

• РАДИО И СВЯЗЬ •

СПРАВОЧНИК

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

КНИГА

2

**МОДЕЛИ
И МЕТОДЫ**

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

2

СПРАВОЧНИК

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

В ТРЕХ КНИГАХ

1

**СИСТЕМЫ ОБЩЕНИЯ
И ЭКСПЕРТНЫЕ СИСТЕМЫ**

2

МОДЕЛИ И МЕТОДЫ

3

**ПРОГРАММНЫЕ
И АППАРАТНЫЕ СРЕДСТВА**

СПРАВОЧНИК

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

КНИГА

2

МОДЕЛИ И МЕТОДЫ

**ПОД РЕДАКЦИЕЙ
ПРОФЕССОРА
Д.А. ПОСПЕЛОВА**



**МОСКВА „РАДИО И СВЯЗЬ”
1990**

ББК 32.81
И86
УДК 681.3→007

Рецензент докт. техн. наук О. П. Кузнецов

Редакция литературы по информатике и вычислительной технике

Искусственный интеллект. — В 3-х кн. Кн. 2. Модели и
И86 методы: Справочник/ Под ред. Д. А. Поспелова — М.: Радио
и связь, 1990.—304 с.: ил.

ISBN 5-256-00368-2 (Кн. 2).

Дается общая характеристика данной проблемной области, вводятся основные понятия и определения, приводится классификация моделей и методов. Рассматриваются отдельные направления теории интеллектуальных систем: представление знаний, логический вывод на знаниях, языки общения, понимание текстов, модели обучения, планирование действий и обработка зрительной информации.

Для специалистов в области управления, информационных систем и вычислительной техники, использующих методы искусственного интеллекта.

И 1402070000-100
046(01)-90 28-90

ББК 32.81

Справочное издание

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ
В 3-х кн.

Книга 2

МОДЕЛИ И МЕТОДЫ

Под редакцией Д. А. Поспелова

Справочник

Заведующая редакцией Г. И. Козырева Редактор Н. Г. Давыдова Переплет
художника Н. А. Пашуро Художественный редактор Н. С. Шейн Технический
редактор Т. Н. Зыкина Корректор Т. В. Демидович

ИБ № 2230

Сдано в набор 27.12.89 Подписано в печать 3.04.90 Т-06096 Формат 60×88¹/₁₆
Бумага офсетная № 2 Гарнитура литературная Печать офсетная Усл. печ. л. 18,62
Усл. кр.-отт. 18,62 Уч.-изд. л. 28,67 Тираж 25 000 экз. Изд. № 22828 Зак. № 6943 Цена 1 р. 80 к.

Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693

Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая
Образцовая типография» Государственного комитета СССР по печати. 113054, Москва,
Валовая, 28.

ISBN 5-256-00368-2 (Кн. 2)
ISBN 5-256-00756-4

© Коллектив авторов, 1990

Предисловие

В настоящее время в исследованиях по искусственному интеллекту выделились шесть основных направлений.

1. Представление знаний. В рамках этого направления решаются задачи, связанные с формализацией и представлением знаний в памяти интеллектуальной системы (ИС). Для этого разрабатываются специальные модели представления знаний и языки для описания знаний, выделяются различные типы знаний. Изучаются источники, из которых ИС может черпать знания, и создаются процедуры и приемы, с помощью которых возможно приобретение знаний для ИС. Проблема представления знаний для ИС чрезвычайно актуальна, так как ИС — это система, функционирование которой опирается на знания о проблемной области, которые хранятся в ее памяти.

2. Манипулирование знаниями. Для того чтобы знаниями можно было пользоваться при решении задач, надо научить ИС оперировать ими. В рамках данного направления строятся способы пополнения знаний на основе их неполных описаний, изучаются системы классификации хранящихся в ИС знаний, разрабатываются процедуры обобщения знаний и формирования на их основе абстрактных понятий, создаются методы достоверного и правдоподобного вывода на основе имеющихся знаний, предлагаются модели рассуждений, опирающихся на знания и имитирующих особенности человеческих рассуждений. Манипулирование знаниями очень тесно связано с представлением знаний. Многие исследователи считают, что эти два направления можно разделить лишь условно. Создающаяся в настоящее время теория баз знаний включает исследования, относящиеся как к первому, так и ко второму направлениям.

3. Общение. В круг задач этого направления входят: проблема понимания связанных текстов на ограниченном и неограниченном естественном языке, синтез связанных текстов, понимание речи и синтез речи, теория моделей коммуникации между человеком и ИС. К этому же кругу проблем примыкают задачи формирования объяснений действий ИС, которые она должна уметь порождать по просьбе человека, а также комплекс задач, связанных с интеграцией в единый внутренний образ сообщений различной модальности (речевых, текстовых, зрительных и т. п.), полученных в процессе коммуникации. На основе исследований в этом направлении формируются методы построения лингвистических процессоров, вопросно-ответных систем, диалоговых систем и других ИС, целью которых является обеспечение комфортных условий для общения человека с ИС.

4. Восприятие. Это направление традиционно включает: проблемы анализа трехмерных сцен, разработку методов представления информации о зрительных образах в базе знаний, создание методов перехода от зрительных сцен к их текстовому описанию и методов обратного перехода, разработку процедур когнитивной графики, создание средств для порождения зрительных сцен на основе внутренних представлений в ИС. Специалисты в области искусственного интеллекта видят большие возможности в повышении уровня интеллектуальности ИС за счет обработки зрительной (образной) информации и соотнесения ее с обработкой символично: (текстовой) информации.

5. Обучение. Предполагается, что ИС подобно человеку будут способны к обучению — решению задач, с которыми они ранее не встречались. Для того чтобы это стало возможным, необходимо: создать методы формирования условий задачи по описанию проблемной ситуации или по наблюдению за этой ситуацией, научиться переходу от известного решения частных задач (примеров) к решению общей задачи, создать приемы декомпозиции исходной для ИС задачи на более мелкие так, чтобы они оказались для ИС уже известными, разработать нормативные и декларативные модели самого процесса обучения, создать теорию подражательного поведения. Перечень задач можно было бы продолжить, ибо в области обучения ИС сделано еще весьма мало. Однако важность работ в этом направлении не вызывает никаких сомнений.

6. Поведение. Так как ИС должны действовать в некоторой окружающей среде, то необходимо разработать специальные поведенческие процедуры, которые позволили бы им адекватно взаимодействовать с окружающей средой, другими ИС и людьми. Для достижения такого взаимодействия надо провести исследование в ряде направлений и создать: модели целесообразного поведения, нормативного поведения, ситуативного поведения, специальные методы многоуровневого планирования и коррекции планов в динамических ситуациях. Лишь после этого можно будет говорить о возможности привычного взаимодействия между людьми и ИС.

Практически все перечисленные направления искусственного интеллекта рассмотрены в настоящем томе справочника. Полнота изложения результатов, касающихся того или иного направления, не одинакова. Это объясняется разным уровнем исследований в них. Данный том отражает тот взгляд на исследование в области искусственного интеллекта, который характерен для специалистов, разрабатывающих теоретические модели и методы решения разнообразных задач, возникающих при создании ИС. Этим он отличается от первого тома, в котором проблемы искусственного интеллекта рассматривались с точки зрения пользователей ИС.

Как и остальные тома справочника, второй том подготовлен коллективом специалистов, который объединяется вокруг Научного Совета по проблеме «Искусственный интеллект» АН СССР. В общую редколлегия справочника входят: академик Г. С. Поспелов, д-р техн. наук, проф. Э. В. Попов, д-р техн. наук, проф. Д. А. Поспелов, канд. техн. наук, доц. В. Н. Захаров, канд. техн. наук, доц. В. Ф. Хорошевский. При подготовке данной книги большую помощь оказали члены рабочей группы: канд. техн. наук М. Г. Гаазе-Рапопорт, канд. физ.-мат. наук С. М. Ефимова, канд. физ.-мат. наук Л. В. Литвинцева, Н. В. Руссова, Н. В. Чудова.

