

Министерство образования и науки РФ
ФГБОУ ВО «Томский государственный университет
систем управления и радиоэлектроники»
Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

И.А. Ходашинский, К.С. Сарин, А.Е. Анфилофьев

Базы знаний

Методические указания

для выполнения практических и самостоятельных работ

для студентов специальностей

10.05.03 – «Информационная безопасность автоматизированных систем»,

10.05.04 – «Информационно-аналитические системы безопасности»

В-Спектр
Томск, 2017

УДК 004.82
ББК 32.973.26-018.2
X 69

X 69 **Ходашинский И.А., Сарин К.С., Анфилофьев А.Е.** Базы знаний: методические указания для выполнения практических и самостоятельных работ. – Томск: В-Спектр, 2017. – 68 с.
ISBN 978-5-91191-370-0

Методические указания содержат описания практических и самостоятельных работ по дисциплине «Базы знаний» для специальностей 10.05.03 – «Информационная безопасность автоматизированных систем», 10.05.04 – «Информационно-аналитические системы безопасности», задания, методические указания по выполнению, требования по представлению отчетности, вопросы для самоконтроля.

Работа выполнена в рамках выполнения базовой части государственного задания Минобрнауки России, проект 8.9628.2017/8.9.

УДК 004.82
ББК 32.973.26-018.2

ISBN 978-5-91191-370-0

© И.А. Ходашинский, К.С. Сарин,
А.Е. Анфилофьев, 2017
© ТУСУР, каф. КИБЭВС, 2017

Содержание

Введение	5
Тема 1. Логика нулевого порядка	5
1.1. Основы.....	5
1.2. Символизация естественного языка.....	6
1.3. Синтаксис и семантика	7
1.4. Вывод в логических моделях нулевого порядка.....	10
1.5. Аудиторные задания.....	13
1.6. Самостоятельная работа.....	14
1.7. Контрольные вопросы.....	14
Тема 2. Логика первого порядка	15
2.1. Синтаксис	15
2.2. Символизация естественного языка.....	16
2.3. Семантика.....	18
2.4. Нормальные формы.....	19
2.5. Выводы в логических моделях первого порядка.....	21
2.6. Аудиторные задания.....	25
2.7. Самостоятельная работа.....	26
2.8. Контрольные вопросы.....	26
Тема 3. Продукционная модель представления знаний	27
3.1. Продукционная система.....	27
3.2. Аудиторные задания.....	28
3.3. Самостоятельная работа.....	28
3.4. Контрольные вопросы.....	29
Тема 4. Семантические сети	30
4.1. TLC-модель	30
4.2. Аудиторные задания.....	30
4.3. Самостоятельная работа.....	31
4.4. Контрольные вопросы.....	31
Тема 5. Фреймовая модель представления знаний	32
5.1. Фреймовая модель представления статических знаний.....	32
5.2. Фреймовая модель представления динамических знаний	33
5.3. Аудиторные задания.....	33
5.4. Самостоятельная работа.....	34
5.5. Контрольные вопросы.....	34
Тема 6. Нечеткие множества и нечеткие отношения	35
6.1. Нечеткие множества.....	35
6.2. Нечеткие отношения	38
6.3. Аудиторные задания.....	40
6.4. Самостоятельная работа.....	41

6.5. Контрольные вопросы	42
Тема 7. Нечеткий композиционный вывод	43
7.1. Описание нечеткого композиционного вывода	43
7.2. Пример	45
7.3. Аудиторные задания	46
7.4. Самостоятельная работа	47
7.5. Контрольные вопросы	47
Тема 8. Разработка нечеткой системы	48
8.1. Нечеткая система	48
8.2. Нечеткий вывод	50
8.3. Аудиторные задания	51
8.3.1. Задание 1	51
8.3.2. Задание 2	52
8.3.3. Задание 3	62
8.4. Самостоятельная работа	64
8.5. Контрольные вопросы	65
ЛИТЕРАТУРА	66

Введение

Практические и самостоятельные работы по дисциплине преследуют следующие цели:

- обобщение, систематизация, углубление, закрепление полученных теоретических знаний и основных положений проектирования баз знаний;
- формирование умений применять полученные знания на практике.

Тема 1. Логика нулевого порядка

Цель занятия – изучение представления знаний и способа вывода на основе логических моделей нулевого порядка.

1.1. Основы

В логике высказываний предполагается, что мир может быть описан элементарными предложениями или высказываниями и логическими связями между ними. Кроме этого допущения приняты еще два следующих:

□ простому предложению или высказыванию можно приписать истинностное значение;

□ сложные предложения, образуются путем видоизменения некоторого предложения с помощью слова «НЕ» (\sim) или путем связывания простых предложений с помощью слов «И» (\wedge), «ИЛИ» (\vee), «ЕСЛИ, ТО» (\rightarrow), «ТОГДА И ТОЛЬКО ТОГДА, КОГДА» (\leftrightarrow). Эти пять слов называют сентенциональными, пропозициональными или логическими связками, каждая из них имеет свое название « \sim » – отрицание, « \wedge » – конъюнкция, « \vee » – дизъюнкция, « \rightarrow » – импликация, « \leftrightarrow » – эквиваленция.

Таким образом, *высказывание* – это грамматически правильное, неразлагаемое и неанализируемое повествовательное предложение, которое может быть истинными или ложными, но не тем и другим одновременно. Высказывание, состоящее из одного предложения, называют простым или элементарным.

Существует два подхода к установлению истинности высказываний: эмпирический и логический. Первый устанавливает истинность высказываний путем выполнения некоторых действий, таких как наблюдение, измерение, эксперимент. Во втором подходе «истинна» или «ложь», которая приписывается высказыванию, и есть *истинностное значение* высказывания. Для краткости «истинна» обозначается как И, а «ложь» – Л. Высказывания обозначаются заглавными буквами или цепочкой таких букв. Например, Козьма Прутков считает, что

Р: «Военные люди защищают отечество».

Q: «Ветер есть дыхание природы».

R: «Новые сапоги всегда жмут».

Символы *P*, *Q*, *R* и др., которые используются для обозначения элементарных высказываний, называются *атомарными формулами* или *атомами*.

Примеры сложных высказываний от Козьмы Прутковка: «чиновник умирает, а его ордена остаются на лице земли», «хочешь быть красивым, поступи в гусары». Примем следующие обозначения:

M: чиновник умирает;

L: ордена чиновника остаются на лице земли;

B: хотеть быть красивым;

G: поступать в гусары.

Тогда два последних предложения могут быть записаны в виде формул как $M \wedge L, B \rightarrow G$, соответственно.

1.2. Символизация естественного языка

Существуют проблемы отображения рассуждений на основе здравого смысла в предложения формальной логики. Например, логическая связка \rightarrow оперирует только значениями истинности и полностью игнорирует корреляционную специфику естественноречевой конструкции «если ..., то ...».

Рассмотрим некоторые аспекты символизации естественного языка и применение логических связок. Операция конъюнкции в логике высказываний и союз «и» в повседневной речи употребляются в одном и том же смысле. Однако в обыденной речи не принято соединять союзом «и» два далеких по смыслу предложения (*ироничное*: «в огороде бузина, а в Киеве дядька»). В то время как в логике высказываний операция конъюнкции соединяет два любых высказывания. Операции конъюнкции соответствуют следующие выражения:

- A* и *B*;
- не только *A*, но и *B*;
- B*, хотя и *A*;
- A*, а также *B*;
- как *A*, так и *B*;
- A* вместе с *B*.

В повседневной речи союз «или» употребляется в двух различных смыслах: исключающем и не исключающем, а операция дизъюнкции всегда употребляется в не исключающем смысле.

Употребление слов «если ..., то ...» в повседневной речи существенно отличается от применения в логике высказываний. В предложении «если *A*, то *B*» обыденной речи подразумевается, что *B* логически следует из *A*, в то время как в логике высказываний, не рассматривающей смысла предложений, этого не требуется. Кроме того ложность предложения *A* в повседнев-

ной речи влечет либо ложность B , либо потерю смысла всего предложения «если A , то B ». Таким образом, трансляция естественно языкового предложения в предложение логики высказываний при кажущейся простоте таковой не является. Здесь требуется понять смысл предложения, а затем уже конструировать формулы логики высказываний. Операции импликации ($A \rightarrow B$) соответствуют следующие выражения естественного языка:

- B , если A ;
- A влечет B ;
- A является причиной B ;
- B является следствием A ;
- в случае A имеет место B ;
- коль скоро A , то B ;
- B , так как A ;
- B , потому что A .

Операции эквиваленции ($A \leftrightarrow B$) соответствуют следующие выражения естественного языка:

- A , если и только если B ;
- если A , то B и обратно;
- A , если B , и B , если A ;
- A эквивалентно B ;
- A равносильно B .

1.3. Синтаксис и семантика

В логике высказываний используются следующие символы:

- 1) символы высказываний $P, Q, R, S \dots$;
- 2) логические связки $\leftrightarrow, \rightarrow, \vee, \wedge, \sim$;
- 3) символы значений истинности И, Л.

Правильно построенные формулы или просто *формулы* в логике высказываний определяются рекурсивно следующим образом:

- Атом есть формула.
- Если G – формула, то $(\sim G)$ – формула.
- Если G и H – формулы, то $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$ и $(G \leftrightarrow H)$ – формулы.

- Других формул нет.

Логическим связкам приписан следующий убывающий ранг:

$$\leftrightarrow, \rightarrow, \vee, \wedge, \sim,$$

таким образом, связка с большим рангом имеет большую область действия. Формула

$$P \leftrightarrow \sim Q \vee R \rightarrow S \text{ означает } (P \leftrightarrow (((\sim Q) \vee R) \rightarrow S)).$$

Присвоение значения истинности логическим предложениям называется интерпретацией. Другими словами, интерпретация – это утверждение отно-

сительно истинности некоторого логического предложения в некотором возможном мире, формально интерпретация – это отображение логических символов на множество $\{И, Л\}$.

Если G, H – две формулы, тогда истинность формул $(\sim G)$, $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$, $(G \leftrightarrow H)$ определяется по истинностным значениям атомов, входящих в эту формулу и следующей истинностной таблице.

Истинностная таблица

G	H	$\sim G$	$G \wedge H$	$G \vee H$	$G \rightarrow H$	$G \leftrightarrow H$
И	И	Л	И	И	И	И
И	Л	Л	Л	И	Л	Л
Л	И	И	Л	И	И	Л
Л	Л	И	Л	Л	И	И

Интерпретацией формулы является такое приписывание истинностных значений атомам, входящим в формулу, при котором каждому из них приписано либо И, либо Л, но не оба одновременно. Таким образом, если истинностные значения атомов известны, то истинностное значение формулы может быть определено индуктивным способом в соответствии с приведенной выше истинностной таблицей. Если формула содержит n различных атомов, то эта формула имеет 2^n интерпретаций. Ниже приведена истинностная таблица для формул $P \rightarrow (Q \rightarrow P)$ и $(P \rightarrow Q) \rightarrow P$.

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$P \rightarrow (Q \rightarrow P)$	$(P \rightarrow Q) \rightarrow P$
И	И	И	И	И	И
И	Л	Л	И	И	И
Л	И	И	Л	И	Л
Л	Л	И	И	И	Л

Интерпретацию формулы, содержащей атомы A_1, A_2, \dots, A_n , удобно представлять в виде

$$I = \{m_1, m_2, \dots, m_n\},$$

где m_j есть A_j или $\sim A_j$. Если m_j есть A_j , то атому A_j присвоено значение И, в противном случае Л. Например, в интерпретации $\{\sim P, Q\}$ формула $(P \rightarrow Q) \rightarrow P$ «ложна», а в интерпретации $\{P, \sim Q\}$ эта же формула «истинна».

Формула истинная при всех возможных интерпретациях называется *общезначимой формулой* или *тавтологией*. Формула ложная при всех возможных интерпретациях называется *противоречивой* (или *невыполнимой*). Общезначимую и противоречивую формулу будем обозначать \blacksquare , \square , соответственно. Формула, которая не является общезначимой или противоречивой, называется *необщезначимой, непротиворечивой или выполнимой*.

Общезначимость и противоречивость формулы может быть определена с использованием таблицы истинности, но в силу экспоненциального роста размерности числа интерпретаций с ростом числа входящих в формулу атомов, такой метод не всегда является приемлемым. Такое положение привело к необходимости разработки правил преобразования формул. Преобразования выполняются путем замены в преобразуемой формуле некоторой ее части на подформулу, эквивалентную заменяемой. Две формулы эквивалентны, если их истинностные значения совпадают при всех интерпретациях. Например, формула $(P \rightarrow Q) \rightarrow P$ эквивалентна формуле P . Для ведения преобразований необходимо иметь минимальный запас эквивалентных формул, ниже приведены одиннадцать законов преобразования, здесь F , G и H являются формулами, символ « \Leftrightarrow » это знак эквивалентности.

1. $F \leftrightarrow G = (F \rightarrow G) \wedge (G \rightarrow F)$;
2. $F \rightarrow G = \sim F \vee G$;
3. $F \vee G = G \vee F$; $F \wedge G = G \wedge F$;
4. $(F \vee G) \vee H = F \vee (G \vee H)$; $(F \wedge G) \wedge H = F \wedge (G \wedge H)$;
5. $F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H)$; $F \wedge (G \vee H) = (F \wedge G) \vee (F \wedge H)$;
6. $F \vee \square = F$; $F \wedge \blacksquare = F$;
7. $F \vee F = F$; $F \wedge F = F$;
8. $F \vee \blacksquare = \blacksquare$; $F \wedge \square = \square$;
9. $F \vee \sim F = \blacksquare$; $F \wedge \sim F = \square$;
10. $\sim(\sim F) = F$;
11. $\sim(F \vee G) = \sim F \wedge \sim G$; $\sim(F \wedge G) = \sim F \vee \sim G$.

Правила преобразования под номером три называются коммутативными законами; законы под номером четыре – ассоциативными законами; законы под номером пять – дистрибутивными законами; семь – закон идемпотентности; девять – законы дополнения; десять – закон двойного отрицания; одиннадцать – законами де Моргана.

В логике высказываний определены две нормальные формы: дизъюнктивная и конъюнктивная. Формула находится в *дизъюнктивной нормальной форме*, если она имеет следующий вид: $F_1 \vee F_2 \vee \dots \vee F_n$, где каждая подформула F_i – это конъюнкция атомов или отрицания атомов. Формула находится в *конъюнктивной нормальной форме*, если она имеет следующий вид: $F_1 \wedge F_2 \wedge \dots \wedge F_n$, где каждая подформула F_i – это дизъюнкция атомов или отрицания атомов. *Литера* – это атом или отрицание атома. *Дизъюнкт* – это дизъюнкция литер. *Единичный дизъюнкт* – это дизъюнкт, состоящий из одной литеры. Дизъюнкт, не содержащий никаких литер, называется *пустым дизъюнктом* и обозначается \emptyset . Формула, находящаяся в конъюнктивной нормальной форме, может быть представлена как множество входящих в форму дизъюнктов. Например, формула $(P \vee Q) \wedge (P \vee R \vee \sim Q) \wedge$

$\wedge (\sim Q \vee P) \wedge \sim P$ может быть представлена как множество $\{P \vee Q, P \vee R \vee \sim Q, \sim Q \vee P, \sim P\}$.

1.4. Вывод в логических моделях нулевого порядка

Задача логики – описать мир в виде логических формул и дать теорию вывода. Практически теория выводов сводится к получению критериев и алгоритмов для решения вопроса о том, можно ли некоторую цепочку рассуждений, основываясь на ее форме, считать правильной. Цепочка рассуждений представляет собой конечную последовательность высказываний, приводимых в обоснование утверждения того, что последнее высказывание в этой последовательности (заключение) может быть выведено из некоторых начальных высказываний (посылок). Это приводит к понятию «логического следствия».

ОПРЕДЕЛЕНИЕ. Пусть даны формулы F_1, F_2, \dots, F_n и формула G . Формула G есть *логическое следствие* формул F_1, F_2, \dots, F_n , если для всякой интерпретации, в которой формула $F_1 \wedge F_2 \wedge \dots \wedge F_n$ истинна, G также истинна. Формулы F_1, F_2, \dots, F_n называются *посылками*, G – *заключением*.

ТЕОРЕМА 1. Пусть даны формулы F_1, F_2, \dots, F_n и формула G . Тогда G есть логическое следствие формул F_1, F_2, \dots, F_n , если формула $((F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow G)$ общезначима.

ТЕОРЕМА 2. Пусть даны формулы F_1, F_2, \dots, F_n и формула G . Тогда G есть логическое следствие формул F_1, F_2, \dots, F_n , если формула $(F_1 \wedge F_2 \wedge \dots \wedge F_n \wedge \sim G)$ противоречива.

Таким образом, вопрос о том, какие высказывания представляют собой логические следствия других высказываний, сводится к вопросу о том, какие высказывания общезначимы или противоречивы. Это, в свою очередь, дает возможность превратить ОПРЕДЕЛЕНИЕ и ТЕОРЕМЫ 1 и 2 в рабочий аппарат для логического вывода. Ниже приведены восемь способов логического вывода в логике высказываний. В качестве примера рассматриваются следующие формулы: $F_1: P$; $F_2: R$; $F_3: Q \wedge R \rightarrow \sim R$; $G: \sim Q$.

Способ 1: *вычисление истинностного значения*. Здесь вывод основан на определении логического следствия и на истинностных таблицах.

P	R	Q	$Q \wedge R$	$\sim R$	$Q \wedge R \rightarrow \sim R$	$P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R)$	$\sim Q$
И	И	И	И	Л	Л	Л	Л
И	И	Л	Л	Л	И	И	И
И	Л	И	Л	И	И	Л	Л
И	Л	Л	Л	И	И	Л	И
Л	И	И	И	Л	Л	Л	Л
Л	И	Л	Л	Л	И	Л	И
Л	Л	И	Л	И	И	Л	Л
Л	Л	Л	Л	И	И	Л	И

Из таблицы видно, что есть только одна интерпретация, где формула $P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R)$ является истинной, в этой же интерпретации формула $\sim Q$ является истинной, следовательно, формула $\sim Q$ является логическим следствием формул $P, R, Q \wedge R \rightarrow \sim R$.

Способ *таблиц истинности 2*. Здесь в качестве аппарата для логического вывода может быть использована ТЕОРЕМА 1 и метод истинностных таблиц.

