВ

Системы, базы знаний которых насчитывают сотни правил, отнюдь не считаются чем-то необычным [1]. При такой сложности системы для инженера знаний процесс обновления состава правил и контроль связей между ними становится весьма затруднительным, поскольку добавляемые правила могут дублировать имеющиеся знания или вступать с ними в противоречие. Для выявления подобных фактов можно использовать программные средства, но включение их в работу системы приводит к еще более тяжелым последствиям – потере работоспособности, так как в этом случае инженер знаний теряет представление о том, как взаимодействуют правила; возрастает количество связей между понятиями, инженеру знаний трудно их контролировать.

Представление знаний, основанных на фреймах [1 – 3], является альтернативным по отношения к системам, основанным на правилах: оно дает возможность хранить иерархию понятий в базе знаний в явной форме.

Фреймом называется структура для описания стереотипной ситуации, состоящая из характеристик этой ситуации и их значений.

Характеристики называются слотами, а значения – заполнителями слотов. Слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одно или несколько правил, с помощью которых это значение можно найти.

В слот может входить не одно, а несколько значений. Иногда слот включает компонент, называемый фасетом, который задает диапазон или перечень его возможных значений.

Как уже отмечалось, помимо конкретного значения, в слоте могут храниться процедуры и правила, которые вызываются при необходимости вычисления этого значения. Если, например, фрейм, описывающий человека, включает слоты "*Дата рождения*" и "*Возраст*", и в первом из них находится некоторое значение, то во втором слоте может стоять прцедура, вычисляющая возраст по дате рождения и текущей дате.

Процедуры, располагающиеся в слоте, называются связанными процедурами. В предыдущем примере связанная процедура будет активизироваться при каждом изменении текущей даты.

Чаще всего используются процедуры трех видов:

1) "если – добавлено" – выполняется, когда новая информация помещается в слот;
2) "если – удалено" – выполняется, когда информация удаляется из слота;
3) "если – нужно" – выполняется, когда запрашивается информация из слота, а он пустой.

Эти процедуры могут проверять, что при изменении значения производятся соответствующие действия.

Совокупность фреймов, моделирующая какую-нибудь предметную область, представляет собой иерархическую структуру, в которую соединяются фреймы. На верхнем уровне иерархии находится фрейм, содержащий наиболее общую информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать значения характеристик своих родителей, находящихся на более высоком уровне иерархии. Значения характеристик фреймов могут передаваться по умолчанию фреймам, находящимся ниже них в иерархии, но если последние содержат собственные значения данных характеристик, то в качестве истинных данных принимаются именно они. Это обстоятельство позволяет легко учитывать во фреймовых системах различного рода исключения.

Различают статические и динамические системы фреймов. В системах статических фреймы не могут быть изменены в процессе решения задачи, в динамических системах это допустимо.

Наиболее ярко достоинства фреймовых систем представления знаний проявляются в том случае, если связи между объектами изменяются нечасто и предметная область насчитывает немного исключений. Значения слотов представляются в системе в единственном экземпляре, поскольку включается только в один фрейм, описывающий наиболее общее понятие из всех тех, которые содержат слот с данным именем. Такое свойство систем фреймов дает возможность уменьшить объем памяти, необходимый для их размещения в компьютере. Однако основное достоинство состоит не в экономии памяти, а в представлении в базе знаний связей, существующих между понятиями предметной области.

На рис. 5 приведен фрагмент базы знаний о свойствах горения нитей.

Рис. 5

Данный фрагмент основан на следующих знаниях:

1) анидные и капроновые нити являются нитями из полиамидного волокна;
2) нити из синтетических волокон включают полиамидные, лавсановые и акриловые нити;
3) нити из синтетических волокон горят;
4) продуктом сгорания большинства нитей из синтетических волокон является твердый шарик;
5) нити из полиамидного волокна горят медленно;
6) нити из лавсана и акрила горят быстро;
7) при сгорании акриловой нити образуется рыхлый шарик;
8) при горении анидной и акриловой нити запах отсутствует;
9) при горении капроновой нити чувствуется запах сургуча;
10) при горении лавсановой нити чувствуется сладковатый запах.

В результате с помощью базы знаний можно получить новые знания, например: капроновая нить горит медленно, при горении чувствуется запах сургуча, после сгорания образуется твердый шарик.

2.3 Представление знаний с использованием семантических сетей

Термин "семантическая сеть" используется для описания метода представления знания, основанного на сетевой структуре [1 – 3]. Этот метод является одним из наиболее эффективных методов хранения знаний. Семантические сети состоят из:

- *узлов*, соответствующих объектам, понятиям и событиям;
- *дуг*, связывающих узлы и описывающих отношения между ними.

Иными словами, семантическая сеть отображает совокупность объектов предметной области и отношений между ними. При этом, объектам соответствуют вершины сети, а отношениям – соединяющие их дуги.

В семантическую сеть включаются только те объекты предметной области, которые необходимы для решения прикладных задач. В качестве объектов могут выступать события, действия, обобщенные понятия или свойства объектов.

Вершины сети соединяются дугой, если соответствующие объекты предметной области находятся в каком-либо отношении. Наиболее распространенными являются следующие типы отношений:

- "является" – означает, что объект входит в состав данного класса;
- "имеет" – позволяет задавать свойства объектов.

Возможны также отношения вида:

- "является следствием" – отражает причинно-следственные связи;
- "имеет значение" – задает значение свойств объектов.

П р и м е р 2.2: Требуется составить семантическую сеть для отображения следующих знаний:

1) оборудование для выполнения основных операций электрохимических покрытий представляет собой основное технологическое оборудование гальванических комплексов;

2) ванны цинкования и меднения – это оборудование для выполнения операций электрохимических покрытий;

3) ванны цинкования имеют защитную футеровку и вытяжную вентиляцию;

4) в состав вытяжной вентиляции входят бортовые отсосы;

5) оборудование для выполнения основных операций электрохимических покрытий имеет аппаратуру управления и регулирования р-тока и теплообменные аппараты.

Из построенной семантической сети (рис. 6) вытекают, например, следующие дополнительные факты:

1) ванны цинкования являются основным технологическим оборудованием гальванических комплексов;

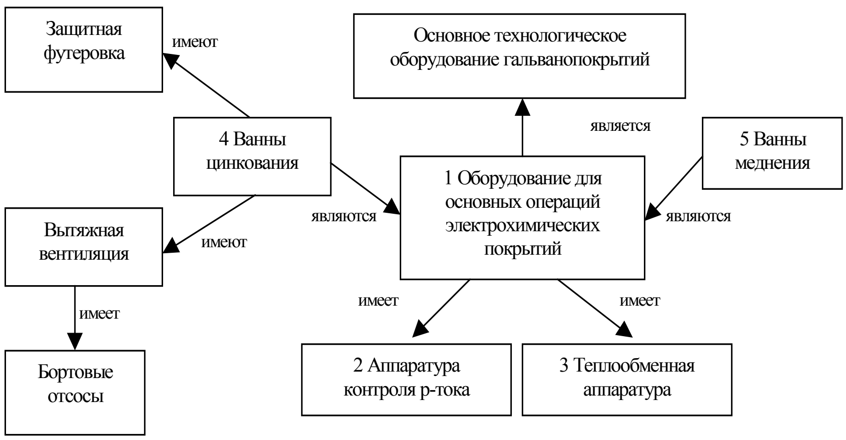


Рис. 6

2) бортовые отсосы входят в состав ванны цинкования;

3) ванны меднения имеют теплообменную аппаратуру.

Свойства семантических сетей наследовать узлами более высокого уровня свойств узлов более низкого уровня, принято называть иерархией наследования. Принцип иерархии наследования позволяет исключать дублирование информации в семантических сетях. Например, достаточно один раз связать узлы 1, 2 и 3 (рис. 6), чтобы не повторять информацию для узлов 4 и 5 и т.д.

Основной недостаток такого способа представления знаний – сложность обработки исключений.

2.4 НЕЙРОННЫЕ СЕТИ

Моделирование сложных систем требует большого числа знаний об объекте, в том числе экспериментальных и экспертных. Для их обработки в последнее время широко используются нейронные сети. В литературе встречаются несколько типов информационных моделей на основе нейронных сетей [8, 9]:

– моделирование отклика системы на внешнее воздействие;
– классификация внутренних состояний системы;
– прогноз динамики изменения системы;
– оценка полноты описания системы и определение значимости параметров системы;
– оптимизация параметров системы по отношению к заданной целевой функции;
– управление системой.

В ряде случаев нейронные сети и физико-математические модели могут составлять единую модель, например, когда внешние условия описываются уравнениями кинетики, а отклик системы – нейронной сетью. Иногда используются гибридные нейронные модели, параметры которых являются нечеткими.

В основе теории нейронных сетей лежит желание воспроизвести функции мозга при решении конкретной задачи. Однако создающиеся системы не полностью воспроизводят функции мозга, а, скорее, представляют математическую модель, воспроизводящую отдельные возможности человеческого мозга, по аналогии с которым искусственные нейтронные сети характеризуются следующими свойствами [8]:

– обучение (т.е. изменение поведения в зависимости от окружающей среды).
– обобщение (реакция сети после обучения будет, до известной степени, нечувствительна к малым изменениям входящих сигналов).
– абстрагирование (способность выявления различий во входных сигналах).

2.4.1 Описание биологического нейрона

Из нейробиологии известно, что человеческий мозг состоит из $10^{10}…10^{11}$ нейронов. На рис. 7 схематично представлен один биологический нейрон. Он содержит клеточное тело и отростки (аксон и дендриты).

Клеточное тело состоит из ядра и окружающей его цитоплазмы. На внешней поверхности содержится мембрана, включающая три слоя. Она отделяет клеточное тело от окружающих его крайних окончаний аксона.

Аксон (выход) – отросток нейрона, который служит для передачи нервных импульсов к другим нейронам или эффекторным органам (мускульным волокнам, клеткам желез).

Дендриты (входы) – отростки, которые связывают нейрон с другими нейронами. Связь осуществляется через специальные контакты, называемыми *синапсами*.

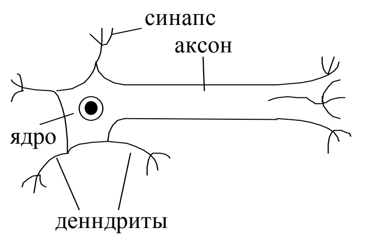


Рис. 7

В упрощенном виде работу нейрона можно представить так. Клеточное тело принимает входной сигнал от других нейронов через синаптические связи дендритов, преобразует его и передает выходной сигнал через аксон другим нейронам. Скорость передачи зависит и от значений входных сигналов, и от силы синаптических связей. Несмотря на то, что функция нейрона – нелинейная, нейробиологи считают, что большинство нейронов производят линейную аппроксимацию, т.е. выходной сигнал нейрона пропорционален, в некоторой степени, линейной комбинации значений входных сигналов.

2.4.2 Искусственный нейрон

Отдельный обрабатываемый элемент искусственной нейронной сети называется искусственным нейроном [8]. Каждый нейрон производит относительно простую работу. На его вход поступает набор сигналов
$X = [x_1, x_2, ..., x_n]$, каждый из которых может быть выходом от другого нейрона или другого источника. Каждый вход умножается на соответствующий угловой коэффициент $W = [w_1, w_2, ..., w_n]$, который соответствует силе синапса биологического нейрона и поступает на вход суммирующего блока, где все произведения $w_i x_i$ суммируются. По этой величине определяется общий вход нейрона

$$h = \sum_i (w_i x_i + \theta_i), \tag{3}$$

где θ_i – пороговая величина i-го нейрона.

Для определения выхода нейрона O (рис. 8) используется функция активации

$$O = F(h) = F\left[\sum_i (w_i x_i + \theta_i)\right]. \qquad (4)$$

Наиболее типичными функциями активации являются:
– экспоненциальная

$$F(h) = \frac{2}{1+\exp(-\lambda h)} - 1, \quad \lambda > 0; \qquad (5)$$

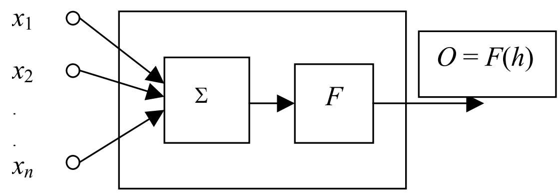


Рис. 8

– функция знака

$$F(h) = \mathrm{sgn}(h) = \begin{cases} +1, & h > 0; \\ -1, & h < 0. \end{cases} \qquad (6)$$

Функции, записанные в таком виде, называются биполярными. Возможно использование униполярных функций:

$$F(h) = \frac{1}{1+\exp(-\lambda h)}, \quad \lambda > 0; \qquad (7)$$

$$F(h) = \begin{cases} 1, & h > 0; \\ 0, & h < 0. \end{cases} \qquad (8)$$

Следует отметить, что при $\lambda \to \infty$ экспоненциальная функция приближается к функции знака.

2.4.3 Многослойные нейронные сети

Для решения практических задач часто используются многослойные нейронные сети. Обычно, в таких сетях все нейроны в слое связаны со всеми нейронами в предыдущем слое через однонаправленную связь [18]. При решении задач аппроксимации чаще используется нейронная сеть с одним скрытым слоем (рис. 9).

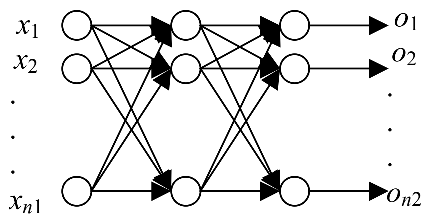


Рис. 9

Многослойная нейронная сеть имеет разное количество нейронов в слоях и разные весовые коэффициенты нейронов. Каждый нейрон характеризуется множеством входов и одним выходом. Связь вход-выход для сети, представленной на рис. 9, можно представить в матричной форме:

$$O = F(X) = f\,[W^2 f\,[W^1 X]], \qquad (9)$$

где X – вектор входных параметров; O – вектор выходных параметров; W^1, W^2 – матрицы весовых коэффициентов для скрытого и выходного слоя соответственно; f – функция активации.

При решении задачи аппроксимации обычно используется экспоненциальная функция (3), (5).

3 ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ВИДЕ НЕЧЕТКИХ ВЫСКАЗЫВАНИЙ

Методы построения математических моделей часто основаны хотя и не неточной, но в целом объективной информации об объекте. Однако возможны ситуации, когда при построении моделей решающее значение имеют сведения, полученные от эксперта, обычно качественного характера. Они отражают содержательные особенности изучаемого объекта и формулируются на естественном языке. Описание объекта в таком случае носит нечеткий характер. Например, экспертом предоставлена следующая информация:

- *Если* концентрация кислоты мала и температура раствора не высокая, *то* выбирается первый вариант конструкции аппарата.
- *Если* концентрация кислоты мала и температура раствора высокая, *то* выбирается второй вариант конструкции аппарата.
- *Если* концентрация кислоты большая и температура раствора высокая, *то* выбирается третий вариант конструкции аппарата.

Пусть ставится задача определить конструкцию аппарата, если концентрация кислоты – 25 %, а температура раствора 30…40 °С.

В данном случае, "высокая", "мала", "невысокая", "большая" являются нечеткими переменными. Каждой нечеткой переменной соответствуют определенные значения в некотором интервале. Использование нечетких переменных для построения и анализа правил называют нечеткой логикой, в основе которой лежит понятие нечеткого множества.

3.1 НЕЧЕТКИЕ МНОЖЕСТВА И ОПЕРАЦИИ НАД НИМИ

Обозначим через $X = \{ x \}$ – универсальное множество.

Нечетким множеством $\tilde{A}$ на множестве X называется совокупность пар вида

$$\tilde{A} = \{\langle \mu_A(x)/x \rangle\}, \tag{10}$$

где $\mu_A(x) \to [0,1]$ – отображение множества X в единичный отрезок [0,1]. Эта функция называется функцией принадлежности нечеткого множества $\tilde{A}$.

Значение функции принадлежности $\mu_A(x)$ для конкретного элемента $x \in X$ называется *степенью принадлежности.*

Можно сказать, что степень принадлежности $\mu_A(x)$ является субъективной мерой того, насколько элемент $x \in X$ соответствует понятию, смысл которого формируется нечетким множеством $\tilde{A}$.

Носителем нечеткого множества $\tilde{A}$ называется множество

$$S_A = \{x \mid x \in X \;\&\; \mu_A(x) > 0\}, \tag{11}$$

т.е. носителем нечеткого множества $\tilde{A}$ является подмножество S_A универсального множества X, для элементов которого функция принадлежности $\mu_A(x)$ строго больше нуля.

П р и м е р 3.1: Пусть универсальное множество X соответствует множеству возможных значений толщин изделия от 10 до 40 мм с дискретным шагом 1 мм. Нечеткое множество $\tilde{A}$, соответствующее нечеткому понятию "малая толщина изделия", может быть представлено в виде:

$\tilde{A}$ = {<1/10>, <0,9/11>, <0,8/12>, <0,7/13>, <0,5/14>, <0,3/15>, <0,1/16>, <0/17>, <0/18>, <0/19> …}.

Носителем нечеткого множества А будет являться конечное подмножество:

$$S_A = \{10, 11, 12, 13, 14, 15, 16\}.$$

Нечеткое множество A называется *нормальным*, если границы

$$\sup_{x \in X} \mu_A(x) = 1. \tag{12}$$

Мы будем рассматривать только нормальные нечеткие множества, так как если нечеткое множество ненормально, то его всегда можно превратить в нормальное, разделив все значения функции принадлежности на ее максимальное значение.

Для нечетких множеств вводятся операции объединения, пересечения и дополнения.

Пусть $\tilde{A}$ и $\tilde{N}$ два нечетких множества, заданных на универсальном множестве X с функциями принадлежности $\mu_A(x)$ и $\mu_N(x)$.

Рассмотрим основные операции над нечеткими множествами, которые будут использоваться в дальнейшем.

Объединением нечетких множеств $\tilde{A}$ и $\tilde{N}$ называется нечеткое множество

$$\tilde{A} \cup \tilde{N} = \left\{\left\langle \mu_{A \cup N}(x)/x \right\rangle\right\}, \tag{13}$$

где $(\forall \in X)$ $\mu_{A \cup N}(x) = \max\left\{\mu_A(x),\ \mu_N(x)\right\}$.