Глава 1

Представление знаний

1.1. Данные и знания

Д. А. Поспелов

Основные определения

Информация, с которой имеют дело ЭВМ, разделяется на *процедурную* и *декларативную*. Процедурная информация овеществлена в *программах*, которые выполняются в процессе решения задач, декларативная информация — в *данных*, с которыми эти программы работают. Стандартной формой представления информации в ЭВМ является *машинное слово*, состоящее из определенного для данного типа ЭВМ числа двоичных разрядов — *битов*. Машинное слово для представления данных и машинное слово для представления команд, образующих программу, могут иметь одинаковое или разное число разрядов. В последнее время для представления данных и команд используются одинаковые по числу разрядов машинные слова. Однако в ряде случаев машинные слова разбиваются на группы по восемь двоичных разрядов, которые называются *байтами*.

Одинаковое число разрядов в машинных словах для команд и данных позволяет рассматривать их в ЭВМ в качестве одинаковых информационных единиц и выполнять операции над командами, как над данными. Содержимое памяти образует *информационную базу*.

В большинстве существующих ЭВМ возможно извлечение информации из любого подмножества разрядов машинного слова вплоть до одного бита. Во многих ЭВМ можно соединять два (или более) машинных слова в слово с большей длиной. Однако машинное слово является основной характеристикой информационной базы, так как его длина такова, что каждое машинное слово хранится в одной стандартной ячейке памяти, снабженной индивидуальным именем — адресом ячейки. По этому имени происходит извлечение информационных единиц из памяти ЭВМ и запись их в нее.

Параллельно с развитием структуры ЭВМ происходило развитие информационных структур для представления данных. Появились способы описания данных в виде векторов и матриц, возникли списочные структуры, нерархические структуры. В настоящее время в языках программирования высокого уровня используются *абстрактные типы данных*, структура которых задается программистом. Появление *баз данных* (БД) знаменовало собой еще один шаг на пути организации работы с декларативной информацией. В базах данных могут одновременно храниться большие объемы информации, а специальные средства, образующие *систему управления базами данных* (СУБД), позволяют эффективно манипулировать с данными, при необходимости извлекать их из базы данных и записывать в нужном порядке в базу.

Таблица 1.1

Фамилия	Год рождения	Специальность	Стаж, число лет
Попов	1965	Слесарь	5
Сидоров	1946	Токарь	20
Иванов	1925	»	30
Петров	1937	Сантехник	25

По мере развития исследований в области ИС возникла концепция *знаний*, которые объединили в себе многие черты процедурной и декларативной информации.

Особенности знаний

Перечислим ряд особенностей, присущих этой форме представления информации в ЭВМ.

1. *Внутренняя интерпретируемость.* Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее, а также отвечает на запросы, в которых это имя упомянуто. Когда данные, хранящиеся в памяти, были лишены имен, то отсутствовала возможность их идентификации системой. Данные могла идентифицировать лишь программа, извлекающая их из памяти по указанию программиста, написавшего программу. Что скрывается за тем или иным двоичным кодом машинного слова, системе было неизвестно.

Если, например, в память ЭВМ нужно было записать сведения о сотрудниках учреждения, представленные табл. 1.1, то без внутренней интерпретации в память ЭВМ была бы занесена совокупность из четырех машинных слов, соответствующих строкам этой таблицы. При этом информация о том, какими группами двоичных разрядов в этих машинных словах закодированы сведения о специалистах, у системы отсутствуют. Они известны лишь программисту, который использует данные табл. 1.1 для решения возникающих у него задач. Система не в состоянии ответить на вопросы типа «Что тебе известно о Петрове?» или «Есть ли среди специалистов сантехник?».

При переходе к знаниям в память ЭВМ вводится информация о некоторой *протоструктуре информационных единиц*. В рассматриваемом примере она представляет собой специальное машинное слово, в котором указано, в каких разрядах хранятся сведения о фамилиях, годах рождения, специальностях и стаже. При этом должны быть заданы специальные словари, в которых перечислены имеющиеся в памяти системы фамилии, года рождения, названия специальностей и продолжительности стажа. Все эти *атрибуты* могут играть роль имен для тех машинных слов, которые соответствуют строкам таблицы. По ним можно осуществлять поиск нужной информации. Каждая строка таблицы будет экземпляром протоструктуры. В настоящее время СУБД обеспечивают реализацию внутренней интерпретируемости всех информационных единиц, хранящихся в базе данных.

2. *Структурированность.* Информационные единицы должны обладать гибкой структурой. Для них должен выполняться «принцип матрешки», т. е. рекурсивная вложенность одних информационных единиц в другие. Каждая информационная единица может быть включена в состав любой другой, и из каждой информационной единицы можно выделить некоторые составляющие ее информационные единицы. Другими словами, должна существовать возможность произвольного установления между отдельными информационными единицами отношений типа «часть — целое», «род — вид» или «элемент — класс».

3. *Связность.* В информационной базе между информационными единицами должна быть предусмотрена возможность установления связей различного ти-

па. Прежде всего эти связи могут характеризовать отношения между информационными единицами. Семантика отношений может носить декларативный или процедурный характер. Например, две или более информационные единицы могут быть связаны отношением «одновременно», две информационные единицы — отношением «причина — следствие» или отношением «быть рядом». Приведенные отношения характеризуют декларативные знания. Если между двумя информационными единицами установлено отношение «аргумент — функция», то оно характеризует процедурное знание, связанное с вычислением определенных функций. Далее будем различать *отношения структуризации, функциональные отношения, каузальные отношения и семантические отношения*. С помощью первых задаются иерархии информационных единиц, вторые несут процедурную информацию, позволяющую находить (вычислять) одни информационные единицы через другие, третьи задают причинно-следственные связи, четвертые соответствуют всем остальным отношениям.

Между информационными единицами могут устанавливаться и иные связи, например, определяющие порядок выбора информационных единиц из памяти или указывающие на то, что две информационные единицы несовместимы друг с другом в одном описании.

Перечисленные три особенности знаний позволяют ввести общую модель представления знаний, которую можно назвать *семантической сетью*, представляющей собой иерархическую сеть, в вершинах которой находятся информационные единицы. Эти единицы снабжены индивидуальными именами. Дуги семантической сети соответствуют различным связям между информационными единицами. При этом иерархические связи определяются отношениями структуризации, а неиерархические связи — отношениями иных типов [Hendrix, 1975; Schubert, 1975].

4. *Семантическая метрика*. На множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т. е. силу ассоциативной связи между информационными единицами. Его можно было бы назвать *отношением релевантности* для информационных единиц. Такое отношение дает возможность выделять в информационной базе некоторые типовые ситуации (например, «покупка», «регулирование движения на перекрестке»). Отношение релевантности при работе с информационными единицами позволяет находить знания, близкие к уже найденным.

5. *Активность*. С момента появления ЭВМ и разделения используемых в ней информационных единиц на данные и команды создалась ситуация, при которой данные пассивны, а команды активны. Все процессы, протекающие в ЭВМ, инициируются командами, а данные используются этими командами лишь в случае необходимости. Для ИС эта ситуация неприемлема. Как и у человека, в ИС актуализации тех или иных действий способствуют знания, имеющиеся в системе. Таким образом, выполнение программ в ИС должно инициироваться текущим состоянием информационной базы. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы.

Перечисленные пять особенностей информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в *базы знаний* (БЗ). Совокупность средств, обеспечивающих работу с знаниями, образует *систему управления базой знаний* (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы внутренняя интерпретируемость, структуризация, связность, введена семантическая мера и обеспечена активность знаний.

Модели представления данных *

В 70-х годах различали три основные модели представления данных: *реляционные, сетевые и иерархические*. В настоящее время появилось второе по-

* В написании данного раздела участвовала С. М. Ефимова.

коление таких моделей, в рамках которых происходит постепенное слияние данных и знаний [Цикритис и др., 1985]. В развитых моделях представления данных можно выделить два компонента: *интенциональные представления* и *экстенциональные представления*. Оба компонента хранятся в базе данных. При этом в ее экстенциональную часть входят конкретные факты, касающиеся предметной области (например, строки табл. 1.1), а в интенциональную часть — схемы связей между атрибутами (например, между именами столбцов табл. 1.1.). Таким образом, экстенциональные представления описывают конкретные объекты из предметной области, конкретные события, происходящие в ней, или конкретные явления и процессы, а интенциональные представления фиксируют те закономерности и связи, которым все эти конкретные объекты, события, явления или процессы обязаны в данной проблемной области удовлетворять. Экстенциональные представления относятся к данным. Относительно интенциональных представлений единого мнения нет. Апологеты баз данных говорят в этом случае о *схемах базы данных*, а представители искусственного интеллекта — о *знаниях о проблемной области*.