P	R	Q	$Q \wedge R$	$\sim R$	$Q \wedge R \rightarrow \sim R$	$P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R)$	$\sim Q$	$F_1 \wedge F_2 \wedge F_3 \rightarrow G$
И	И	И	И	Л	Л	Л	Л	И
И	И	Л	Л	Л	И	И	И	И
И	Л	И	Л	И	И	Л	Л	И
И	Л	Л	Л	И	И	Л	И	И
Л	И	И	И	Л	Л	Л	Л	И
Л	И	Л	Л	Л	И	Л	И	И
Л	Л	И	Л	И	И	Л	Л	И
Л	Л	Л	Л	И	И	Л	И	И

Из таблицы видно, что есть формула $P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R) \rightarrow \sim Q$ является общезначимой, значит формула $\sim Q$ является логическим следствием формул $P, R, Q \wedge R \rightarrow \sim R$.

Способ *таблиц истинности 3*. В качестве аппарата для логического вывода может быть использована ТЕОРЕМА 2 и метод истинностных таблиц.

P	R	Q	$Q \wedge R$	$\sim R$	$Q \wedge R \rightarrow \sim R$	$P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R)$	$F_1 \wedge F_2 \wedge F_3 \wedge \sim G$
И	И	И	И	Л	Л	Л	Л
И	И	Л	Л	Л	И	И	Л
И	Л	И	Л	И	И	Л	Л
И	Л	Л	Л	И	И	Л	Л
Л	И	И	И	Л	Л	Л	Л
Л	И	Л	Л	Л	И	Л	Л
Л	Л	И	Л	И	И	Л	Л
Л	Л	Л	Л	И	И	Л	Л

Из таблицы видно, что есть формула $P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R) \wedge \sim Q$ является противоречивой, значит формула $\sim Q$ является логическим следствием формул $P, R, Q \wedge R \rightarrow \sim R$.

Способ 4, *алгебраический*, здесь в качестве аппарата для логического вывода используется ТЕОРЕМА 1, а для доказательства общезначимости формулы – десять законов эквивалентных преобразований.


$$\begin{aligned}
 P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R) \rightarrow \sim Q &= \sim(P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R)) \vee \sim Q = \\
 &= \sim(P \wedge R \wedge (\sim(Q \wedge R) \vee \sim R)) \vee \sim Q = \sim P \vee \sim R \vee (Q \wedge R \wedge R) \vee \sim Q =
 \end{aligned}$$

$$\begin{aligned}
 &= \sim P \vee \sim R \vee (Q \wedge R) \vee \sim Q = \sim P \vee \sim R \vee (Q \wedge \sim Q) \vee (R \vee \sim Q) = \\
 &= \sim P \vee \sim R \vee R \vee \sim Q = \sim P \vee \blacksquare \vee \sim Q = \blacksquare.
 \end{aligned}$$


Способ 5, также *алгебраический*, только здесь использована ТЕОРЕМА 2, а для доказательства противоречивости формулы – десять законов эквивалентных преобразований.

$$\begin{aligned}
 P \wedge R \wedge ((Q \wedge R) \rightarrow \sim R) \wedge Q &= P \wedge R \wedge (\sim(Q \wedge R) \vee \sim R) \wedge Q = \\
 &= P \wedge R \wedge (\sim Q \vee \sim R \vee \sim R) \wedge Q = P \wedge R \wedge (\sim Q \vee \sim R) \wedge Q = \\
 &= P \wedge R \wedge ((\sim Q \wedge Q) \vee (\sim R \wedge \sim Q)) = P \wedge R \wedge \sim R \wedge \sim Q = \\
 &= P \wedge \square \wedge \sim Q = \square.
 \end{aligned}$$

Способ 6, *принцип резолюций*. Данный метод основан на использовании конъюнктивной нормальной формы, представленной как множество дизъюнктов S . Обобщение правила связано с тем, что оно применяется к любой паре дизъюнктов, не обязательно единичных. Правило это называется правилом резолюций и формулируется следующим образом:

 **Если в дизъюнкте C_1 существует литера L , а в дизъюнкте C_2 существует литера $\sim L$, то, вычеркнув литеры L и $\sim L$ из C_1 и C_2 , соответственно, построим дизъюнкцию оставшихся дизъюнктов, которая называется резольвентой дизъюнктов C_1 и C_2 .**

Рассмотрим, например, дизъюнкты $C_1: P \vee Q$, $C_2: \sim Q \vee \sim R \vee U$, резольвентой дизъюнктов C_1 и C_2 будет следующий дизъюнкт: $P \vee \sim R \vee U$.

 **Важное свойство резольвенты. Любая резольвента дизъюнктов C_1 и C_2 является логическим следствием дизъюнктов C_1 и C_2 . Для невыполнимого множества дизъюнктов применением правила резолюций можно получить пустой дизъюнкт \emptyset .**

Метод резолюций основан на проверке того, содержит ли исходное множество дизъюнктов пустой дизъюнкт, если множество содержит пустой дизъюнкт, то это множество невыполнимо, в противном случае проверяется, можно ли получить пустой дизъюнкт из исходного множества. Такой процесс проверки называется выводом.

Выводом из конечного множества дизъюнктов S называется конечная последовательность C_1, C_2, \dots, C_n дизъюнктов, всякий элемент C_i которой или принадлежит множеству S , или является резольвентой дизъюнктов, предшествующих данному элементу C_i . Вывод пустого дизъюнкта \emptyset из S называется опровержением или доказательством невыполнимости S .

Используя метод резолюций, проведем вывод и покажем, что формула $(P \vee Q) \wedge (P \vee R) \wedge (\sim Q \vee \sim R) \wedge \sim P \wedge U$ невыполнима. Запишем множество дизъюнктов $S = \{P \vee Q, P \vee R, \sim Q \vee \sim R, \sim P, U\}$ следующим образом:

1. $P \vee Q$.

2. $P \vee R$.
3. $\sim Q \vee \sim R$.
4. $\sim P$.
5. U .

Из п. 2 и 3 получим резольвенту

6. $P \vee \sim Q$.

Из п. 1 и 6 получим резольвенту

7. P .

Из п. 4 и 7 получим резольвенту

8. \emptyset .

Так как \emptyset есть логическое следствие множество дизъюнктов S , то S невыполнимо.

1.5. Аудиторные задания

1. Если Степан не знал о необходимости декларировать доход, то он плохой законодатель. Если он знал и не декларировал, то он – мошенник. Если Степан является плохим законодателем или мошенником, то ему нет места в Думе. Степан не декларировал свой доход. Следовательно, ему нет места в Думе. Доказать всеми возможными способами.

2. Если Мери бросила Джона, то она уехала или в Россию, или в Израиль. Если Мери уехала в Россию, то ее арестовал КГБ. Если Мери уехала в Израиль, то ее арестовал Моссад. Мери не арестовал ни Моссад, ни КГБ. Значит, Мери не бросила Джона. Доказать всеми возможными способами.

3. Если я пойду завтра на первое занятие, то должен буду встать рано, а если я пойду вечером на танцы, то лягу спать поздно. Если я лягу спать поздно, а встану рано, то я буду вынужден довольствоваться пятью часами сна. Я не могу довольствоваться пятью часами сна. Следовательно, я или не пойду завтра на первое занятие, или не пойду вечером на танцы. Доказать всеми возможными способами.

4. Если я поеду автобусом, а автобус опоздает, то я пропущу назначенное свидание. Если я пропущу назначенное свидание и буду огорчен, то мне не следует ехать домой. Если я не получу эту работу, то я буду огорчен и мне следует поехать домой. Следовательно, если я поеду домой автобусом, и автобус опоздает, то я получу эту работу. Доказать всеми возможными способами.

1.6. Самостоятельная работа

1. Халиф Омар, сжегший Александрийскую библиотеку, рассуждал так: если ваши книги согласны с Кораном, то они излишни; если они не согласны с Кораном, то они вредны; но вредные или излишние книги следует уничтожить; значит, ваши книги следует уничтожить. Доказать правильность рассуждений халифа.

2. Или Маша и Ваня одного возраста, или Маша старше Вани. Если Маша и Ваня одного возраста, то Наташа и Ваня не одного возраста. Если Маша старше Вани, то Ваня старше Пети. Следовательно, или Наташа и Ваня не одного возраста, или Ваня старше Пети. Доказать всеми возможными способами.

3. Если завтра будет холодно, я надену шубу, если рукав будет починен. Завтра будет холодно, а рукав не будет починен. Следовательно, я не надену шубу. Доказать всеми возможными способами.

1.7. Контрольные вопросы

1. Какие допущения приняты при описании мира с помощью логических моделей?

2. Что такое правильно построенная формула в логике высказываний?

3. Что есть интерпретация в логике высказываний?

4. Что есть логическое следствие?

5. Охарактеризуйте способы логического вывода в логике высказываний.

Тема 2. Логика первого порядка

Цель занятия – изучение представления знаний и способа вывода на основе логических моделей первого порядка.

2.1. Синтаксис

Класс задач, решаемых с использованием логики высказываний, очень ограничен. Например, из посылок следующего классического примера:

«Все люди смертны»;

«Сократ – человек»;

интуитивно следует заключение «Сократ смертен».

Однако в рамках логики высказываний решить эту задачу не удастся. Объясняется это тем, что утверждение в логике высказываний – это неделимый объект (атом), а приведенный пример требует анализа внутренней структуры предложения. Логика предикатов первого порядка позволяет выразить большее разнообразие утверждений благодаря тому, что в неё добавлены *термы*, *предикаты* и *кванторы*.

Если проанализировать приведенные выше три утверждения, то можно обнаружить, что рассуждения здесь ведутся на некоторой предметной области (множестве людей). «Сократ» является объектом этой предметной области. Кроме того, в первом утверждении есть неявное указание на принадлежность к этой предметной области («если некто принадлежит к множеству людей, значит он смертен»). Такое неконкретизированное указание объекта предметной области соответствует понятию *«переменная»*, а явное указание «Сократ» – понятию *«константа»*. Принадлежность объекта к предметной области можно задать в виде *«логической функции»* или предиката, например, предикат ЧЕЛОВЕК(x) указывает на то, что если x является человеком, то высказывание ЧЕЛОВЕК(x) является истинным, и, соответственно, предикат СМЕРТЕН(x) указывает на то, что x смертен. Здесь формы записи ЧЕЛОВЕК и СМЕРТЕН называются предикатными символами. Выражение «все люди» служит примером квантификации, такая запись символически может быть представлена, как \forall и называется квантором всеобщности (общности). Запись $(\forall x)$ читается как «для всех x », «для всякого x », «для каждого x ». Тогда приведенные выше три высказывания можно формально записать следующим образом:

$$(\forall x)(\text{ЧЕЛОВЕК}(x) \rightarrow \text{СМЕРТЕН}(x)).$$

$$\text{ЧЕЛОВЕК}(\text{Сократ}).$$

$$\text{СМЕРТЕН}(\text{Сократ}).$$

Кроме квантора всеобщности, в логике предикатов используется квантор существования, символически записываемый в виде $(\exists x)$, что означает «существует x », «для некоторых x », «по крайней мере, для одного x ».

Константы и переменные образуют более общее понятие *терм*, который определяется следующим образом:

- константа – это терм;
- переменная – это терм;
- если f – n -местный функциональный символ и t_1, t_2, \dots, t_n – термы, то $f(t_1, t_2, \dots, t_n)$ – терм;
- других термов нет.

Синонимом для сложного терма вида $f(t_1, t_2, \dots, t_n)$ будет «функция». Функция есть отображение списка констант в константу. Предикат – это тоже отображение списка констант, но не в константу, а в элемент множества $\{И, Л\}$.

Основными символьными конструкциями логики предикатов являются константы, переменные, термы, предикатные символы, логические связки, кванторы.

После определения терма, можно дать понятие атома и формулы в логике предикатов. Если P – n -местный предикатный символ и t_1, t_2, \dots, t_n – термы, то $P(t_1, t_2, \dots, t_n)$ – атом. Понятие *формулы* рекурсивно определяется следующим образом:

- атом – это формула;
- если G и H – формулы, то $(\sim G)$, $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$ и $(G \leftrightarrow H)$ – формулы;
- если G – формула и x – переменная, то $(\forall x)G$, $(\exists x)G$ – формулы;
- других формул нет.

Здесь, как и в логике высказываний, круглые скобки могут быть опущены в соответствии с принятыми соглашениями о ранге логических связок, считая ранг кванторов наименьшим.

Примем следующие соглашения по использованию мнемонических имен, которые будут действовать по умолчанию:

- константы обозначаются строчными буквами a, b, c, d, e возможно с индексами;
- переменные – это обычно строчные буквы x, y, z, u, v, w , возможно с индексами;
- функциональные символы представляются буквами f, g, h или словами из строчных букв;
- предикатные символы обозначаются прописными буквами P, Q, R, S или словами из прописных букв.

2.2. Символизация естественного языка

Для перевода предложения русского языка на язык исчисления предикатов первого порядка необходимо обратить внимание на отношения и объекты, присутствующие в предложении, и описать их в виде предикатов и тер-

мов. Часто предикат задается глаголом предложения, а терм – объектом. Существует неоднозначность при переводе предложения русского языка на язык исчисления предикатов.

Рассмотрим несколько примеров *формализации предложений естественного языка*. Пусть $R(x)$ обозначает « x есть рациональное число», а $Q(x)$ – « x есть действительное число», тогда предложение «каждое рациональное число есть число действительное» можно переформулировать в такое: «для любого x , если x есть рациональное число, то x есть действительное число» и символически представить в следующем виде:

$$(\forall x)(R(x) \rightarrow Q(x)).$$

Смысл этого предложения сводится к тому, что множество рациональных чисел является подмножеством множества действительных чисел, что записывается как $R \subseteq Q$. Такая запись является типичной для высказываний «всякое нечто есть то-то».

Теперь рассмотрим утверждение «некоторые действительные числа являются рациональными». Формула, правильно отражающая это предложение, будет иметь вид:

$$(\exists x)(Q(x) \wedge R(x)),$$

что означает, что пересечение двух множеств действительных и рациональных чисел не является пустым $R \cap Q \neq \emptyset$. Типичная ошибка при формализации такого типа предложений заключается в том, что исходя из правильности формализации предложений типа «всякое нечто есть то-то» в виде $(\forall x)(R(x) \rightarrow Q(x))$, делается ошибочное предположение в правильности формализации предложения типа «некоторое нечто есть то-то» формулой $(\exists x)(Q(x) \rightarrow R(x))$.


И еще одно предложение «не каждое действительное число есть число рациональное», что может быть прочтено как «не верно, что каждое действительное число является рациональным», формальная запись которого выглядит так

$$\sim((\forall x)(Q(x) \rightarrow R(x))).$$

Переформулировав это предложение еще и так: «существуют числа, которые являются действительными, но не являются рациональными», можно это же предложение представить и в виде такой формулы

$$(\exists x)(Q(x) \wedge \sim R(x)).$$

Таким образом, имеется две формулы отражающих одно и то же предложение естественного языка, однако, в дальнейшем станет ясно, что эти формулы являются эквивалентными.

 **Не существует простых синтаксических правил перевода предложений естественного языка в язык формул логики предикатов. Для перевода необходимо установить смысл предложения, а потом транслировать этот смысл в язык логики предикатов.**

2.3. Семантика

Формулы логики предикатов могут быть интерпретированы, т.е. им может быть приписано истинностное значение. Однако наличие термов требует задать и им значения из предметной области. Таким образом, *интерпретация* формулы F логики предикатов состоит из непустой предметной области D и задания значений из данной предметной области всем константам, переменным и функциям, встречающимся в формуле F ; каждому предикату, определенному на D , приписывается либо И, либо Л. Указанные приписывания определяют семантику языка исчисления предикатов.

Правила интерпретации для неквантифицированных и квантифицированных формул будут следующие:

□ если истинностные значения формул G и H заданы, то истинностные значения формул $(\sim G)$, $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$ и $(G \leftrightarrow H)$ определяются по таблицам истинности.

□ формула $(\forall x)G$ получит значение И, если значение И получит формула G для каждого x из области D , в противном случае формула G получит значение Л;

□ формула $(\exists x)G$ получит значение И, если формула G получит значение И хотя бы для одного x из области D , в противном случае формула G получит значение Л.

Рассмотрим следующий пример: пусть дана формула

$$(\forall x)(\exists y)(P(x, a) \rightarrow Q(f(y))),$$

в следующей интерпретации: область $D = \{1, 2\}$, константа $a = 1$, функция f принимает следующие значения:

$f(1)$	$f(2)$
2	1

оценка для предикатов следующая:

$P(1, 1)$	$P(1, 2)$	$P(2, 1)$	$P(2, 2)$	$Q(1)$	$Q(2)$
И	Л	И	Л	И	Л

Пусть $x = 1$, $y = 1$, определим истинностное значение формулы $P(x, a) \rightarrow Q(f(y))$ при этих значениях x и y

$$P(x, a) \rightarrow Q(f(y)) = P(1, a) \rightarrow Q(f(1)) = P(1, 1) \rightarrow Q(2) = \text{И} \rightarrow \text{Л} = \text{Л};$$

положим $y = 2$

$$P(x, a) \rightarrow Q(f(y)) = P(1, a) \rightarrow Q(f(2)) = P(1, 1) \rightarrow Q(1) = \text{И} \rightarrow \text{И} = \text{И};$$

т.е. для $x = 1$ существует такой y (а именно $y = 2$), что формула $P(1, a) \rightarrow Q(f(y))$ принимает значение И.

Пусть теперь $x = 2$, $y = 1$, определим истинностное значение формулы $P(x, a) \rightarrow Q(f(y))$ при данных значениях x и y :

$$P(x, a) \rightarrow Q(f(y)) = P(2, a) \rightarrow Q(f(1)) = P(2, 1) \rightarrow Q(2) = \text{И} \rightarrow \text{Л} = \text{Л};$$

положим $y = 2$


$$P(x, a) \rightarrow Q(f(y)) = P(2, a) \rightarrow Q(f(2)) = P(2, 1) \rightarrow Q(1) = \text{И} \rightarrow \text{И} = \text{И}.$$

Таким образом, и для $x = 2$ существует такой y (а именно $y = 2$), что формула $P(2, a) \rightarrow Q(f(y))$ принимает значение И. Так как для всех x из области D существует такое значение y , что формула $P(x, a) \rightarrow Q(f(y))$ истинна, то формула $(\forall x)(\exists y)(P(x, a) \rightarrow Q(f(y)))$ истинна в указанной интерпретации.

Формулы логики предикатов, также как и формулы логики высказываний, делятся на три класса: *общезначимые* – истинные во всех своих интерпретациях, *противоречивые (невыполнимые)* – ложные во всех своих интерпретациях, и *непротиворечивые (выполнимые)* – истинные только в отдельных своих интерпретациях.