Пересечением нечетких множеств $\tilde{A}$ и $\tilde{N}$ называется нечеткое множество вида

$$\tilde{A} \cap \tilde{N} = \left\{\left\langle \mu_{A \cap N}(x)/x \right\rangle\right\}, \tag{14}$$

где $(\forall \in \mathrm{X})$ $\mu_{A \cap N}(x) = \min\left\{\mu_A(x),\ \mu_N(x)\right\}$.

Очевидно, что при выполнении операции пересечения над нечеткими множествами получается множество не всегда являющееся нормальным.

Дополнением нечеткого множества $\tilde{A}$ называется нечеткое множество

$$\neg\tilde{A} = \left\{\left\langle \mu_{\neg A}(x)/x \right\rangle\right\}, \tag{15}$$

где $(\forall\ \mathrm{x} \in \mathrm{X})$ $\mu_{\neg A}(x) = 1 - \mu_A(x)$.

Носителем нечеткого множества $\neg\tilde{A}$ будет являться множество тех элементов $x \in X$, для которых функция принадлежности $\mu_A(x) \neq 1$.

Обозначим $\tilde{A}i$ – нечеткое множество, определенное на Xi ($i = 1, \ldots, n$).

Декартовым произведением нечетких множеств $\tilde{A}i$ называется множество

$$\tilde{A}1 \cdot \tilde{A}2 \cdot \ldots \cdot \tilde{A}n = \left\{\left\langle \mu_A(x1, x2, \ldots, xn)/(x1, x2, \ldots, xn) \right\rangle\right\}, \tag{16}$$

где $xi \in Xi$, $\mu_A(x1, \ldots xn) = \min\left\{\mu_{A1}(x1), \ldots, \mu_{An}(xn)\right\}$.

П р и м е р 3.2: Пусть на множестве $X = \{10, 15, 20, 25\}$ и $Y = \{5, 6, 7\}$ заданы множества $\tilde{A}1$ и $\tilde{A}2$, имеющие вид:

$\tilde{A}1$ = {<1/10>, <0,8/15>, <0,5/20>, <0,3/25>};

$\tilde{A}2$ = {<1/5>, <0,5/6>, <0,2/7>}.

Тогда множество $\tilde{A}1 . \tilde{A}2$ будет иметь вид

$\tilde{A}1 . \tilde{A}2$ = {<1/(10,5)>, <0,8/(15,5)>, <0,5/(20,5)>, <0,3/(25,5)>, <0,5/(10,6)>, <0,5/(15,6)>, <0,5/(20,6)>, <0,3/(25,6)>, <0,2/(10,7)>, <0,2/(15,7)>, <0,2/(20,7)>, <0,2/(25,7)> }.

Степенью е множества $\tilde{A}$ называется нечеткое множество

$$\tilde{A}^e = \left\{\left\langle \mu_{A^e}(x)/x \right\rangle\right\}. \tag{17}$$

При $e = 2$ получается частный случай операции возведения в степень – операция *концентрации*, обозначаемая CON.

$$\mathrm{CON}(\tilde{A}) = \tilde{A}^2. \tag{18}$$

П р и м е р 3.3: Условие см. пример 3.2:

CON($\tilde{A}1$) = {<1/10>, <0,64/15>, <0,25/20>, <0,09/25>}.

Операция CON снижает степень нечеткости описания.

При e = 0,5 получается операция *растяжения* DIL:

$$\mathrm{DIL}(\tilde{A}) = \tilde{A}^{0,5}. \tag{19}$$

П р и м е р 3.4: Условие см. пример 3.2:

D*I*L($\tilde{A}2$) = {<1/5>, <0,7/6>, <0,45/7>}.

Операция D*I*L повышает степень нечеткости описания.

Множеством α – *уровня* нечеткого множества $\tilde{A}$ называется множество

$$S_\alpha = \{x \in X \mid \mu_A(x) >= \alpha\}, \quad (20)$$

где $\alpha \in [0,1]$.

П р и м е р 3.5: Условие см. пример 3.2.

Множество 0,5 – уровня нечеткого множества $\tilde{A}1$ имеет вид:

$$S_{0,5} = \{10, 15, 20\}.$$

3.2 НЕЧЕТКОЕ ВКЛЮЧЕНИЕ И РАВЕНСТВО МНОЖЕСТВ. НЕЧЕТКОЕ БИНАРНОЕ ОТНОШЕНИЕ

Понятие нечеткого включения и равенств множеств, нечеткого бинарного отношения являются одними из основных понятий процесса принятия решений при наличии экспертной информации.

Пусть $\tilde{A}1$ и $\tilde{A}2$ – нечеткие множества.

Степенью включения множества $\tilde{A}1$ в $\tilde{A}2$ называется величина

$$\eta(\tilde{A}1, \tilde{A}2) = \underset{x \in X}{\&} (\mu_{A1}(x) \to \mu_{A2}(x)). \quad (21)$$

Операция $\to$ есть импликация, определяемая как

$$\mu_{A1}(x) \to \mu_{A2}(x) = 1\& (1 - \mu_{A1}(x) + \mu_{A2}(x)) = \min\{1, 1 - \mu_{A1}(x) + \mu_{A2}(x)\}.$$

Степенью равенства нечетких множеств $\tilde{A}1$ и $\tilde{A}2$ называется величина $\rho(\tilde{A}1, \tilde{A}2)$, определяемая как логическая сумма эквивалентностей.

$$\rho(\tilde{A}1, \tilde{A}2) = \underset{x \in X}{\&} (\mu_{A1}(x) \leftrightarrow \mu_{A2}(x)). \quad (22)$$

Здесь $\leftrightarrow$ операция эквивалентности

$$\mu_{A1}(x) \leftrightarrow \mu_{A2}(x) = (\mu_{A1}(x) \to \mu_{A2}(x))\ \&\ (\mu_{A2}(x) \leftrightarrow \mu_{A1}(x)).$$

Очевидно, что $\rho(\tilde{A}1, \tilde{A}2) = \eta(\tilde{A}1, \tilde{A}2)\ \&\ \eta(\tilde{A}2, \tilde{A}1)$.

Степень включения и степень равенства могут принимать любые значения из отрезка [0, 1].

П р и м е р 3.6: Даны нечеткие множества

$\tilde{A}1 = \{<0,3/x2>, <0,6/x3>, <0,4/x5>\}$, $\tilde{A}2 = \{<0,8/x1>, <0,5/x2>, <0,7/x3>, <0,6/x5>\}$, определенные на универсальном множестве $X = \{x1, x2, x3, x4, x5\}$,

Определим степень включения множества $\tilde{A}1$ в $\tilde{A}2$ ($\eta(\tilde{A}1, \tilde{A}2)$) и множества $\tilde{A}2$ и $\tilde{A}1$ ($\eta(\tilde{A}2, \tilde{A}1)$):

$\eta(\tilde{A}1, \tilde{A}2)$ = (0 –> 0,8) & (0,3 –> 0,5) & (0,6 –> 0,7) & (0 –> 0) & (0,4 –> 0,6) = = (1&(1– 0 + 0,8)) & (1&(1– 0,3 + 0,5)) & (1&(1– 0,6 + 0,7)) & (1&(1– 0 + 0)) & (1&(1– 0,4 + 0,6)) = 1 & 1 & 1 & 1 & 1 = 1

$\eta(\tilde{A}2, \tilde{A}1)$= (0,8 –> 0) & (0,5 –> 0,3) & (0,7 –> 0,6) & (0 –> 0) & (0,6 –> 0,4) = (1&(1– 0,8 + 0)) & (1&(1– 0,5 + 0,3)) & (1&(1– 0,7 + 0,6)) & (1&(1– 0 + 0)) & (1&(1– 0,6 + 0,4)) = 0,2 & 0,8 & 0,9 & 1 & 0,8 = 0,2.

Тогда степень равенства множеств $\tilde{A}1$ и $\tilde{A}2$ будет равна 0,2.

Нечетким бинарным отношением $\tilde{R}$ на множестве X называется нечеткое множество вида $\{<\mu(x_i, x_j)/(x_i, x_j)>\}$, где xi, xj – некоторая пара элементов из множества X; $\mu(xi, xj)$ – функция принадлежности, определяемая субъективной мерой того, насколько пара (xi, xj) соответствует бинарному отношению $\tilde{R}$.

Если множество X конечно (счетно) и невелико, то нечеткое бинарное отношение удобно представить в виде матрицы $M(R)$. Матрица M – квадратная с числом строк и столбцов, равным количеству элементов множества X. В матрице строки и столбцы помечаются элементами множества X. В сроке, соответствующей элементу xi на пересечении ее со столбцом, соответствующим элементу xj, располагается значение функции принадлежности $\mu(xi, xj)$.

П р и м е р 3.7. Пусть X = {1, 3, 5, 7, 9}. Определить на множестве X нечеткое бинарное отношение "намного больше". Тогда матрица $M(R)$ может иметь вид

$$M(\widetilde{R})=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0 & 0 & 0 & 0 \\ 0{,}5 & 0{,}1 & 0 & 0 & 0 \\ 0{,}8 & 0{,}4 & 0{,}1 & 0 & 0 \\ 1 & 0{,}8 & 0{,}5 & 0{,}2 & 0 \end{bmatrix}. \tag{23}$$

3.3 НЕЧЕТКАЯ И ЛИНГВИСТИЧЕСКАЯ ПЕРЕМЕННЫЕ

Понятие нечеткой и лингвистической переменной используется экспертом при описании сложных объектов и явлений, а также при формализации процессов принятия решений на трудно формализуемых этапах проектирования.

Нечеткой переменной называется тройка объектов вида

$$< \alpha, X, C_\alpha >, \tag{24}$$

где α – наименование нечеткой переменной, $X = \{x\}$ – область ее определения; $C_\alpha = \{< \mu_\alpha(x)/x >\}$ – нечеткое множество на X, описывающее ограничения на возможные значения нечеткой переменной α (т.е. ее семантику).

Лингвистической переменной называется пятерка объектов

$$< \beta, T, X, G, M >, \tag{25}$$

где β – наименование лингвистической переменной; T – множество ее значений (терм – множество), нечеткие переменные, областью определения каждой из которых является множество X; G – синтаксическая процедура, позволяющая оперировать элементами терм – множества T, в частности, генерировать новые осмысленные термы (при традиционном подходе процедура G определяет новые значения лингвистической переменной, исходя из ее базового терм – множества T и логических операций *и*, *или*, *не*, *очень*, *слегка*); M – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G, в нечеткую переменную путем формирования соответствующего нечеткого множества. Например, семантические процедуры могут иметь вид:

$M(\widetilde{C}1$ *или* $\widetilde{C}2) = \widetilde{C}1 \cup \widetilde{C}2$ – объединение нечетких множеств;

$M(\widetilde{C}1$ *и* $\widetilde{C}2) = \widetilde{C}1 \cap \widetilde{C}2$ – пересечение нечетких множеств;

$M($*не* $\widetilde{C}1) = \neg\widetilde{C}1$ – дополнение нечетких множеств;

$M($*очень* $\widetilde{C}1) = \mathrm{CON}(\widetilde{C}1)$ – концентрация нечетких множеств;

$M($*слегка* $\widetilde{C}1) = \mathrm{DIL}(\widetilde{C}1)$ – растяжение нечетких множеств,

где $\widetilde{C}1$ *и* $\widetilde{C}2$ – нечеткие множества, соответствующие нечетким переменным $\alpha1$ и $\alpha2$ рассматриваемой лингвистической переменной.

Пример 3.8. Пусть эксперт оценивает толщину выпускаемого изделия с помощью понятий: "малая толщина", "средняя толщина", "большая толщина"; при этом минимальная толщина изделия равна 10 мм, а максимальная – 80 мм.

Формализация такого описания может быть проведена с помощью лингвистической переменной

$$< \beta, T, X, G, M >,$$

где β – "толщина изделия"; $T = \{\alpha1, \alpha2, \alpha3\}$ = {малая, средняя, большая}; $X = [10, 80]$.

Пусть нечеткие множества $C1$, $C2$, $C3$ описывают семантику базовых значений переменной β. Функции принадлежности представлены на рис. 10.

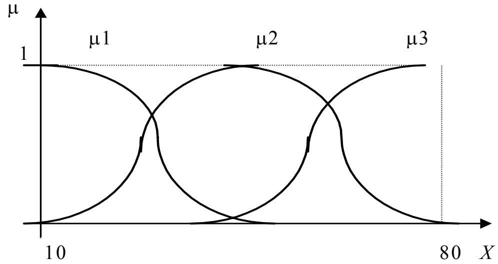


Рис. 10

Значение α^1 = "малая или средняя толщина ", будет определяться нечетким множеством C^1, т.е C^1 = M(α1 *или* α2) = $C1 \cup C2$, функция принадлежности которой имеет вид, представленный на рис. 11.

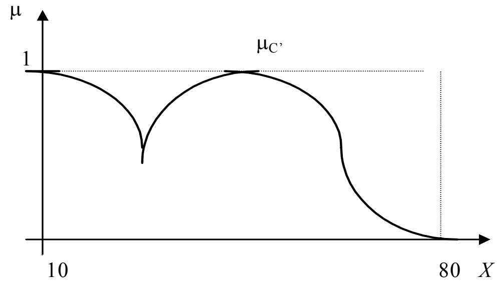


Рис. 11

Значение α^2 = "небольшая толщина" будет определяться нечетким множеством C^2 ($C^2 = M$(*не* α3) = $\neg C3$), функция принадлежности которой имеет вид, представленный на рис. 12.

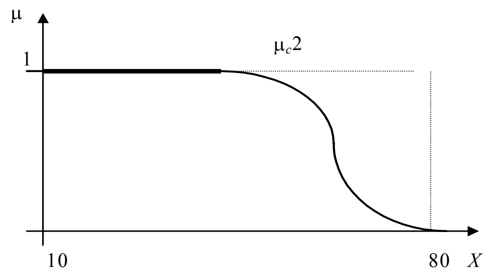


Рис. 12

Лингвистическая переменная, у которой процедура образования нового значения G зависит от множества базовых терм-значений T называется *синтаксически зависимой лингвистической переменной*.

Наряду с рассмотренными выше синтаксически зависимыми лингвистическими переменными существуют переменные, у которых процедура образования новых значений зависит не от множества базовых терм – значений T, а от области определения X (универсальное множество), т.е. $G = G(X)$. Например, значение лингвистической переменной "толщина изделия" может быть определено как "близкое к 20 мм" или "приблизительно к 75 мм". Такие лингвистические переменные называются *синтаксически независимыми.*

Произвольные значения синтаксически независимой лингвистической переменной взаимно однозначно определяется некоторыми значениями x области определения X. Поэтому произвольное значение (нечеткую переменную α) синтаксически независимой лингвистической переменной задается в виде тройки объектов

$$\alpha = \langle x, X, C_\alpha \rangle. \qquad (26)$$

Например, нечеткая переменная α, определенная как "приблизительно 75 мм", запишется в виде $\alpha = \langle 75, [10, 80], C_\alpha \rangle$.

3.4 МЕТОДЫ ПОСТРОЕНИЯ ТЕРМ-МНОЖЕСТВ

В основании теории из любой области естествознания лежит очень важное понятие элементарного объекта. Для теории нечетких множеств основополагающим понятием является понятие нечетного множества, которое характеризуется функцией принадлежности. Но основной трудностью мешающей интенсивному применению теории нечетного множества при решении практических задач, является то, что функция принадлежности должна быть задана вне самой теории u, следовательно, ее адекватность не может быть проверена непосредственно средствами самой теории.

Будем считать, что функция принадлежности $\mu_A(x)$ элемента x к нечеткому множеству A – это субъективная мера того, насколько $x \in X$ соответствует понятию, смысл которого формируется нечетким множеством A. Под субъективной мерой понимается определяемая опросом экспертов степень соответствия элемента x понятию, формализуемому нечетким множеством A. При этом степень соответствия – не условная вероятность наблюдения события A при возникновении события x, а, скорее, возможность интерпретации понятия x понятием A.

Утверждается, что для практических задач достаточно наличия нечеткого языка с фиксированным конечным словарем. Это ограничение не слишком сильное с точки зрения практического использования. Лингвистическая переменная β на практике имеет базовое терм-множество $T = \{T_i\}$, состоящее из 2 – 10 нечетких переменных. Каждый терм описывается нечетким подмножеством множества значений и некоторой базовой переменной. Предполагается, что объединение всех элементов терм-множества покрывает всю область определения лингвистической переменной. Это гарантирует, что любой элемент $u \in U$ описывается некоторым $T_i \in T$. На практике значения на входе часто сильно искажены (шум, помехи, ошибки измерения и т.д.), поэтому функции принадлежности должны выбираться достаточно широкими, чтобы искажения не давали ощутимого эффекта (см. рис. 13).

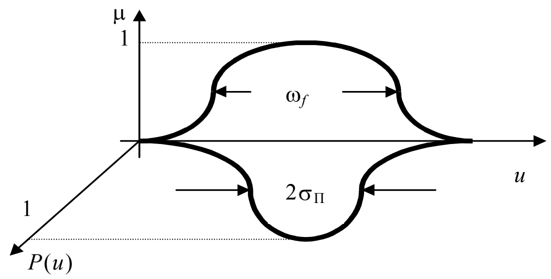


Рис. 13

Все термы нумеруются $T_i \in T$ $T = \{T_j\}^n{}_1$ на множестве действительных чисел $u \in R$, так что имеющий левее расположенный носитель, имеет меньший номер. Правила для выбора терм-множества сведены в табл. 1.

Таблица 1

	Критерий	Типичные значения
$\vert T \vert$	Выбирается в результате компромисса между сложностью и простотой	2…10
$U_{\max}$, $U_{\min}$	Для измеримых переменных на основании априорных знаний определяют область значений базовых переменных	
ω_t	Должно быть достаточно широким, чтобы избежать чрезмерного влияния погрешностей при переходе от нечеткой переменной к лингвисти-	$\omega_t > 5\sigma_п$

	ческой переменой	

Вводятся более строгие условия:

1) $\mu_{T1}(U_{\min}) = 1$; $\mu_{T1}(U_{\max}) = 1$;

2) $\forall i, i+1 <= n$
$0 < \max\limits_{u \in U} \mu_{Ti \cap Ti+1}(U) < 1$;

3) $\forall i$ существует $u \in U$:
$\mu_{Ti}(U) = 1$;

4) $\forall i$ и U $\sum \mu_{Ti}(U) > 1$.