Элементами схемы базы данных являются схемы вида $Q(A_1, A_2, \dots, A_n)$, где Q — некоторое отношение, а множество элементов, входящих в Q , — множество атрибутов. В зависимости от того, какие отношения Q допускаются в схемах баз данных, возникают различные представления данных.

Введем понятие *домена* D_i для некоторого атрибута A_i . Элементами D_i будем считать все конкретные факты в экстенциональном представлении, соответствующие этому атрибуту. Например, в простейшем случае табл. 1.1 домены образуются из элементов столбцов таблицы, а соответствующий каждому домену атрибут показан как обозначение столбца. Домены, относящиеся к различным атрибутам, могут иметь пустое пересечение, непустое пересечение или включаться друг в друга. В предельном случае домены двух различных атрибутов могут полностью совпадать. При этом атрибуты оказываются синонимами. Например, если все токари на данном предприятии (и только они) имеют по трое детей, то атрибуты «Тожари» и «Отцы, имеющие троих детей» в данной БД окажутся синонимичными.

Экстенциональное отношение — это определенным образом выделенное подмножество декартова произведения доменов, относящихся к некоторому набору атрибутов. Например, содержимое табл. 1.1 есть подмножество декартова произведения доменов, соответствующих атрибутам: «Фамилия», «Год рождения», «Специальность» и «Стаж». Из всех возможных кортежей, полученных в этом декартовом произведении, в таблицу отобраны лишь те, которые отражают реальных людей, работающих на предприятии. В схеме баз данных такому экстенциональному отношению будет соответствовать *интенциональное отношение* $R(A_1, A_2, \dots, A_m)$. Имя этого отношения совпадает с именем экстенционального отношения (именем той таблицы, которым оно задано, например R может выглядеть как «Список сотрудников данного предприятия»), а в качестве его аргументов выступают атрибуты, домены которых использовались для образования соответствующего экстенционального отношения.

Описанная модель представления данных, т. е. табличное представление данных, протоструктура которых определяется экстенциональными отношениями, а схема базы данных — интенциональными отношениями [Codd, 1979, 1981], характерна для реляционных баз данных.

В сетевых моделях представления данных используются табличные и графовые представления. Вершинам графа соответствуют информационные единицы, называемые *записями*, которые представляют собой специальным образом организованные таблицы, а дугам графа — типы отношений между записями [Мартин, 1980; Ульман, 1983; Цикритис и др., 1985]. В отличие от реляционных моделей, в которых запрещается повторение одинаковых строк в множестве, определяющем экстенциональное отношение, в сетевых моделях такие повторения допустимы. Другим отличием является явное задание отношений между записями с помощью графа, а не опосредованное, как в реляционной модели.

В иерархических моделях представления данных в отличие от реляционных и сетевых моделей схема базы данных представляет собой множество схем отношений, упорядоченных в деревообразную структуру. В отличие от сетевой модели, в которой экстенциональные отношения могут быть определены на доменах, элементы которых сами могут быть агрегированными записями (т. е. представлять собой целые фрагменты графа), в иерархических моделях (как и в реляционных) элементами доменов могут быть лишь простые факты [Дейт, 1980; Мартин, 1980; Ульман, 1983].

В базах данных выделяют две составляющие: *язык описания данных* (ЯОД) и *язык манипулирования данными* (ЯМД). Средства ЯОД ориентированы, с одной стороны, на то, как на физическом уровне в ЭВМ будут представляться данные, интенциональные и экстенциональные отношения, а с другой стороны, на семантику проблемной области, так как в них входят операции по классификации, обобщению и агрегированию экстенциональных и интенциональных представлений. Эти средства позволяют вводить обобщенные атрибуты и записи, устанавливать новые схемы отношений на интенциональном уровне и строить многоуровневые иерархии на множествах обобщенных атрибутов. Таким образом, ЯОД позволяет реализовать в базах данных такие особенности знаний, которые ранее назывались структурированностью и связностью [Цикритис и др., 1985].

При работе с базами данных, т. е. при поиске ответов на запросы пользователей, пополнении содержимого экстенционального или интенционального компонентов, необходимо уметь выполнять ряд операций, которые входят в ЯМД. Различия в наборе операций определяют различия в ЯМД.

Многие авторы, например [Дейт, 1980; Дрибас, 1982; Калиниченко, 1983], определяют базы данных как совокупность ЯОД и ЯМД, что сблизжает их с понятиями алгебры и универсальной алгебраической системы [Мальцев, 1970]. На сегодняшний день можно считать завершенной теорию реляционных баз данных [Maier, 1983; Цаленко, 1985]. Сейчас уже ясно, что все три типа баз данных равномогущи [Jacoby, 1982; Филиппов, 1983, 1985].

Второе поколение баз данных характеризуется рядом особенностей моделей представления данных [Цаленко, 1985; Цикритис и др., 1985]. Расширение возможностей таких моделей происходит из-за ослабления требований к виду отношений Q в экстенциональных представлениях и к виду отношений R в интенциональных представлениях.

Семантические сети являются наиболее общей моделью представления знаний, так как в них имеются средства для выполнения всех пяти требований, предъявляемых к знаниям. Но такая универсальность семантических сетей имеет и негативную сторону. Если допускать в них произвольные типы отношений и связей, не являющиеся отношениями в математическом смысле (например, ассоциативные связи), то сложность работы с таким образом организованной информацией резко возрастает. Поэтому в базах данных второго поколения вводятся ограничения на характер структур и типов информационных единиц, находящихся в вершинах семантической сети, и на характер связей, задаваемых ее дугами.

Например, в *бинарных моделях* представления данных используются лишь бинарные отношения между информационными единицами. Близкой к таким моделям является модель, опирающаяся на язык синтагматических цепей, применявшийся в ситуационном управлении [Поспелов Д., 1981]. Наиболее известной бинарной моделью является модель Сенко [Senko, 1980].

В модели *П-графов* [Ефимова, 1985; Ефимова и др., 1986] фиксирована топология семантической сети, используемой в схеме базы данных: отношения вида $Q(A_0; A_1, \dots, A_m)$ описывают фрагменты семантической сети, называемые *звездами*. Тело звезды — вершина семантической сети с именем A_0 . Вершины с именами A_1, \dots, A_m связаны с вершиной A_0 бинарными отношениями с некоторыми именами R_1, \dots, R_m . Такой способ представления информации имеет некоторые аналоги с моделью RX -кодов, предлагавшейся для информационно-поисковых систем [Скороходько, 1968].

Модели представления знаний

В ИС используются различные способы описания знаний.

1. *Логические модели.* В основе моделей этого типа лежит *формальная система*, задаваемая четверкой вида $M = \langle T, P, A, B \rangle$. Множество T есть *множество базовых элементов* различной природы, например слов из некоторого ограниченного словаря, деталей детского конструктора, входящих в состав некоторого набора и т. п. Важно, что для множества T существует некоторый способ определения принадлежности или непринадлежности произвольного элемента к этому множеству. Процедура такой проверки может быть любой, но за конечное число шагов она должна давать положительный или отрицательный ответ на вопрос, является ли x элементом множества T . Обозначим эту процедуру $\Pi(T)$.

Множество P есть *множество синтаксических правил*. С их помощью из элементов T образуют *синтаксически правильные совокупности*. Например, из слов ограниченного словаря строятся синтаксически правильные фразы, из деталей детского конструктора с помощью гаек и болтов собираются новые конструкции. Декларируется существование процедуры $\Pi(P)$, с помощью которой за конечное число шагов можно получить ответ на вопрос, является ли совокупность X синтаксически правильной.

В множестве синтаксически правильных совокупностей выделяется некоторое подмножество A . Элементы A называются *аксиомами*. Как и для других составляющих формальной системы, должна существовать процедура $\Pi(A)$, с помощью которой для любой синтаксически правильной совокупности можно получить ответ на вопрос о принадлежности ее к множеству A .