2.4. Нормальные формы

Понятие логического следствия формул, определенное для формул логики высказываний, справедливо и в логике предикатов; справедливы здесь и отношения между общезначимостью, противоречивостью и логическим следствием, заданные в виде ТЕОРЕМ 1 и 2. В общем случае язык логики высказываний является подмножеством языка логики предикатов первого порядка. Формула логики высказываний может рассматриваться как формула логики предикатов без кванторов, функций и переменных.

 **Формула логики предикатов имеет в общем случае бесконечное число областей интерпретации или предметных областей и, как следствие, бесконечное число интерпретаций, а значит невозможно доказать противоречивость или общезначимость формулы перебором всех ее интерпретаций.**

Цель дальнейшего изложения – определить процедуру проверки противоречивости формул логики предикатов. Для решения этой задачи введены еще две нормальные формы: *предваренная* и *сколемовская стандартная*, помимо двух ранее определенных: конъюнктивной и дизъюнктивной. Введение нормальных форм позволит упростить алгоритмы поиска противоречивости формул логики предикатов.

Предваренной (префиксной) нормальной формой называется формула, состоящая из префикса и матрицы, здесь префикс – это конечная последовательность кванторных комплексов, а матрица – это формула, не содержащая кванторных комплексов, т.е. формула имеет следующий вид:

$$(Q_1x_1)(Q_2x_2) \dots (Q_nx_n) M,$$

где Q_i есть \forall или \exists для $i = 1, 2, \dots, n$.

Рассмотрим метод преобразования формулы логики предикатов в предваренную нормальную форму. Десять правил преобразования были определены ранее при рассмотрении логики высказываний, эти правила дей-

ствуют и в логике предикатов. Ниже рассмотрим дополнительные пары эквивалентных формул, содержащих кванторы. Пусть F и H – это формулы, содержащие переменную x , G есть формула, которая не содержит переменной x , а Q – это \forall или \exists .

$$12. (Qx)F(x) \wedge G = (Qx)(F(x) \wedge G); \quad (Qx)F(x) \vee G = (Qx)(F(x) \vee G);$$

$$13. \sim((\forall x) F(x)) = (\exists x)(\sim F(x)); \quad \sim((\exists x)F(x)) = (\forall x)(\sim F(x)).$$

$$14. (\forall x)F(x) \wedge (\forall x)H(x) = (\forall x)(F(x) \wedge H(x));$$

$$(\exists x)F(x) \vee (\exists x)H(x) = (\exists x)(F(x) \vee H(x)).$$

Переменная в формулах логики предикатов может быть переименована. На использовании этого приема основаны обобщающие законы тринадцать следующие правила преобразования:

$$15. (Q_1x)F(x) \vee (Q_2x)H(x) = (Q_1x)F(x) \vee (Q_2z)H(z) = (Q_1x)(Q_2z) (F(x) \vee H(z));$$

$$(Q_3x)F(x) \wedge (Q_4x)H(x) = (Q_3x)F(x) \wedge (Q_4z)H(z) = (Q_3x)(Q_4z) (F(x) \wedge H(z)).$$

Любая формула логики предикатов допускает эквивалентную предваренную нормальную форму.

Введение кванторов в формулы логики предикатов позволило увеличить выразительную мощь, одновременно усложнив язык данного формализма. Упрощение языка возможно за счет приемлемого уменьшения выразительной мощи. Идея упрощения состоит в том, чтобы, сохранив противоречивость формулы, в ней исключаются кванторы существования путем использования сколемовских функций. Тогда произвольной формуле F логики предикатов сопоставляется некоторая формула S_F , причем, если формула F невыполнима, то и формула S_F также невыполнима. Такая связь между формулами F и S_F строго слабее, чем логическая эквивалентность, однако именно такое сопоставление позволяет провести доказательство невыполнимости формулы более эффективно.

Сколемовской стандартной формой называется формула, находящаяся в предваренной форме, у которой матрица приведена к конъюнктивной нормальной форме.

Константы и функции, используемые для замены переменных квантора существования, называются *сколемовскими функциями*.

Рассмотрим пример, пусть необходимо получить сколемовскую стандартную форму формулы

$$\begin{aligned} & (\forall x)(\exists y)P(x, y) \rightarrow (\forall z)(\exists v)((\forall w)Q(z, w) \rightarrow (\exists w)R(z, v, w)) = \\ & = \sim((\forall x)(\exists y)P(x, y)) \vee (\forall z)(\exists v)((\forall w)Q(z, w) \rightarrow (\exists w)R(z, v, w)) = \\ & = \sim((\forall x)(\exists y)P(x, y)) \vee (\forall z)(\exists v)(\sim((\forall w)Q(z, w)) \vee (\exists w)R(z, v, w)) = \\ & = ((\exists x)(\forall y)\sim P(x, y)) \vee (\forall z)(\exists v)(\sim((\forall w)Q(z, w)) \vee (\exists w)R(z, v, w)) = \\ & = (\exists x)(\forall y)(\sim P(x, y)) \vee (\forall z)(\exists v)((\exists w)(\sim Q(z, w)) \vee (\exists w)R(z, v, w)) = \\ & = (\exists x)(\forall y)(\sim P(x, y)) \vee (\forall z)(\exists v)(\exists w)(\sim Q(z, w) \vee R(z, v, w)) = \end{aligned}$$

$$= (\exists x)(\forall y)(\forall z)(\exists v)(\exists w) (\sim P(x, y) \vee \sim Q(z, w) \vee R(z, v, w)) = \{x = a, v = f(y, z), w = g(y, z)\} = (\forall y)(\forall z)(\sim P(a, y) \vee \sim Q(z, g(y, z)) \vee R(z, f(y, z), g(y, z))).$$

Сколемовская стандартная форма – это предваренная форма, префикс которой содержит только кванторы всеобщности. Поэтому удобно префикс опустить, считая, что каждая переменная в матрице управляется квантором всеобщности, а так как конъюнктивная форма – это конъюнкция дизъюнктов, то сколемовскую стандартную форму можно представить в виде множества дизъюнктов.

📖 Сколемовская стандартная форма – это множество дизъюнктов.

Например, стандартная форма

$$(\forall y)(\forall z)(\sim P(a, y) \vee \sim Q(z, g(y, z)) \vee R(z, f(y, z), g(y, z)))$$

может быть представлена множеством

$$S = \{\sim P(a, y) \vee \sim Q(z, g(y, z)) \vee R(z, f(y, z), g(y, z))\}.$$

2.5. Выводы в логических моделях первого порядка

Формула логики предикатов противоречива в том случае, когда противоречиво множество дизъюнктов, представляющих стандартную форму этой формулы. А противоречиво множество дизъюнктов тогда, когда оно ложно при всех интерпретациях во всех предметных областях.

📖 Черч и Тьюринг: Не существует общего универсального метода, позволяющего определить общезначимость или противоречивость формулы логики предикатов первого порядка.

📖 Однако существуют алгоритмы, подтверждающие общезначимость или противоречивость формулы, если формула действительно общезначима или противоречива. Для необщезначимых или непротиворечивых формул алгоритмы, в общем случае, свою работу не заканчивают.

Частные, но интересные, с точки зрения определения процедуры поиска доказательства противоречивости формулы, результаты были получены Эрбраном. Так как рассмотрение всех возможных интерпретаций формулы логики предикатов, в общем случае, невозможно, Эрбраном была найдена специальная универсальная область интерпретации. Стандартная форма формулы противоречива тогда и только тогда, когда форма ложна при всех интерпретациях в этой области. Называют эту область *эрбрановским универсумом* и определяется она для множества дизъюнктов S следующим образом:

□ множество констант нулевого уровня H_0 состоит из констант, встречающихся в S ; если S не содержит констант, тогда H_0 содержит одну произвольно выбранную константу, допустим $H_0 = \{c\}$;

□ множество констант i -го уровня ($i = 1, 2, \dots, \infty$) определяется как объединение констант $i-1$ -го уровня и множества всех термов $f(t_1, t_2, \dots, t_n)$, где f – все функциональные символы, встречающиеся в S , а t_1, t_2, \dots, t_n константы $(i-1)$ -го уровня;

□ множество H_∞ есть эрбрановский универсум множества дизъюнктов S .

Элементы эрбрановского универсума – это абстрактные объекты, не имеющие конкретной интерпретации. Если во множестве S отсутствуют функции, то эрбрановский универсум всегда конечен и состоит из множества констант. Если же множество S содержит хотя бы одну функцию, то универсум всегда бесконечен.

Рассмотрим несколько примеров. Пусть множество дизъюнктов $S = \{P(x, y) \vee \sim Q(x, z, u), \sim P(u, y) \vee R(y) \vee Q(x, y, u), S(x)\}$, тогда $H_\infty = \{c\}$.

Для множество дизъюнктов

$S = \{Q(a, g(y)), P(x, y)\}$, универсум $H_\infty = \{a, g(a), g(g(a)), g(g(g(a))), \dots\}$.

Если множество дизъюнктов

$S = \{\sim P(a, y) \vee \sim Q(z, g(y, z)) \vee R(z, f(y, z)), g(y, z)\}$, то $H_\infty = \{a, g(a, a), f(a, a), g(a, g(a, a)), g(g(a, a), a), g(g(a, a), g(a, a)), g(a, f(a, a)), g(f(a, a), a), g(f(a, a), f(a, a)), f(a, g(a, a)), f(g(a, a), a), f(g(a, a), g(a, a)), \dots\}$.

Эрбрановской интерпретацией или *H-интерпретацией* для множества дизъюнктов S называется интерпретация, удовлетворяющая следующим условиям:

- предметной областью является эрбрановский универсум;
- интерпретация отображает все константы из S в соответствующую эрбрановскую константу;
- если f – это n -местный функциональный символ и h_1, h_2, \dots, h_n – константы эрбрановского универсума, то в эрбрановской интерпретации через f обозначается функция, отображающая (h_1, h_2, \dots, h_n) в $f(h_1, h_2, \dots, h_n)$.

В общем случае, эрбрановских интерпретаций на множестве S может быть бесконечно много, так как интерпретации предикатов и функций могут быть выбраны произвольно. Пусть множество дизъюнктов

$S = \{P(x, y) \vee \sim Q(x, z, u), \sim P(u, y) \vee R(y) \vee Q(x, y, u), S(x)\}$,

эрбрановский универсум для этого множества дизъюнктов $H_\infty = \{c\}$. Некоторые интерпретации приведены ниже

$I_1 = \{P(c, c), Q(c, c, c), R(c), S(c)\}$, $I_2 = \{\sim P(c, c), Q(c, c, c), R(c), S(c)\}$,


$I_3 = \{P(c, c), \sim Q(c, c, c), R(c), S(c)\}$, $I_4 = \{P(c, c), Q(c, c, c), \sim R(c), S(c)\}$,

так как здесь имеется четыре атома $P(c, c), Q(c, c, c), R(c), S(c)$, то всего существует $2^4 = 16$ эрбрановских интерпретаций.

Для множество дизъюнктов $S = \{Q(a, g(y)), P(x, y)\}$, эрбрановский универсум бесконечен $H_\infty = \{a, g(a), g(g(a)), g(g(g(a))), \dots\}$, и, соответственно, эрбрановских интерпретаций будет бесконечное множество, четыре из них приведены ниже:

$$\begin{aligned}
I_1 &= \{P(a, a), Q(a, a), P(a, g(a)), Q(a, g(a)), P(g(a), a), Q(g(a), a), P(g(a), g(a)), \\
&\quad Q(g(a), g(a)), \dots\}, \\
I_2 &= \{\sim P(a, a), Q(a, a), \sim P(a, g(a)), Q(a, g(a)), \sim P(g(a), a), Q(g(a), a), \\
&\quad \sim P(g(a), g(a)), \dots\}, \\
I_3 &= \{P(a, a), \sim Q(a, a), P(a, g(a)), \sim Q(a, g(a)), P(g(a), a), \sim Q(g(a), a), \\
&\quad P(g(a), g(a)), \dots\}, \\
I_4 &= \{\sim P(a, a), \sim Q(a, a), \sim P(a, g(a)), \sim Q(a, g(a)), \sim P(g(a), a), \sim Q(g(a), a), \\
&\quad \sim P(g(a), g(a)), \dots\}.
\end{aligned}$$

Доказано, что множество дизъюнктов S невыполнимо тогда и только тогда, когда оно ложно при всех своих эрбрановских интерпретациях.

 **Для проверки выполнимости множество дизъюнктов необходимо рассматривать только эрбрановские интерпретации.**

Выражение – это терм, множество термов, множество атомов, множество дизъюнктов. Если выражение не содержит переменных, то оно называется *основным*.

Основной пример дизъюнкта C множества дизъюнктов S есть дизъюнкт, полученный заменой переменных в C на константы эрбрановского универсума S . Пусть множество дизъюнктов $S = \{Q(a, g(y)), P(x, y)\}$, дизъюнкт $C = P(x, y)$, эрбрановский универсум $H_\infty = \{a, g(a), g(g(a)), g(g(g(a))), \dots\}$, тогда основной пример $C' = P(g(a), a)$.

Основной пример выполняется в данной интерпретации тогда и только тогда, когда в этом основном примере существует основная литера, которая есть и в данной интерпретации. Пусть множество дизъюнктов $S = \{Q(a, g(y)), P(x, y)\}$, дизъюнкт $C = P(x, y)$, основной пример $C' = P(g(a), a)$, шесть интерпретаций:

$$\begin{aligned}
I_1 &= \{P(a, a), Q(a, a), P(a, g(a)), Q(a, g(a)), P(g(a), a), Q(g(a), a), P(g(a), g(a)), \\
&\quad Q(g(a), g(a)), \dots\}; \\
I_2 &= \{\sim P(a, a), Q(a, a), \sim P(a, g(a)), Q(a, g(a)), \sim P(g(a), a), Q(g(a), a), \\
&\quad \sim P(g(a), g(a)), \dots\}; \\
I_3 &= \{P(a, a), \sim Q(a, a), P(a, g(a)), \sim Q(a, g(a)), P(g(a), a), \sim Q(g(a), a), \\
&\quad P(g(a), g(a)), \dots\}; \\
I_4 &= \{\sim P(a, a), \sim Q(a, a), \sim P(a, g(a)), \sim Q(a, g(a)), \sim P(g(a), a), \\
&\quad \sim Q(g(a), a), \sim P(g(a), g(a)), \dots\}; \\
I_5 &= \{\sim P(a, a), Q(a, a), P(a, g(a)), Q(a, g(a)), P(g(a), a), Q(g(a), a), \\
&\quad P(g(a), g(a)), \dots\}; \\
I_6 &= \{\sim P(a, a), \sim Q(a, a), P(a, g(a)), Q(a, g(a)), P(g(a), a), Q(g(a), a), \\
&\quad P(g(a), g(a)), \dots\}.
\end{aligned}$$

Основной пример C' выполняется в интерпретации I_1, I_3, I_5, I_6 и опровергается в I_2, I_4 .

Дизъюнкт выполняется в данной интерпретации тогда и только тогда, когда каждый основной пример выполняется в данной интерпретации.

Дизъюнкт опровергается в данной интерпретации тогда и только тогда, когда существует хотя бы один основной пример данного дизъюнкта, который не выполняется в данной интерпретации.

Вернемся к предыдущему примеру, дизъюнкт $C = P(x, y)$ выполняется в интерпретации I_1, I_3 и опровергается в I_2, I_4, I_5, I_6 .

Множество дизъюнктов невыполнимо тогда и только тогда, когда для каждой интерпретации существует хотя бы один основной пример дизъюнкта, невыполнимый в данной интерпретации.

Пусть $S = \{P(x) \vee \sim Q(x), \sim P(x), Q(x)\}$, на данном множестве дизъюнктов существуют четыре интерпретации:

$I_1 = \{P(c), Q(c)\}$, $I_2 = \{\sim P(c), Q(c)\}$, $I_3 = \{P(c), \sim Q(c)\}$, $I_4 = \{\sim P(c), \sim Q(c)\}$,

в интерпретации I_1 опровергается дизъюнкт $\sim P(x)$; дизъюнкт $P(x) \vee \sim Q(x)$ опровергается в интерпретации I_2 ; интерпретация I_3 опровергает дизъюнкт $Q(x)$; этот же дизъюнкт $Q(x)$ опровергается в интерпретации I_4 . Рассматриваемое множество дизъюнктов невыполнимо, потому что оно опровергается во всех интерпретациях.

Теорема Эрбрана. **Множество дизъюнктов невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество основных примеров дизъюнктов указанного множества.**

Рассмотрим предыдущий пример, $S = \{P(x) \vee \sim Q(x), \sim P(x), Q(x)\}$, приведенное множество дизъюнктов невыполнимо потому, что существует следующее невыполнимое множество основных примеров:

$$S' = \{P(c) \vee \sim Q(c), \sim P(c), Q(c)\}.$$

Процедуры поиска опровержения, основанные на теореме Эрбрана, имеют один существенный недостаток, они требуют генерации множеств $S'_1, S'_2, \dots, S'_n, \dots$ основных примеров рассматриваемого множества дизъюнктов S . Очень часто размерность множеств основных примеров растет экспоненциально, по этой причине такие процедуры не имеют практического применения на современных компьютерах. И только с появлением метода резолюций, разработанного Дж. Робинсоном, были найдены эффективные компьютерные процедуры доказательства невыполнимости множества дизъюнктов. Идея метода Робинсона состоит в том, чтобы работать напрямую с дизъюнктами, входящими в рассматриваемое множество, не порождая основных примеров. Реализация метода резолюций требует применения операции унификации, но прежде чем объяснить эту операцию, дадим определение понятия подстановка.

Подстановка – это конечное множество вида $\{v_1 = t_1, v_2 = t_2, \dots, v_n = t_n\}$, где v_j – это переменная, а t_j – это терм, отличный от v_j , причем все v_j отличны друг от друга.

Унифицировать два или более выражений значит найти такую подстановку, которая делает выражения одинаковыми (тождественными). Такая подстановка называется *унификатором*.

Рассмотрим два выражения $P(x, a, f(a, g(y)))$ и $P(h(v), z, f(z, u))$. Они не тождественны, однако, если применить к ним операцию унификации с подстановкой $\{x = h(v), z = a, u = g(y)\}$, то получим два тождественных выражения $P(h(v), a, f(a, g(y)))$.