3.4.1 Построение функций принадлежности на счетном множестве точек на основе экспертных оценок

Простейший способ построения функций принадлежности предполагает *опрос нескольких экспертов*.

Пусть имеется m экспертов, часть которых на вопрос о принадлежности элемента $x \in X$ нечеткому множеству A отвечает положительно. Обозначим их число через $n1$. Другая часть экспертов ($n2 = m - n1$) отвечает на вопрос отрицательно. Тогда функция принадлежности принимается: $\mu_A(\text{x}) = n1/(n1+n2)$.

П р и м е р 3.9. Пусть имеется множество X: $X = \{1, 2, 3, 4, 5\}$
Требуется построить нечеткое множество A, которое формализует нечеткое понятие "немного больше двух".

Пусть имеется шесть экспертов. Результаты их опроса имеют вид:

	1	2	3	4	5
$n1$	0	0	6	4	1
$n2$	6	6	0	2	5

Тогда нечеткое множество A имеет вид

$$\tilde{A} = \{<0/1>, <0/2>, <1/3>, <0{,}7/4>, <0{,}2/5>\}.$$

Необходимо отметить, что данная схема определения функции принадлежности самая простая, но и самая грубая.

Более точно функцию принадлежности можно построить на основе *количественного парного сравнения степеней принадлежности*. Такая схема допускает и одного эксперта.

Результатом опроса эксперта является матрица $\text{M} = \|m_{ij}\|$, $i, j = 1, \ldots, n$, где n – число точек, в которых сравниваются значения функции принадлежности. Число m_{ij} показывает, во сколько раз, по мнения эксперта, степень принадлежности $\mu_A(x_i)$ больше $\mu_A(x_j)$. При этом количество вопросов, на которые надо ответить эксперту составляет не n^2, а лишь $(n^{2-}n)/2$, так как по определению $m_{ii} = 1$ и $m_{ij} = 1/m_{ji}$.

При этом эксперт оперирует понятиями, представленными в табл. 2.

Таблица 2

Смысл	M_{ij}
$\mu(x_i)$ равна $\mu(x_j)$	1
$\mu(x_i)$ немного больше $\mu(x_j)$	3
$\mu(x_i)$ больше $\mu(x_j)$	5

$\mu(x_i)$ заметно больше $\mu(x_j)$	7
$\mu(x_i)$ намного больше $\mu(x_j)$	9
Значения, промежуточные по степени между перечисленными	2, 4, 6, 8

Далее, определить значение функции принадлежности μ_A в точках $x1$, $x2$, …, xn можно, используя формулу

$$\mu_A(x_i) = \frac{m_{ij}}{\sum_{i=1}^{n} m_{ij}}, \tag{27}$$

где j – произвольный столбец матрицы М.

П р и м е р 3.10: Пусть для описания расстояния между двумя точками используется лингвистическая переменная β – "расстояние" с множеством базовых значений $T = \{$"малое", "среднее", "большое"$\}$.

Базовое множество лингвистической переменной β: $X = \{1, 3, 6, 8\}$. Терм "малое" характеризуется нечеткой переменной <малое, X, C>. Требуется построить функцию принадлежности нечеткого множества C, т.е. определить значение $\mu_C(x)$, $x \in X$.

Пусть опросом экспертов получена матрица парных сравнений

$$\mathrm{M} = \begin{array}{c} \\ 1 \\ 3 \\ 6 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 3 & 6 & 8 \end{array} \\ \begin{bmatrix} 1 & 5 & 6 & 6 \\ 1/5 & 1 & 4 & 6 \\ 1/6 & 1/4 & 1 & 4 \\ 1/7 & 1/6 & 1/4 & 1 \end{bmatrix} \end{array}.$$

Здесь, например, на пересечении первой строки и второго столбца стоит число 5, т.е. $m_{12} = 5$, т.е. вследствие оценки эксперта $\mu_C(1)$ больше $\mu_C(3)$ в соответствии с таблицей.

Зафиксируем первый столбец матрицы M: $M1 = \{1, 1/5, 1/6, 1/7\}$ и по формуле, приведенной выше найдем значения функций принадлежности в точках x_i, $i = 1, 2, 3, 4$:

$$\mu_C(1) = \mu_C(x_1) = \frac{m_{11}}{\sum_{i=1}^{4} m_{i1}} = \frac{1}{1{,}55} = 0{,}64\,; \quad \mu_C(3) = \mu_C(x_2) = \frac{m_{21}}{\sum_{i=1}^{4} m_{i1}} = \frac{0{,}2}{1{,}55} = 0{,}13\,;$$

$$\mu_C(6) = \mu_C(x_3) = \frac{m_{31}}{\sum_{i=1}^{4} m_{i1}} = \frac{0{,}16}{1{,}55} = 0{,}1\,; \qquad \mu_C(8) = \mu_C(x_4) = \frac{m_{41}}{\sum_{i=1}^{4} m_{i1}} = \frac{0{,}14}{1{,}55} = 0{,}08\,.$$

Таким образом, нечеткое множество C имеет вид

$$C = \{<0{,}64/1>, <0{,}13/3>, <0{,}1/6>, <0{,}08/8>\}.$$

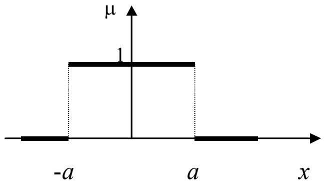
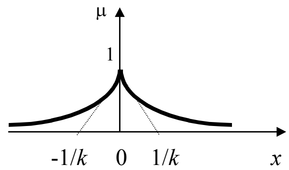
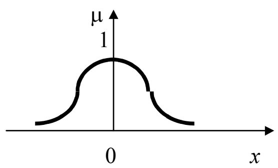
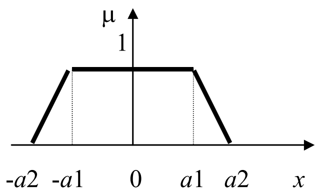
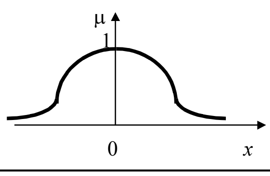
3.4.2. Построение функции принадлежности на непрерывном множестве точек

Выбор вида функции принадлежности и их параметров определяется в большей степени опытом, интуицией и другими субъективными факторами лица, принимающего решение. Именно здесь возникают новые, связанные с неоднозначностью и другого рода нечеткостью, неопределенности, которые носят субъективный характер.

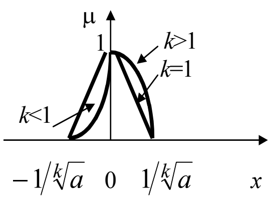
В табл. 3 приведены некоторые простейшие функции принадлежности, которые можно предложить эксперту. Все функции определены на множествах действительных и целых чисел.

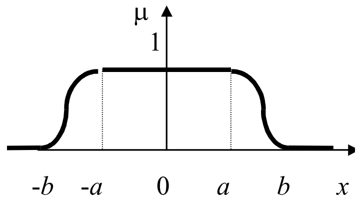
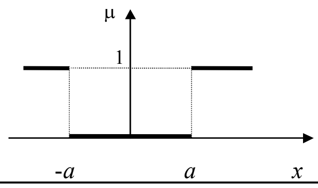
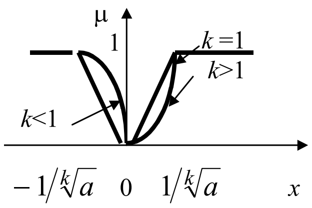
Задание функций степеней принадлежности является центральным вопросом формализации качественной информации. От корректности его выполнения в конечном итоге зависит степень достоверности результата решения задачи с использованием качественной информации.

Таблица 3

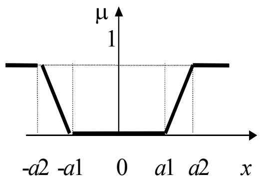
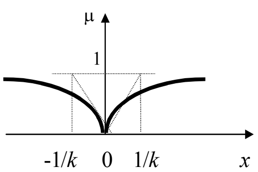
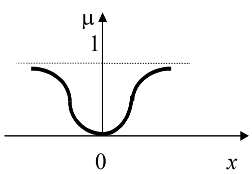
График	Функция
Функции степеней принадлежности утверждения "величина $\lvert x \rvert$ малая"	
	$\mu(x) = \begin{cases} 0, -\infty < x < -a; \\ 1, -a \le x \le a; \\ 0, a < x < \infty \end{cases}$
	$K > 1$ $\mu(x) = \begin{cases} e^{kx}, -\infty < x \le 0; \\ e^{-kx}, 0 \le x < \infty \end{cases}$
	$\mu(x) = e^{-kx^2}$
	$\mu(x) = \begin{cases} 0, -\infty < x \le -a_2; \\ \dfrac{a_2 + x}{a_2 - a_1}, -a_2 \le x \le a_1; \\ 1, -a_1 \le x \le a_1; \\ \dfrac{a_2 - x}{a_2 - a_1}, a_1 \le x < \infty \end{cases}$
	$\mu(x) = \dfrac{1}{1 + kx^2}$, $k > 1$

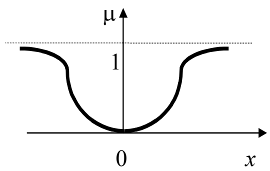
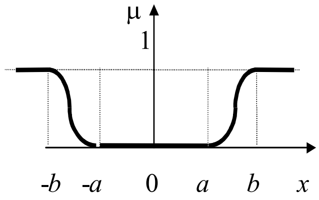
Продолжение табл. 3

График	Функция
	$\mu(x) = \begin{cases} 0, -\infty < x \le -1/\sqrt[k]{a}; \\ 1 - a(-x)^k, -1/\sqrt[k]{a} \le x \le 0; \\ 1 - ax^k, 0 \le x \le 1/\sqrt[k]{a}; \\ 0, 1/\sqrt[k]{a} \le x \le \infty \end{cases}$

	$\mu(x) = \begin{cases} 0, \ -\infty < x \le -b; \\ \frac{1}{2} + \frac{1}{2}\sin\frac{\pi}{b-a}\left(x + \frac{a+b}{2}\right); \\ -b \le x \le -a; \\ 1, \ -a \le x \le a; \\ \frac{1}{2} - \frac{1}{2}\sin\frac{\pi}{b-a}\left(x - \frac{a+b}{2}\right); \\ a \le x \le b; \\ 0, \ b \le x < \infty \end{cases}$
Функции степеней принадлежности утверждения "величина $\lvert x \rvert$ большая"	
	$\mu(x) = \begin{cases} 1, \ -\infty \le x < -a; \\ 0, \ -a \le x \le a; \\ 1, \ a < x \le \infty \end{cases}$
	$\mu(x) = \begin{cases} 1, \ -\infty < x < -1/\sqrt[k]{a}; \\ a(-x)^k, \ -\frac{1}{\sqrt[k]{a}} \le x \le 0; \\ ax^k, \ 0 \le x \le \frac{1}{\sqrt[k]{a}}; \\ 1, \ \frac{1}{\sqrt[k]{a}} \le x < \infty \end{cases}$

Продолжение табл. 3

График	Функция
	$\mu(x) = \begin{cases} 1, -\infty < x < -a_2; \\ -\frac{x+a_1}{a_2-a_1}, -a_2 \le x \le -a_1; \\ 0, -a_1 x \le a_1; \\ \frac{x-a_1}{a_2-a_1}, a_1 \le x \le a_1; \\ 1, a_2 \le x < \infty \end{cases}$
	$k > 1$ $\mu(x) = \begin{cases} 1 - e^{kx}, -\infty < x \le 0; \\ 1 - e^{-kx}, 0 \le x < \infty \end{cases}$
	$k > 1$ $\mu(x) = 1 - e^{-kx^2}$

	$k > 1$ $\mu(x) = \dfrac{kx^2}{1+kx^2} = \dfrac{1}{1+\dfrac{1}{kx^2}}$
	$\mu(x) = \begin{cases} 1, \ -\infty < x \le -b; \\ \frac{1}{2} - \frac{1}{2}\sin\frac{\pi}{b-a}\left(x + \frac{a+b}{2}\right); \\ -b \le x \le -a; \\ 0, \ -a \le x \le a; \\ \frac{1}{2} + \frac{1}{2}\sin\frac{\pi}{b-a}\left(x - \frac{a+b}{2}\right); \\ a \le x \le b; \\ 1, \ b \le x < \infty \end{cases}$

Задание функции степеней принадлежности в нечетких подмножествах осуществляют несколькими способами.

- В ряде случаев исследователь может задать самостоятельно функцию, исходя из личного опыта. Например, проводя сопоставление результатов измерений, выполненных на различных технологических системах, исследователь наряду с количественными данными оперирует качественными факторами и описывает результаты сопоставления словесно.
- В более сложных и ответственных случаях задание функций принадлежности в нечетких подмножествах выполняется с привлечением группы экспертов с последующей обработкой их оценок. Так при оценке качества изделий, контроль которого осуществляется визуально, возникает задача выбора эталона. В этом случае к выбору и классификации эталонов целесообразно привлечь экспертов.

Рассмотрим процесс задания функции принадлежности. Пусть диапазон изменения величины $x \in X$ определяется отрезком $[x_n, x_k]$.Обычно на этом отрезке выделяют значение $x_0 \in X$, характеризующее понятие "норма". Кроме этого на отрезке $[x_n, x_k]$ существуют противоположные по смысловому содержанию (с точки зрения нечеткого множества) термины. Иными словами, множество $[x_н, x_k]$должно обладать симметрией относительно элемента x_0. Требуется, кроме того, выполнение следующих асимптотических свойств:

$$\lim_{x \to x_n} \mu(x) = a; \qquad \lim_{x \to x_k} \mu(x) = b,$$

где a, b – постоянные для данного термина.

Например, на рис. 14 представлена функция принадлежности $\mu(x)$, формализующая понятие "высокий".

На оси абсцисс отмечен опорный элемент x_0, соответствующий понятию "норма". Обычно полагают $\mu(x) = 0{,}5$. Выбор x_0 подвержен субъективизму каждого исследователя и определяется уровнем знаний о конкретной системе. Выполнение условия $\lim\limits_{x \to x_n} \mu(x) = 0$ отражает тот факт, что элементы $x < x_0$ в меньшей степени, чем x_0, относятся к понятию "высокий".

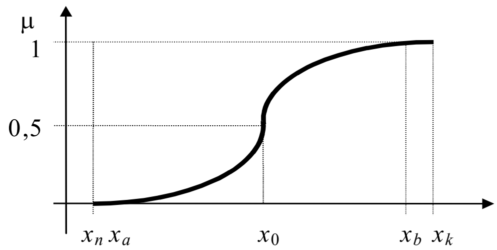


Рис. 14

Здесь важно заметить, что функции принадлежности должны быть сформированы с точностью до качественных различий первичных терминов (например, понятие высокий" и "сверхвысокий"). Кроме того, заметим, что формируемое нечеткое множество предполагается нормальным, т.е. $\sup\limits_{x \in X} \mu(x) = 1$.

Исходя из асимптотических свойств функции $\mu(x)$, исследователем может быть установлены интервалы $[x_н, x_a]$и $[x_b, x_k]$, на которых функция задается путем четкой классификации:

$$\mu(x) = \begin{cases} 0; x \in [x_н, x_a]; \\ 1; x \in [x_b, x_к]. \end{cases} \tag{28}$$

Наиболее сложным является задание $\mu(x)$ при $x \in [x_a, x_b]$. Предполагается, что $\mu(x)$ является монотонной функцией.

Отмечается, что человек с достаточно хорошей точностью может запомнить в памяти и анализировать от пяти до семи признаков. Поэтому необходимо минимизировать психологическую нагрузку эксперта, который выполняет формализацию первичных терминов.

Например, проиллюстрируем способ задания функции принадлежности для формализации понятий "низкий", "средний", "высокий".

Процедура задания функций принадлежности, которой должны придерживаться эксперты, заключается в следующем (рис. 15):

1) выделение точки $x_1 \in X$, которая, с точки зрения, эксперта точно соответствует нечеткому подмножеству. В этом случае $\mu(x) = 1$;

2) нахождение точек слева и справа от x_1, которые с точки зрения эксперта не могут быть отнесены к рассматриваемому термину. Для них $\mu(x_2) = \mu(x_3) = 0$;

3) графическое построение функций по выбранным точкам с использованием линейной аппроксимации;

4) выделение подмножества $X_1 \in X$, на котором определена формализация термина, $X_1 \in [x_2, x_3]$. Следует отметить, что в ряде случаев точки x_2, x_3 могут быть отнесены в бесконечность.

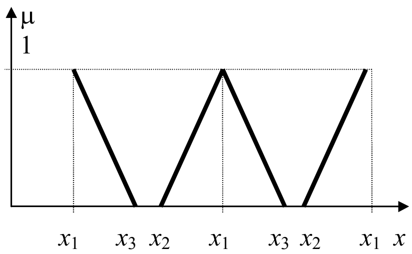


Рис. 15

Такой способ задания функций принадлежности обладает следующими особенностями:

– простотой выполнения экспертной оценки с точки зрения психологической нагрузки;
– компактность задания функций;
– простотой математических средств при переходе от одного термина к другому;

В ряде случаев функцию степеней принадлежности $\mu(x)$ нечеткого подмножества некоторого множества задают в виде функциональной зависимости, например экспоненциальной, полинома и т.п. с одним или несколькими неизвестными переменными.

Вообще, задание функций принадлежности требует знаний особенностей объекта исследований, принятой в данной отрасли терминологии и использование, по возможности, простых функциональных зависимостей. Для идентификации неизвестных параметров в функции принадлежности нечеткого подмножества могут быть использованы метод наименьших квадратов, симплекс-метод и другие.

П р и м е р ы:

1) Параметр "расход сырья на установку" (G) определен на отрезке [70-100] и имеет три нечетких значения:

– малый (70…80) с функцией принадлежности

$$\mu_1(x) = \exp\left(-\frac{1}{5}\ln\frac{1}{2}|x-75|\right);$$

– средний (80…90) с функцией принадлежности

$$\mu_2(x) = \exp\left(-\frac{1}{5}\ln\frac{1}{2}|x-85|\right);$$

– большой (90…100) с функцией принадлежности

$$\mu_3(x) = \exp\left(-0{,}1\ln\frac{1}{2}\left|x-100\right|\right).$$

Здесь принятые термины описываются зависимостью вида

$$\mu(x) = \exp(-Q\left|x-a_p\right|),$$

где Q – постоянная величина, которая находится при идентификации функции принадлежности: $a_p = (a_r + a_{r+1})/2$.