Множество B есть *множество правил вывода*. Применяя их к элементам A , можно получать новые синтаксически правильные совокупности, к которым снова можно применять правила из B . Так формируется *множество выводимых* в данной формальной системе *совокупностей*. Если имеется процедура $\Pi(B)$, с помощью которой можно определить для любой синтаксически правильной совокупности, является ли она выводимой, то соответствующая формальная система называется *разрешимой*. Это показывает, что именно правила вывода являются наиболее сложной составляющей формальной системы.

Для знаний, входящих в базу знаний, можно считать, что множество A образуют все информационные единицы, которые введены в базу знаний извне, а с помощью правил вывода из них выводятся новые *производные знания*. Другими словами, формальная система представляет собой генератор порождения новых знаний, образующих множество *выводимых* в данной системе *знаний*. Это свойство логических моделей делает их притягательными для использования в базах знаний. Оно позволяет хранить в базе лишь те знания, которые образуют множество A , а все остальные знания получать из них по правилам вывода.

2. *Сетевые модели.* В основе моделей этого типа лежит конструкция, названная ранее семантической сетью. Сетевые модели формально можно задать в виде $H = \langle I, C_1, C_2, \dots, C_n, \Gamma \rangle$. Здесь I есть множество информационных единиц; C_1, C_2, \dots, C_n — множество типов связей между информационными единицами. Отображение Γ задает между информационными единицами, входящими в I , связи из заданного набора типов связей.

В зависимости от типов связей, используемых в модели, различают *классифицирующие сети*, *функциональные сети* и *сценарии*. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют *вычислительными моделями*, так как они позволяют описывать процедуры «вычислений» одних информационных единиц через другие. В сценариях используются каузальные отношения, а также отношения типов «средство — результат», «орудие — действие» и т. п. Если в сетевой модели допускаются связи различного типа, то ее обычно называют семантической сетью.

3. *Продукционные модели.* В моделях этого типа используются некоторые элементы логических и сетевых моделей. Из логических моделей заимствована идея правил вывода, которые здесь называются *продукциями*, а из сетевых моделей — описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, в продукционных моделях появляется *вывод на знаниях*.

4. *Фреймовые модели.* В отличие от моделей других типов во фреймовых моделях фиксируется жесткая структура информационных единиц, которая называется *протофреймом*. В общем виде она выглядит следующим образом:

(Имя фрейма:

Имя слота 1 (значение слота 1)
Имя слота 2 (значение слота 2);
Имя слота K (значение слота K)).

Значением *слота* может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов). В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать «принцип матрешки».

При конкретизации фрейма ему и слотам присваиваются конкретные имена и происходит заполнение слотов. Таким образом, из протофреймов получаются *фреймы-экземпляры*. Переход от исходного протофрейма к фрейму-экземпляру может быть многошаговым, за счет постепенного уточнения значений слотов.

Например, структура табл. 1.1, записанная в виде протофрейма, имеет вид (Список работников:

Фамилия (значение слота 1);
Год рождения (значение слота 2);
Специальность (значение слота 3);
Стаж (значение слота 4)).

Если в качестве значений слотов использовать данные табл. 1.1, то получится фрейм-экземпляр

(Список работников:

Фамилия (Попов—Сидоров—Иванов—Петров);
Год рождения (1965—1946—1925—1937);
Специальность (Слесарь—токарь—токарь—сантехник);
Стаж (5—20—30—25)).

Связи между фреймами задаются значениями специального слота с именем «Связь». Часть специалистов по ИС считает, что нет необходимости специально выделять фреймовые модели в представлении знаний, так как в них объединены все основные особенности моделей остальных типов. В справочнике фреймовые модели будут рассматриваться в общем контексте с сетевыми.

1.2. Логические модели

Г. С. Плесневич

Основные определения

Постановка и решение любой задачи всегда связаны с ее «погружением» в подходящую предметную область. Так, решая задачу составления расписания обработки деталей на металлорежущих станках, мы вовлекаем в предметную область такие объекты, как конкретные станки, детали, интервалы времени, и общие понятия «станок», «деталь», «тип станка» и т. п. Все предметы и события, которые составляют основу общего понимания необходимой для решения задачи информации, называются *предметной областью*. Мысленно предметная область представляется состоящей из реальных или абстрактных объектов, называемых *сущностями*.

Сущности предметной области находятся в определенных *отношениях* друг к другу (ассоциации), которые также можно рассматривать как сущности и включать в предметную область. Между сущностями наблюдаются различные отношения подобия. Совокупность подобных сущностей составляет *класс сущностей*, являющийся новой сущностью предметной области.

Отношения между сущностями выражаются с помощью суждений. *Суждение* — это мысленно возможная ситуация, которая может иметь место для предъявляемых сущностей или не иметь места. В языке (формальном или естественном) суждениям отвечают *предложения*. Суждения и предложения также можно рассматривать как сущности и включать в предметную область.

Языки, предназначенные для описания предметных областей, называются *языками представления знаний*. Универсальным языком представления знаний является естественный язык. Однако использование естественного языка в системах машинного представления знаний наталкивается на большие трудности ввиду присущих ему нерегулярностей, двусмысленностей, presuppositions и т. п. Но главное препятствие заключается в отсутствии формальной семантики естественного языка, которая имела бы достаточно эффективную операционную поддержку.

Для представления математического знания в математической логике давно пользуются логическими формализмами — главным образом *исчислением предикатов*, которое имеет ясную формальную семантику и операционную поддержку в том смысле, что для него разработаны механизмы вывода. Поэтому исчисление предикатов было первым логическим языком, который применили для формального описания предметных областей, связанных с решением прикладных задач.

Описания предметных областей, выполненные в логических языках, называются (формальными) *логическими моделями*.

Рассмотрим логический язык, называемый *многосортовым исчислением предикатов первой ступени* (или многосортовой логикой первого порядка). Приведем пример описания в этом языке предметной области, связанной с обработкой деталей на станках.

В логических моделях, основанных на исчислении предикатов, классам сущностей предметной области отвечают так называемые *сорта* (или типы). Точнее, в логических моделях участвуют имена сортов, которые интерпретируются как классы сущностей. Для примера введем такие имена сортов: Деталь, Станок, Операция, Время, ТипДетали, ТипСтанка, а также следующие функции и предикаты:

дет:Операция → Деталь, ст:Операция → Станок (1)

нач:Операция → Время, кон:Операция → Время (2)

тип_дет:Деталь → ТипДетали, тип_ст:Станок → ТипСтанка (3)

ток: → ТипСтанка, фрез → ТипСтанка (4)

вал шест: \rightarrow ТипДетали, ст-вал: \rightarrow ТипДетали (5)

0: \rightarrow Время, 1: \rightarrow Время, ..., t : \rightarrow Время. (6)

+ :Время \times Время \rightarrow Время (7)

\leq :Время \times Время $\rightarrow T$ (8)

фрез-торц:Операция $\rightarrow T$, ток-обр:Операция $\rightarrow T$ (9)

Выражения (1) — (9), составляющие так называемую сигнатуру, имеют следующий смысл:

(1) задает две функции дет (деталь) и ст (станок), которые, будучи применены к объекту e сорта Операция, дают деталь дет(e) и станок ст(e), участвующие в операции e ;

(2) задает функции начала и конца операции e , причем нач(e) и кон(e) есть объекты сорта Время;

(3) задает функции, значениями которых служат типы деталей и типы станков;

(4) задает две константы: ток (токарный станок) и фрез (фрезерный станок) сорта ТипСтанка (эти константы рассматриваются как нуль-местные функции);

(5) задает константы вал-шест (вал-шестерня) и ст-вал (ступенчатый вал) сорта ТипДетали;

(6) задает счетное множество констант, обозначающих натуральные числа (эти константы принадлежат сорту Время);

(7) задает двуместную функцию, аргументами которой служат объекты сорта Время (предполагаемая интерпретация этой функции — сложение натуральных чисел, измеряющих моменты времени);

(8) задает двуместный предикат на объектах сорта Время (предполагаемая интерпретация предиката — двуместное отношение «меньше или равно» между натуральными числами);

(9) задает два одноместных предиката фрез-торц (фрезерование торцов) и ток-обр (токарная обработка); эти предикаты представляют свойства операций: фрез-торц(e) = 1 тогда и только тогда, когда e есть операция фрезерования торцов (и аналогично для ток-обр).