В логическом исчислении удобно работать с дизъюнктами, не содержащими повторяющихся литер. Устранение повторения в результирующем дизъюнкте выполняют с помощью операции склейки. Если две или более литер с одинаковым знаком в дизъюнкте C имеют общий унификатор $\{v_1 = t_1, v_2 = t_2, \dots, v_n = t_n\}$, то дизъюнкт, полученный из C заменой одновременно всех вхождений переменных v_i на термы t_i , называется *склейкой* дизъюнкта C .

Применение операции унификации и склейки делает возможным использование метода резолюций в логике предикатов первого порядка.

Пусть C_1 и C_2 – дизъюнкты, не имеющие общих переменных. Если в дизъюнкте C_1 существует литера L_1 , а в дизъюнкте C_2 существует литера $\sim L_2$, и литеры L_1 и $\sim L_2$ унифицируемы, то, вычеркнув литеры L_1 и $\sim L_2$ из C_1 и C_2 , соответственно, построим дизъюнкцию оставшихся дизъюнктов, которая называется *бинарной резольвентой* дизъюнктов C_1 и C_2 .

Рассмотрим пример, пусть $C_1 = P(x, g(b)) \vee \sim Q(x)$ и $C_2 = \sim P(a, g(x)) \vee R(z)$. Сначала переименуем переменную x в первом дизъюнкте:

$$C_1 = P(y, g(b)) \vee \sim Q(y).$$

Выберем в дизъюнкте C_1 литеру $C_1 = P(y, g(b))$ и литеру $C_2 = \sim P(a, g(x))$ в C_2 . Литеры C_1 и $\sim C_2$ имеют общий унификатор $\{y = a, x = b\}$, тогда бинарная резольвента имеет следующий вид: $\sim Q(a) \vee R(z)$.

Резольвентой дизъюнктов C_1 и C_2 является одна из следующих резольвент:

бинарная резольвента C_1 и C_2 ;

бинарная резольвента C_1 и склейки C_2 ;

бинарная резольвента C_2 и склейки C_1 ;

бинарная резольвента склейки C_1 и склейки C_2 .

Пусть $C_1 = \sim P(x, y) \vee \sim P(f(z), z) \vee \sim Q(x, y)$ и $C_2 = P(f(g(a)), g(a)) \vee R(a)$. Склейка C_1 имеет следующий вид: $\sim P(f(z), z) \vee \sim Q(f(z), z)$. Бинарная резольвента склейки C_1 и C_2 равна $\sim Q(f(g(a)), g(a)) \vee R(a)$.

2.6. Аудиторные задания

1. Ни один человек не является четвероногим. Все женщины – люди. Следовательно, ни одна женщина не является четвероногой. Доказать.

2. Некоторые республиканцы любят всех демократов. Ни один республиканец не любит ни одного социалиста. Следовательно, ни один демократ не является социалистом. Доказать.

3. Никакой торговец подержанными автомобилями не покупает подержанный автомобиль для своей семьи. Некоторые люди, покупающие подержанные автомобили для своих семей – жулики. Следовательно, некоторые жулики не являются торговцами подержанными автомобилями. Доказать.

4. Некоторые пациенты любят своих докторов. Ни один пациент не любит знахаря. Следовательно, никакой доктор не является знахарем. Доказать.

5. Используя процедуру Эрбрана, доказать невыполнимость множества дизъюнктов

$$\text{а) } S = \{\sim S(y) \vee \sim C(y), S(b), V(a, b), \sim C(z) \vee V(a, z)\};$$

$$\text{б) } S = \{\sim S(x, y) \vee \sim M(y) \vee I(f(x)), \sim S(x, y) \vee \sim M(y) \vee E(x, f(x)), \sim I(z), S(a, b), M(b)\};$$

$$\text{в) } S = \{\sim E(x) \vee V(x) \vee S(x, f(x)), \sim E(x) \vee V(x) \vee C(f(x)), P(a), E(a), \sim S(a, y) \vee P(y), \sim P(x) \vee \sim V(x), \sim P(x) \vee \sim C(x)\}.$$

2.7. Самостоятельная работа

1. Ни один первокурсник не любит второкурсников. Все, живущие на шестом этаже, – второкурсники. Следовательно, ни один первокурсник не любит никого из живущих на шестом этаже. Доказать.

2. Ни один торговец наркотиками не является наркоманом. Некоторые наркоманы привлекались к ответственности. Следовательно, некоторые люди, привлекавшиеся к ответственности, не являются торговцами наркотиками. Доказать.

3. Все первокурсники встречаются со всеми второкурсниками. Ни один первокурсник не встречается ни с одним студентом предпоследнего курса. Существуют первокурсники. Следовательно, ни один второкурсник не является студентом предпоследнего курса. Доказать.

4. Используя процедуру Эрбрана, доказать невыполнимость множества дизъюнктов

$$\text{а) } S = \{P(x, a, g(x, b)), \sim P(f(y), z, g(f(a), b))\};$$

$$\text{б) } S = \{P(x), Q(x, f(x)) \vee \sim P(x), \sim Q(g(y), z)\}.$$

2.8. Контрольные вопросы

1. Чем отличается логика высказываний от логики первого порядка?
2. Что такое правильно построенная формула в логике первого порядка?
3. Что есть интерпретация в логике первого порядка?
4. Что такое подстановка?
5. Что такое унификация?
6. Охарактеризуйте способы логического вывода в первого порядка.

Тема 3. Продукционная модель представления знаний

Цель занятия – систематизация, углубление, закрепление полученных теоретических знаний о механизмах вывода в продукционных системах.

3.1. Продукционная система

Продукционная система состоит из трех основных компонентов. Первый из них – это набор правил, используемый как база знаний, иногда его еще называют *базой правил*. Второй компонент – это *база фактов* или *рабочая память* – память для временного хранения, в которой хранятся предпосылки, касающиеся конкретных задач предметной области, и результаты выводов, получаемые на их основании. Третий компонент реализует *механизм логического вывода*, обрабатывающий правила в соответствии с содержанием рабочей памяти; другое название этого компонента – машина логического вывода.

Пусть *база правил* в продукционной системе имеет содержимое:

если F и B то Z ; если C и D то F ; если A то D ;

рабочая память: A, B, H, C .

Рассмотрим, каким образом «работают» правила. Система построена так, что один раз выбранное правило из базы правил выполняться будет только один раз. Оно как бы «выгорает». Первым выгорает правило «если A то D », так как A уже имеется в базе данных. В качестве следствия этого правила получается логический вывод о наличии ситуации D , которая заносится в рабочую область. Это вызывает выгорание правила «если C и D то F », и, как следствие, выводится ситуация F и она заносится в базу данных. Это, в свою очередь, вызывает выгорание правила «если F и B то Z » с занесением Z в базу данных. Такой способ называется прямым выводом. Графически вывод можно изобразить в виде И/ИЛИ дерева (рис. 3.1).

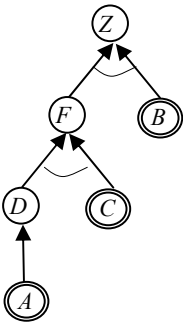


Рис. 3.1

Существует и другой способ вывода, называемый *обратным выводом*. При его использовании система начинает работу с формулировки того, что требуется доказать, например, предполагая, что ситуация Z присутствует, и выполняет только те правила, которые имеют отношение к доказательству предположения.

Если на каждом этапе логического вывода существует множество применяемых правил, то это множество носит название *конфликтного набора*, а выбор одного из них называется разрешением конфликта.

Чтобы повысить эффективность продукционной системы, необходимо решить проблему управления последовательностью применения правил или управления выводом.

3.2. Аудиторные задания

База правил и рабочая память в продукционной системе имеет содержимое, заданное в вариантах. Проиллюстрировать графически механизм прямого и обратного логического вывода факта A . Обратите внимание на изменение содержимого рабочей памяти в процессе вывода. Проведите упорядочение правил вывода. Рассмотрите возможные конфликты при прямом и обратном выводе.

Вариант 1. База правил:

если F и D и E то B ; если G то C ; если B и C то A ; если R то D ; если S то A ;

если F и G то M ;

рабочая память: G, E, R, F .

Вариант 2. База правил:

если B и C и D то A ; если E то B ; если G и H то C ; если F то B ; если E то A ;

рабочая память: G, H, D, F .

Вариант 3. База правил:

если B и C и D то A ; если F и G то B ; если H и D то E ; если E то A ;

рабочая память: G, H, D, F .

Вариант 4. База правил:

если C и D то B ; если E то B ; если F и G то E ; если B то A ; если H то C ;

рабочая память: G, H, D, F .

3.3. Самостоятельная работа

База правил и рабочая память в продукционной системе имеет содержимое, заданное в вариантах. Проиллюстрировать графически механизм прямого и обратного логического вывода факта A . Обратите внимание на изменение содержимого рабочей памяти в процессе вывода. Проведите упорядочение правил вывода. Рассмотрите возможные конфликты при прямом и обратном выводе.

Вариант 1. База правил:

если D то B ; если F и H то D ; если B и C то A ; если G и R то E ; если E то B ; если F и G то C ;

рабочая память: G, H, F .

Вариант 2. База правил:

если B то A ; если E и F и D то B ; если G то C ; если H то C ; если C то A ;
если D то K ;

рабочая память: G, H, D .

Вариант 3. База правил:

если B то A ; если E и F и G то C ; если H то D ; если C то B ; если D то B ;
если E то D ;

рабочая память: G, H, F .

3.4. Контрольные вопросы

1. Что есть продукционная система?
2. Каковы функции рабочей памяти и машины логического вывода?
3. Что есть цепочка логического вывода?
4. Обоснуйте преимущества и недостатки прямого и обратного вывода.

Тема 4. Семантические сети

Цель занятия – систематизация, углубление, закрепление полученных теоретических знаний о моделях представления знаний на основе семантических сетей (TLC-модель).

4.1. TLC-модель

В качестве модели семантической сети рассмотрим модель, получившую название TLC-модели (Teachable Language Comprehender: доступный механизм понимания языка). Здесь основой для определения значения того или иного понятия является множество взаимосвязей с другими понятиями (т.е. определение понятия дается через отношения с другими понятиями). Среди этих связей основными являются следующие:

- класс, к которому принадлежит данное понятие;
- свойства, выделяющие понятие из всех прочих понятий этого класса;
- примеры данного понятия.

В качестве примера рассмотрим простейшую сеть для представления концептуального объекта «автомобиль» (рис. 4.1).

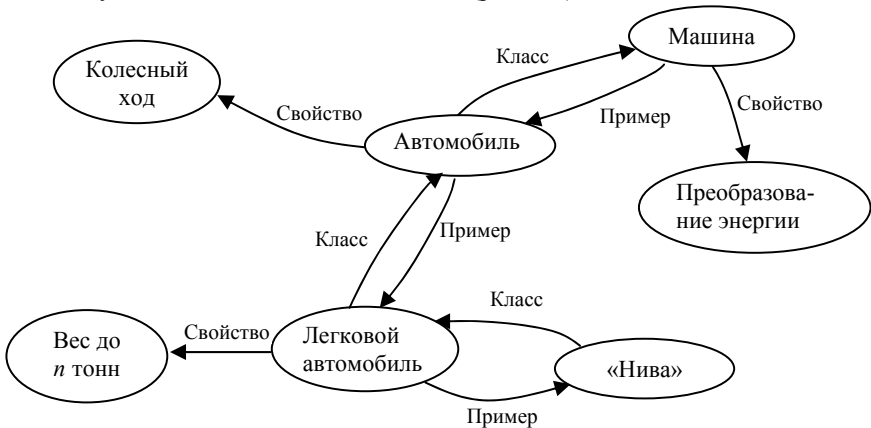


Рис. 4.1

4.2. Аудиторные задания

Постройте TLC-модель для определения понятия, заданного в варианте; поскольку слова, используемые в определении понятия, сами обозначают понятия, то определив их, построите некоторую структуру, определяющую

каждое понятие через взаимосвязи с другими имеющимися понятиями. Рассмотрите не менее десяти понятий в сети.

Вариант 1. Понятие «студент».

Вариант 2. Понятие «профессор».

Вариант 3. Понятие «шкаф».

Вариант 4. Понятие «компьютер».

Вариант 5. Понятие «стол».

Вариант 6. Понятие «журнал».

Вариант 7. Понятие «книга».

4.3. Самостоятельная работа

Постройте TLC-модель для определения понятия, заданного в варианте; поскольку слова, используемые в определении понятия, сами обозначают понятия, то определив их, постройте некоторую структуру, определяющую каждое понятие через взаимосвязи с другими имеющимися понятиями. Рассмотрите не менее десяти понятий в сети.

Вариант 1. Понятие «ребенок».

Вариант 2. Понятие «трактор».

Вариант 3. Понятие «посуда».

4.4. Контрольные вопросы

1. Определите основную концепцию представления знаний на основе семантической сети.
2. Назовите основные отношения, принятые в TLC-модели.
3. Выделите достоинства и недостатки TLC-модели.

Тема 5. Фреймовая модель представления знаний

Цель занятия – изучение представления знаний на основе фреймов.

5.1. Фреймовая модель представления статических знаний

Фреймовая модель, или модель представления знаний, основанная на фреймовой теории М.Минского, представляет собой систематизированную в виде единой теории психологическую модель памяти человека и его сознания. Важным моментом в этой теории является понятие фрейма – структуры данных для представления некоторого концептуального объекта. Информация, относящаяся к этому фрейму, содержится в слотах. Все фреймы взаимосвязаны и образуют единую сеть фреймов. Однако четкого определения связи между фреймами и слотами может и не быть. Рассмотрим пример фреймовой системы, описывающей аудиторию 426.

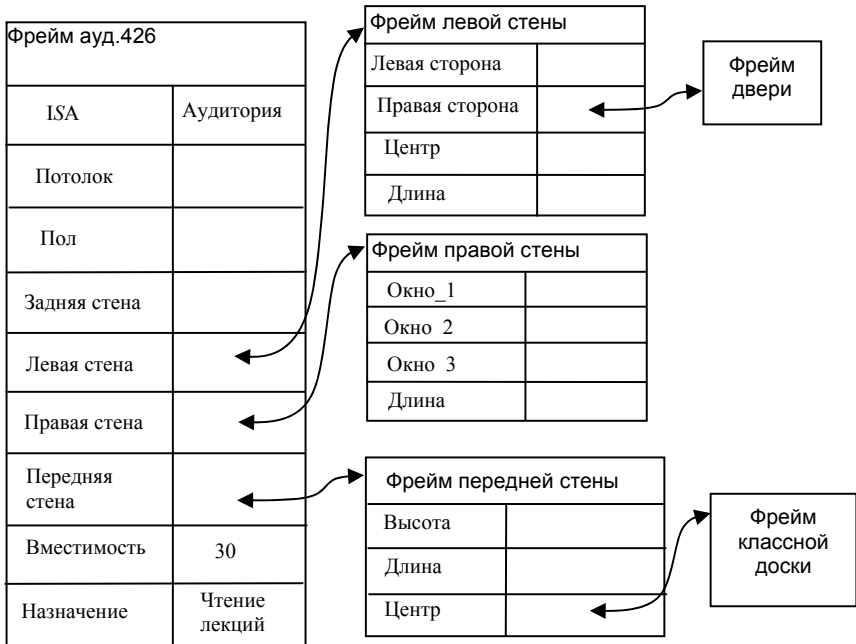


Рис. 5.1

5.2. Фреймовая модель представления динамических знаний

С помощью фреймов описываются не только статические ситуации (аудитория), но и динамические – празднование дня рождения, подготовка к дню рождения. Каждый фрейм можно рассматривать как сеть, состоящую из нескольких вершин и отношений.

Фрейм прокол	
ISA	Действие
Инструмент	Остроконечный предмет
Объект	Надутый предмет
Способ	(Столкновение инструмент объект)
Результат	(Разрушение объект)
Свидетельство	Взрыв

Значениями слота являются процедуры, параметрами которых являются значения слотов этого же фрейма. Иерархическая структура сети фреймов построена на отношениях «род – вид» и «часть – целое». Отношения «род-вид» характерны тем, что на верхних уровнях расположены абстрактные объекты (концепты), а на нижних уровнях – конкретные объекты. Причем объекты нижних уровней наследуют значения атрибутов объектов верхних уровней. Эти отношения еще называют отношениями типа ISA. Такое название объясняется формой записи «ПРОКОЛ» IS а «действие».

Отношение «часть – целое» касается структурированных объектов и показывает, что объект нижнего уровня является частью объекта верхнего уровня. Например, стена является структурным элементом аудитории.

5.3. Аудиторные задания

1. Построить фреймы для следующих статических понятий:

Вариант 1. Понятие «студент».

Вариант 2. Понятие «профессор».

Вариант 3. Понятие «шкаф».

Вариант 4. Понятие «компьютер».

Вариант 5. Понятие «стол».

Вариант 6. Понятие «журнал».

2. Построить фреймы для следующих динамических понятий:

Вариант 1. Понятие «вращение».

Вариант 2. Понятие «падение».

Вариант 3. Понятие «взлет».

Вариант 4. Понятие «вычисление».

5.4. Самостоятельная работа

1. Построить фреймы для следующих понятий:

Вариант 1. Понятие «книга».

Вариант 2. Понятие «ребенок».

Вариант 3. Понятие «трактор».

Вариант 4. Понятие «посуда».

2. Построить фреймы для следующих динамических понятий:

Вариант 1. Понятие «плавание».

Вариант 2. Понятие «испарение».

5.5. Контрольные вопросы

1. Определите характер описания информации во фреймовых система.
2. Какие значения может принимать слот фрейма?
3. Что означает ISA-отношение?

Тема 6. Нечеткие множества и нечеткие отношения

Цель занятия – систематизация, углубление, закрепление полученных теоретических знаний о задании нечетких множеств и нечетких отношений, а также операциях над ними.

6.1. Нечеткие множества

Задание нечетких множеств

Знания не всегда могут быть описаны точно – часто встречаются так называемые «нечеткие знания». Люди повседневно решают проблемы и делают заключения в среде «нечетких знаний», а для того чтобы интеллектуальные системы обладали такими возможностями как гибкость, широкий кругозор, адаптируемость, необходимо представление и использование нечетких знаний. Все нечеткости, с которыми имеет дело инженерия знаний, можно классифицировать следующим образом: недетерминированность выводов; неоднозначность; ненадежность; неполнота; собственно нечеткость.

Одним из методов изучения множеств без уточнения их границ является теория нечетких множеств, которая была предложена Л. Заде. Много исследований связано также с нечеткими выводами, которые выполняются с использованием правил, представленных как нечеткие множества.