2 Параметр "вес ткани " характеризуется нечетким знанием "хороший". Параметр определен на отрезке [5, 20]. При $w = 5$ вес считается лучшим, а при $w = 20$ – худшим. Функция принадлежности имеет вид

$$\mu(x) = \begin{cases} 1, & 0 \le \omega \le 5; \\ 0{,}5 - \sin\frac{\pi}{15}(\omega - 12{,}5), & 5 < \omega < 20; \\ 0, & \omega > 20 \end{cases}$$

3.5 НЕЧЕТКИЕ ВЫСКАЗЫВАНИЯ. ПРАВИЛА ПРЕОБРАЗОВАНИЯ НЕЧЕТКИХ ВЫСКАЗЫВАНИЙ

Нечеткими высказываниями называются высказывания следующего вида:

1 Высказывание <β есть α>,

где β – наименование лингвистической переменной, отражающей некоторый объект или параметр реальной действительности; α – наименование нечеткой переменной, которая является нечеткой оценкой β.

Например: <давление большое>

<толщина равна 14> (в этом случае значение $\alpha = 14$ является четкой оценкой лингвистической переменной β (толщина)).

2 Высказывания вида:

$<\beta$ есть $m\alpha>$;

$<\beta$ есть $Q\alpha>$;

$<Q\beta$ есть $m\alpha>$;

$<m\beta$ есть $Q\alpha>$.

где m – модификатор (ему соответствуют такие слова как *очень*, *средний*, *более или менее*, *незначительный …*); Q – квантификатор (ему соответствуют слова типа: *большинство*, *несколько*, *много*, *немного*, *очень много* и др.)

Например: <давление очень большое>

<большинство значений параметра очень мало>

3 Высказывания, образованные из высказываний 1-го и 2-гр видов и союзов: *и*, *или*, *если … то*, *если … то … иначе …*

Например: *если* давление большое *то* толщина не мала.

Предположим, имеются некоторые нечеткие высказывания $\tilde{C}$ и $\tilde{D}$ относительно одной ситуации A. Эти высказывания имеют вид:

$< \beta$ есть $\alpha_C >$;

$< \beta$ есть $\alpha_D >$,

где α_C и α_D – нечеткие переменные, определенные на универсальном множестве $X = \{x\}$.

Истинностью высказывания $\tilde{D}$ относительно $\tilde{C}$ называется значение функции $T(\tilde{D}/\tilde{C})$, определяемое степенью соответствия высказываний $\tilde{D}$ и $\tilde{C}$:

$$T(\tilde{D}/\tilde{C}) = \{\mu_T(\tau)/\tau\}, \tag{29}$$

где $\tau = \mu_D(x)\ \forall\ x \in X$; $\mu_T(\tau) = \max\limits_{x \in X'} \mu_C(x)$; $X' = \{x \in X | \mu_D(x) = \tau\}$,

т.е. функция принадлежности значения истинности $\mu_T(\tau)$ для любого $0 \le \tau \le 1$ определяется как максимальное из $\mu_C(x)$ (функция принадлежности нечеткой переменной α_C) для тех x, у которых $\mu_D(x) = \tau$ ($\mu_D(x)$ – функция принадлежности нечеткой переменной α_D).

П р и м е р 3.11. Имеются два высказывания:

$\tilde{C}$: < β имеет значение приблизительно 6 >; $\tilde{D}$: < β находится близко к 5 >.

Нечеткое множество определено на универсальном множестве X = {2, 3, 4, 5, 6, 7, 8, 9, 10}.

Нечеткие переменные:

α_C – "приблизительно 6" с функцией принадлежности:

C_C = {<0,1/3>, <0,4/4>, <0,8/5>, <1/6>, <0,7/7>, <0,4/8>, <0,3/9>, <0,1/10>}.

α_D – "близко 5" с функцией принадлежности:

C_D = {<0,1/2>, <0,3/3>, <0,7/4>, <1/5>, <0,8/6>, <0,6/7>, <0,3/8>, <0,1/9>},

Требуется определить истинность высказывания $\tilde{D}$ относительно $\tilde{C}$.

Определим значения τ, для которых будут вычисляться функции принадлежности.

$\tau \in \{0, 0{,}1, 0{,}3, 0{,}6, 0{,}7, 0{,}8, 1\}$; $(\tau = \mu_D(x))$

τ = 0 $x = \{10\}$ max $\mu_C(x) = 0{,}1$;

τ = 0,1 $x = \{2, 9\}$ max $\mu_C(x) = 0{,}3$;

τ = 0,3 $x = \{3, 8\}$ max $\mu_C(x) = 0{,}4$;

τ = 0,6 $x = \{7\}$ max $\mu_C(x) = 0{,}7$;

τ = 0,7 $x = \{4\}$ max $\mu_C(x) = 0{,}4$;

τ = 0,8 $x = \{6\}$ max $\mu_C(x) = 1$;

τ = 1 $x = \{5\}$ max $\mu_C(x) = 0{,}8$.

Таким образом, истинность высказывания $\tilde{D}$ относительно $\tilde{C}$ имеет вид

$T(\tilde{D}/\tilde{C})$ = {<0,1/0>; <0,3/0,1>; <0,4/0,3>; <0,7/0,6>; <0,4/0,7>; <1/0,8>; <0,8/1>}.

Мы рассмотрели нахождение истинности высказываний вида < β есть α >. Чтобы определить истинность более сложных высказываний, необходимо привести эти высказывания к виду < β есть α >. Такое приведение осуществляется по определенным правилам.

(1) Правило преобразования конъюнктивной формы:

$$<\beta_x \text{есть } \alpha_{x1} \text{ и } \beta_y \text{есть } \alpha_{y1}> \rightarrow <(\beta_x, \beta_y) \text{ есть } \ddot{\alpha}_{x1} \cap \ddot{\alpha}_{y1}>, \qquad (30)$$

Здесь $\ddot{\alpha}_{x1} \cap \ddot{\alpha}_{y1}$ – это значение лингвистической переменной (β_x, β_y) с нечетким множеством $C_\cap = \ddot{C}_{x1} \cap \ddot{C}_{y1}$, где $\ddot{C}_{x1}, \ddot{C}_{y1}$ – цилиндрические продолжения нечетких множеств C_x и C_y: $\ddot{C}_{x1} = \{<\ddot{\mu}_{x1}(x,y)/(x,y)>\}$; $\ddot{C}_{y1} = \{<\ddot{\mu}_{y1}(x,y)/(x,y)>\}$.

Причем $(x,y) \in X * Y$, $(\forall x \in X), (\forall y \in Y)$, $\ddot{\mu}_{x1}(x,y) = \mu_{x1}(x)$, $\ddot{\mu}_{y1}(x,y) = \mu_{y1}(y)$.

П р и м е р 3.12. Пусть имеется нечеткое высказывание вида: <давление большое и диаметр малый>. Здесь лингвистические переменные β_x – давление, β_y – диаметр принимают значения α_{x1} – большое, α_{y1} – малый.

Лингвистическая переменная β_x определена на множестве X = {3, 5, 6}, а нечеткое множество C_{x1} , соответствующие значению α_{x1} имеет вид

$$C_{x1} = \{<0{,}3/3>, <0{,}7/5>, <1/6>\},$$

β_y определена на множестве Y = {10, 15, 20, 25}, а нечеткое множество C_{y1} соответствующее α_{y1} имеет вид

$$C_{y1} = \{<1/10>, <0{,}8/15>, <0{,}4/20>, <0{,}2/25>\}.$$

Найдем цилиндрические продолжения:

$\ddot{C}_{x1}$ = {<0,3/(3,10)>; <0,3/(3,15)>; <0,3/(3,20)>; <0,3/(3,25)>; <0,7/(5,10)>; <0,7/(5,15)>; <0,7/(5,20)>; <0,7/(5,25)>; <1/(6,10)>; <1/(6,15)>; <1/(6,20)>; <1/(6,25)>};

$\ddot{C}_{y1}$ = {<1/(3,10)>; <1/(5,10)>; <1/(6,10)>; <0,8/(3,15)>; <0,8/(5,15)>; <0,8/(6,15)>; <0,4/(3,20)>; <0,4/(5,20)>; <0,4/(6,20)>; <0,2/(3,25)>; <0,2/(5,25)>; <0,2/(6,25)>}.

Тогда получим преобразование исходного высказывания:

<давление большое и диаметр малый> $\rightarrow <(\beta_x, \beta_y) \text{ есть } \ddot{\alpha}_{x1} \cap \ddot{\alpha}_{y1}>$,

где $\ddot{\alpha}_{x1} \cap \ddot{\alpha}_{y1}$ значение лингвистической переменной (β_x, β_y) с нечетким множеством

$C_\cap = \ddot{C}_{x1} \cap \ddot{C}_{y1} =$ {<0,3/(3,10)>; <0,3/(3,15)>; <0,3/(3,20)>; <0,2/(3,25)>; <0,7/(5,10)>; <0,7/(5,15)>; <0,4/(5,20)>; <0,2/(5,25)>; <1/(6,10)>; <0,8/(6,15)>; <0,4/(6,20)>; <0,2/(6,25)>}.

(2) Правило преобразования дизъюнктивной формы:

$$<\beta_x \text{есть } \alpha_{x1} \text{ или } \beta_y \text{ есть } \alpha_{y1} > \to <(\beta_x,\ \beta_y) \text{ есть } \ddot{\alpha}_{x1} \cup \ddot{\alpha}_{y1} > . \quad (31)$$

Здесь $\ddot{\alpha}_{x1} \cup \ddot{\alpha}_{y1}$ – это значение лингвистической переменной (β_x, β_y) с нечетким множеством $C_\cup = \ddot{C}_{x1} \cup \ddot{C}_{y1}$ (объединение цилиндрических продолжений).

П р и м е р 3.13. Смотри задание примера 3.12.

Пусть имеется нечеткое высказывание:

<давление большое ИЛИ диаметр малый>$\to <(\beta_x,\ \beta_y)$ есть $\ddot{\alpha}_{x1} \cup \ddot{\alpha}_{y1}$>,

где $\ddot{\alpha}_{x1} \cup \ddot{\alpha}_{y1}$ значение лингвистической переменной $(\beta_x,\ \beta_y)$ с нечетким множеством $C_\cup = \ddot{C}_{x1} \cup \ddot{C}_{y1} =$ {<1/(3,10)>; <0,8/(3,15)>; <0,4/(3,20)>; <0,3/(3,25)>; <1/(5,10)>; <0,8/(5,15)>; <0,7/(5,20)>; <0,7/(5,25)>; <1/(6,10)>; <1/(6,15)>; <1/(6,20)>; <1/(6,25)>}.

(3) Правило преобразования высказываний импликативной формы.

$$<\text{если } \beta_x \text{ есть } \alpha_{x1} \text{ то } \beta_y \text{ есть } \alpha_{y1} > \to <(\beta_x,\ \beta_y) \text{ есть } \ddot{\alpha}_{x1} \Diamond \ddot{\alpha}_{y1} > \quad (32)$$

Знак $\Diamond$ означает пороговую сумму, определяемую как

$$(\forall\ x \in X)\ (\forall\ y \in Y)\ \mu_\Diamond(x,\ y) = 1 \wedge (1 - \mu_{\ddot{\alpha}\ X1}(x,\ y) + \mu_{\ddot{\alpha}\ Y1}(x, y)),$$

где $\mu_{\ddot{\alpha}X1}(x,\ y), \mu_{\ddot{\alpha}Y1}(x,\ y)$ – функции принадлежности, соответствующие нечетким множествам $\ddot{C}_{x1}, \ddot{C}_{y1}$.

П р и м е р 3.14. Рассмотрим нечеткое высказывание

<если давление большое то диаметр малый>.

Это высказывание можно записать в виде $<(\beta_x,\ \beta_y)$ есть $\ddot{\alpha}_{x1} \Diamond \ddot{\alpha}_{y1}>$.

Определим функцию принадлежности $\mu_\Diamond(x, y)$ (смотри задание примера 12):

$\mu_\Diamond(3,10) = 1 \wedge (1 - 0,3 + 1) = 1$; $\mu_\Diamond(3,15) = 1 \wedge (1 - 0,3 + 0,8) = 1$;

$\mu_\Diamond(3,20) = 1 \wedge (1 - 0,3 + 0,4) = 1$; $\mu_\Diamond(3,25) = 1 \wedge (1 - 0,3 + 0,2) = 0,9$;

$\mu_\Diamond(5,10) = 1 \wedge (1 - 0,7 + 1) = 1$; $\mu_\Diamond(5,15) = 1 \wedge (1 - 0,7 + 0,8) = 1$;

$\mu_\Diamond(5,20) = 1 \wedge (1 - 0,7 + 0,4) = 0,7$; $\mu_\Diamond(5,25) = 1 \wedge (1 - 0,7 + 0,2) = 0,5$;

$\mu_\Diamond(6,10) = 1 \wedge (1 - 1 + 1) = 1$; $\mu_\Diamond(6,15) = 1 \wedge (1 - 1 + 0,8) = 0,8$;

$\mu_\Diamond(6,20) = 1 \wedge (1 - 1 + 0,4) = 0,4$; $\mu_\Diamond(6,25) = 1 \wedge (1 - 1 + 0,2) = 0,2$;

т.о. нечеткая переменная $\ddot{\alpha}_{x1} \Diamond \ddot{\alpha}_{y1}$ будет характеризоваться нечетким множеством:

$C_\Diamond$={<1/(3,10)>; <1/(3,15)>; <1/(3,20)>; <0,9/(3,25)>; <1/(5,10)>; <1/(5,15)>; <0,7/(5,20)>; <0,5/(5,25)>; <1/(6,10)>; <0,8/(6,15)>; <0,4/(6,20)>; <0,2/(6,25)>}.

Рассмотрим более сложное высказывание импликативной формы

$$<\text{если } \beta_x \text{ есть } \alpha_{x1} \text{ то } \beta_y \text{ есть } \alpha_{y1} \text{ иначе } \alpha_{y2} > .$$

Представляя его в конъюнктивной форме получим:

<если β_x есть α_{x1} то β_y есть α_{y1} и если β_x
есть не α_{x1} то β_y есть α_{y2} >

Согласно ранее приведенным формулам получаем

$$\to <(\beta_x, \beta_y) \text{ есть } (\ddot{\alpha}_{x1} \Diamond \ddot{\alpha}_{y1}) \cap (\neg \ddot{\alpha}_{x1} \Diamond \ddot{\alpha}_{y2}) >$$

3.6 ПРЕДСТАВЛЕНИЕ ЭКСПЕРТНОЙ ИНФОРМАЦИИ В ВИДЕ СИСТЕМ НЕЧЕТКИХ ВЫСКАЗЫВАНИЙ

Обозначим через X, Y, Z… – множество значений входных параметров процесса проектирования, существенно влияющих на выбор выходного параметра V. Введем лингвистические переменные: $<\beta_x, T_x, X_x, G_x, M_x>$, $<\beta_y, T_y, X_y, G_y, M_y>$, $<\beta_z, T_z, X_z, G_z, M_z>$, и $<\beta_v, T_v, X_v, G_v, M_v>$, определенные на множествах X, Y, Z … и V.

Системы логических высказываний, отражающие опыт эксперта в типовых ситуациях, представим в виде

$$\tilde{L}^{(1)} = \begin{cases} \tilde{L}_1^{(1)} :< \text{если } \tilde{E}_{11} \text{ или...или } \tilde{E}_{1n_1} \text{ то } \beta_v \text{ есть } \mu\alpha_{v1} > \\ \ldots \\ \tilde{L}_m^{(1)} : \langle \text{если } \tilde{E}_{m1} \text{ или...или } \tilde{E}_{mn_m} \ \beta_v \text{ есть } \mu\alpha_{vm} > \end{cases} \tag{33}$$

или в виде

$$\tilde{L}^{(1)} = \begin{cases} \tilde{L}_1^{(1)} :< \text{если } \beta_v \text{ есть } \mu\alpha_{v1} \text{ то } \tilde{E}_{11} \text{ или...или } \tilde{E}_{1n_1} > \\ \ldots \\ \tilde{L}_m^{(1)} : \langle \text{если } \beta_v \text{ есть } \mu\alpha_{vm} \text{ то } \tilde{E}_{m1} \text{ или...или } \tilde{E}_{mn_m} > \end{cases} \tag{34}$$

где m – число базовых значений лингвистической переменной β_v; E_{ji} ($i = 1…n$, $j = 1...m$– высказывания вида

$$< \beta_x \text{ есть } \mu\alpha_{x_{ji}} \, и \ \beta_y \text{ есть } \mu\alpha_{y_{ji}} \, и \ \beta_z \text{ есть } \mu\alpha_{z_{ji}} \ldots > .$$

Высказывание E_{ij} представляет собой i-ю входную нечеткую ситуацию, которая может иметь место, если лингвистическая переменная β_v примет значение α_{vj}. Значения α_{Xji}, α_{Yji}, α_{Zji}, … α_{Vji} – нечеткие переменные с функциями принадлежности соответственно: $\mu_{Xji}(x)$, $\mu_{Yji}(y)$, $\mu_{Zji}(z)$, … $\mu_{Vji}(v)$ ($x\in X$, $y\in Y$, $z\in Z$, $v\in V$).

Обе приведенные системы нечетких высказываний, так же как и ранее рассмотренные четкие системы, отражают два разных случая взаимосвязи между значениями входных и выходных параметров процесса проектирования. В первом случае в зависимости от базовых значений входных лингвистических переменных делается вывод о базовом значении выходной лингвистической переменной. Во втором случае в зависимости от возможных значений выходного параметра делается предположение о возможных значениях входных параметров.

Представим системы в более компактном виде.

Используя правило преобразования конъюнктивной формы, высказывание E_{ji} можно записать в более компактном виде:

$$E_{ji} :< \beta_W \text{ есть } \alpha_{Eji}>,$$

где β_W – лингвистическая переменная, определенная на множестве $W = X * Y * Z * …$ и принимающая базовые значения α_{Eji} с функцией принадлежности $\mu_{Eji}(w) = \min\{\mu_{Xji}(x), \mu_{Yji}(y), \mu_{Zji}(z), …\}$.