В общем случае *сигнатурой* называется множество Σ выражений вида $f:A_1 \times A_2 \times \dots \times A_n \rightarrow B$, где A_i , B — сорта, а f — функция, или *предикат*. Имя f есть имя *предиката* в B , если $B = T$, т. е. если значением f служит «истина» или «ложь» (обычно обозначаемые 1 и 0).

Сорта сигнатуры интерпретируются как множества. Сорт T является особым в том смысле, что он всегда интерпретируется как множество истинностных значений $\{0, 1\}$. Остальные сорта сигнатуры могут интерпретироваться по-разному. Таким образом, предикат рассматривается как функция, заданная на объектах (вообще говоря, разных сортов) и принимающая значение «истина» или «ложь». Предикат можно рассматривать также как многоместное отношение между объектами (разных) сортов: объекты e_1, e_2, \dots, e_n находятся в отношении r , что записывается как $r(e_1, e_2, \dots, e_n)$ тогда и только тогда, когда $r(e_1, e_2, \dots, e_n) = 1$. Заметим, что для двуместных предикатов часто используется инфиксная запись: вместо $r(e_1, e_2)$ пишут $e_1 r e_2$ (например, пишут $x \leq y$ для предиката «меньше или равно»).

Сигнатура задает структурные связи между понятиями предметной области, представленными предикатами и функциями. Логические связи между этими понятиями задаются формулами, которые записываются в сигнатуре. Структурные и логические связи выражают некоторое знание о предметной области. Таким образом, сигнатура формально представляет одну часть знания о предметной области, а формулы, записанные в этой сигнатуре, представляют другую часть знания.

Рассмотрим, например, как формально представляется в языке многосортного исчисления предикатов следующее знание:

Зн1 — каждая деталь должна сначала пройти фрезерование торцов, а затем токарную обработку;

Зн2 — станки не должны все вместе простаивать ни в какой момент времени до того момента, когда все станки прекратят работу;

Зн3 — токарная обработка ступенчатого вала длится 7 мин, а вала-шестерни — 12 мин; фрезерование торцов ступенчатого вала длится 5 мин, а вала-шестерни — 6 мин;

Зн4 — на производственном участке имеется три станка: два токарных и один фрезерный;

Зн5 — всего было совершено восемь операций;

Зн6 — было обработано четыре детали: два ступенчатых вала и два вала-шестерни;

Зн7 — станок ст3 является фрезерным;

Зн8 — операция опер3 совершалась над деталью дет2 на станке ст3 в течение интервала времени от момента 5 до момента 10.

Для того чтобы формально представить Зн1, перефразируем его следующим образом: для каждой детали x существуют операции y и z , такие, что конец операции y наступает не позже начала операции z , операция y есть фрезерование торцов, операция z — токарная обработка и в обеих операциях участвует одна и та же деталь x . Соответствующая формула будет такой:

$$(\forall x \in \text{Деталь}) (\exists y, z \in \text{Операция}). [(\text{кон}(y) \leq \text{нач}(z)) \wedge (\text{дет}(y) = x) \wedge (\text{дет}(z) = x) \wedge \text{фрез_торц}(y) \wedge \text{ток_обр}(z)] \quad (10)$$

В (10) использованы следующие формальные обозначения:

$(\forall x \in A)$ — *квантор общности* („для всех x из A “); $(\exists x \in A)$ — *квантор существования* („существует x из A “); \wedge — *конъюнкция* („ α и β “). Вместо блока кванторов $(\forall x_1 \in A) (\forall x_2 \in A) \dots (\forall x_p \in A)$ сокращенно можно писать $(\forall x_1, x_2, \dots, x_p \in A)$; аналогично для блока кванторов \exists .

Перефразируем знание Зн2: существует момент времени x , такой, что: а) существует операция y , заканчивающаяся в момент x , б) каждая операция z заканчивается не позже момента x , в) не существует момента u , такого, что $u < x$, и какова бы ни была операция v , интервал ее исполнения не содержит u (т. е. неверно, что $\text{нач}(v) \leq u$ и $u \leq \text{кон}(v)$). Соответствующая формула

$$(\exists x \in \text{Время}). (\exists y \in \text{Операция}) (\text{кон}(y) = x) \wedge (\forall z \in \text{Операция}) (\text{кон}(z) \leq x) \wedge \neg (\exists u \in \text{Время}). (u < x) \wedge (\forall v \in \text{Операция}) \neg [(\text{нач}(v) \leq u) \wedge (u \leq \text{кон}(v))] \quad (11)$$

(Здесь точка выполняет роль скобок, отделяя соответствующую подформулу, стоящую в ряду слева от точки.)

В (11) использован символ \neg , обозначающий *логическое отрицание* ($\neg \alpha$ читается: «не верно, что α »).

Перефразируем знание Зн3: для любой операции x , если x есть токарная обработка и тип детали, участвующей в операции x , есть ступенчатый вал, то $\text{нач}(x) + 7 = \text{кон}(x)$ и т. д. Соответствующая формула

$$(\forall x \in \text{Операция}). [\text{ток_обр}(x) \wedge (\text{тип_дет}(\text{дет}(x)) = \text{ст_вал}) \rightarrow (\text{нач}(x) + 7 = \text{кон}(x))] \wedge [\text{ток_обр}(x) \wedge (\text{тип_дет}(\text{дет}(x)) = \text{вал_шест}) \rightarrow (\text{нач}(x) + 12 = \text{кон}(x))] \wedge [\text{фрез_торц}(x) \wedge (\text{тип_дет}(\text{дет}(x)) = \text{ст_вал}) \rightarrow (\text{нач}(x) + 5 = \text{кон}(x))] \wedge [\text{фрез_торц}(x) \vee (\text{тип_дет}(\text{дет}(x)) = \text{вал_шест}) \rightarrow (\text{нач}(x) + 6 = \text{кон}(x))] \quad (12)$$

Для того чтобы формально представить знание Зн4, введем три константы ст1, ст2 и ст3 сорта Станок. Получим

$$(\forall x \in \text{Станок}). [(x = \text{ст1}) \vee (x = \text{ст2}) \vee (x = \text{ст3})] \wedge (\text{тип_ст}(\text{ст1}) = \text{ток}) \vee (\text{тип_ст}(\text{ст2}) = \text{ток}) \wedge (\text{тип_ст}(\text{ст3}) = \text{фрез}) \quad (13)$$

В (12) и (13) появились два новых логических символа — \vee и \rightarrow , называемых *дизъюнкция* и *импликация* ($\alpha \vee \beta$ читается: « α или β »; $\alpha \rightarrow \beta$ читается: «если α , то β »).

Знание Зн5 формализуется аналогично, если ввести константы, обозначающие операции:

$$(\forall x \in \text{Операция}). (x = \text{опер1}) \vee (x = \text{опер2}) \vee (x = \text{опер3}) \vee (x = \text{опер4}) \vee (x = \text{опер5}) \vee (x = \text{опер6}) \vee (x = \text{опер7}) \vee (x = \text{опер8}) \quad (14)$$

Знание Зн6 представляется формулой

$$(\forall x \in \text{Деталь}) [(x = \text{дет1}) \vee (x = \text{дет2}) \vee (x = \text{дет3}) \vee (x = \text{дет4})] \wedge (\text{тип_дет}(\text{дет1}) = \text{ст_вал}) \wedge (\text{тип_дет}(\text{дет2}) = \text{ст_вал}) \wedge (\text{тип_дет}(\text{дет3}) = \text{вал_шест}) \wedge (\text{тип_дет}(\text{дет4}) = \text{вал_шест}) \quad (15)$$

Знание Зн7 передается формулой

$$(\text{тип_ст}(\text{ст3}) = \text{фрез}) \quad (16)$$

Наконец, знание Зн8 передается формулой

$$(\text{дет}(\text{опер3}) = \text{дет2}) \wedge (\text{ст}(\text{опер3}) = \text{ст3}) \wedge (\text{нач}(\text{опер3}) = 5) \wedge (\text{кон}(\text{опер3}) = 10) \quad (17)$$

Рассмотрим общий случай, когда задана произвольная сигнатура Σ . Через \mathcal{L}_Σ обозначим множество всех формул языка многосортного исчисления предикатов (первого порядка), построенных на основе сигнатуры Σ . Произвольная формула \mathcal{L}^Σ составляется из атомарных формул (атомов) с использованием логических связок \neg , \wedge , \vee , \rightarrow и кванторов по следующим правилам:

П1. Каждый атом есть формула. Всякая входящая в атом переменная является свободной в этом атоме.