Нечеткое подмножество F множества элементов U определяется функцией принадлежности $\mu_F(u)$. Эта функция отображает элементы u множества U на множество чисел в интервале $[0, 1]$, которые указывают степень принадлежности каждого элемента нечеткому подмножеству F . Если множество U состоит из конечного числа элементов $\{u_1, u_2, \dots, u_n\}$, то нечеткое подмножество F можно представить следующим образом:

$$F = \mu_F(u_1)/u_1 + \mu_F(u_2)/u_2 + \dots + \mu_F(u_n)/u_n = \sum_{i=1}^n \mu_F(u_i)/u_i$$

Следует иметь в виду, что знак плюс в этой формуле означает не суммирование, а объединение или конъюнкцию.

Пример. $U = 1 + 2 + 3 + \dots + 10$. Тогда нечеткое множество A , которое описывается понятием «несколько», можно записать в следующем виде:

$$\text{«несколько»} = 0,4/2 + 0,7/3 + 0,8/9$$

символ « \Leftarrow » означает здесь равенство по определению.

В случае непрерывного множества U вводится следующее обозначение подмножества F :

$$\int_U \mu(u)/\tilde{u}$$

Следует иметь в виду, что знак интеграла в приведенных выше формулах означает не интегрирование, а скорее объединение, а символ «/» показывает, что значение $\mu_A(u)$ относится к элементу, следующему за ним. Заметим также, что если $\mu_A(u)$ принимает значение только 0 или 1, то множество A является обычным множеством. Запись $\mu_A(u) = 1$ означает, что элемент $u \in U$ принадлежит множеству A , т.е. $u \in A$. Запись $\mu_A(u) = 0$ означает, что $u \in U$ не принадлежит множеству A , т.е. $u \notin A$.

Пусть U – множество людей в возрасте 0–100 лет, функции принадлежности нечетких множеств, означающих возраст: «молодой», «средний», «старый» можно определить так

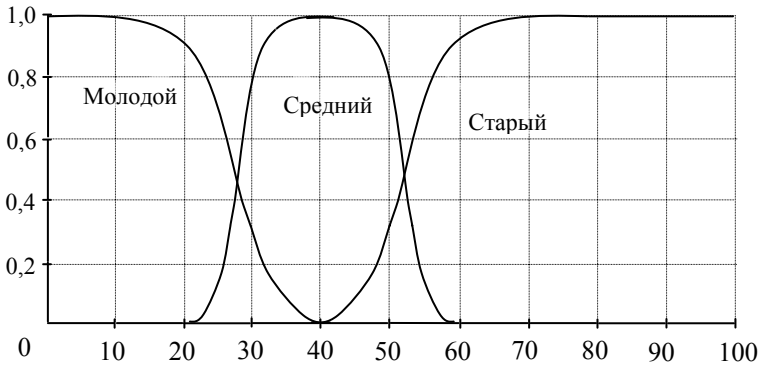


Рис. 6.1. Функции принадлежности

При дискретизации через 10 лет получим приблизительно следующее:

$$\text{молодой} = 1/0 + 1/10 + 0,9/20 + 0,3/30,$$

$$\text{средний} = 0,8/30 + 1/40 + 0,8/50,$$

$$\text{старый} = 0,3/50 + 0,9/60 + 1/70 + 1/80 + 1/90 + 1/100.$$

Здесь элементы множества с функцией принадлежности равной 0 не записываются.

Операции над нечеткими множествами

Над нечеткими множествами выполняются те же операции, что и над обычными множествами.

Понятие *нечеткого подмножества* Л. Заде определил следующим образом: нечеткое подмножество данного конечного множества U – это такое подмножество, значения степеней принадлежности элементов которого лежат в единичном интервале $[0, 1]$. Пусть $A = \int_U \mu_A(u)/u$ и $B = \int_U \mu_B(u)/u$ два

нечетких множества, тогда нечеткое множество A является *подмножеством* нечеткого множества B ($A \subseteq B$), если для всех u справедливо неравенство $\mu_A(u) \leq \mu_B(u)$.

Равенство двух нечетких множеств определяется следующим образом: два нечетких множества A и B равны ($A = B$), если для всех u справедливо неравенство $\mu_A(u) = \mu_B(u)$.

ОБЪЕДИНЕНИЕ нечетких множеств A и B

$$A \cup B = \int_U (\mu_A(u) \vee \mu_B(u)) / u ,$$

где $(\mu_A(u) \vee \mu_B(u)) = \max(\mu_A(u), \mu_B(u))$.

ПЕРЕСЕЧЕНИЕ нечетких множеств A и B

$$A \cap B = \int_U (\mu_A(u) \wedge \mu_B(u)) / u ,$$

где $(\mu_A(u) \wedge \mu_B(u)) = \min(\mu_A(u), \mu_B(u))$.

ДОПОЛНЕНИЕ или *ОТРИЦАНИЕ* определяется следующей формулой:

$$\sim A = \int_U (1 - \mu_A(u)) / u .$$

Квантификатор *не* может интерпретироваться с помощью операции отрицания

$$\text{не } x = \int_U (1 - \mu_x(u)) / u .$$

РАЗНОСТЬ двух нечетких множеств определяется формулой

$$A - B = A \cap \sim B .$$

ДИЗЬЮНКТИВНАЯ СУММА определяется соотношением

$$A + B = (A \cup B) - (A \cap B) .$$

Например, *ПРОИЗВЕДЕНИЕ* нечетких множеств A и B определяется следующим соотношением:

$$A \times B = \int_U (\mu_A(u) \times \mu_B(u)) / u .$$

Операция *возведения в степень*

$$A^b = \int_U (\mu_A(u))^b / u ,$$

где $b > 0$.

Операция *концентрирования* нечеткого множества

$$\text{CON}(A) = A^2 .$$

Эта операция уменьшает степень принадлежности элементов тем больше, чем меньше степень их принадлежности первоначальному множеству A . Квантификатор *очень* может интерпретироваться с помощью операции концентрации, то есть возведения в квадрат

$$\text{очень } x = \int_U \mu_x^2(u) / u .$$

Операция *растяжения* нечеткого множества является противоположной концентрации и определяется соотношением

$$\text{DIL}(A) = A^{0,5}.$$

Операция контрастной интенсивности определяется соотношением

$$\text{INT}(A) = \begin{cases} 2 \cdot A^2, & 0 \leq \mu_A(u) \leq 0,5; \\ \sim(2 \cdot (\sim A)^2), & 0,5 \leq \mu_A(u) \leq 1. \end{cases}$$

Эта операция увеличивает значения $\mu_A(u)$, которые больше 0,5, и уменьшает те значения $\mu_A(u)$, которые меньше 0,5, уменьшая тем самым нечеткость A .

6.2. Нечеткие отношения

Определение. Пусть U и V – универсальные множества, на которых определены X и Y , соответственно, тогда нечеткое отношение

$$R: X \rightarrow Y$$

определяется как подмножество декартова произведения двух нечетких множеств $X \times Y \subseteq U \times V$, которое задается с помощью функции принадлежности двух переменных по формуле

$$R = \int_{X \times Y} \mu_R(u, v)/(u, v).$$

В общем случае n -арное отношение или n -отношение определяется следующим образом. Пусть R – результирующее множество декартова произведения n множеств и μ – его функция принадлежности. Нечеткое n -отношение определяется как нечеткое подмножество R , принимающее какое-либо значение на интервале функции принадлежности, в соответствии со следующей формулой:

$$R = \int_{X_1 \times \dots \times X_n} \mu_R(u_1, \dots, u_n)/(u_1, \dots, u_n) \cdot$$

$$x_i \in X_i, i = 1, \dots, n.$$

Пример. Допустим, что $X = \{\text{Иван, Мария}\}$; $Y = \{\text{Петр, Дарья}\}$, тогда
 Дружба = 0,6/(Иван, Петр) + 0,9/(Иван, Дарья) + 0,8/(Марья, Петр) +
 + 0,1/(Марья, Дарья).

Отношения удобно записывать с помощью матрицы отношений

$$\begin{array}{cc} & \begin{array}{c} \text{Петр} \\ \text{Дарья} \end{array} \\ \begin{array}{c} \text{Иван} \\ \text{Марья} \end{array} & \left| \begin{array}{cc} 0,6 & 0,9 \\ 0,8 & 0,1 \end{array} \right|. \end{array}$$

Допустим, что существует нечеткое знание–правило типа
 если F , то G ,

(если старый, то умный), использующее нечеткие множества $F \subset U$ и $G \subset V$. Тогда один из способов построения нечеткого отношения из соответствующей области полного множества U в область полного множества V состоит в следующем:

$$R = F \times G = \int_{U \times V} (\mu_F(u) \wedge \mu_G(v)) / (u, v),$$

или

$$R = F \times G = \sum_i \sum_j (\mu_F(u_i) \wedge \mu_G(v_j)) / (u_i, v_j).$$

Необходимо отметить, что есть и другие способы построения нечеткого отношения.

Пусть U и V – это области натуральных чисел от 1 до 4, тогда определим следующим образом нечеткие множества

$$F = \text{маленькие} = 1/1 + 0,6/2 + 0,3/3 + 0/4;$$

$$G = \text{большие} = 0/1 + 0,1/2 + 0,6/3 + 1/4.$$

Если есть нечеткое знание-правило

если u – маленькое, то v – большое,

то можно следующим образом построить нечеткое отношение, определяющее данное знание-правило:

$$R = F \times G = \left| \begin{array}{cccc} 0 & 0,1 & 0,6 & 1 \\ 0 & 0,1 & 0,6 & 0,6 \\ 0 & 0,1 & 0,3 & 0,3 \\ 0 & 0 & 0 & 0 \end{array} \right|.$$

Пусть R – нечеткое отношение из области U в область V , S – нечеткое отношение из области V в область W , тогда нечеткое отношение из области U в область W , определяется как свертка max-min:

$$R \bullet S = \sum_{i=1}^n \sum_{k=1}^l \bigcup_{v_j \in V} (\mu_R(u_i, v_j) \wedge \mu_S(v_j, w_k)) / (u_i, w_k).$$

Здесь знак « \bullet » обозначает свертку max-min; \bigcup_{v_j} – взятие максимума для

всех v_j ; \wedge – взятие минимума.

Пример. $U = V = W = \{1, 2, 3, 4\}$.

$$F = \text{маленькие} = 1/1 + 0,6/2 + 0,3/3 + 0/4;$$

$$G = \text{большие} = 0/1 + 0,1/2 + 0,6/3 + 1/4;$$

$$\sim F = \text{не_маленькие} = 0/1 + 0,4/2 + 0,7/3 + 1/4;$$

$$G^2 = \text{очень_большие} = 0/1 + 0,01/2 + 0,36/3 + 1/4 \text{ или, округлив,}$$

$$0/1 + 0/2 + 0,4/3 + 1/4.$$

Тогда если есть знание-правило

если v – не маленькое, то w – очень большое,

то в соответствии с формулой $S = \sim F \times G^2$ можно построить нечеткое отношение S из V в W

$$S = \sim F \times G^2 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0,4 & 0,4 \\ 0 & 0 & 0,4 & 0,7 \\ 0 & 0 & 0,4 & 1 \end{vmatrix}.$$

И далее можно построить нечеткое отношение из U в W .

$$R \cdot S = \begin{vmatrix} 0 & 0,1 & 0,6 & 1 \\ 0 & 0,1 & 0,6 & 0,6 \\ 0 & 0,1 & 0,3 & 0,3 \\ 0 & 0 & 0 & 0 \end{vmatrix} \bullet \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0,4 & 0,4 \\ 0 & 0 & 0,4 & 0,7 \\ 0 & 0 & 0,4 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0,4 & 1 \\ 0 & 0 & 0,4 & 0,6 \\ 0 & 0 & 0,3 & 0,3 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

6.3. Аудиторные задания

1. Построить нечеткие множества для следующих понятий:

- понятие «горячий чай»;
- понятие «высокая скорость автомобиля»;
- понятие «сильный ветер».

2. Дано множество $U = \{1, 2, 3, 4\}$.

Определить: $A \cup B$, $A \cap B$, $\sim A$, $\sim B$, $A - B$, $A + B$, $\text{CON}(A)$, $\text{CON}(B)$, $\sim \text{CON}(A)$, $\text{DIL}(A)$, $\text{DIL}(B)$, $\text{INT}(A)$, $\text{INT}(B)$.

Значения переменных A и B приведены ниже:

- $A = 1/1 + 0,8/2 + 0,4/3 + 0/4$; $B = 0/1 + 0,1/2 + 0,7/3 + 1/4$.
- $A = 1/1 + 0,7/2 + 0,2/3 + 0/4$; $B = 0/1 + 0,2/2 + 0,7/3 + 0/4$.
- $A = 0/1 + 0,4/2 + 0,6/3 + 1/4$; $B = 0/1 + 0,2/2 + 0,6/3 + 1/4$.

3. $U = V = W = \{1, 2, 3, 4\}$.

Определены следующие правила:

- если u – маленькое, то v – не большое,
- если v – не очень маленькое, то w – очень большое.

Построить нечеткое отношение S из V в W .

Значения переменных *маленькое* и *большое* приведены ниже:

- маленькое* = $1/1 + 0,8/2 + 0,4/3 + 0/4$;
большое = $0/1 + 0,1/2 + 0,7/3 + 1/4$.
- маленькое* = $1/1 + 0,75/2 + 0,4/3 + 0/4$;
большое = $0/1 + 0,2/2 + 0,7/3 + 1/4$.
- маленькое* = $1/1 + 0,7/2 + 0,4/3 + 0/4$;
большое = $0/1 + 0,25/2 + 0,75/3 + 1/4$.

4. $U = V = W = \{1, 2, 3, 4, 5\}$.

Определены следующие правила:

- если u – не маленькое, то v – среднее,
- если v – большое, то w – очень маленькое.

Построить нечеткое отношение S из V в W .

Значения переменных *маленькое*, *среднее* и *большое* приведены ниже:

- а) *маленькое* = $1/1 + 0,8/2$; *среднее* = $0,3/2 + 1/3 + 0,2/4$;
большое = $0,1/3 + 0,75/4 + 1/5$.
- б) *маленькое* = $1/1 + 0,7/2$; *среднее* = $0,1/2 + 1/3 + 0,7/4$;
большое = $0,1/3 + 0,5/4 + 1/5$.
- в) *маленькое* = $1/1 + 0,7/2$; *среднее* = $0,25/2 + 1/3 + 0,25/4$;
большое = $0,1/3 + 0,7/4 + 1/5$.

6.4. Самостоятельная работа

1. Построить нечеткие множества для следующих понятий:

- а) понятие «теплая погода»;
 б) понятие «умный человек»;

2. Дано множество $U = \{1, 2, 3, 4\}$.

Определить: $A \cup B$, $A \cap B$, $\sim A$, $\sim B$, $A - B$, $A + B$, $\text{CON}(A)$, $\text{CON}(B)$,
 $\sim \text{CON}(A)$, $\text{DIL}(A)$, $\text{DIL}(B)$, $\text{INT}(A)$, $\text{INT}(B)$.

Значения переменных A и B приведены ниже:

- а) $A = 0,2/1 + 1/2 + 0,3/3 + 0/4$; $B = 0/1 + 0,8/2 + 1/3 + 0,1/4$.
 б) $A = 0/1 + 0,5/2 + 0,8/3 + 1/4$; $B = 1/1 + 0,7/2 + 0,3/3 + 0/4$.

3. $U = V = W = \{1, 2, 3, 4\}$.

Определены следующие правила:

если u – маленькое, то v – не большое,
если v – не очень маленькое, то w – очень большое.

Построить нечеткое отношение S из V в W .

Значения переменных *маленькое* и *большое* приведены ниже:

- а) *маленькое* = $1/1 + 0,8/2 + 0,1/3 + 0/4$;
большое = $0/1 + 0,2/2 + 0,8/3 + 1/4$;
 б) *маленькое* = $1/1 + 0,7/2 + 0,3/3 + 0/4$;
большое = $0/1 + 0,25/2 + 0,6/3 + 1/4$.

4. $U = V = W = \{1, 2, 3, 4, 5\}$.

Определены следующие правила:

если u – не маленькое, то v – среднее,
если v – большое, то w – очень маленькое,

Построить нечеткое отношение S из V в W .

Значения переменных *маленькое*, *среднее* и *большое* приведены ниже:

- а) *маленькое* = $1/1 + 0,8/2$, *среднее* = $0,3/2 + 1/3 + 0,5/4$;
большое = $0,15/3 + 0,5/4 + 1/5$;
 б) *маленькое* = $1/1 + 0,65/2$; *среднее* = $0,15/2 + 1/3 + 0,2/4$;
большое = $0,1/3 + 0,7/4 + 1/5$.

6.5. Контрольные вопросы

1. Дайте определение понятиям «нечеткое множество» и «нечеткое отношение».
2. Какова основная идея, лежащая в основе понятия «нечеткое множество»?
3. В чем отличие операций над традиционными и нечеткими множествами?
4. Как задаются нечеткие отношения?

Тема 7. Нечеткий композиционный вывод

Цель занятия – систематизация, углубление, закрепление полученных теоретических знаний о процедуре нечеткого композиционного вывода.

7.1. Описание нечеткого композиционного вывода

Традиционный дедуктивный вывод, называемый также *модус поненс*, записывается следующим образом:

$$\frac{P \Rightarrow Q}{P} Q$$

Что означает вывод Q из факта P по правилу $P \Rightarrow Q$.

Используя те же обозначения, можно определить нечеткий дедуктивный вывод следующим образом:

$$\frac{P \Rightarrow Q}{P'} Q'$$

Однако эта формулировка имеет существенное отличие от традиционного *модус поненс*. Здесь не требуется совпадения высказывания P' в факте и высказывания P в правиле. В общем случае могут не совпадать и заключения Q и Q' .

Л. Заде предложил нечеткий условный вывод в следующей форме:

$$\frac{\text{если } x \text{ есть } A, \text{ то } u \text{ есть } B, \text{ иначе } u \text{ есть } C}{x \text{ есть } A' \text{ и } u \text{ есть } D.}$$

Здесь x, u – имена объектов; A, A', B, C, D – нечеткие понятия, представленные нечеткими множествами, определенными на множествах U, U, V, V, V , соответственно.

Предложено несколько правил, переводящих нечеткое условное высказывание «если x есть A , то u есть B , иначе u есть C » в нечеткое отношение U

• V .

Пусть A, B, C – нечеткие множества в U, V, V , заданные в виде

$$A = \int_U \mu_A(u)/u; \quad B = \int_V \mu_B(u)/u; \quad C = \int_V \mu_C(u)/u.$$

Тогда имеем.