Далее согласно правилу преобразования дизъюнктивной формы высказывания $L_j^{(1)}$ и $L_j^{(2)}$ могут быть представлены в виде:

$$L_j^{(1)} = < \text{если } \beta_W \text{ есть } \alpha_{Wj} \text{ то } \beta_V \text{ есть } \alpha_{Vi}>,$$
$$L_j^{(2)} = < \text{если } \beta_V \text{ есть } \alpha_{Vj} \text{ то } \beta_W \text{ есть } \alpha_{Wi}>.$$

Здесь α_{Wi} – значение лингвистической переменной β_W с функцией принадлежности: $\mu_{Wj}(w) = \max \mu_{Eji}(w)$.

Обозначим через A_j и N_j высказывания $< \beta_W$ есть $\alpha_{Wj}>$ и $< \beta_V$ есть $\alpha_{Vj}>$.

Тогда системы нечетких высказываний запишутся в виде:

$$\widetilde{L}^{(1)} = \begin{cases} \widetilde{L}_1^{(1)} :< \text{если } \widetilde{A}_1 \text{ то } \widetilde{B}_1 > \\ \widetilde{L}_2^{(1)} :< \text{если } \widetilde{A}_2 \text{ то } \widetilde{B}_2 > \\ \dots \\ \widetilde{L}_m^{(1)} :< \text{если } \widetilde{A}_m \text{ то } \widetilde{B}_m > \end{cases} \tag{35}$$

Эту систему назовем *нечеткой системой первого типа.*

$$\widetilde{L}^{(2)} = \begin{cases} \widetilde{L}_1^{(2)} :< \text{если } \widetilde{B}_1 \text{ то } \widetilde{A}_1 > \\ \widetilde{L}_2^{(2)} :< \text{если } \widetilde{B}_2 \text{ то } \widetilde{A}_2 > \\ \dots \\ \widetilde{L}_m^{(2)} :< \text{если } \widetilde{B}_m \text{ то } \widetilde{A}_m > \end{cases} \tag{36}$$

Эту систему назовем *нечеткой системой второго типа.*

Системы нечетких экспертных высказываний представим в виде соответствий:

1 Система высказываний первого типа может быть задана соответствием:

$$\Gamma^{(1)} = (T_V, T_W, F_1), \tag{37}$$

где T_W – область отправления (множество входных ситуаций); T_V – область прибытия (множество выходных ситуаций); $F_1 \subseteq T_W * T_V$ – график соответствия.

2 Система высказываний второго типа задается соответствием

$$\Gamma^{(2)} = (T_V, T_\mathrm{W}, F_2), \tag{38}$$

где $F_2 \subseteq T_V * T_\mathrm{W}$.

Графики соответствия представляются в виде графа, в левой части которого вершинам соответствуют области отправления, а в правой – области прибытия.

Пример приведен на рис. 16.

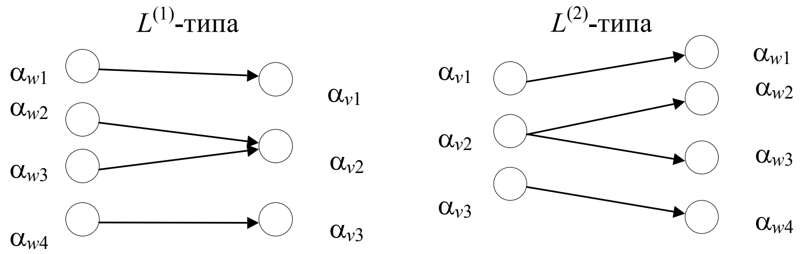


Рис. 16

Для анализа нечеткой информации вводится ряд понятий.

1 Система нечетких высказываний называется лингвистически не избыточной, если граф соответствия не содержит повторяющихся пар вершин.

2 Система нечетких высказываний называется лингвистически полной, если граф системы первого типа в правой части, а системы второго типа в левой части не содержит изолированных вершин. В противном случае система является лингвистически вырожденной (пример на рис. 17):

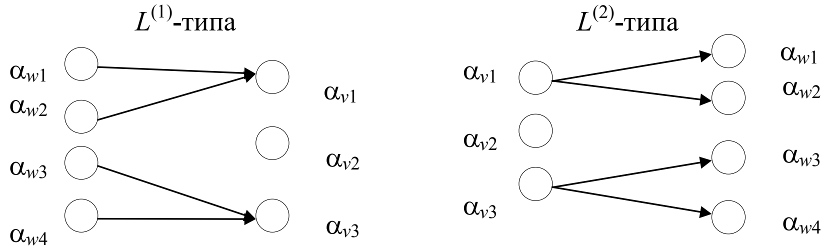


Рис. 17

3 Система нечетких высказываний называется лингвистически непротиворечивой, если в графе соответствия системы первого типа из каждой вершины левой части выходит не более одного ребра, а для системы второго типа в каждую вершину правой части входит не более одного ребра. Примеры противоречивых систем приведены на рис. 18.

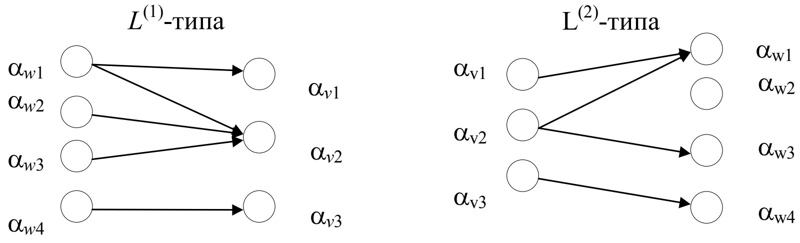


Рис. 18

Рассмотренные понятия позволяют качественно оценить экспертную информацию. Естественным требованием к ней является то, что система нечетких высказываний должна быть лингвистически полной невырожденной и непротиворечивой.

3.7 ПРЕДСТАВЛЕНИЕ ЭКСПЕРТНОЙ ИНФОРМАЦИИ НА ТРУДНО ФОРМАЛИЗУЕМЫХ ЭТАПАХ ПРОЕКТИРОВАНИЯ

Процесс автоматизации проектирования, в настоящее время, в основном, охватывает этапы связанные с поиском наилучших конструктивных и технологических решений; созданием баз данных отдельных частей объектов проектирования; использованием эффективных технических средств, обеспечивающих оперативную работу проектировщика. Однако остаются задачи, решение которых ищет сам пользователь-конструктор. И эти решения зависят от его физического и психологического состояния в разные отрезки времени проектирования. Возникает противоречие: с одной стороны, требуется сократить сроки и повысить качество проектных работ, а с другой, – некоторые этапы проектирования невозможно автоматизировать. Известно, что большую часть изделий составляют типовые изделия, при проектировании которых у конструкторов имеется опыт. Поэтому при разработке САПР типовых изделий указанное выше противоречие разрешается включением в САПР экспертной системы.

Использование ЭС позволит:

1) ускорить процесс проектирования;
2) повысить качество проектирования за счет использования накопленного опыта;
3) уменьшить субъективное влияние пользователя САПР на процесс проектирования.

К настоящему времени известно достаточно большое число работ, посвященных исследованиям различных моделей, имитирующих действия эксперта. При построении моделей используются самые различные математические теории, такие как:

– теория информации;
– теория автоматов;
– теория статических решений;
– теория полезности;
– векторная оптимизация.

Однако, результаты исследований позволяют сделать вывод, что не одна из перечисленных теорий не дает возможность корректно описать действия эксперта и построить адекватные модели для всех случаев. Специфика выбора модели принятия решения состоит в том, что разрабатываемые алгоритмы должны учитывать качественную информацию, исходящую от эксперта и представленную в лингвистической форме.

В настоящее время известно довольно большое число моделей принятия решений, в которых в качестве математического аппарата используется теория нечетких множеств. В основном, эти алгоритмы предполагают задание:

– множества альтернатив выбора,
– критериев выбора,
– ограничений,
– отношений предпочтения и т.д.

В зависимости от выбора решений все этапы проектирования можно разделять на два класса:

1 К первому классу относятся этапы, в результате которых происходит выбор значений параметров проектирования. В этом случае значениями определяемого параметра является подмножество множества действительных чисел. Для этих задач разработаны модели принятия решений, использующие нечеткие правила MODUS PONENS и индуктивную схему вывода.

2 Ко второму классу относятся этапы, цель которых – выбор варианта (схемы) проектирования или значения параметра изделия из конечного достаточно небольшого заранее заданного множества. Для решения таких задач также используется нечеткое правило MODUS PONENS, нечеткое индуктивная схема вывода, а также модель, использующую нечеткую экспертную информацию второго рода.

4 ПРИНЯТИЕ РЕШЕНИЙ В СИСТЕМАХ, ОСНОВАННЫХ НА ПРАВИЛАХ

4.1 ВЫБОР РЕШЕНИЙ НА ОСНОВЕ ЧЕТКОЙ ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Использование моделей и алгоритмов принятия решений на основе экспертной информации связано с решением задачи представления данной информации в виде, пригодном для использования.

При выборе решений в четких условиях экспертная информация представила в виде системы условных высказываний, устанавливающих взаимосвязь между четкими значениями входных и выходных параметров процесса принятия решения [7].

Если в зависимости от возможных четких значений входных параметров делается вывод о значениях выходного параметра, то такая система называется системой $L^{(1)}$-типа и представляется в виде

$$L^{(1)} = \begin{cases} L_1^{(1)} :< \text{если } A_1 \text{ то } B_1 > \\ L_2^{(1)} :< \text{если } A_2 \text{ то } B_2 > \\ \dots \\ L_m^{(1)} :< \text{если } A_m \text{ то } B_m > \end{cases}, \tag{39}$$

где m – число экспертных высказываний; A_j – четкое значение входного параметра; B_j – четкое значение выходного параметра или некоторое конкретное действие процесса проектирования.

В случаях, когда в зависимости от возможных значений выходной ситуации (B_j) экспертом делается предположение о возможной входной ситуации (A_j), система экспертных высказываний называется системой $L^{(2)}$-типа и представляется в виде

$$L^{(2)} = \begin{cases} L_1^{(2)} :< \text{если } B_1 \text{ то } A_1 > \\ L_2^{(2)} :< \text{если } B_2 \text{ то } A_2 > \\ \dots \\ L_m^{(2)} :< \text{если } B_m \text{ то } A_m > \end{cases}. \tag{40}$$

Рассмотрим теперь механизм выбора решений в четких условиях.

(1) При задании экспертной информации системой высказываний $L^{(1)}$-типа выбор решения основывается на правиле MODUS PONENS.

Пусть A и B – произвольные четкие высказывания. Правило <если A_j то B_j> – правило из системы $L^{(1)}$. Обозначим через $T(A/A_j)$ – истинность высказывания A относительно A_j; $T(B/B_j)$ – истинность высказывания B относительно B_j:

$$T(A/A_j)=\begin{cases}1, \text{при } A=A_j;\\ 0, \text{в других случаях,}\end{cases} \qquad (41)$$

$$T(B/B_j)=\begin{cases}1, \text{при } B=B_j;\\ 0, \text{в других случаях.}\end{cases} \qquad (42)$$

(В данном случае значения истинностей являются частным случаем значений истинности для нечетких высказываний).

Согласно правилу MODUS PONENS из высказываний <если A_j то B_j> И <A> выводимо высказывание <B>.

Формально правило MODUS PONENS записывается в виде:

<если A_j то B_j>;

<A – истинно>; (43)

<B – истинно>;

Истинностью правила modus ponens для схемы вывода (43) называется величина

$$T(L_j^{(1)}, A, B)=\begin{cases}1, \text{если } A=A_j \text{ и } B=B_j;\\ 0, \text{в других случаях.}\end{cases} \qquad (44)$$

(2) При задании экспертной информации системой высказываний $L^{(2)}$-типа выбор решения основывается на индуктивной схеме вывода. Согласно ей из высказываний <если A_j то B_j> И <A> следует правдоподобность высказывания <B>. Формально такая схема запишется в виде:

<ЕСЛИ A_j ТО B_j>;

<A – истинно>; (45)

<B – более правдоподобно>.

Понятие истинности данной схемы вывода выводится аналогично схеме modus ponens (44).

Тогда, при выборе решений в четких условиях в случае, когда информация задана полной непротиворечивой системой первого типа, правило modus ponens соответствует выбору такого выходного высказывания, при котором истинность (44) схемы вывода (43) достигает своего наибольшего значения, т.е. единицы. Для принятия решения используется алгоритм прямой цепочки рассуждений.

Аналогично, при задании экспертной информации системой четких высказываний второго типа индуктивная схема вывода соответствует выбору такого высказывания B, при котором истинность (44) схемы вывода (45) также достигает своего максимума (= 1). В данном случае для принятия решения используется алгоритм обратной цепочки рассуждений.

4.1.1 Алгоритм прямой цепочки рассуждения

Прямой алгоритм, в ходе рассуждения движется от исходных данных, через систему правил к результату. При этом из исходных данных поочередно выбираются все факты, последовательно сравниваются с частью "*если*" всех: правил базы знаний. Прямая цепочка рассуждений позволяет пользователю экспертной системы получить всю информацию, сходящую из входных данных.

Базу знаний экспертной системы можно условно разбить на две части: Базу правил и Базу данных. База правил включает в себя статические знания о предметной области и формируется на основе экспертной информации. Процесс построения базы правил является наиболее трудоемким и трудно формализуемым этапом разработки экспертной системы. База данных – знания в виде фактов, которые, в свою очередь, подразделяются на постоянные и временные факты. Временные факты описывают определенную ситуацию, в процессе работы экспертной системы и зависят от входных данных. Постоянные факты определяют логические зависимости – между объектами, предметной области и не меняются в ходе эксплуатации экспертной системы [2, 4].

Работу алгоритма рассмотрим на основе знаний, представленных в виде фактов и правил, являющихся фрагментом базы знаний системы проектирования оборудования автоматизированной линии гальванопокрытий.

База знаний содержит три факта:

1) кислота •= агрессивная жидкость;
2) азотная кислота = кислота;

3) сернистая кислота = кислота
и 9 правил:

1 Если среда = кислота" и концентрация > 70 %, то среда "концентрированная кислота.
2 Если среда = кислота и концентрация < 70 %, то среда = разбавленная кислота.
3 Если среда = концентрированная кислота, то материал ванны = хромоникелевая сталь.
4 Если среда = разбавленная кислота, то материал ванны =в углеродистая сталь.
5 Если среда = агрессивная жидкость, то футеровка = есть.
6 Если футеровка = есть и среда = сернистая кислота, то материал футеровки – свинец.
7 Если футеровка = есть и среда = азотная кислота, то материал футеровки = винипласт.
8 Если температура = меньше 100 °С, то тип обогрева = пароводяная рубашка.
9 Если температура = больше 100 °С, то тип обогрева = электронагреватель.

После формирования базы знаний определяются вспомогательные структуры данных, необходимые для работы алгоритма рассуждений. Для удобства работы представим эти структуры в виде табл. 4 – 7.

4 Список переменных условия

Переменная	Переменная
1. среда	11.футеровка
2. концентрация	12. среда
3. среда	13. футеровка
4. концентрация	14. среда
5 среда	15. температура
6.	16.
7. среда	17. температура
8.	18.
9. среда	
10.	

В эту таблицу заносятся имена переменных, присутствующих в условной части правил.

Способ организации: для каждого правила базы знаний в списке резервируется N позиций, где N = max [N }, i = 1...k; N – количество переменных в условной части первого правила; k – количество правил.

5 Список переменных условия расширенный

Имя	Признак инициализации	Значение
1	2	3
Среда	Не определена	
Концентрация	Не определена	
Футеровка	Не определена	
Температура	Не определена	

Графы 2 и 3 заполняются по ходу работы алгоритма.

В графу "Имя" заносятся имена переменных условия, причем каждая переменная заносится только один раз. Графа "Признак инициализации" заполняется до начала работы экспертной системы значением "не определена". Графа "Значение" в начале работы пуста.

6 Указатель переменных условия

Номер правила	Номер условия

Указатель предназначен для отслеживания текущего события в цепочке рассуждений и состоит из двух частей:

а) номера правила, с которым в настоящий момент работает система;
б) номера условия соответствующего правила: данная графа необходима в связи с тем, что правило, в общем случае, может содержать более одного условия.

7 Очередь переменных логического вывода

Переменная

Эта таблица представляет собой список переменных условия и определяет последовательность их сопоставления с условными частями правил.

Определив систему фактов и правил и сформировав все необходимые вспомогательные таблицы, получим состав базы знаний, обеспечивающий работу алгоритма прямой цепочки рассуждений. Теперь система готова принять от пользователя входные данные. Полученная информация интерпретируется системой и заносится в соответствующие таблицы.

Пусть пользователю требуется получить консультацию о конструктивных особенностях проектируемой ванны при условии, что в ней будет находиться сернистая кислота с концентрацией до 75 % при температуре от 80 до 90 °С.

Входная информация представляется следующим образом:

среда = сернистая кислота
концентрация = более 70 %
температура = меньше 100 °С

На основе списка переменных условия и системы правил можно определить допустимые значения каждой переменной условия, которые используются при интерпретации исходной информации во внутреннее представление системы. Эта информация заносится в очередь переменных логического вывода и в расширенный список переменных условия (табл. 5).

Далее выбирается переменная, стоящая в очереди первой ("Среда") и отыскивается правило, содержащее в условной части эту переменную. Поиск осуществляется по списку переменных условия.

Первым правилом, содержащим переменную "Среда", является правило 1. При этом указатель переменных условия устанавливается как первое условие первого правила (табл. 8).

Указатель переменных условия при анализе первого правила

1	**1**

Из списка переменных условия следует такие, что для правила 1 в условной части содержится две переменных ("Среда" и "Концентрация"). Обратившись к расширенному списку переменных условия, определяем, что обе эти переменные проинициализированы. При этом имеется логическое несоответствие в значениях переменной "Среда" в расширенном списке и в условной части правила. Такое несоответствие возможно по двум причинам:

1) по причине смыслового различия (сернистая кислота – азотная кислота, концентрированная кислота – разбавленная кислота);

2) по причине того, что одно из значений является более общим и включает в себя другое (агрессивная жидкость –" концентрированная кислота •–" кислота → азотная кислота).