П2. Если α и β — уже построенные формулы, то выражения $\neg \alpha$, $(\beta \wedge \alpha)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$ — также формулы. Всякая переменная, входящая свободно (связанно) в α или β , входит свободно и в новые формулы.

П3. Если α — уже построенная формула, в которую свободно входит переменная x , значения которой принадлежат к сорту A , то выражения $(\forall x \in A)\alpha$ и $(\exists x \in A)\alpha$ также являются формулами. Переменная x считается связанной в новых формулах.

Атомы \mathcal{L}^Σ составляются из термов, предикатных символов и символа равенства по следующим правилам:

П4. Если сигнатура Σ содержит выражение $p: A_1 \times A_2 \times \dots \times A_n \rightarrow T$ и $\tau_1, \tau_2, \dots, \tau_n$ — термы сортов A_1, A_2, \dots, A_n соответственно, то $p(\tau_1, \tau_2, \dots, \tau_n)$ — атом;

П5. Если τ и σ — термы одинакового сорта, то выражение $\tau = \sigma$ есть атом.

Термы составляются по следующим правилам:

П6. Переменная, принимающая значения из сорта A , есть терм сорта A .

П7. Если сигнатура Σ содержит выражение $f: A_1 \times A_2 \times \dots \times A_n \rightarrow B$, причем $B \neq T$, и $\tau_1, \tau_2, \dots, \tau_n$ — уже построенные термы сортов A_1, A_2, \dots, A_n соответственно, то выражение $f(\tau_1, \tau_2, \dots, \tau_n)$ есть терм сорта B .

Рассмотренный пример механообработки иллюстрирует общую ситуацию построения логической модели с использованием языка многосортного исчисления предикатов. Разработчик на основе предварительного анализа предметной области дает имена для констант, сортов, функций и предикатов. Имена эти выбирают так, чтобы их мнемоника соответствовала названиям неформаль-

ных понятий предметной области. Структурные связи между понятиями предметной области формально представляются в сигнатуре Σ , составленной из введенных имен.

Другие связи между понятиями предметной области представляются формулами из Σ . Пусть Ω обозначает множество формул, построенных для предметной области. Эти формулы должны быть истинны в той формальной структуре, которая представляет предметную область (если это представление ей адекватно). Структура, интерпретирующая заданную сигнатуру Σ , строится следующим образом. Указывается непустое множество U , называемое универсумом. Для каждого имени сорта A из сигнатуры Σ указывается некоторое подмножество универсума, которое также обозначается A ($A \subseteq U$). Для каждого выражения $f: A_1 \times A_2 \times \dots \times A_n \rightarrow B$ из сигнатуры указывается функция, отображающая множество $A_1 \times A_2 \times \dots \times A_n$ в множество B ; эта функция также обозначается f .

Для примера формализации фрагмента предметной области «механообработка» структура может быть построена следующим образом:

$U = \{\text{дет}_j, \text{ст}_j, \text{опер}_j | j \in N, j \geq 1\} \cup \{\text{ток}, \text{фрез}, \text{ст-вал}, \text{вал-шест}\} \cup N$ (где $N = 0, 1, 2, \dots$);

Деталь = $\{\text{дет}1, \text{дет}2, \text{дет}3, \text{дет}4\}$;

Станок = $\{\text{ст}1, \text{ст}2, \text{ст}3\}$;

Операция = $\{\text{опер}1, \text{опер}2, \dots, \text{опер}8\}$;

ТипСтанка = $\{\text{ток}, \text{фрез}\}$;

ТипДетали = $\{\text{ст-вал}, \text{вал-шест}\}$;

Время = N

Функции дет, ст, нач, кон, тип-дет, тип-ст, фрез-торц и ток-обр указаны в табл. 1.2—1.4. Предикат \leq имеет смысл отношения «меньше или равно» на множестве N натуральных чисел; функция $+$ — смысл операции сложения натуральных чисел. В указанной структуре истинны все формулы (10)—(17). Как говорят в математической логике, структура эта является моделью множества формул $\Omega = \{(10), (11), \dots, (17)\}$. Но здесь термин «модель» означает не то, что в словосочетании «логическая модель».

Таблица 1.2

Операция	дет	ст	нач	кон	ток_обр	фрез_торц
опер1	дет1	ст3	0	5	0	1
опер2	дет1	ст1	5	12	1	0
опер3	дет2	ст3	5	10	0	1
опер4	дет2	ст2	10	17	1	0
опер5	дет3	ст3	10	16	0	1
опер6	дет3	ст1	16	24	1	0
опер7	дет4	ст3	16	22	0	1
опер8	дет4	ст2	22	28	1	0

Таблица 1.3

Деталь	тип_дет
дет1	ст-вал
дет2	ст-вал
дет3	вал-шест
дет4	вал-шест

Таблица 1.4

Станок	тип_ст
ст1	ток
ст2	ток
ст3	фрез

Пусть Σ — произвольная сигнатура и M — интерпретирующая ее структура. Каждая замкнутая формула $\alpha \in \mathcal{L}^\Sigma$ (т. е. формула, не включающая свободные переменные) в структуре M истинна или ложна. Если α истинна в M , то пишут $M \models \alpha$. Формальное определение истинности или ложности замкнутой формулы α в структуре M выполняется по индуктивным правилам П8 и П9. Для применимости этих правил нужно расширить исходную сигнатуру Σ , присоединив к ней выражения $\alpha: \rightarrow A$ для каждого элемента a сорта A и каждого сорта A из Σ . Другими словами, расширенная сигнатура Σ' включает все литеральные константы, обозначающие элементы тех множеств, которые интерпретируют в M сорта. Благодаря такому расширению множество всех формул $\mathcal{L}^{\Sigma'}$ оказывается замкнутым относительно подстановок любых литеральных констант вместо свободных переменных.

Базу индуктивных правил П8 и П9 составляют константные (т. е. не содержащие переменных) атомарные формулы. Истинность или ложность каждого такого атома устанавливается в M непосредственно.

П8. Для любых замкнутых формул $\alpha, \beta \in \mathcal{L}^{\Sigma'}$:

$M \models \neg \alpha$ тогда и только тогда, когда не имеет места $M \models \alpha$;

$M \models (\alpha \wedge \beta)$ тогда и только тогда, когда $M \models \alpha$ и $M \models \beta$;

$M \models (\alpha \vee \beta)$ тогда и только тогда, когда $M \models \alpha$ или $M \models \beta$;

$M \models (\alpha \rightarrow \beta)$ тогда и только тогда, когда не имеет места $M \models \alpha$ или имеет место $M \models \beta$.

П9. Для любой формулы $\alpha \in \mathcal{L}^{\Sigma'}$, содержащей одну свободную переменную x , т. е. $\alpha = \alpha[x]$, полагаем

$M \models (\exists x \in A) \alpha$ тогда и только тогда, когда $M \models \alpha[a]$, для хотя бы одной константы a сорта A из Σ' .

Структура M , интерпретирующая сигнатуру Σ , называется модельной структурой для заданной совокупности замкнутых формул $\Omega \subseteq \mathcal{L}^\Sigma$, если все формулы из Ω истинны в M , т. е. $M \models \alpha$ для всех $\alpha \in \Omega$.

Обращаясь к примеру сигнатуры $\Sigma = \{(1), (2), \dots, (9)\}$, которая была построена для фрагмента предметной области «механообработка», видим, что указанная для этой сигнатуры структура является модельной структурой для совокупности формул $\Omega = \{(10), (11), \dots, (17)\}$.

Совокупность замкнутых формул может не иметь ни одной модельной структуры. Тогда она называется *логически невыполнимым* (или просто *невыполнимым*) множеством. Другими словами, для невыполнимого множества формул нельзя указать такую интерпретацию, что в ней будут истинны все формулы множества. Синонимом термина «невыполнимое» является термин «противоречивое».