А. Максиминное правило Rm' :

$$Rm' = (A \times B) \cup (\sim A \times C) = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v)) \vee ((1 - \mu_A(u)) \wedge \mu_C(v)) / (u, v).$$

Б. Арифметическое правило Ra' :

$$\begin{aligned} Ra' &= (\sim A \times V + U \times B) \cap (A \times V + U \times C) = \\ &= \int_{U \times V} 1 \wedge (1 - \mu_A(u) + \mu_B(v)) \wedge ((\mu_A(u) + \mu_C(v)) / (u, v)). \end{aligned}$$

В. Размытое бинарное правило

$$\begin{aligned} Rb^e &= (\sim A \times V \cup U \times B) \cap (A \times V \cup U \times C) = \\ &= \int_{U \times V} (1 - \mu_A(u) \vee \mu_B(v)) \wedge ((\mu_A(u) \vee \mu_C(v)))/(u, v). \end{aligned}$$

Г. Правила Танака–Мидзумото–Фуками

$$\begin{aligned} RGG' &= (A \times V \xRightarrow{G} U \times B) \cap (\sim A \times V \xRightarrow{G} U \times C) = \\ &= \int_{U \times V} (\mu_A(u) \xrightarrow{g} \mu_B(v)) \wedge ((1 - \mu_A(u)) \xrightarrow{g} \mu_C(v))/(u, v), \end{aligned}$$

$$\text{где } \mu_A(u) \xrightarrow{g} \mu_B(v) = \begin{cases} 1, & \text{если } \mu_A \leq \mu_B; \\ \mu_B, & \text{если } \mu_A > \mu_B. \end{cases}$$

$$\begin{aligned} Rss' &= (A \times V \xRightarrow{s} U \times B) \cap (\sim A \times V \xRightarrow{s} U \times C) = \\ &= \int_{U \times V} (\mu_A(u) \xrightarrow{s} \mu_B(v)) \wedge ((1 - \mu_A(u)) \xrightarrow{s} \mu_C(v))/(u, v), \end{aligned}$$

$$\text{где } \mu_A(u) \xrightarrow{s} \mu_B(v) = \begin{cases} 1, & \text{если } \mu_A \leq \mu_B; \\ 0, & \text{если } \mu_A \geq \mu_B; \end{cases}$$

$$\begin{aligned} RsG' &= (A \times V \xRightarrow{s} U \times B) \cap (\sim A \times V \xRightarrow{G} U \times C) = \\ &= \int_{U \times V} (\mu_A(u) \xrightarrow{s} \mu_B(v)) \wedge ((1 - \mu_A(u)) \xrightarrow{g} \mu_C(v))/(u, v); \end{aligned}$$

$$\begin{aligned} RGs' &= (A \times V \xRightarrow{G} U \times B) \cap (\sim A \times V \xRightarrow{s} U \times C) = \\ &= \int_{U \times V} (\mu_A(u) \xrightarrow{g} \mu_B(v)) \wedge ((1 - \mu_A(u)) \xrightarrow{s} \mu_C(v))/(u, v). \end{aligned}$$

Таким образом, возвращаясь к исходной постановке задачи

если x есть A , то u есть B , иначе u есть C

$$\frac{x \text{ есть } A'}{y \text{ есть } D}$$

используя max-min композицию, следствие D можно вывести следующим образом:

$$Dm = A' \bullet Rm',$$

$$Da = A' \bullet Ra',$$

$$Db = A' \bullet Rb',$$

$$Dss = A' \bullet Rss',$$

$$DsG = A' \bullet Rsg',$$

$$DGs = A' \bullet RGs',$$

$$DGG = A' \bullet Rgg'.$$

7.2. Пример

Пусть имеются следующие посылки:

x – не очень маленькое;

если x – маленькое, то y – большое, иначе y – маленькое.

Найти значения y' .

Множество $U = 1+2+3$.

маленькое = $1/1 + 0,4/2$; большое = $0,5/2 + 1/3$.

Тогда термин *очень маленькое* = $1/1 + 0,16/2$, а *не очень маленькое* = $0,84/2 + 1/3$.

Отношение для максиминного правила

$$Rm' = (A \times B) \cup (\sim A \times C) = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v) \vee ((1 - \mu_A(u)) \wedge \mu_C(v)) / (u, v),$$

здесь A = маленькое, B = большое, C = маленькое.

Пример вычисления значений элементов матрицы Rm' приведены ниже:

$$\begin{aligned} (u_1, v_1) &= (1 \wedge 0) \vee (0 \wedge 1) = 0; \\ (u_2, v_1) &= (0,4 \wedge 0) \vee (0,6 \wedge 1) = 0,6; \\ (u_3, v_1) &= (0 \wedge 0) \vee (1 \wedge 1) = 1; \\ (u_1, v_2) &= (1 \wedge 0,5) \vee (0 \wedge 0,4) = 0,5; \\ (u_2, v_2) &= (0,4 \wedge 0,5) \vee (0,6 \wedge 0,4) = 0,4; \\ (u_3, v_2) &= (0 \wedge 0,5) \vee (1 \wedge 0,4) = 0,4; \\ (u_1, v_3) &= (1 \wedge 1) \vee (0 \wedge 0) = 1; \\ (u_2, v_3) &= (0,4 \wedge 1) \vee (0,6 \wedge 0) = 0,4; \\ (u_3, v_3) &= (0 \wedge 1) \vee (1 \wedge 0) = 0; \end{aligned}$$

$$Rm' = \begin{vmatrix} 0 & 0,5 & 1 \\ 0,6 & 0,4 & 0,4 \\ 1 & 0,4 & 0 \end{vmatrix}.$$

Тогда значение y' может быть определено следующим образом

$$y' = \text{не очень маленькое} \bullet Rm' = \begin{vmatrix} 0 & 0,5 & 1 \\ 0,6 & 0,4 & 0,4 \\ 1 & 0,4 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0,4 & 0,4 \end{vmatrix},$$

т.е. $y' = 1/1 + 0,4/2 + 0,4/3$, что может быть интерпретировано (с некоторой натяжкой) как *довольно таки маленькое*.

Далее рассмотрим для указанных выше посылок арифметическое правило Ra' :

$$Ra' = \begin{vmatrix} 0 & 0,5 & 1 \\ 0,6 & 0,8 & 0,4 \\ 1 & 0,4 & 0 \end{vmatrix},$$

тогда значение y' , используя арифметическое правило, может быть определено следующим образом

$$y' = \text{не очень маленькое} \bullet Ra' = \left| \begin{array}{ccc} 0 & 0,84 & 1 \\ 0,6 & 0,8 & 0,4 \\ 1 & 0,4 & 0 \end{array} \right| = \left| \begin{array}{ccc} 1 & 0,8 & 0,4 \end{array} \right|,$$

т.е. $y' = 1/1 + 0,8/2 + 0,4/3$.

Вывод с использованием размытого бинарного правила приведен ниже:

$$Rb' = \left| \begin{array}{ccc} 0 & 0,5 & 1 \\ 0,6 & 0,4 & 0,4 \\ 1 & 0,4 & 0 \end{array} \right|,$$

тогда значение y' может быть определено следующим образом

$$y' = \text{не очень маленькое} \bullet Rb' = \left| \begin{array}{ccc} 0 & 0,5 & 1 \\ 0,4 & 0,4 & 0,4 \\ 1 & 0,4 & 0 \end{array} \right| = \left| \begin{array}{ccc} 1 & 0,4 & 0,4 \end{array} \right|,$$

т.е. $y' = 1/1 + 0,4/2 + 0,4/3$, что может быть также интерпретировано как *довольно таки маленькое*.

Последний нечеткий вывод проведем с использованием правила Танака–Мидзумото

$$Rgg' = \left| \begin{array}{ccc} 0 & 0,5 & 1 \\ 0 & 0,4 & 0 \\ 1 & 0,4 & 0 \end{array} \right|,$$

$$y' = \text{не очень маленькое} \bullet Rgg' = \left| \begin{array}{ccc} 0 & 0,5 & 1 \\ 0 & 0,4 & 0 \\ 1 & 0,4 & 0 \end{array} \right| = \left| \begin{array}{ccc} 1 & 0,4 & 0 \end{array} \right|,$$

т.е. $y' = 1/1 + 0,4/2 + 0/3$, что интерпретируется как *маленькое*.

7.3. Аудиторные задания

Дано множество $U = 1 + 2 + 3 + 4$.

Значения переменных и посылки приведены в вариантах.

Найти значения y' , используя последовательно все четыре правила нечеткого вывода (максиминное, арифметическое, размытое бинарное и правило Танака–Мидзумото). Сравнить результаты, насколько сильно они отличаются от ожидаемых.

Вариант 1. Имеются следующие посылки:

если x – маленькое, то y – большое, иначе y – маленькое, x – не маленькое.

Определены следующие переменные

маленькое = $1/1 + 0,6/2 + 0,1/3$; *большое* = $0,2/2 + 0,7/3 + 1/4$.

Вариант 2. Имеются следующие посылки:

если x – маленькое, то y – среднее, иначе y – большое, x – не маленькое.

Значения переменных: *маленькое* = $1/1 + 0,6/2 + 0,1/3$; *среднее* = $0,1/1 + 0,4/2 + 0,8/3$; *большое* = $0,2/3 + 1/4$.

Вариант 3. Имеются следующие посылки:

если x – маленькое, то y – большое, иначе y – среднее, x – не очень маленькое.

Значения переменных: *маленькое* = $1/1 + 0,4/2$; *среднее* = $0,1/1 + 0,5/2 + 0,6/3$; *большое* = $0,5/3 + 1/4$.

7.4. Самостоятельная работа

Дано множество $U = 1 + 2 + 3 + 4$.

Значения переменных и посылки приведены в вариантах.

Найти значения y' , используя последовательно все четыре правила нечеткого вывода (максиминное, арифметическое, размытое бинарное и правило Танака–Мидзумото). Сравнить результаты, насколько сильно они отличаются от ожидаемых.

Вариант 1. Имеются следующие посылки:

если x – маленькое, то y – среднее, иначе y – маленькое, x – очень маленькое

Значения переменных: *маленькое* = $1/1 + 0,25/2$; *среднее* = $0,1/1 + 1/2 + 0,6/3$; *большое* = $0,4/3 + 1/4$.

Вариант 2. Имеются следующие посылки:

если x – большое, то y – большое, иначе y – среднее, x – не маленькое.

Значения переменных: *маленькое* = $1/1 + 0,25/2$; *среднее* = $0,1/1 + 1/2 + 0,6/3$; *большое* = $0,4/3 + 1/4$.

7.5. Контрольные вопросы

1. В чем заключается принципиальное отличие традиционного правила модуса поненс от обобщенного?

2. Как изменятся правила вывода, если в них будет отсутствовать часть «иначе y есть C »?

Тема 8. Разработка нечеткой системы

Цель занятия – систематизация, углубление, закрепление полученных теоретических знаний о механизмах вывода на нечетких правилах.

8.1. Нечеткая система

Рассмотрим формализацию для более сложных форм вывода, таких как

если x есть A_1 , то y есть B_1
 если x есть A_2 , то y есть B_2

 если x есть A_n , то y есть B_n

y есть B'

В этом случае простое применение композиционного правила невозможно. В этом контексте рассмотрим нечеткий вывод в нечетких системах.

Исследуемая предметная область может не иметь аналитического описания, однако эксперты могут описать взаимодействие объектов предметной области посредством лингвистических переменных и правил естественного языка, содержащих качественную оценку ситуации. Основой для описания ситуации является нечеткое высказывание следующего вида:

x_i есть X_i или $x_i = X_i$,

где x_i – некоторая величина, X_i – элемент терм-множества лингвистической переменной из исследуемой предметной области.

Нечеткая система выполняет отображение из входного пространства $A \subset \mathfrak{R}^m$ в выходное пространство $B \subset \mathfrak{R}^r$. Такая система является системой типа «много_входов – много_выходов» (MIMO – multiple_input-multiPle_output). Если система имеет m входов и R выходов и входное и выходное пространства являются многомерными, то входное пространство определяется как $A = A_1 \times \dots \times A_m$, а выходное пространство – как $B = B_1 \times \dots \times B_r$, где $A_i, B_j \subset \mathfrak{R}$. Обозначим $a = [a_1 a_2 \dots a_m]^T$ и $b = [b_1 b_2 \dots b_r]^T$ как входной и выходной векторы, соответственно. Отображение вход/выход может быть представлено как множество нечетких правил типа «ЕСЛИ–ТО». Каждое правило состоит из двух частей: условной и заключительной частей. Антецедент или условная часть (ЕСЛИ-часть) содержит утверждение относительно значений входных переменных, в консеквенте или заключительной части (ТО-части) указываются значения, которые принимают выходные переменные. Таким образом, нечеткая система типа «мно-

го_входов – много_выходов» может быть задана нечеткими правилами следующего вида:

Правило 1: ЕСЛИ $a_1 = A_{11}$ И $a_2 = A_{21} \dots a_m = A_{m1}$ ТО $b_1 = B_{11}$ И $b_2 = B_{21} \dots b_r = B_{r1}$;

Правило 2: ЕСЛИ $a_1 = A_{12}$ И $a_2 = A_{22} \dots a_m = A_{m2}$ ТО $b_1 = B_{12}$ И $b_2 = B_{22} \dots b_r = B_{r2}$.

.....
Правило n : ЕСЛИ $a_1 = A_{1n}$ И $a_2 = A_{2n} \dots a_m = A_{mn}$ ТО $b_1 = B_{1n}$ И $b_2 = B_{2n} \dots b_r = B_{rn}$.

Где a_1, a_2, \dots, a_m – входные переменные, b_1, b_2, \dots, b_r – выходные переменные, A_{ij} и B_{js} – нечеткие области определения входных и выходных переменных, соответственно. Каждая нечеткая область A_{ij} связана с функцией принадлежности $\mu_{A_{ij}}(a_i)$.

Вход A нечеткой системы активизирует каждое из правил, хранимых в нечеткой ассоциативной памяти. Чем больше вход A соответствует antecedенту i -го правила, тем больше выход соответствует консеквенту этого правила.

Весьма популярными в практическом применении в настоящее время являются нечеткие системы типа «много_входов – один_выход». Система такого типа выполняет отображение из входного пространства $A \subset \mathfrak{R}^m$ в выходное пространство $B \subset \mathfrak{R}$. Известно два основных типа нечетких систем «много_входов–один_выход». Системы типа Мамдани имеют правила:

Правило i : ЕСЛИ $a_1 = A_{1i}$ И $a_2 = A_{2i} \dots$ И $a_m = A_{mi}$ ТО $b_i = B_i$;

Другой тип – системы типа Сугено с правилами следующего вида:

Правило i : ЕСЛИ $a_1 = A_{1i}$ И $a_2 = A_{2i} \dots a_m = A_{mi}$ ТО $b_i = f_i(a_1, \dots, a_m)$;

где f_i – функция, определенная на переменных $a_1 \dots a_m$.

Для описания отображения входного вектора a в значение b используются методы нечеткой логики, например, аппроксимация Мамдани или метод, основанный на формальном логическое доказательстве. В процессе вывода участвуют операции конъюнкции и дизъюнкции. Задание этих операций на основе триангулярных норм позволяет более гибко настраивать нечеткую систему на исследуемую предметную область.

В настоящее время широкой популярностью пользуются системы типа Такаги–Сугено, которые имеют правила следующего вида:

Правило i : ЕСЛИ $a_1 = A_{1i}$ И $a_2 = A_{2i} \dots a_m = A_{mi}$ ТО $b = d_{i0} + d_{i1} \cdot a_1 + \dots + d_{im} \cdot a_m$, где выход правила b определяется значением линейной функции от входных переменных с вещественными коэффициентами d_{i1}, \dots, d_{im} при переменных a_1, \dots, a_m соответственно и свободным членом d_{i0} .

В общем случае процесс создания нечетких систем состоит из следующих шагов:

- 1) определение входных и выходных переменных системы;

- 2) задание функций принадлежности каждой переменной;
 - 3) определение нечетких правил;
 - 4) настройка параметров функций принадлежности и нечетких правил.
- Нечеткие системы имеют два основных недостатка:

1) правила, сформулированные экспертом или экспертами, могут оказаться неполными или противоречивыми;

2) функции принадлежности, которые собственно и определяют множества A_i и B_{js} , чаще всего задаются на основе здравого смысла, поэтому, в силу сложности исследуемой предметной области, могут не вполне адекватно отражать существующую действительность.

8.2. Нечеткий вывод

Рассмотрим нечеткую систему типа «два входа – один выход». Для осуществления вывода в такой системе можно воспользоваться композиционным правилом. Однако предварительно нужно выполнить операции конъюнкции «И» и далее операцию объединения (агрегации) n правил.

Пусть на вход системы поступают четкие значений x_1, x_2 . Требуется определить четкий выход y . Для этого необходимо выполнить следующие операции:

1) *фаззификация* – для каждого правила вычисляется значения $\mu_{A_i}(x_1)$ и $\mu_{A_{2i}}(x_2)$;

2) *конъюнкция* – объединение посылок в antecedенте каждого правила, используя t -нормальную функцию, получим $T(\mu_{A_i}(x_1), \mu_{A_{2i}}(x_2))$;

3) *импликация* – $I(T(\mu_{A_i}(x_1), \mu_{A_{2i}}(x_2)), \mu_{B_i}(y))$;

4) *агрегация* – получение нечеткого выходного значения из множества объединенных правил, т.е. выполнение операции «А_ТАКЖЕ» или определение итоговой функции принадлежности $\mu_B(y)$.

5) *дефаззификация* – преобразование итоговой функции принадлежности $\mu_B(y)$ в четкое значение y .

Структура нечеткой системы оценивания величин представлена на рис. 8.1.

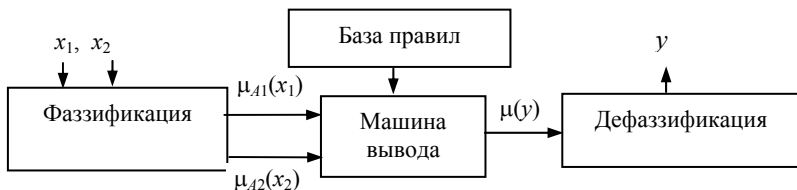


Рис. 8.1. Структура нечеткой системы

Представим нечеткое правило в базы правил в виде следующего отношения:

$$R_i: A_{1i} \cap A_{2i} \rightarrow B_i.$$

Объединив все правила $R = \bigcup_{i=1}^n R_i$, где $\bigcup_{i=1}^n$ – операция, соответствующая

союзу «А_ТАКЖЕ», представим нечеткую систему оценивания величин в виде следующего отношения:

$$R: A_1 \cap A_2 \rightarrow B,$$

где R – нечеткое отношение, определенное на $X_1 \times X_2 \times Y$.