Выявление причин несоответствия значений выполняется на основе анализа группы фактов базы знаний. В рассматриваемом примере в базе знаний имеется факт.

Сернистая кислота = кислота, откуда следует, что понятие "сернистая кислота" является частным случаем понятия "кислота". В результате этого первое условие первого правила является выполненным. Указатель переменных условия устанавливается на второе условие первого правила. Значения переменной "Концентрация" в расширенном списке переменных условия и в условной части правила совпадают, следовательно, все правило считается выполненным. Согласно части "То" этого правила, переменной "Среда" присваивается значение "концентрированная кислота". Изменение значения переменной отражается в расширенном списке переменных условия (табл. 9).

Изменения в расширенном списке

Среда	Определена	Концентрированная кислота

В части "то" правила содержится переменная условия. Она должна быть помещена в очередь переменных логического вывода. В данном случае этого не происходит, так как переменная "Среда" уже есть в очереди. Кроме того, в базу знаний добавляется новый факт:

Сернистая кислота = концентрированная кислота.

Этот факт является временным. Он необходим только для работы: при конкретных исходных данных и удаляется из базы знаний при завершении работы системы. Временные факты формируют логическую цепочку, на основе которой система в дальнейшем выясняет причины возникшего несоответствия значений переменных.

Далее система проверяет, имеется ли еще какое-либо правило, в условной части которого содержится переменная "Среда". Если такого правила не существует, то переменная удаляется из очереди. В противном случае, указатель устанавливается на первое условие найденного правила (в данном случае правило 2). Анализ найденного правила выполняется системой аналогично схеме, рассмотренной выше. Такой анализ показывает, что правило 2 не выполняется. Рассмотрим теперь правило 3. Око считается выполненным. Согласно части "То" этого правила переменной "Материал ванны" присваивается значение "хромоникелевая сталь". Эта переменная не содержится в списке переменных условия, следовательно, она является выводом.

Переменные вывода помещаются в список переменных вывода, который при завершении работы система определяет результат ее работы.

После анализа правила 3 состояние структур данных в базе знаний следующее:

– очередь не изменилась;
– расширенный список переменных условия не изменился;
– список переменных вывода содержит одну строку;
– число фактов в базе знаний – 4;
– указатель переменных условия (табл. 10).

Указатель переменных условия (правило 3)

3	1

Дальнейший анализ показывает, что правило 4 не выполняется из-за несоответствия значения переменной "Среда" ("Разбавленная кислота" – "Концентрированная кислота"). Выполняем рассмотрение следующего правила, содержащего переменную условия "Среда" (правило 5). Это правило выполняется. В структуре данных возникли изменения:

- в очередь переменных логического вывода добавляется "Футеровка";
- изменяется значение переменной в расширенном списке переменных условия (табл. 11);
- указатель переменных условия.

Изменения в расширенном списке

Футеровка	Определена	Есть

- в список переменных вывода добавляется переменная "Материал Футеровки" со значением "свинец";
- очередь не изменилась;
- списки переменных условия не изменились;
- указатель переменных условия (табл. 12).

Указатель переменных условия (правило 6)

6	2

Правило 7 не выполняется.

После этого система не находит больше правила, содержащего переменную "Среда". Эта переменная исключается из очереди. Первой в очереди становится переменная "Концентрация". Для нее должны быть выполнены все предыдущие шаги. В нашем примере правила, содержащие в условной части переменную "Концентрация", уже были проверены (правила 1 и 2), действия, предусмотренные частью "То" были выполнены, поэтому переменная условия "Концентрация" исключается из очереди.

Теперь первая в очереди переменная "Температура". Анализ правил, содержащих переменную "Температура", приводит к следующему состоянию структур данных:

- очередь пустая (после рассмотрения переменной "Температура" в очереди остается переменная условия "Футеровка" , которая исключается аналогично переменной "Концентрация");
- списки переменных условия не изменились;
- указатель (табл. 13);
- к списку переменных вывода добавили "Тип обогрева", равную "пароводяная рубашка".

13 Указатель переменных условия (правило 9)

9	1

Так как очередь пуста, процесс рассуждения заканчивается. Результатом работы алгоритма являются следующие решения:

- материал ванны – хромоникелевая сталь;
- ванна имеет защитную футеровку;
- материал футеровки – свинец;
- тип обогрева – пароводяная рубашка.

В простейшем случае результатом работы является список логических выводов, содержащий переменные вывода и их значения. Однако существует ряд случаев, когда в качестве выходной информации используются и переменные условия. Это позволяет обеспечить более логичное и наглядное представление исходных данных системы в виде, удобном пользователю. Так, в рассмотренном выше примере переменная вывода "Материал футеровки" входит в список выводов только при определенных значениях переменкой условия "Футеровка". Поэтому переменная условия "Футеровка" также может быть рассмотрена как результат работы системы.

Предложенный алгоритм не является самым универсальным и допускает ряд модификаций. В частности, допускается различное толкование недостатка входных данных. В ходе работы алгоритма может возникнуть ситуация, когда при обращении к некоторому правилу выяснится, что переменная условия в нем не была определена.

В данной ситуации система может либо запросить недостающие данные у пользователя, либо выдать сообщение о недостающей информации и прекратить рассмотрение данного правила. Каждый из этих подходов имеет свои преимущества и недостатки. В первом случае достигается более полный список выводов, но пользователю приходится вводить большее количество информации. В противном случае, недостаток входной информации может сделать невозможным получение достаточно важных выводов, поэтому возникает задача выявления минимально необходимого комплекта входной информации.

Существенного ускорения работы системы можно достичь за счет исключения из дальнейшего рассмотрения уже проверенных правил. Для тех правил, часть "то" которых была выполнена, устанавливается признак "Выполнено" и они, в дальнейшем, не обрабатываются системой. При этом следует помнить, что правило может не выполняться из-за недостатка входной информации. По ходу работы алгоритма может быть найдена дополнительная информация, что позволит обработать эти правила.

4.1.2 Алгоритм обратной цепочки рассуждений

В отличие от алгоритма прямой цепочки рассуждений, при обратной цепочке рассуждений система ведет работу от гипотез и результатов к фактам и входной информации. Выдвинутая гипотеза проверяется на основе правил базы знаний. В случае, если входной информации недостаточно для подтверждения гипотезы, происходит обращение к пользователю.

Работу алгоритма обратной цепочки рассуждений [2, 4] рассмотрим на основе примера, приведенного в предыдущем разделе – на основе фрагмента базы знаний проектирования гальванического оборудования. Набор фактов и правил в данном случае остается без изменений.

Как и в прямом алгоритме, введем ряд вспомогательных таблиц, необходимых для работы алгоритма.

1 Список переменных условия (табл. 4).

2 Расширенный список переменных условия (табл. 5) Построение этих таблиц рассмотрено ранее.

3 Список логических выводов. В список заносятся последовательно номера **всех** правил базы знаний и имена переменных, расположенных в части "то" соответствующих правил.

4 Стек логических выводов (табл. 14)

14 Стек логических выводов

Номер правила	Номер условия

Пусть теперь системе требуется найти ответ на вопрос: Какой материал необходим для изготовления защитной *i* футеровки проектируемой ванны?

Алгоритм ответа на вопрос будет включать в себя следующие шаги:

1 Определяем переменную, значение которой должно быть найдено. В нашем примере это "Материал футеровки".

2 Находим переменную "Материал футеровки" в списке логических выводов. В стек логических выводов заносим номера правила и условия, содержащие вывод по материалу. Данный шаг алгоритма рассуждений допускает некоторые модификации:

а) В стек заносятся в обратном порядке номера всех правил, содержащих в части "то" искомую переменную (табл. 15).

15 Стек логических выводов после первого обращения к базе знаний

6	1
7	1

Затем выбираем из стека верхнее правило и начинаем его обработку. Если правило доказать не удалось, из стека выбираем следующее правило и т.д. В дальнейшем будем использовать данную модификацию алгоритма.

б) В стек заносится номер первого найденного правила, а на соответствующую строку в списке логических выводов установим указатель. Если правило недоказуемо, повторяем поиск в базе знаний, начиная его с текущей позиции указателя (табл. 16).

16 Стек и список логических выводов после первого обращения к базе знаний

Стек

6	**1**

Список

Правила	Переменная
6	Материал футеровки

После этого система приступает к рассмотрению найденного правила. Из списка переменных условия определяем, что переменными условия в правиле 6 являются "Футеровка" и "Среда". Выбираем первую переменную – "Футеровка" из расширенного списка переменных условия определяем, что переменная не была проинициализирована. Переменная "Футеровка" есть в списке логических выводов (правило 5). В этом случае система помещает номер правила в стек логических выводов (табл. 17).

Стек логических выводов (шаг 2)

7 5	1
6	1

7	1

3 В правиле 5, через список переменных условия, система определяет переменную условия "Среда" и через расширенный список переменных условия проверяет, имеет ли она какое-либо значение. В данном примере, переменная "Среда" не проинициализирована, но в то же время она содержится в списке логических выводов (правило 1). Заносим в стек логических выводов информацию по найденному правилу (табл. 18).

18 Стек логических выводов (шаг 3)

1	1
5	1
6	1
7	1

4 Через список переменных условия определяем, что для первого правила переменными условия являются "Среда" и "Концентрация". Выбираем первую переменную – "Среда". Из расширенного списка видно, что она не проинициализирована, но в то же время она содержится в другой строке списка логических выводов (правило 2). Информация по найденному правилу заносится в стек логических выводов (табл. 19).

19 Стек логических выводов (шаг 4)

2	1
1	1
5	1
6	1
7	1

Первой переменной условия правила 2 является переменная "Среда", Она не проинициализирована и в списке логических выводов нет больше строк, содержащих **ее**. Система должна задать вопрос о конкретном значении этой переменной. Предположим, что на вопрос системы был дан ответ "сернистая кислота". После этого в расширенном списке переменных условия появляется информация, представленная в табл. 20.

20 Изменения в расширенном списке (шаг 4)

Среда	Определена	Сернистая кислота

5 Далее система проверяет значение переменной "Среда" в условной части правила и в расширенном списке переменных условия Эти значения не совпадают. О причине несовпадения можно узнать из анализа фактов базы знаний. В нашем примере база знаний содержит факт:

Сернистая кислота = кислота.

Следовательно, первое условие правила 2 можно считать выполненным. Если же подобного факта найти не удается, условие и все правило считается невыполненным.

6 В стеке логических выводов для правила 2 меняется указатель условия (табл. 21).

21 Изменения в стеке логических выводов

2	2

В этом случае система приступает к анализу второго условия правила 2. Анализ этого условия приводит к необходимости запросить пользователя о значении переменной "Концентрация". Предположим, что на запрос был получен ответ:

Концентрация = более 70 %

При этом в расширенном списке переменных условия изменяется значение переменной "Концентрация" (табл. 22).

22 Изменения в расширенном списке (шаг 6)

Концентрация	Определена	более 70 %

Анализ значений переменной "Концентрация" в условной части второго правила и в расширенном списке переменных условия и далее на группе фактов показывает, что условие 2, а следовательно, и все правило 2 не выполняется. При этом из стека логических выводов исключается верхняя строка. Теперь правилом, требующим дальнейшего рассмотрения, является правило 1 (согласно стеку).

7 Анализ правила 1 показывает, что оно выполняется. При этом значение переменной "Среда" в расширенном списке переменных условия изменяется (табл. 23).

23 Изменения в расширенном списке (шаг 7)

Среда	Определена	Концентрированная кислота

А в базу знаний добавляется временный факт:

Сернистая кислота = концентрированная кислота.

После этого исключаем из стека строку, содержащую правило

8 Продолжается рассмотрение первого условия правила 5. Анализ показывает, что оно выполняется (с помощью группы фактов устанавливается соответствие между "концентрированной кислотой" из расширенного списка и "агрессивной жидкостью" из условной части правила). При этом в расширенном списке переменных условия переменной "Футеровка" присваивается значение "Есть" (табл. 24).

24 Изменения в расширенном списке (шаг 8)

Футеровка	Определена	Есть

Правило 5 также исключается из стека к система переходит к анализу правила 6. Анализ этого правила показывает, что первое условие выполняется ("Футеровка" = "есть"); переменной условия 2 правила 6 является переменная "Среда". Также с помощью группы фактов устанавливается соответствие между "концентрированной кислотой" из расширенного списка и "сернистой кислотой" из условной

части правила. Условие выполняется, следовательно, выполняется и все правило. Согласно этому правилу:

Материал футеровки = свинец.

Ответ на поставленный вопрос получен и система завершает свою работу.

Вообще говоря, работа системы завершается в двух случаях:

а) Проинициализирована переменная вывода, значение которой требовалось определить. В этом случае система сработала корректно, значение переменной есть результат работы.

б) Переменная не проинициализирована и при очередном обращении к стеку логических выводов выявляется, что он пуст. В этом случае результата достичь не удалось, выдается сообщение об ошибке.

Отличие в алгоритме работы с базой знаний при обратной цепочке рассуждений по сравнению с алгоритмом при прямой цепочке рассуждений состоит, прежде всего, в том, что при прямой цепочке анализ выполняется от переменных условия к переменным вывода; в обратной же все иначе – от переменных вывода к переменным условия. В соответствии с этим несколько изменяется и структура данных базы знаний.

4.2 ВЫБОР ЗНАЧЕНИЙ ПАРАМЕТРОВ НА ОСНОВЕ НЕЧЕТКОГО ПРАВИЛА MODUS PONENS

После построения функции принадлежности, которые определяют все нечеткие значения входных и выходных параметров, переходим к формализации отношений между входными и выходными параметрами. Наиболее простой способ задания таких отношений – это представление их в виде системы нечетких высказываний вида

$$\widetilde{L} = \begin{cases} \widetilde{L}_1 : <\text{если } \widetilde{E}_{11} \text{ или...или } \widetilde{E}_{1n_1} \text{ то } \widetilde{B}_1 > \\ ... \\ \widetilde{L}_m : <\text{если } \widetilde{E}_{m1} \text{ или...или } \widetilde{E}_{mn_m} \text{ то } \widetilde{B}_m >, \end{cases} \tag{46}$$

где m – число базовых нечетких значений выходной лингвистической переменной β_v; $\widetilde{B}_i\left(i = 1,\overline{m}\right)$– нечеткое высказывание $\langle\beta_i \text{ есть } \alpha_{vi}\rangle$, где α_{vi} – нечеткое значение (переменная) лингвистической переменной β_v; $\widetilde{E}_{ji}\left(i = \overline{1, n_i}; j = \overline{1, m}\right)$– нечеткое высказывание, которое отражает i-ю входную ситуацию, которая может возникнуть, если выходная лингвистическая переменная примет нечеткое значение α_{vi};

$$\widetilde{E}_{ji} : \langle\beta_x \text{ есть } \alpha_{xij} \text{ и } \beta_y \text{ есть } \alpha_{yij} \text{ и } \beta_z \text{ есть } \alpha_{zij} \text{ и...}\rangle;$$

где $\beta_x, \beta_y, \beta_z$ ...– входные лингвистические переменные, а $\alpha_{xji}, \alpha_{yji}, \alpha_{zij}$ ...– их нечеткие значения.

Возможно представить исходную систему нечетких высказываний в более компактной форме, используя правила преобразования нечетких высказываний. Вводится нечеткое высказывание $\widetilde{A}_j$, вида: $\widetilde{A}_j : \langle\beta_W is \alpha_{Wj}\rangle$, где β_W – обобщенная входная лингвистическая переменная; α_{Wj} – ее нечеткое значение с функцией принадлежности:

$$\mu_{Wj}(\omega) = \max\left\{\min\left[\mu_{xij}(x), \mu_{yij}(x), \mu_{zij}(x), ...\right]\right\}. \tag{47}$$

Тогда система нечетких высказываний примет вид

$$\widetilde{L}^{(1)} = \begin{cases} \widetilde{L}_1 : \langle\text{если } \widetilde{A}_1 \text{ то } \widetilde{B}_1\rangle \\ .... \\ \widetilde{L}_m : \langle\text{если } \widetilde{A}_m \text{ то } \widetilde{B}_m\rangle. \end{cases} \tag{48}$$

Таким образом, получили нечеткую систему, которая связывает обобщенный входной параметр $\omega \in W = X^*Y^*Z^*$...и выходной параметр $v \in V$.

Для выбора решений в нечетких условиях, когда экспертная информация представлена нечеткой системой первого типа (46), можно использовать схему вывода, основанную на нечетком правиле modus ponens.

Нечетким правилом modus ponens называется следующая схема вывода:

$$\begin{array}{l} \widetilde{L}^{(1)}; \\ \widetilde{A}'\text{–истина}; \\ \hline \widetilde{B}'\text{–истина.} \end{array} \qquad (49)$$

Истинность данной схемы представляет собой нечеткое множество

$$T_{mp}(\widetilde{L}^{(1)}, \widetilde{A}', \widetilde{B}') = \{< \mu_{mp}^{(1)}(\tau) / \tau >\} = \bigcap_{j=1,m} T_{mp}(\widetilde{L}_j^{(1)}, \widetilde{A}', \widetilde{B}'),$$

где $\forall \tau \in [0,1] \mu_{mp}^{(1)}(\tau) = \dfrac{\min \mu_j^{(1)}(\tau)}{j = 1, m}$; $\min \mu_j^{(1)}(\tau)$ – функция принадлежности нечеткого множества $T_{mp}(\widetilde{L}_j^{(1)}, \widetilde{A}', \widetilde{B}')$ – определяется как

$$\mu_j^{(1)}(\tau) = 1 \cap (1 - \mu_{TA}(\tau) + \mu_{Tb}(\tau)),$$

где TA – истинность высказывания $\widetilde{A}'$ относительно $\widetilde{A}_j$; TB – истинность высказывания $\widetilde{B}'$ относительно $\widetilde{B}_j$.

Для определения значений выходного параметра при конкретных значениях (четких) параметров возможно использовать нечеткое правило "modus ponens":

$$\begin{array}{l} \widetilde{L}^{(1)}; \\ A'\text{– истина}; \\ \hline B'\text{ – истина,} \end{array} \qquad (50)$$

здесь A' = <β_W есть w'>, B' = <β_v есть v'> – четкие высказывания о значении обобщенного входного и выходного параметров.