Логическое следствие и логический вывод

С помощью понятия модельной структуры уточняется понятие логического следствия, которое является интуитивно достаточно ясным. Пусть Ω и β — произвольное множество замкнутых формул и произвольная замкнутая формула, записанные в сигнатуре Σ . Говорят, что β *логически следует* из Ω , и пишут $\Omega \models \beta$, если β истинно во всякой модельной структуре для Ω . Иными словами, отношение логического следствия $\Omega \models \beta$ имеет место тогда и только тогда, когда нет ни одной интерпретации, в которой все формулы из Ω были бы истинны, а формула β ложна. Таким образом, символ \models имеет двойное употребление: для обозначения отношения $M \models \beta$ между структурой и формулой и для обозначения отношения $\Omega \models \beta$ между множеством формул и формулой.

Например, рассмотрим следующее утверждение:

Утв1 — Операция опер3 выполнялась на фрезерном станке. Очевидно, что Утв1 логически следует из Зн7 и Зн8. Утв1 формально представляется так:

$$\text{тип_ст}(\text{ст}(\text{опер3})) = \text{фрез.} \quad (18)$$

Поэтому неформальное логическое следствие Утв1 из Зн7 и Зн8 формально представляется в виде

$$(16), (17) \models (18). \quad (19)$$

(Правильнее следовало бы писать $\{(16), (17)\} \models (18)$, но обычно фигурные скобки здесь не пишутся.)

Справедливость логического следствия (19) можно доказать математически. Пусть в некоторой структуре M истинны формулы (16) и (17), т. е. $M \models (16)$ и $M \models (17)$. Из последнего соотношения следует, что (см. П8)

$$M \models (\text{дет}(\text{опер3}) = \text{дет2}), \quad M \models (\text{ст}(\text{опер3}) = \text{ст3}),$$

$$M \models (\text{нач}(\text{опер3}) = 5), \quad M \models (\text{нач}(\text{опер3}) = 10).$$

В частности, формула $\text{ст}(\text{опер3}) = \text{ст3}$ истинна в структуре M . Отсюда следует, что в M истинна также формула $\text{тип_ст}(\text{ст}(\text{опер3})) = \text{тип_ст}(\text{ст3})$. С другой стороны, так как в M истинна формула, $\text{тип_ст}(\text{ст3}) = \text{фрез.}$, в M также истинна формула $\text{тип_ст}(\text{ст}(\text{опер3})) = \text{фрез.}$, т. е. $M \models (18)$. Итак, во всякой структуре M , в которой истинны (16) и (17), будет также истинна и (18). Следовательно, $(16), (17) \models (18)$.

Для исчисления предикатов основной является проблема логического следствия: разработать методы и алгоритмы, позволяющие для произвольных Ω и β — множества замкнутых формул и замкнутой формулы — выяснить, имеет ли место отношение логического следствия $\Omega \models \beta$.

Проблема логического следствия эквивалентна проблеме логической невыполнимости: разработать методы и алгоритмы, позволяющие для произвольного множества Ω замкнутых формул выяснить, является ли Ω невыполнимым множеством. Действительно, $\Omega \models \beta$ имеет место тогда и только тогда, когда множество $\Omega \cup \{\neg \beta\}$ невыполнимо. Докажем последнее утверждение. По определению $\Omega \models \beta$ имеет место тогда и только тогда, когда нет ни одной структуры M , такой, что для всех $\alpha \in \Omega$ верно $M \models \alpha$, но неверно, что $M \models \beta$, т. е. верно, что $M \models \neg \beta$. Последнее означает, что нет ни одной структуры M , такой, что $M \models \gamma$ для всех $\gamma \in \Omega \cup \{\neg \beta\}$, т. е. множество $\Omega \cup \{\neg \beta\}$ невыполнимо.

Проблема логического следствия тесно связана также с проблемой общезначимости логических формул. Замкнутая формула $\alpha \in \mathcal{L}^x$ называется *общезначимой* (или торжественно истинной), если α истинна во всякой структуре сигнатуры Σ . Очевидно, что соотношение $\alpha_1, \alpha_2, \dots, \alpha_p \models \beta$ имеет место тогда и только тогда, когда формула $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_p \rightarrow \beta$ общезначима. В самом деле, указанное логическое следствие справедливо тогда и только тогда, когда нет ни одной структуры M , такой, что $M \models \alpha_1, M \models \alpha_2, \dots, M \models \alpha_p$, и не имеет места $M \models \beta$, а это означает, что нет ни одной структуры M , такой, что не имеет места $M \models \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_p \rightarrow \beta$. Очевидно, что соотношение $\alpha_1, \alpha_2, \dots, \alpha_p \models \beta$ справедливо тогда и только тогда, когда формула $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \neg \beta$ невыполнима, т. е. не имеет модельных структур.

А. Черч доказал алгоритмическую неразрешимость проблемы общезначимости формул исчисления предикатов: нельзя построить алгоритм, который был бы применим к любой замкнутой формуле α в произвольной сигнатуре и который выяснял бы, является α общезначимой формулой или нет. Теорема Черча не запрещает существования алгоритмов общезначимости для отдельных сигнатур или классов сигнатур. Однако имеются сигнатуры с неразрешимой проблемой общезначимости.

В математической логике известна теорема Геделя, из которой следует, что можно построить частичный (т. е. «зацикливающийся» на некоторых предъявляемых ему формулах) алгоритм, область определенности которого (т. е. мно-

жество формул, на которых он дает результат) включает множество всех общезначимых формул. Теорема Геделя утверждает, что проблему общезначимости для исчисления предикатов можно решить с помощью системы логического вывода (дедукции) так называемого гильбертовского типа.

Система дедукции гильбертовского типа состоит из аксиом и правил вывода. Аксиомами служат некоторые общезначимые формулы. Правило вывода — это отношение $R(a_1, a_2, \dots, a_n)$ между формулами a_i (чаще всего $n=2$ или $n=3$), обладающее так называемым свойством состоятельности: если формулы a_1, a_2, \dots, a_{n-1} истинны в какой-либо структуре M и имеет место $R(a_1, a_2, \dots, a_n)$, то формула a_n также истинна в M .

Вместо $R(a_1, a_2, \dots, a_n)$ обычно пишут $a_1, a_2, \dots, a_{n-1} \vdash a_n$ и говорят, что формула a_n получена из формул a_1, a_2, \dots, a_{n-1} применением к ним правила вывода R . Выводом формулы β из множества формул Ω называется последовательность формул, такая, что ее последняя формула совпадает с β и каждая другая формула либо есть аксиома, либо получается применением к некоторым предыдущим формулам одного из правил вывода. Если существует вывод формулы β из множества формул Ω , то говорят, что β выводима из Ω , и пишут $\Omega \vdash \beta$.

Система дедукции называется *полной*, если для любого множества формул Ω формула β выводима из Ω тогда и только тогда, когда β логически следует из Ω , т. е. $\Omega \vdash \beta$ тогда и только тогда, когда $\Omega \models \beta$. Теорема Геделя устанавливает факт полноты некоторой системы дедукции.

Для практических задач теории логических моделей представления знаний более эффективны системы дедукции не гильбертовского, а резолюционного типа (§ 2.4).

Проблемы логического следования и логической невыполнимости важны не только для исчисления предикатов, но и для других формализмов представления знаний.

Пусть Δ — какое-либо описание предметной области в некотором формализме представления знаний. Если этот формализм является логическим языком, то Δ есть логическая модель предметной области. Δ состоит из двух частей: Δ' и Δ'' , которые описывают соответственно структурные свойства предметной области и логические ее свойства. (Например, если формализмом представления знаний является многосортное исчисление предикатов, часть Δ' есть сигнатура Σ , а часть Δ'' — совокупность замкнутых формул Ω , записанных в сигнатуре Σ .) Часть Δ'' служит для представления ограничений, которым удовлетворяют сущности и отношения предметной области, закономерностей, описывающих поведение сущностей, и т. п.

Цель разработки логической модели — получить такое описание Δ , чтобы его часть Δ'' имела модельную структуру M , формально приближающую заданную предметную область. Но разработчик логической модели может совершить ошибки, в результате которых Δ'' не будет иметь модельных структур, адекватных предметной области. Ошибки могут быть столь критичными, что Δ'' вообще может не иметь модельных структур. Ясно, что методы выяснения невыполнимости могут быть использованы для обнаружения таких критических ошибок.