Дефазификацию можно провести, воспользовавшись одним из следующих методов:

Метод центра тяжести

Центр тяжести или центр масс вычисляется по следующей формуле:

$$y = \frac{\int_{\min}^{\max} z \cdot \mu(z) dz}{\int_{\min}^{\max} \mu(z) dz},$$

где y – количественное значение, z – переменная, соответствующая выходной переменной B , изменяющаяся в интервале от \min до \max .

В *методе центра площади* (CenteR-of-ARea) или методе медиан значение y определяется из следующего уравнения:

$$\int_{\min}^y \mu(z) dz = \int_y^{\max} \mu(z) dz.$$

Метод среднего максимума определяется как среднее арифметическое между левым и правым максимальным значением функции принадлежности.

8.3. Аудиторные задания

8.3.1. Задание 1

Дана следующая база правил

1. Если x – маленькое, то y – маленькое.
2. Если x – среднее, то y – среднее.
3. Если x – большое, то y – большое.

Ниже приведены функции принадлежности для переменных x и y .

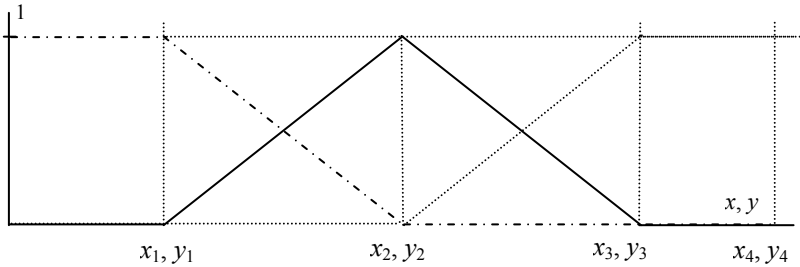


Рис. 8.2. Функции принадлежности

Определить y при заданных значениях $x_{\text{вх}}$.

Вариант 1. $x_1 = 8, y_1 = 5, x_2 = 16, y_2 = 10, x_3 = 24, y_3 = 15, x_4 = 32, y_4 = 20$;

а) $x_{\text{вх}} = 11$, б) $x_{\text{вх}} = 15$.

Вариант 2. $x_1 = 10, y_1 = 4, x_2 = 20, y_2 = 8, x_3 = 30, y_3 = 12, x_4 = 40, y_4 = 16$;

а) $x_{\text{вх}} = 15$, б) $x_{\text{вх}} = 37$.

Вариант 3. $x_1 = 4, y_1 = 5, x_2 = 8, y_2 = 10, x_3 = 12, y_3 = 15, x_4 = 16, y_4 = 20$;

а) $x_{\text{вх}} = 5$, б) $x_{\text{вх}} = 15$.

Вариант 4. $x_1 = 5, y_1 = 8, x_2 = 10, y_2 = 16, x_3 = 15, y_3 = 24, x_4 = 20, y_4 = 32$;

а) $x_{\text{вх}} = 8$, б) $x_{\text{вх}} = 16$.

Вариант 5. $x_1 = 16, y_1 = 25, x_2 = 32, y_2 = 50, x_3 = 48, y_3 = 75, x_4 = 64, y_4 = 100$;

а) $x_{\text{вх}} = 25$, б) $x_{\text{вх}} = 56$.

Вариант 6. $x_1 = 25, y_1 = 8, x_2 = 50, y_2 = 16, x_3 = 75, y_3 = 24, x_4 = 100, y_4 = 32$;

а) $x_{\text{вх}} = 30$, б) $x_{\text{вх}} = 85$.

8.3.2. Задание 2

Цель работы: Аппроксимация функции по экспериментальным данным в одномерном входном пространстве с помощью нечетких систем типа Такаги–Сугено.

Необходимые сведения для работы в пакете MathCad

Для создания нового проекта необходимо зайти в меню «File»–«New» и в открывшемся диалоговом окне с выбором типов проекта выбрать «Normal» и нажать «Ok». На экране появится рабочая область нового проекта. Эта область предназначена для ввода команд, рисования графиков и отображения полученных результатов.

Для того чтобы решать поставленные задачи MathCad предлагает набор инструментов, распределенных по типам на специальных панелях. Чтобы панели отображались в пакете нужно зайти в меню «View»–«Toolbars» и выбрать нужные. Понадобятся следующие: «Matrix», «Calculator», «Programming» и «Graph». После выбора панели отображаются в главном окне пакета (рис. 8.3).

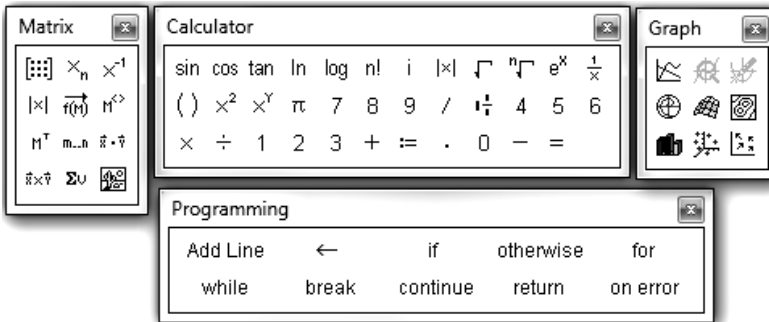


Рис. 8.3. Необходимый набор панелей инструментов

Панель «Matrix» содержит инструменты для работы с матричными вычислениями, «Calculator» содержит базовый набор математических операций, «Programming» содержит инструменты для написания функций с помощью высокоуровневых операций, таких как циклы и условия, а «Graph» – все необходимое для построения графиков и диаграмм.

MathCad позволяет работать с переменными скалярного и матричного типа. Для того чтобы переменной присвоить какое-либо значение, нужно в рабочей области ввести имя переменной, оператор «Definition» из панели «Calculator» и значение, которое получает переменная, причем это значение может быть тоже переменной. Чтобы узнать имя оператора на панели инструментов нужно навести на него курсор мыши. Чтобы получить значение переменной, нужно ввести имя переменной, а затем выбрать оператор «Evaluate Numerically» из панели «Calculator».

Для того чтобы переменной присвоить матричное значение, необходимо воспользоваться оператором «Matrix or Vector» из панели Matrix. Чтобы получить значения элемента матрицы, нужно использовать оператор «Subscript» из этой же панели. Далее все обозначения переменных будут представлены так, как они выглядят в рабочей области. На рис. 8.4 переменной x присвоено скалярное значение, переменной y – векторное, переменной z – матричное. Значение переменной f является суммой значений переменной x , первого элемента вектора y (нумерация элементов в векторах и матрицах начинается с 0) и элемента $z_{2,3}$.

В пакете MathCad существует возможность рисования графиков функций, а так же вывода экземпляров данных в виде точек декартовой системы координат. Продемонстрируем эту работу. Предположим, что нам даны векторы X и Y , каждый элемент которых хранит соответствующую координату экземпляра данных в двумерном пространстве. Для того чтобы отобразить множество экземпляров в декартовой системе координат, на рабочую область нужно поместить инструмент «X-Y Plot» из панели инструментов

«Graph». На полученной канве указываются верхние и нижние границы отображения по каждой переменной, а так же соответствующие каждой переменной векторы с данными. Чтобы данные отображались в виде точек, нужно воспользоваться контекстным меню и выбрать пункт «Format», в появившемся диалоговом окне перейти на вкладку «Traces», где определяется форматирование каждого графика на канве (рис. 8.5). Для первого графика установим параметр «Symbol» в значение точки, параметр «Line» оставим пустым, «Type» установим в значение «Points», параметром «Symbol Weight» можно регулировать размер точки на канве. Полученное графическое представление можно увидеть на рис. 8.6.

$$x := 9 \quad y := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad z := \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

$$f := x + y_0 + z_{1,2}$$

$$f = 18$$

Рис. 8.4. Пример матричных операций в пакете MathCad

Попробуем на этой же канве отобразить график функции $f(x) = x$. Сначала необходимо объявить функцию (рис. 8.7), а затем, в области ввода данных для каждой переменной после запятой для координаты X указать значение независимой переменной x , а для координаты Y значение функции $f(x)$.

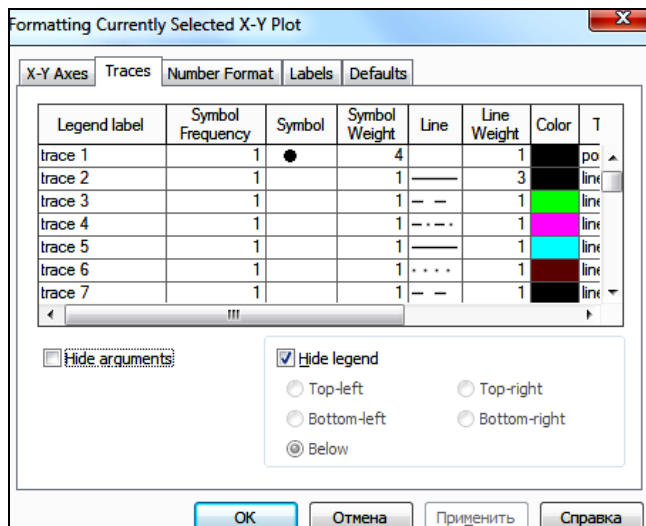


Рис. 8.5. Форматирование графиков

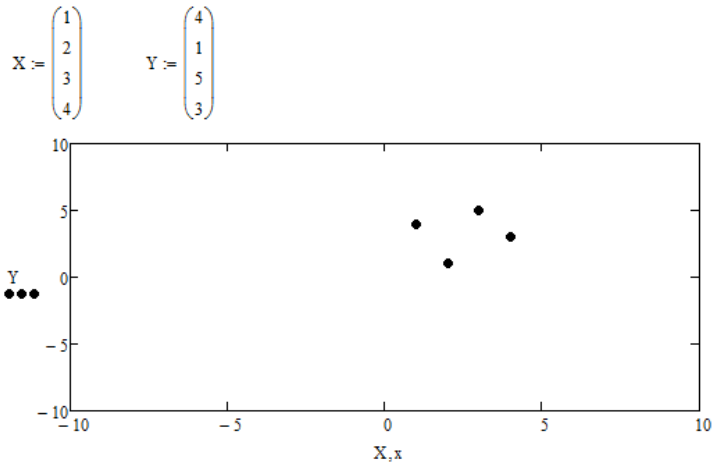
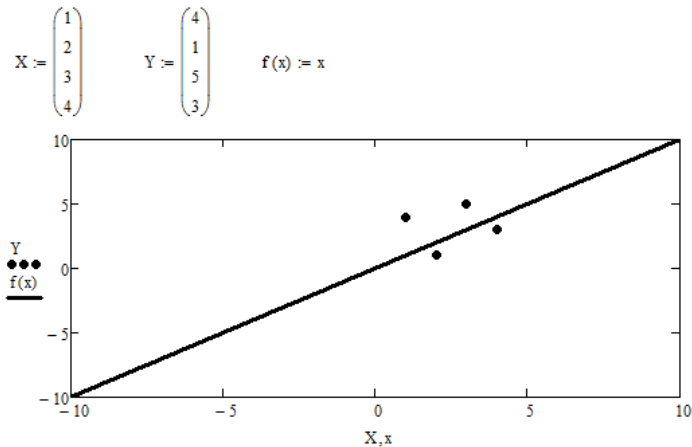


Рис. 8.6. Графическое представление экземпляров данных в MathCad

Параметр форматирования «Line» устанавливаем в значение непрерывной линии, «Type» в значение «lines». Полученное изображение графика показано на рис. 8.7.

Рис. 8.7. График функции $f(x) = x$ и экземпляры данных в пакете MathCad

Пакет MathCad предусматривает подключение к рабочей области библиотек с функциями, которые можно в дальнейшем использовать в своей работе. Для выполнения практических работ нам понадобятся пять библиотек с готовыми функциями. Архив с библиотеками можно взять у препода-

вателя. Библиотеки подключаются с помощью меню «Insert»-«References» и после выбора библиотеки, в рабочей области появляется команда «Reference имя_библиотеки». Подключать библиотеки следует в самом начале рабочей области и в том порядке, в котором они указаны ниже.

1. `utils.xmcd` – содержит вспомогательные функции, такие как импорт данных из файла, нормировка данных и т.д.

2. `TS.xmcd` – содержит функцию вывода нечеткой системы типа «Такаги-Сугено».

3. `KLI.xmcd` – содержит функцию, реализующую алгоритм генерации структуры нечетких систем типа Такаги-Сугено, называемый кусочно-линейной инициализацией (КЛИ).

4. `TS_GD.xmcd` – содержит функцию, реализующую метод градиентного спуска для оптимизации параметров части ЕСЛИ нечетких систем типа Такаги-Сугено.

5. `TS_MNK.xmcd` – содержит функцию, реализующую рекуррентный алгоритм наименьших квадратов (РАНК) для оптимизации параметров частей ТО нечетких систем типа Такаги-Сугено.

Пример 1. Пример аппроксимации функции с помощью нечетких систем в одномерном входном пространстве по экспериментальным данным. Экспериментальные данные находятся в файлах «tra.dat» и «tst.dat».

Подключение библиотек

Подключаем библиотеки для работы с функциями построения нечетких систем (смотрите описание выше).

Импорт данных из файла

Экспериментальные данные в файлах необходимо импортировать в переменные рабочей области. Для этого воспользуемся функциями `ReadInputData` и `ReadOutputData`, которые возвращают входные данные эксперимента и выходные соответственно. Параметром функций является имя файла с данными. Если файл не расположен в папке с файлом проекта, то указывается полный путь. Выход первой – матрица входных данных размерностью $n \times m$, где n – размерность входного пространства, m – число экземпляров данных. Выход второй – матрица $1 \times m$, где каждый элемент является выходом советуемого входа матрицы входных данных.

В переменные `Xtra` и `Ytra` импортируем входные и выходные данные обучения, а в `Xtst` и `Ytst` импортируем данные теста (рис. 8.8).

Генерация структуры нечеткой системы типа Такаги-Сугено

Этап генерации структуры предназначен для формирования структурных характеристик системы, таких как определения числа нечетких правил, выбор используемых переменных в правилах, выбор типа функций принадлежности и установка начальных значений параметров функций принадлежности и линейных функций выходов правил.


```

➤ Reference:D:\Test\utils.xmcd
➤ Reference:D:\Test\TS.xmcd
➤ Reference:D:\Test\KLI.xmcd
➤ Reference:D:\Test\TS_GD.xmcd
➤ Reference:D:\Test\TS_MNK.xmcd

Xtra := ReadInputData("tra.dat")
Ytra := ReadOutputData("tra.dat")
Xtst := ReadInputData("tst.dat")
Ytst := ReadOutputData("tst.dat")

```

Рис. 8.8. Импорт данных из файла в пакете MathCad

Для генерации структуры воспользуемся функцией `Init_KLI`, которая возвращает нечеткую систему, сформированную алгоритмом КЛИ. Входными параметрами этой функции являются: 1 – обучающие данные входов, 2 – обучающие данные выходов, 3 – порог алгоритма КЛИ, 4 – вариант алгоритма КЛИ (0 – обратный, 1 – прямой), 5 – Т-нормальная функция для операции конъюнкции (1 – \min , 2 – произведение). Используются функции принадлежности гауссового типа, которые имеют следующую формулу

$$\mu(x, s, \sigma) = e^{-\left(\frac{x-s}{\sigma}\right)^2},$$

где параметр s – среднее и σ – отклонение, x – элемент вектора входных значений.

Создадим в переменной P нечеткую систему, выполнив команду:

$$P := \text{Init_KLI}(Xtra, Ytra, 0,05, 0, 1),$$

где порог алгоритма выбран в значении $0,05$, вариант алгоритма – обратный, Т-нормальная функция – произведение.

Для того чтобы определить выход нечеткой системы для определенного входа можно воспользоваться функцией `TS`, которая имеет два параметра: 1 – матрица входов, 2 – нечеткая система. Например, команда `TS(Xtra, P)` возвращает выходы, соответствующие входам `Xtra` в виде строки, но уже сформированные нечеткой системой P . Здесь следует обратить внимание на то, что первый параметр является матрицей, и если в одномерном входном пространстве используется скаляр, то его необходимо преобразовать в матрицу размерностью 1×1 с помощью инструмента «Matrix or Vector» из панели `Matrix`. Например, для вывода графика вывода нечеткой системы P в

зависимости от переменной x необходимо применять функцию, указанную на рис. 8.9 для жирной линии. Независимая переменная здесь является матрицей 1×1 . Поскольку выходом функции TS является строка, а на графике нужен скаляр, поэтому функция берется с индексом 0.

```
P := Init_KLI(Xtra, Ytra, 0.05, 0, 1)
```

```
MSE(Ytra, TS(Xtra, P)) = 0.029
```

```
MSE(Ytst, TS(Xtst, P)) = 0.02
```

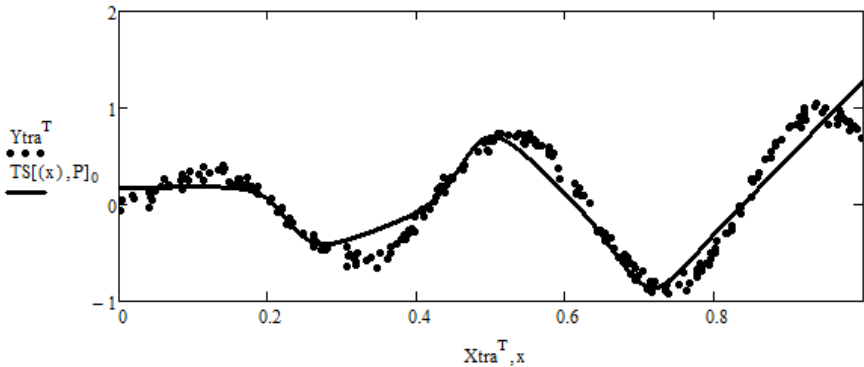


Рис. 8.9. Генерация структуры нечеткой системы в MathCad

На рис. 8.9 показан пример генерации структуры нечеткой системы, а также результаты оценок ее точности на тестовых и обучающих данных с помощью функции MSE. Входными параметрами функции являются выходы эксперимента и выходы нечеткой системы, а выходом среднеквадратичная ошибка между ними. На рисунке представлены обучающих данных в виде точек, а так же выход нечеткой системы в виде линии. Диапазон вывода ограничивался диапазоном обучающих данных на отрезке $[0, 1]$.