Так как четкое значение v можно рассмотреть как нечеткую переменную α_v, характеризуемую нечетким множеством с функцией принадлежности

$$\mu_{v'}(v) = \begin{cases} 1, \text{ если } v = v'; \\ 0, \text{в других случаях}; \end{cases} \qquad (51)$$

то истинность четкого высказывания относительно нечеткого высказывания полностью определяется одним значением $\mu_{v'}(v') = 1$. Тогда истинность правила modus ponens

$$T_{mp}(\widetilde{L}^{(1)}, A', B') = \{< \mu_{mp}^{(1)}(1) / 1 >\}, \qquad (52)$$

где

$$\begin{aligned} &\mu_{mp}^{(1)}(1) = 1 \cap (1 - \mu_{W1}(w') + \mu_{v1}(v')) \cap (1 - \mu_{W2}(w') + \mu_{v2}(v')) \\ &\cap \ldots \cap (1 - \mu_{Wm}(w') + \mu_{vm}(v')) \end{aligned} \qquad (53)$$

Величина (53) называется степенью истинности правила modus ponens для нечеткой системы высказываний первого типа. Данное понятие отражает степень соответствия четкого значения v' выходного параметра V четкому значению w' обобщенного входного параметра W при задании экспертной информации системой нечетких высказываний первого типа.

Введение понятия степени истинности нечеткого правила modus ponens позволяет сформулировать постановку задачи выбора выходного параметра: найти такие значения выходного параметра V_0, для которых степень истинности (53) схемы (49) на основе нечеткого правила "modus ponens" будет максимальной.

4.3 НЕЧЕТКАЯ МОДЕЛЬ ВЫБОРА ПАРАМЕТРОВ ПРИ ИНДУКТИВНОМ ЛОГИЧЕСКОМ ВЫВОДЕ

Рассмотрим теперь случай, когда экспертная информация задана системой нечетких высказываний второго типа (36). Задача заключается в том, чтобы для заданных значений входных параметров x, y, z,… входных параметров выбрать значения выходного параметра V.

Тогда индуктивную схему выбора можно записать:

$$\frac{\tilde{L}^{(2)}; \quad A' \text{– истина;}}{B' \text{ – истина.}} \qquad (54)$$

Здесь $A' = <\beta_W$ есть $w'>$, $B' = <\beta_v$ есть $v'>$ – четкие высказывания о значении обобщенного входного и выходного параметров.

4.4 ВЫБОР ЗНАЧЕНИЙ ПАРАМЕТРОВ ПРИ ИНДУКТИВНОМ ВЫВОДЕ НА ОСНОВЕ НЕЧЕТКОГО ПРАВИЛА MODUS PONENS

Следует отметить, что дедуктивная (49) и индуктивная (54) схемы вывода принципиально отличаются друг от друга. В дедуктивной схеме высказывания о значениях входных параметров ($\tilde{A}'$ и $\tilde{A}_j$) являются посылками как в самой схеме вывода, так и внутри системы $\tilde{L}^{(1)}$, а высказывания о значениях выходных параметров (B' и B_j) являются следствиями.

В индуктивной схеме вывода высказывания о значениях входных параметров являются посылкой для самой схемы (A') и следствием внутри системы $\tilde{L}^{(2)}$ (A_j), а высказывания о значениях выходного параметра являются следствием для схемы вывода (B'), но посылкой внутри системы $\tilde{L}^{(2)}$ (B_j).

Поэтому для выбора значений выходного параметра V на основе правила modus ponens необходимо индуктивную схему вывода преобразовать в дедуктивную. Для этого систему высказываний первого типа преобразуем в эквивалентную ей систему первого типа, используя правило контрапозиции:

$$\text{если } A \text{ то } B \equiv \text{если} \neg B \text{ то} \neg A\,. \qquad (55)$$

Применяя это правило к высказываниям из системы (34), получаем:

$$\text{если } \tilde{B}_j \text{ то } \tilde{A}_j \equiv \text{если} \neg\tilde{A}_j \text{ то} \neg\tilde{B}_j\,, \qquad (56)$$

где $\neg\tilde{A}_j = <\beta_W \text{ есть } \alpha^*_{Wj}>$; $\neg\tilde{B}_j = <\beta_V \text{ есть } \alpha^*_{Vj}>$ – нечеткие переменные с функциями принадлежности:

$$\forall w \in W = X * Y * Z * \ldots \quad \mu^*_{Wj}(w) = 1 - \mu_{Wj}(w)\,; \qquad (57)$$

$$\forall v \in V \quad \mu^*_{Vj}(v) = 1 - \mu_{Vj}(v)\,. \qquad (58)$$

Введем обозначения $\tilde{A}^*_j = \neg\tilde{A}_j$ и $\tilde{B}^*_j = \tilde{B}_j$. Тогда система нечетких высказываний второго типа (34) примет вид

$$\tilde{L}^{(*)} = \begin{cases} \tilde{L}^{(*)}_1 : \langle \text{если } \tilde{A}^*_1 \text{ то } \tilde{B}^*_1 \rangle \\ \ldots\ldots\ldots\ldots \\ \tilde{L}^*_m : \langle \text{если } \tilde{A}^*_m \text{ то } \tilde{B}^*_m \rangle \end{cases} \qquad (59)$$

Тогда схема вывода (54) запишется в виде:

$$\frac{\tilde{L}^{(*)}; \quad A' \text{ – истина;}}{B' \text{ – истина.}} \qquad (60)$$

Степень истинности правила modus ponens (50) для этой схемы примет вид

$$\begin{aligned} \mu^{(2)}_{mp}(1) = 1 \cap (1 - \mu^*_{W1}(w') + \mu^*_{v1}(v')) \cap (1 - \mu^*_{W2}(w') + \mu^*_{v2}(v')) \\ \cap \ldots \cap (1 - \mu^*_{Wm}(w') + \mu^*_{vm}(v')). \end{aligned} \qquad (61)$$

Таким образом, при задании экспертной информации системой нечетких высказываний второго типа (34), математическая постановка задачи выбора значений выходного параметра формулируется так: при заданной системе нечетких высказываний второго типа (34) для значений x, y, z, … входных параметров, найти такие значения выходного параметра v, для которых индуктивная схема вывода (60) имеет наибольшую степень истинности (61) правила modus ponens.

Для решения данной задачи выражение (60) в соответствии с (57), (59) приведем к виду

$$\mu_{mp}^{(2)}(1) = 1 \cap (1 + \xi_1 - \mu_{v1}(v)) \cap \ldots \cap (1 + \xi_m - \mu_{vm}(v)), \qquad (62)$$

где $\xi_j = \mu_{Wj}$, $j = \overline{1, m}$.

Такая запись степени истинности несет в себе много полезной информации.

Для достижения максимального значения степени истинности (=1), необходимо выполнение условий:

$$\begin{gathered} 1 + \xi_1 - \mu_{v1}(v) \geq 1, \\ 1 + \xi_2 - \mu_{v2}(v) \geq 1, \\ \ldots \\ 1 + \xi_m - \mu_{vm}(v) \geq 1. \end{gathered}$$

Отсюда имеем:

$$\begin{gathered} \mu_{v1}(v) \leq \xi_1, \\ \mu_{v2}(v) \leq \xi_2, \\ \ldots \\ \mu_{vm}(v) \leq \xi_m. \end{gathered} \qquad (63)$$

Теперь можно из области V всех возможных значений выходного параметра определить значения, которые удовлетворяют всем этим условиям. Это и будут значения выходного параметра, оптимальные с точки зрения выбора.

Следует отметить, что применение правила modus ponens для системы нечетких высказываний второго типа имеет следующую особенность: множество рекомендуемых значений в произвольном случае состоит из нескольких отдельных интервалов, причем их количество может меняться в пределах от 1 до m + 1. Это связано с тем, что выбор решения максимизирует значение степени истинности (62), что в свою очередь связано с уменьшением значений $\mu_j(v)$ (63). В силу унимодальности функций принадлежности, уменьшение их значений возможно как влево, так и вправо от точки максимума, что и порождает указанную особенность.

4.5 ВЫБОР ЗНАЧЕНИЙ ПАРАМЕТРОВ НА ОСНОВЕ НЕЧЕТКОЙ ИНДУКТИВНОЙ СХЕМЫ ВЫВОДА

Нечеткой индуктивной схемой вывода называется схема

$$\frac{\begin{gathered} <\text{если } \widetilde{B} \text{ то } \widetilde{A}> \\ \widetilde{A}' - \text{истина}; \end{gathered}}{<\widetilde{B}' - \text{истина}> - \text{более правдоподобно}.} \qquad (64)$$

Для системы нечетких высказываний второго типа индуктивная схема вывода принимает вид:

$$\frac{\begin{gathered} \widetilde{L}^{(2)}; \\ A' - \text{истина}; \end{gathered}}{<B' - \text{истина}> - \text{более правдоподобно},} \qquad (65)$$

где $A' = <\beta_W$ есть $w'>$; $B' = <\beta_v$ есть $v'>$ – четкие высказывания о значении обобщенного входного и выходного параметров.

Истинность данной схемы определяется как

$$T_{\text{ис}}(\widetilde{L}^{(2)}, A', B') = \{<\mu_{\text{ис}}^{(2)}(1)/1>\},$$

где

$$\mu^{(2)}_{\text{ис}}(1) = 1 \cap (1-\mu_{W1}(w') + \mu_{v1}(v')) \cap (1-\mu_{W2}(w') + \mu_{v2}(v')) \\ \cap \ldots \cap (1-\mu_{Wm}(w') + \mu_{vm}(v')). \quad (66)$$

Величина (66) называется степенью истинности индуктивной схемы вывода для нечеткой системы высказываний второго типа. Данное понятие отражает степень соответствия четкого значения v' выходного параметра V четкому значению w' обобщенного входного параметра W при задании экспертной информации системой нечетких высказываний второго типа.

Тогда математическая постановка задачи выбора выходного параметра имеет вид: найти такие значения выходного параметра V_0, для которых степень истинности (66) схемы (65) на основе нечеткой индуктивной схемы вывода будет максимальной.

4.6 ВЫБОР ВАРИАНТА

Ранее рассматривались задачи выбора значений параметров объекта. Другим типом задач является выбор некоторого варианта из заранее заданного, достаточно небольшого числа вариантов. В этом случае экспертная информация может быть представлена одним из следующих способов:

$$\widetilde{L}^{(1)} = \begin{cases} \widetilde{L}_1 : \langle \text{если } \widetilde{A}_1 \text{ то } B_1 \rangle \\ \ldots\ldots\ldots\ldots \\ \widetilde{L}_m : \langle \text{если } \widetilde{A}_m \text{ то } B_m \rangle \end{cases} \quad (67)$$

или

$$\widetilde{L}^{(2)} = \begin{cases} \widetilde{L}^{(2)}_1 :< \text{если } B_1 \text{ то } \widetilde{A}_1 > \\ \cdots \\ \widetilde{L}^{(2)}_m :< \text{если } B_m \text{ то } \widetilde{A}_m > \end{cases} \quad (68)$$

Здесь высказывания B_j выражают суть выбора того или иного варианта проектирования, т.е. являются четкими высказываниями: $B_j :< \beta_v \text{ есть } v_j >$, где v_j – четкое значение из конечного множества вариантов.

4.6.1 Выбор варианта при дедуктивном выводе

Дедуктивная схема вывода имеет вид:

$$\frac{\begin{array}{c}\widetilde{L}^{(1)}; \\ A' - \text{истина;}\end{array}}{B' - \text{истина.}} \quad (69)$$

Степень истинности данной схемы для произвольного j-го высказывания системы (65) определяется

$$\mu^{(1)}_{mp}(v_j) = 1 \cap [1-\mu_{w1}(w') + \mu_{v1}(v')] \cap \ldots \\ \cap [1-\mu_{wj}(w') + \mu_{vj}(v')] \cap \ldots \cap [1-\mu_{wm}(w') + \mu_{vm}(v')]. \quad (70)$$

При этом учитываем, что B_j – четкие высказывания и, следовательно:

$$\mu_{vj}(v') = \begin{cases} 1, \text{при } v' = v_j \\ 0, \text{в др. случ.} \end{cases} \quad (71)$$

Для выбора варианта $v_0 \in V$ на основе правила *m*odus ponens необходимо:

1) для каждого $v_j \in V$ ($j = 1 \ldots m$) определить степень истинности по формуле (70);

2) в качестве решения (v_0) выбрать такое $v_j \in V$, при котором степень истинности (70) имеет максимальное значение.

4.6.2 Выбор варианта при индуктивном выводе

Индуктивная схема вывода имеет вид

$$\begin{array}{l}\widetilde{L}^{(2)}; \\ A' - \text{истина}; \\ \hline B' - \text{истина}.\end{array} \qquad (72)$$

Для выбора варианта проектирования на основе правила modus ponens преобразуем систему (68) второго типа в систему первого типа, согласно правилу контрапозиции (52)

$$\widetilde{L}^{(2)} \equiv \widetilde{L}^{*} = \{L_j^{*}\},\ j = \overline{1,m}\,, \qquad (73)$$

где $\widetilde{L}_j^{*} :<$ если $\widetilde{A}_j^{*}$ то $B_j^{*} >$; $\widetilde{A}_j^{*} :< \beta_W$ есть $\alpha_{Wj}^{*} >$, $B_j^{*} :< \beta_V$ есть $\alpha_{Vj}^{*} >$.

Значения α_{Wj}^{*} и α_{Vj}^{*} характеризуются функциями принадлежности:

$$\mu_{Wj}^{*}(w) = 1 - \mu_{Wj}(w) \ \text{ и } \ \mu_{Vj}^{*}(v) = 1 - \mu_{Vj}(v) = \begin{cases} 1, \text{при } v \neq v_j \\ 0, \text{в др. случ.} \end{cases}$$

Задача принятия решения, как и ранее, заключается в выборе такого варианта, при котором степень истинности нечеткого правила modus ponens имеет наибольшее значение.

Для произвольного варианта $v_j \in V$ степень истинности имеет вид:

$$\begin{array}{l}\mu_{mp}^{(2)}(v_j) = 1 \cap [1 - \mu_{W1}^{*}(w) + \mu_{V1}^{*}(v_j)] \cap \ldots \\ \cap [1 - \mu_{Wj}^{*}(w) + \mu_{Vj}^{*}(v_j)] \cap \ldots \cap [1 - \mu_{Wm}^{*}(w) + \mu_{Vm}^{*}(v_j)] = \\ 1 \cap [1 - (1 - \mu_{W1}(w)) + (1 - 0)] \cap \ldots \\ \cap [1 - (1 - \mu_{Wj}(w)) + (1 - 1)] \cap \ldots \cap [1 - (1 - \mu_{Wm}(w)) + (1 - 0)] = \\ 1 \cap [1 + \mu_{W1}(w)] \cap \ldots \cap [\mu_{Wj}(w)] \cap \ldots \cap [1 + \mu_{Wm}(w)]\end{array}$$

Таким образом, в качестве варианта выбирается такой вариант $v_k \in V$, для которого степень истинности $\mu_{Wk}(w)$ имеет наибольшее значение, т.е.:

$$v_0 = v_k \left| \mu_{Wk}(w) = \max_{j = \overline{1,m}} (\mu_{Wj}(w) \right. . \qquad (74)$$

4.7 НЕЧЕТКАЯ МОДЕЛЬ ВЫБОРА ГОТОВОГО РЕШЕНИЯ

Рассмотрим пример использования теории принятия решений в системах автоматизированного проектирования.

Одним из первых этапов при исследовании, расчете или проектировании некоторого объекта является поиск ранее выполненных разработок, которые аналогичны заданию, и описания которых хранятся в базе данных. Любой объект характеризуется совокупностью входных параметров, которая используются для поиска готовых решений. Этот процесс можно представить в виде последовательности следующих этапов:

1) определить степень близости по каждому отдельно взятому входному параметру между новым объектом и объектом, информация о котором хранится в базе данных;
2) определить степень аналогичности объектов по всем входным параметрам по вычисленным на первом этапе степеням близости;
3) ранжировать изделия по степеням аналогичности и выдать объекты с наибольшей степенью в качестве готового решения.

ОПРЕДЕЛЕНИЕ СТЕПЕНИ БЛИЗОСТИ ИЗДЕЛИЙ ПО ОДНОМУ ПАРАМЕТРУ. *ОБОЗНАЧИМ ЧЕРЕЗ* X*1,* X*2,……,* XN *МНОЖЕСТВА ЗНАЧЕНИЙ ПА-*

РАМЕТРОВ. ТОГДА ПРОИЗВОЛЬНЫЙ ОБЪЕКТ Q *ХАРАКТЕРИЗУЕТСЯ НАБОРОМ ВХОДНЫХ ПАРАМЕТРОВ* X_{Q1}, X_{Q2}, ……, X_{QN}. *СТЕПЕНЬ БЛИЗОСТИ ОБЪЕКТОВ P И Q ПО ОДНОМУ ПАРАМЕТРУ МОЖНО ОПРЕДЕЛИТЬ ПО ФОРМУЛЕ*

$$r_i(p/q) = 1 - \frac{|x_i^p - x_i^q|}{x_i^{\max} - x_i^{\min}}, \qquad (75)$$

где $x_i^{\max}$, $x_i^{\min}$ – наибольшее и наименьшее значение i-го параметра (область определения).

Определение степени аналогичности объектов по всем параметрам. При выборе готового решения информация, получаемая от эксперта представляется в виде следующих двух высказываний:

$$\left\langle \text{если}\, \widetilde{C}_1 \text{ или} \ldots \text{или}\, \widetilde{C}_k \text{ то изделие} - \text{аналог} \right\rangle; \qquad (76)$$

$$\left\langle \text{если не}(\widetilde{C}_1 \text{ или} \ldots \text{или}\, \widetilde{C}_k) \text{ то изделие} - \text{не аналог} \right\rangle, \qquad (77)$$

где k – число эталонных ситуаций, при которых соответствующий рассматриваемый объект p является аналогом нового объекта q.

Выражения $\widetilde{C}_j$ $(j = 1...k)$ являются высказываниями следующего вида:

$$\widetilde{C}_j : \left\langle \beta_1^A \text{ есть } \alpha_{ji}^A \text{ и} \ldots \text{и } \beta_N^A \text{ есть } \alpha_{jN}^A \right\rangle, \qquad (78)$$

где $\alpha_{ji}^{(A)}$, $i = 1,\ldots, N$ – нечеткие переменные (например <близко к 1>, <больше 0,5>, <не равно 0> и т.д.), являющиеся значениями лингвистических переменных $\beta_i^{(A)}$ – "аналогичность объекта по i-му параметру"; N – число входных параметров, по которым происходит сравнение объектов.

Приведенные высказывания (76), (77) представляют собой систему нечетких эталонных высказываний первого типа, которая может быть записана в следующем виде:

$$\widetilde{L} = \begin{cases} \left\langle \text{если} \quad \widetilde{A}_1 \text{ то } \widetilde{B}_1 \right\rangle; \\ \left\langle \text{если} \quad \widetilde{A}_2 \text{ то } \widetilde{B}_2 \right\rangle, \end{cases} \qquad (79)$$

где $\widetilde{A}_1 = \left\langle \beta_\Sigma \text{ есть } \alpha_{\Sigma 1} \right\rangle$; $\widetilde{A}_2 = \left\langle \beta_\Sigma \text{ есть } \alpha_{\Sigma 2} \right\rangle$; $\alpha_{\Sigma 1}$ и $\alpha_{\Sigma 2}$ – нечеткие значения лингвистической переменной $\beta_\Sigma = (\beta_1^{(A)}, \beta_2^{(A)}, \ldots\ldots, \beta_N^{(A)})$, функции принадлежности которых согласно правилам преобразования высказываний конъюнктивной и дизъюнктивной форм записываются в виде:

$$\mu_{\Sigma 1}(r_1, r_2, \ldots, r_N) = \max_{j=1,K} \min_{i=1,N} \{\mu_{ji}(r_i)\}; \qquad (80)$$

$$\mu_{\Sigma 2}(r_1, r_2, \ldots, r_N) = 1 - \mu_{\Sigma 1}(r_1, r_2, \ldots, r_N); \qquad (81)$$

$$\widetilde{B}_1 = \left\langle \beta_A \text{ есть } \alpha_{A1} \right\rangle, \ \widetilde{B}_2 = \left\langle \beta_A \text{ есть } \alpha_{A2} \right\rangle,$$

β_A – лингвистическая переменная "аналог изделия" принимает базовые значения α_{A1} = "аналог", α_{A2} = "не аналог".

Функции принадлежности $\mu_{A1}(r)$, $\mu_{A2}(r)$ ($r \in [0,1]$ нечетких переменных α_{A1} и α_{A2} определяются опросом эксперта. При этом справедливы выражения: $\mu_{A1}(0) = 0$; $\mu_{A1}(1) = 1$; $\mu_{A2}(0) = 1$; $\mu_{A2}(1) = 0$;

Для рассматриваемого примера расчета процесса конвективной сушки волокнистых материалов высказывание $\widetilde{A}_1$ может иметь вид:

Пусть A', B' – четкие высказывания: $A' = \langle \beta_1^{(A)}$ есть r_1 и …… и $\beta_N^{(A)}$ есть $r_N \rangle$; $B' = \langle \beta_A$ есть $k_A \rangle$, $k_A \in [0,1]$.

Рассмотрим схему вывода [1, 3]:

$$\frac{\begin{array}{c}\widetilde{L}; \\ A' - \text{истина};\end{array}}{B' - \text{истина}.} \qquad (82)$$

Степень истинности правила modus ponens для схемы вывода (82) имеет вид

$$\mu_{mp}^{(1)}(k_A) = 1 \wedge [1 - \xi_1 + \mu_{A1}(k_A)] \wedge [1 - \xi_2 + \mu_{A2}(k_A)]\,,$$

где $\xi_1 = \mu_{\Sigma 1}(r_1, r_2, ..., r_N)$; $\xi_2 = \mu_{\Sigma 2}(r_1, r_2, ..., r_N)$. Учитывая выражение (81) запишем

$$\mu_{mp}^{(1)}(k_A) = 1 \wedge [1 - \xi_1 + \mu_{A1}(k_A)] \wedge [1 + \xi_1 - \mu_{A1}(k_A)]\,,$$

а так как справедливо равенство $(1 - a + b)\,\&(1 + a - b) = 1 - |a + b|$, то имеем

$$\mu_{mp}^{(1)}(k_A) = 1 \wedge [1 - |\,\xi_1 + \mu_{A1}(k_A)\,|]\,.$$

Отсюда следует, что функция $\mu_{mp}{}^{(1)}$ достигает своего наибольшего значения единицы при таком k_A, при котором $\mu_{A1}(k_A) = \xi_1$, т.е.

$$k_A = \mu_A^{-1}(\xi_1)\,, \tag{83}$$

где

$$\xi_1 = \max_{j=1,\,K}\ \min_{i=1,\,N} \left\{\mu_{ji}(r_i)\right\}\ . \tag{84}$$

Тогда степень аналогичности входных параметров объекта p относительно входных параметров нового объекта q это такое значение $k_{\text{A}} \in [0,1]$, определяемое выражением (83), при котором схема вывода (82) имеет наибольшую степень истинности правила modus ponens.

5 ПРИНЯТИЕ РЕШЕНИЯ В НЕЙРОСЕТЕВЫХ СИСТЕМАХ

Очевидно, чтобы система хорошо работала и решала практические задачи, необходимо ее обучить. Если говорить в общем, то обучение это относительно постоянный процесс изменения поведения при поступлении жизненного опыта. Если говорить о человеке, то результат его обучения оценивается по действиям и поступкам. Обучение же нейронных сетей – более прямой процесс.

Обучение нейронных сетей рассматривается как процесс аппроксимации непрерывной функции $y(X)$ другой функцией $Y(W, X)$, где $X = [x_1, x_2, ..., x_n]^t$ – входной вектор, а $W = [w_1, w_2, ..., w_n]^t$ – вектор весовых коэффициентов.

Задача обучения состоит в выборе вектора W, такого что достигается лучшая аппроксимация, т.е.

$$\rho(Y(W^*, X), y(X)) \le \rho\ (Y(W^*, X), y(X)), \tag{85}$$

где $\rho(Y(W, X), y(X)\ )$ – функция расстояния, которая определяет значение качества аппроксимации между $Y(W, X)$ и $y(X)$).

Все алгоритмы обучения делятся на две большие группы: с учителем и без учителя.

Рассмотрим алгоритм обучения с учителем многослойной нейронной сети. Предполагается, что в каждый момент времени вместе с входами формируется желаемое значение выхода d, которое поступает от учителя (рис. 20).

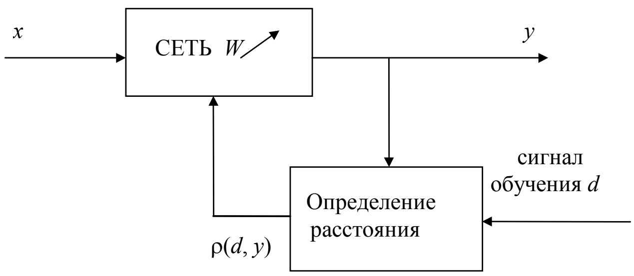


Рис. 20

По значениям реального выхода и желаемого выхода формируется ошибка, которая используется для корректировки параметров нейронной сети. Множество входных и выходных образцов, называемых обучающим множеством, является необходимым для такого способа обучения. "Учитель" оценивает негативное направление градиента ошибки и соответственно сеть уменьшает ошибку. Во многих ситуациях, входы, выходы и вычисляемые градиенты являются детерминистическими, однако, минимизация осуществляется по случайному закону. И, как результат, большинство алгоритмов обучения с учителем используют стохастическую минимизацию ошибки в многомерном пространстве весов.

Рассмотрим стандартный алгоритм обучения многослойных нейронных сетей на основе обратного распространения ошибки (back-propagation).

Цель обучения состоит в определении всех весовых коэффициентов, при которых ошибка вычислений будет минимальной. Обучение сети осуществляется на основе множества пар "вход-выход". Каждый пример обучения состоит из вектора $X = [x_1, x_2, ..., x_{n1}]$ входных сигналов и вектора $D = [d_1, d_2, ..., d_{n3}]$ желаемых результатов. Обучение состоит в определении всех весовых коэффициентов, таких, что значение ошибки между желаемыми и действительными выходными сигналами будет минимальной (близкой к 0).

Рассматриваемый метод использует пошаговый градиентный подход для минимизации функции квадрата ошибки. Тогда локальная функция ошибки для p-го примера обучения формулируется как

$$E_p = \frac{1}{2}\sum_{j=1}^{n_2}(d_{jp} - y_{jp})^2 = \frac{1}{2}\sum_{j=1}^{n_2} e_{jp}^2 \,; \tag{86}$$

Тогда общая функция ошибки имеет вид

$$E = \sum_p E_p = \frac{1}{2}\sum_p \sum_j (d_{jp} - y_{jp})^2, \tag{87}$$

где d_{jp} и y_{jp} – желаемый и действительный выходные сигналы j-го выходного нейрона для p-го образца, соответственно.

Подход, который используется нами, предполагает, что для каждого примера обучения синаптические веса w_{ji}^s (s – число уровней сети) изменяются на величину Δw_{ji}^s пропорционально отрицательному градиенту локальной функции E_p:

$$\Delta w_{ji}^s = -\eta \frac{\partial E_p}{\partial w_{ji}^s}; \ \eta > 0\,, \tag{88}$$

где η – параметр обучения (малое число).

Или в непрерывной форме

$$\frac{dw_{ji}^s}{dt} = -\mu \frac{\partial E_p}{\partial w_{ji}^s}, \ \mu > 0. \tag{89}$$

По этой процедуре минимизируется общая функция ошибки $E = \sum_p E_p$. Рассмотрим нахождение решения на примере двухслойной нейронной сети.

Определим синаптические веса w_{ji}^s ($s = 2$) выходного уровня. Имеем

$$\Delta w_{ji}^2 = -\eta \frac{\partial E_p}{\partial w_{ji}^2} = -\eta \frac{\partial E_p}{\partial o_j^/} \frac{\partial o_j^/}{\partial w_{ji}^2}\,. \tag{90}$$

При этом

$$o_j^/ = \sum_{i=1}^{n_2} w_{ji}^2 h_i\,. \tag{91}$$

Локальная ошибка, называемая "дельта", определяется как

$$\delta_j^2 = \frac{\partial E_p}{\partial o_j^/} = -\frac{\partial E_p}{\partial e_{jp}} \frac{\partial e_{jp}}{\partial o_j^/} = e_{jp} \frac{\partial f^2}{\partial o_{jp}^/} = (d_p - y_p)\frac{\partial f^2}{\partial o_{jp}^/}\,. \tag{92}$$

Общая формула для определения весов в выходном слое

$$\Delta w_{ji}^2 = \eta \delta_j^2 h_i\,. \tag{93}$$

Теперь определим синаптические веса в скрытом слое. Можем записать

$$\Delta w_{ji}^1 = -\eta \frac{\partial E_p}{\partial w_{ji}^1} = -\eta \frac{\partial E_p}{\partial h_j^{/}} \frac{\partial h_j^{/}}{\partial w_{ji}^1} = \eta \delta_j^1 x_i \, , \tag{94}$$

где локальная ошибка для скрытого слоя определяется по формуле

$$\delta_j^1 = -\frac{\partial E_p}{\partial h_j^{/}}, \;\; j = \overline{1, n2}. \tag{95}$$

В данном случае локальная ошибка определяется не напрямую через известные значения. Поэтому, продолжим рассуждения для определения локальной ошибки в скрытом слое. Запишем

$$\delta_j^1 = -\frac{\partial E_p}{\partial h_j^{/}} = -\frac{\partial E_p}{\partial h_j} \frac{\partial h_j}{\partial h_j^{/}} \, . \tag{96}$$

Принимая $h_j = f^1(h_j^1)$, имеем

$$\delta_j^1 = -\frac{\partial E_p}{\partial h_j} \frac{\partial f^1}{\partial h_j^{/}} \, . \tag{97}$$

Фактор $\partial E_p / \partial h_j$ определяется в соответствии с локальной ошибкой выходного слоя:

$$-\frac{\partial E_p}{\partial h_j} = -\sum_{i=1}^{n2} \frac{\partial E_p}{\partial o_i^{/}} \frac{\partial o_i^{/}}{\partial h_j} = \sum_{i=1}^{n2} \left(-\frac{\partial E_p}{\partial o_i^{/}} \right) \frac{\partial}{\partial h_j} \left(\sum_{k=1}^{n2} w_{ik}^2 h_k \right) = \sum_{i=1}^{n2} \delta_i^2 w_{ij}^2 \, ; \tag{98}$$

Тогда локальную ошибку в скрытом слое можно определить, используя формулу

$$\delta_j^1 = \frac{\partial f^1}{\partial h_j^{/}} \sum_{i=1}^{n2} \delta_i^2 w_{ij}^2 \, . \tag{99}$$

Локальная ошибка внутреннего скрытого слоя определяется на базе локальных ошибок следующего слоя. Стартуя с высшего выходного слоя, мы вычисляем δ^2. Эта ошибка возвращается на нижний уровень. На рис. 21 представлена функциональная схема алгоритма обучения с обратным распространением ошибки. Главным при данном способе обучения является определение локальной ошибки δ_j^s (s = 1, 2). В выходном слое ошибка определяется как функция от желаемого и действительного результатов и сигмоидальной функции активации. Для скрытого слоя локальная ошибка определяется на базе ранее определенных локальных ошибок выходного слоя.

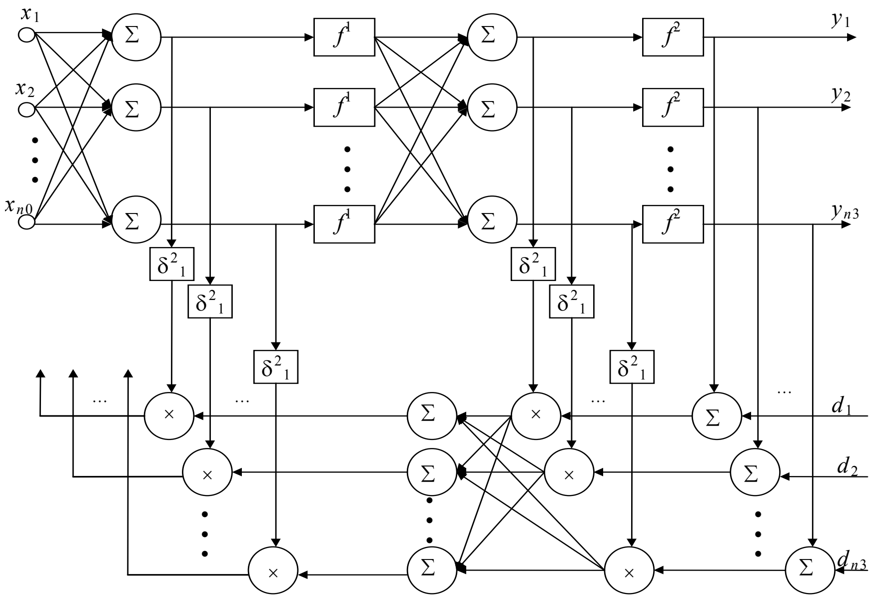


Рис. 21

Тогда алгоритм реализации обучения с обратным распространением ошибки включает следующую последовательность шагов [9]:

1 Инициализируются все синаптические веса w_{ij}^s как малое случайное число.

2 Задаются все примеры обучения в виде пар "вход-выход"; вычисляются действительные значения выходов всех нейронов, используя заданные значения w_{ij}^s и значения входов.

3 Используя значения желаемого и действительного выходов определяются локальные ошибки δ_j^s для всех уровней (92), (99).

4 Пересчитываются синаптические веса по итерационной формуле:

$$\Delta w_{ji}^s = \eta \delta_j^s x_i^s, \; s = 1, 2. \qquad (100)$$

5 Осуществляется переход ко второму примеру обучения и возврат к п. 2.

Заметим, что все тренировочные примеры обрабатываются циклически, пока ошибка обучения не станет малой. После обучения многослойный персептрон (нейронная сеть) обычно обладает свойствами объекта, для которого он обучался. Теперь можно вводить любые входные значения и получать выходные без дополнительного обучения.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 *Искусственный* интеллект: В 3 кн. Кн.2. Модели и методы: Справочник / Под ред. Э.В. Попова. – М.: Радио и связь, 1990. 303 с.

2 *Левин Р., Дранг Д., Эделсон Б.* Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бейсике. М.: Финансы и статистика, 1990. 239 с.

3 *Нильсон Н.* Принципы искусственного интеллекта. М.: Радио и связь. 1990. 376 с.

4 *Экспертные* системы в САПР: Лаб. раб. / Сост.: А.А. Кузнцов, О.П. Федосов. Тамбов: Тамб. гос. техн. ун-т. 1995. 33 с.

5 *Методы* представления знаний: Метод. указ. / Сост. И.Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. 30 с.

6 *Основы* теории нечетких множеств: Метод. указ. / Сост. И.Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. 30 с.

7 *Малышев Н.Г., Берштейн Л.С., Боженюк А.В.* Нечеткие модели для экспертных систем в САПР. М.: Энергоатомиздат, 1991. 264 с.

8 *Горбань А.Н., Дунин-Барковский В.Л., Кирдин Н., Миркес Е.М., Новоходько А.Ю., Россиев Д.А., Терехов С.А., Сенашова М.Ю.* Нейроинформатика. <http//www.bmstu.ru//facult/iu/iu4/rus/stst/book2/ann.htm>

9 *Вопросы* приближения функций: Метод. указ. / Авт.-сост.: Ю.В. Литовка, А.В. Романенко, И.Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2001. 20 с.

10 *Заде Л.* Основы нового подхода к анализу сложных систем и процессов принятия решения // Математика сегодня: (Сб. ст.). М.: Знание, 1974. 48 с.

11 *Zimmerman H.J.* Fuzzy Set Theory and its Applications. Boston etc. 1992.

12 *Кафаров Б.Б., Дорохов И.Н., Марков Е.П.* Системный анализ процессов химической технологии. Применение метода нечетких множеств. М.: Наука, 1986.

13 *Кофман Л.* Введение в теорию нечетких множеств. М.: Радио и связь, 1982.

14 *Нечеткие* множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. М.: Наука, 1986.

15 *Вощинин А.П., Сотиров Г.Р.* Оптимизация в условиях неопределенности. Изд-во: МЭИ (Россия), Техника (НРБ), 1990.