Оптимизация параметров нечеткой системы типа Такаги–Сугено

Этап оптимизации параметров предназначен для подбора параметров функций принадлежности нечеткой системы и выходов правил таким образом, чтобы среднеквадратичная ошибка аппроксимации между выходами обучающих данных и нечеткой системы стремилась к минимуму:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \rightarrow \min ,$$

где m – число обучающих данных, y_i – выход i -го экземпляра обучающих данных, \hat{y}_i – выход нечеткой системы для входа i -го экземпляра обучающих данных.

Оптимизация параметров осуществляется функциями OPT_TS_GD и OPT_TS_MNK. Обе функции ведут подбор параметров нечеткой системы таким образом, чтобы среднеквадратичная ошибка аппроксимации устремлялась к минимуму. Причем первая оптимизирует параметры функций принадлежности частей ЕСЛИ с помощью алгоритма градиентного спуска, а вторая – параметры линейных функций частей ТО с помощью РАНК. OPT_TS_GD имеет следующие входные параметры: нечеткая система, матрица входов, строка выходов, число итераций. OPT_TS_MNK в качестве параметров использует матрицу входов, строку выходов и нечеткую систему. Обе функции возвращают нечеткую систему с обновленными параметрами.

Оптимизируем сформированную алгоритмом КЛИ нечеткую систему, последовательно вызвав функции оптимизации, и посмотрим, происходит ли уменьшение среднеквадратичной ошибки. На рис. 8.10 показаны данные действия.

```
P := OPT_TS_GD(P, Xtra, Ytra, 100)
MSE(Ytra, TS(Xtra, P)) = 0.027
P := OPT_TS_MNK(Xtra, Ytra, P)
MSE(Ytra, TS(Xtra, P)) = 0.019
```

Рис. 8.10. Оптимизация параметров в пакете MathCad

Для достижения высокой точности необходимо итерационно оптимизировать параметры нечеткой системы. Для этого удобно сформировать функцию, в которой последовательно вызываются OPT_TS_GD и OPT_TS_MNK заданное число итераций. На вход подаются нечеткая система, входные и выходные данные, а так же число итераций. Для добавления в функцию управляющих элементов, таких как новая строка и цикл типа «for» воспользуемся инструментами «Add Line» и «For» из панели «Programming». На рис. 8.11 представлены ее описание, вызов с 20-ю итерациями, и результаты точности аппроксимации после оптимизации. Вывод полученной нечеткой системы приведен на графике.

До оптимизации параметров точности аппроксимации на обучающих и тестовых данных были равны 0,029 и 0,02 соответственно. После оптимизации 0,0042 и 0,003. Функции оптимизации повысили точность нечеткой системы P .

Описание базы правил нечеткой системы.

Попробуем описать базу правил полученной нечеткой системы. Сначала необходимо узнать число нечетких правил, для этого вызовем функцию GetCounRules(P). Затем выясним, чему равны параметры функции принадлежности i -го правила j -й переменной с помощью функции GetParamsAnt(P, i, j). Эта функция возвращает строку из двух элементов, первый – среднее

гаусовой функции принадлежности, второй – отклонение. Далее определяем параметры линейных функций части ТО i -го правила с помощью вызова `GetParamsKons(P, i)`, который возвращает вектор параметров, где первый элемент – свободный член, второй – коэффициент при первой входной переменной, второй элемент – коэффициент при второй входной переменной и т.д. Полученные результаты показаны на рис. 8.12.

```
OptParams(P,X,Y,n) :=  $\left\{ \begin{array}{l} \text{for } i \in 1..n \\ \quad \left\{ \begin{array}{l} P \leftarrow \text{OPT\_TS\_GD}(P,X,Y,100) \\ P \leftarrow \text{OPT\_TS\_MNX}(X,Y,P) \end{array} \right. \\ \text{return } P \end{array} \right.$ 
```

```
P := OptParams(P,Xtra,Ytra,20)
```

```
MSE(Ytra,TS(Xtra,P)) = 0.0042
```

```
MSE(Ytst,TS(Xtst,P)) = 0.003
```

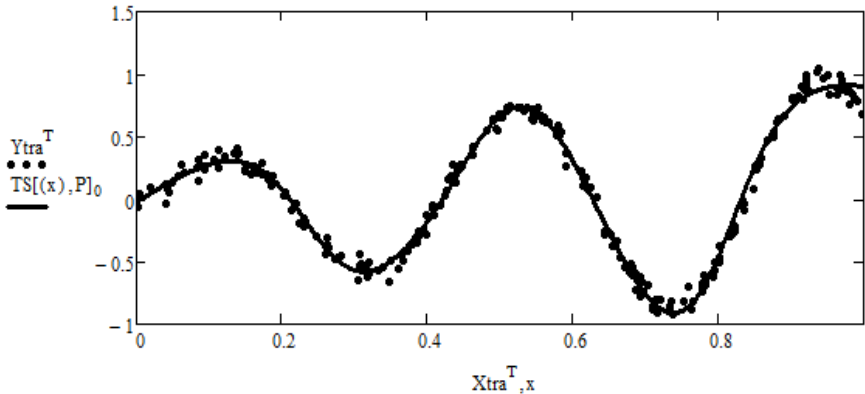


Рис. 8.11. Оптимизация параметров нечеткой системы в пакете MathCad

На основании полученных данных можно описать базу нечеткой системы:

Правило 1: ЕСЛИ x есть $\mu(x; 1,016; 0,233)$ ТО $y = -1,052x + 1,978$.

Правило 2: ЕСЛИ x есть $\mu(x; 0,634; 0,153)$ ТО $y = -14,759x + 9,132$.

Правило 3: ЕСЛИ x есть $\mu(x; 0,385; 0,173)$ ТО $y = -4,023x - 2,138$.

Правило 4: ЕСЛИ x есть $\mu(x; 0,101; 0,14)$ ТО $y = 4,073x + 0,001$.

Функции принадлежности можно представить графически, для этого воспользуемся функцией `FP1(x, Params)`, где `Params` – строка параметров функции принадлежности. Реализация в MathCad показана на рис. 8.13.

```

GetCounRules(P) = 4
GetParams.Ant(P, 1, 1) = (1.016 0.233)
GetParams.Ant(P, 2, 1) = (0.634 0.153)
GetParams.Ant(P, 3, 1) = (0.385 0.173)
GetParams.Ant(P, 4, 1) = (0.101 0.14)

GetParams.Kons(P, 1) =  $\begin{pmatrix} 1.978 \\ -1.052 \end{pmatrix}$ 

GetParams.Kons(P, 2) =  $\begin{pmatrix} 9.132 \\ -14.759 \end{pmatrix}$ 

GetParams.Kons(P, 3) =  $\begin{pmatrix} -2.138 \\ 4.023 \end{pmatrix}$ 

GetParams.Kons(P, 4) =  $\begin{pmatrix} 0.001 \\ 4.073 \end{pmatrix}$ 

```

Рис 8.12. Определение параметров нечеткой системы в пакете MathCad

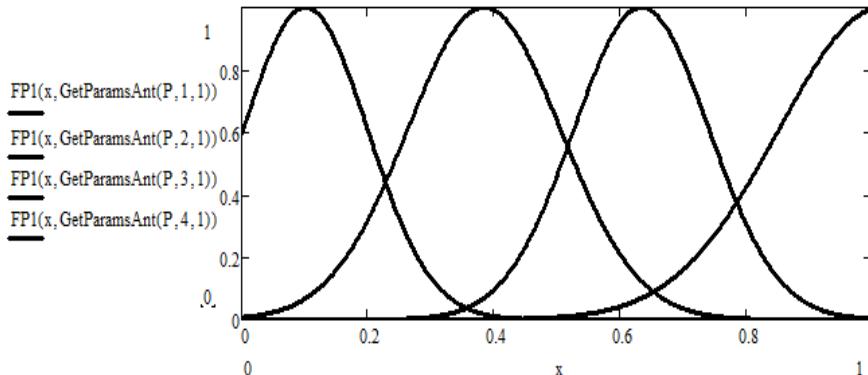


Рис. 8.13. Графическое представление функций принадлежности нечеткой системы в пакете MathCad

Выполнить следующие задачи

1. Согласно своему варианту получить файлы с обучающими и тестовыми данными. Провести импорт данных.
2. Построить нечеткую систему типа Такаги–Сугено с помощью алгоритма КЛИ на обучающих данных и вывести график выхода системы и точек данных в диапазоне $[0, 1]$. Порог алгоритма подобрать из диапазона

[0,02; 1] таким образом, чтобы выход системы лучшим образом описывал поведение данных при минимальном наборе нечетких правил. Оценить точность на обучающих и на тестовых данных.

3. Провести оптимизацию параметров нечеткой системы итерационно с помощью функций OPT_TS_GD и OPT_TS_MNK. Подобрать число итераций, которое достаточно для оптимизации. Оценить полученную точность системы на обучающих и тестовых данных после оптимизации и сравнить с начальной. Построить график вывода нечеткой системы в диапазоне [0, 1], отобразить на графике обучающие данные.

4. Определить параметры нечеткой системы: число правил, параметры частей ЕСЛИ и ТО и описать базу нечетких правил. Построить графики функций принадлежности.

8.3.3. Задание 3

Цель работы: Построение зависимостей входных данных от выходных по реальным экспериментальным наборам данных в многомерном входном пространстве с помощью нечетких систем типа Такаги-Сугено.

Пример 2. Определить по экспериментальным данным зависимость расхода топлива автомобиля в милях на галлон от таких параметров как ускорение, масса, мощность двигателя, год выпуска модели и объем топливного бака.

Создадим новый проект и выполним следующие задачи.

Подключение библиотек

Подключаем библиотеки для работы с функциями построения нечетких систем.

Импорт данных из файла и нормировка входных данных

Произвести импорт обучающих и тестовых данных из файлов autoMPG6-5-1tra.dat и autoMPG6-5-1tst.dat соответственно (см. пример 1).

Обычно при работе с многомерными пространствами входные данные принято нормировать, т.е. масштабировать их значения в диапазон [0, 1] по каждой входной переменной. Это делается с целью приведения всех данных к единой шкале, чтобы уравнивать влияние каждой переменной на результат. Нормирование входных обучающих данных производится с помощью функции NormMatr. На вход подается матрица обучающих входных данных, а на выходе вектор из двух элементов: первый элемент возвращает нормированную входную матрицу, второй – матрицу с коэффициентами для последующего нормирования новых данных.

На рис. 8.14 показан импорт данных и их нормирование. В переменной Kf хранятся коэффициенты нормирования, которые используются для нормирования тестовых данных, не участвующих в обучении. Нормирование тестовых данных производится с помощью функции NormM, которая в ка-

чества параметров использует матрицу тестовых входов и коэффициенты. Обучающие и тестовые нормированные входы расположены в переменных Xtra и Xtst, выходы в Ytra и Ytst.

```
Xtra := NormMatr(ReadInputData("autoMPG6-5-1tra.dat"))0
Ytra := ReadOutputData("autoMPG6-5-1tra.dat")
Kf := NormMatr(ReadInputData("autoMPG6-5-1tra.dat"))1
Xtst := NormM(ReadInputData("autoMPG6-5-1tst.dat"), Kf)
Ytst := ReadOutputData("autoMPG6-5-1tst.dat")
```

Рис. 8.14. Импорт данных и их нормирование в пакете MathCad

Генерация структуры нечеткой системы типа Такаги–Сугено

Выполним генерацию структуры аналогично предыдущему примеру. Порог алгоритма КЛИ эмпирически подберем таким образом, чтобы число правил не превышало 5, а точность была как можно лучше. При пороге равном 7 число правил соответствует значению 5 (рис. 8.15).

```
P := Init_KLI(Xtra, Ytra, 7, 0, 1)
GetCounRules(P) = 5
MSE(Ytra, TS(Xtra, P)) = 7.9322
MSE(Ytst, TS(Xtst, P)) = 8.0767
```

Рис. 8.15. Генерация структуры нечеткой системы в пакете MathCad

Оптимизация параметров нечеткой системы типа Такаги–Сугено

Для оптимизации параметров воспользуемся функцией OptParams, созданной в примере 1. Зададим число итераций равное 20 (рис. 8.16).

```
P := OptParams(P, Xtra, Ytra, 20)
MSE(Ytra, TS(Xtra, P)) = 5.7717
MSE(Ytst, TS(Xtst, P)) = 5.2758
```

Рис. 8.16. Оптимизация параметров нечеткой системы в пакете MathCad

Как видно из рисунка с помощью оптимизации параметров удалось повысить точность.

Выполнить следующие задачи

1. Согласно своему варианту получить файлы с тестовыми и обучающими данными. В данной работе будет использовано пять вариантов обучаю-

щих данных и пять соответствующих им вариантов тестовых данных. Данные относятся к одной предметной области.

2. Провести импорт и нормирование данных. Нормирование тестовых данных следует проводить с учетом коэффициентов, найденных с помощью соответствующих обучающих данных.

3. Построить пять нечетких систему типа Такаги-Сугено с помощью алгоритма КЛИ для каждого файла с обучающими данными. Порог алгоритма необходимо подобрать таким образом, чтобы число правил не превышало 5. Оценить значение точности аппроксимации систем на данных, с помощью которых они построены, а также на соответствующих тестовых данных.

4. Провести оптимизацию параметров систем, построенных выше и снова оценить значение точностей на обучающих и тестовых данных. Рассчитать средние оценки точностей по пяти системам.

5. Определить параметры первой нечеткой системы: число правил, параметры частей ЕСЛИ и ТО и описать базу нечетких правил.

8.4. Самостоятельная работа

Дана следующая база правил

1. Если x – маленькое, то y – большое.
2. Если x – среднее, то y – среднее.
3. Если x – большое, то y – маленькое.

На рис. 8.17 приведены функции принадлежности для переменных x и y .

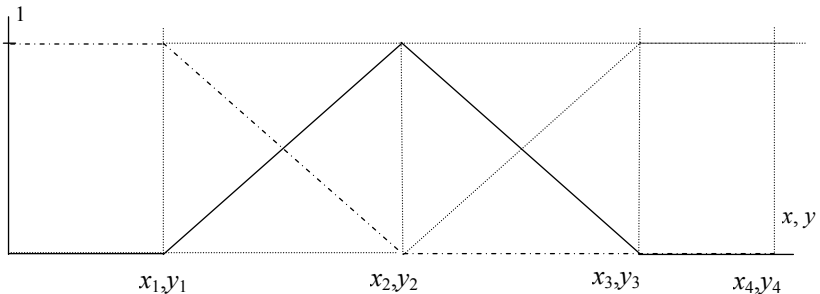


Рис. 8.17. Функции принадлежности

Определить y при заданных значениях $x_{\text{вх}}$ между двумя ансамблями.

Вариант 1. $x_1 = 8, y_1 = 5, x_2 = 16, y_2 = 10, x_3 = 24, y_3 = 15, x_4 = 32, y_4 = 20,$

- а) $x_{\text{вх}} = 11,$ б) $x_{\text{вх}} = 15.$

Вариант 2. $x_1 = 10, y_1 = 4, x_2 = 20, y_2 = 8, x_3 = 30, y_3 = 12, x_4 = 40, y_4 = 16,$

а) $x_{\text{BX}} = 15,$ б) $x_{\text{BX}} = 37.$

Вариант 3. $x_1 = 4, y_1 = 5, x_2 = 8, y_2 = 10, x_3 = 12, y_3 = 15, x_4 = 16, y_4 = 20,$

а) $x_{\text{BX}} = 5,$ б) $x_{\text{BX}} = 15.$

Вариант 4. $x_1 = 5, y_1 = 8, x_2 = 10, y_2 = 16, x_3 = 15, y_3 = 24, x_4 = 20, y_4 = 32,$

а) $x_{\text{BX}} = 8,$ б) $x_{\text{BX}} = 16.$

Вариант 5. $x_1 = 16, y_1 = 25, x_2 = 32, y_2 = 50, x_3 = 48, y_3 = 75, x_4 = 64, y_4 = 100,$

а) $x_{\text{BX}} = 25,$ б) $x_{\text{BX}} = 56.$

Вариант 6. $x_1 = 25, y_1 = 8, x_2 = 50, y_2 = 16, x_3 = 75, y_3 = 24, x_4 = 100, y_4 = 32,$

а) $x_{\text{BX}} = 30,$ б) $x_{\text{BX}} = 85.$

8.5. Контрольные вопросы

1. Что называется нечеткой системой?
2. Какие операции выполняет нечеткая система?

Литература

1. Ходашинский И.А. Методы искусственного интеллекта, базы знаний, экспертные системы: учебное пособие. – Томск: Томск. гос. университет систем управления и радиоэлектроники, 2002. – 139 с.
2. Ходашинский И.А. Методы мягкого оценивания величин. – Томск: Томск. гос. университет систем управления и радиоэлектроники, 2007. – 152 с.
3. Ясницкий Л.Н. Введение в искусственный интеллект: учебное пособие для студ. высш. учеб. заведений. – М.: Изд. центр «Академия», 2005. – 176 с.
4. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2003. – 736 с.
5. Представление и использование знаний / Под ред. Х. Уэно, М. Исудзука. – М.: Мир, 1989. – 220 с.
6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.
7. Корнеев В.В., Гарев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2000. – 352 с.
8. Лорьер Ж.-Л. Системы искусственного интеллекта. – М.: Мир, 1991. – 568 с.
9. Нечеткие множества в моделях управления и искусственного интеллекта. – М.: Наука, 1986. – 312 с.
10. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987. – 288 с.
11. Построение экспертных систем: пер. с англ. / Под ред. Ф. Хейеса-Рота, Д. Уотермана, Д. Лената. – М.: Мир, 1987. – 441 с.

Для заметок

Учебное издание

*Илья Александрович Ходашинский,
Константин Сергеевич Сарин,
Александр Евгеньевич Анфилофьев*

Базы знаний

*Методические указания для выполнения
практических и самостоятельных работ*

для студентов специальностей

10.05.03 – «Информационная безопасность
автоматизированных систем»,

10.05.04 – «Информационно-аналитические системы безопасности»

Верстка – В.М. Бочкаревой

Текст дан в авторской редакции, без корректуры

Издательство «В-Спектр»

Подписано к печати 30.11.2017.

Формат 60×84¹/₁₆. Печать трафаретная.

Печ. л. 4,25. Тираж 100 экз. Заказ 34.

Тираж отпечатан ИП Бочкаревой В.М.

ИНН 701701817754

634055, г. Томск, пр. Академический, 13-24, тел. 49-09-91.

E-mail: bvm@sibmail.com