Вопросы анализа предметной области после ее формализации могут быть сведены к выяснению соответствующих свойств модельных структур для Δ . Многие из этих свойств удается представить предложениями того же логического языка, который был использован для составления описания Δ . Если β обозначает одно из таких предложений, то все модельные структуры для Δ'' будут обладать свойством, выраженным в β , тогда и только тогда, когда $\Delta'' \models \beta$. Таким образом, методы логического вывода могут быть полезны для решения задач анализа предметных областей.

Иногда разработчика интересует логическая модель Δ , которая не содержит избыточной информации в том смысле, что из Δ нельзя исключить ни одного предложения β , не изменив совокупности модельных структур. Очевидно, что β можно исключить из Δ'' тогда и только тогда, когда $\Delta'' \models \beta$. Таким об-

разом, методы логического вывода можно использовать для решения задачи избыточности информации.

Запросы информационно-справочного характера к базам данных, представляющим модельные структуры для логической модели, также можно трактовать в терминах логического следования или невыполнимости. Пусть, например, запрос состоит в нахождении всех объектов (сущностей), которые обладают определенным свойством в предметной области. Часто свойство объектов удается выразить некоторым предложением $\beta[x]$ в логическом языке. Это предложение содержит свободную переменную x , вместо которой разрешается подставлять имена объектов (константы), среди которых разыскиваются обладающие заданным свойством. Таким образом, запрос можно сформулировать так: «Найти все константы c , такие, что предложение $\beta[c]$ истинно в заданной структуре M », или в более формальном виде: «Найти множество $\{c|M|=\beta[c]\}$ ». Ответ на этот запрос может быть вычислен путем перебора всех констант c (данного сорта C) и отбора тех из них, для которых формула $\beta[c]$ истинна в структуре M (последнее выясняется непосредственным применением правил П8 и П9). Но можно свести вычисление этого запроса к логическому выводу формулы

$$(\forall x \in C) \{ \beta[x] \rightarrow \text{ответ}(x) \} \quad (20)$$

из совокупности формул Ω_M , которая однозначно характеризует структуру M . Формула (20) неформально выражает следующее: «Всякий раз, как какая-либо константа сорта C удовлетворяет формуле β , считать ее ответом». Совокупность Ω_M состоит из тех константных формул вида $f(c_1, c_2, \dots, c_n) = c$ или $p(c_1, c_2, \dots, c_n)$, которые истинны в M . Заметим, что полные методы дедукции могут быть модифицированы так, что из вывода (20) можно извлечь все константы c , такие, что $\beta[c]$ выводима из Ω_M .

Усилением рассмотренного запроса является запрос: «Найти все константы c , для которых $\beta[c]$ истинна во всех модельных структурах для Δ' ». Усиленный запрос состоит в нахождении множества $\{c|\Delta'=\beta[c]\}$. Таким образом, для вычисления ответа на этот запрос можно применять логический вывод.

Чаше, однако, рассматриваются запросы «промежуточного» типа, когда структура, моделирующая предметную область, представлена лишь частично. Пусть P обозначает частичную структуру, расширяемую до некоторой модельной структуры для Δ'' . Как правило, расширение P неоднозначно, и запрос формулируется тогда так: «Найти множество $\{c|M|=\beta[c], M \text{ расширяет } P\}$. Частичная структура P также может быть охарактеризована конечным множеством Ω_P константных формул. Ясно, что запрос эквивалентен запросу: «Найти $\{c|\Delta'' \cup \Omega_P = \beta[c]\}$ », для вычисления которого снова можно применить метод логического вывода.

Логические языки и модели данных

В теории баз данных под *моделью данных* понимают формализм, предназначенный для описания структур данных, зависимостей между ними, а также так называемых ограничений целостности, т. е. некоторых условий, которым должны удовлетворять допустимые базы данных. Описание, выполненное в таком формализме, называется *схемой*. Допустимая база данных, т. е. конечная совокупность данных, удовлетворяющих схеме, называется *экземпляром схемы*.

В указанной терминологии язык многосортного исчисления предикатов следует считать моделью данных. В самом деле, $\Sigma \cup \Omega$ можно рассматривать как схему, причем первая ее часть (сигнатура Σ) описывает структуру данных, а вторая часть (множество Ω замкнутых формул) — зависимости и ограничения целостности. Конечную модельную структуру для Ω можно считать экземпляром схемы $\Sigma \cup \Omega$.

Таблица 1.5

Операция				
НомерДетали	НомерСтанка	Начало	Конец	ТипОперации
1	3	0	5	фрез_торц
1	1	5	12	ток_обр
2	3	5	10	фрез_торц
2	2	10	17	ток_обр
3	3	10	16	фрез_торц
3	1	16	24	ток_обр
4	3	16	22	фрез_торц
4	2	22	28	ток_обр

Таблица 1.6

Деталь	
НомерДетали	ТипДетали
1	ст_вал
2	ст_вал
3	вал_шест
4	вал_шест

Таблица 1.7

Станок	
НомерСтанка	ТипСтанка
1	ток
2	ток
3	фрез

Реляционная модель данных [Codd, 1970] — это формализм, непосредственно соответствующий простому фрагменту языка многосортного исчисления. В качестве примера построим реляционную схему для данных фрагмента предметной области «механообработка». Экземпляр этой схемы показан в табл. 1.5—1.7.

В реляционную схему R включим три отношения, назвав их (как и раньше в примере сигнатуры Σ) Деталь, Станок и Операция. Эти отношения имеют атрибуты с именами НомерДетали, ТипДетали, НомерСтанка, ТипСтанка, ТипОперация, Начало, Конец. Связь атрибутов с отношениями указана при помощи структурных предложений:

Деталь[НомерДетали, ТипДетали], Станок[НомерСтанка, ТипСтанка],

Операция [НомерДетали, НомерСтанка, Начало, Конец, ТипОперации].

Структурную часть R' реляционной схемы получаем, дописывая к этим структурным предложениям описания доменов (областей значений атрибутов):

$\text{dom}(\text{НомерДетали}) = \text{dom}(\text{НомерСтанка}) = \{1, 2, 3, \dots\};$

$\text{dom}(\text{ТипДетали}) = \{\text{ст_вал}, \text{вал_шест}\};$

$\text{dom}(\text{ТипСтанка}) = \{\text{ток}, \text{фрез}\};$

$\text{dom}(\text{Начало}) = \text{dom}(\text{Конец}) = \{0, 1, 2, \dots\};$

$\text{dom}(\text{ТипОперации}) = \{\text{ток_обр}, \text{фрез_торц}\}.$

Ясно видна связь структурной части R' реляционной схемы с сигнатурой $\Sigma = \{(1), (2), \dots, (9)\}$. Но непосредственно R' отвечает сигнатуре Σ_1 , представленная следующими выражениями:

№_дет: Операция \rightarrow НомерДетали №_ст: Операция \rightarrow НомерСтанка, нач: Операция \rightarrow Начало, кон: Операция \rightarrow Конец, тип_опер: Операция \rightarrow ТипОперации, №_дет: Деталь \rightarrow НомерДетали, тип_дет: Деталь \rightarrow ТипДетали, №_ст: Станок \rightarrow НомерСтанка, тип_ст: Станок \rightarrow ТипСтанка, фрез_торц: \rightarrow ТипОперации, ток_обр: \rightarrow ТипОперации, ст_вал: \rightarrow ТипДетали, вал_шест: \rightarrow ТипДетали, ток: \rightarrow ТипСтанка, фрез: \rightarrow ТипСтанка, j : \rightarrow Но-

мерДетали, $j \rightarrow \text{НомерСтанка}$ ($1 \leq j < \infty$), $k \rightarrow \text{Начало}$, $k \rightarrow \text{Конец}$ ($1 \leq k < \infty$).

Сигнатура Σ_1 получается из R' трансляцией, при которой отношения и атрибуты реляционной схемы переходят в сорта сигнатуры, а каждая пара отношения — атрибут определяет одноместную функцию, отображающую элементы сорта, соответствующего отношению, в элементы сорта, соответствующего атрибуту (например, пара Операция — НомерДетали определяет функцию Операция \rightarrow НомерДетали, которая обозначена №_дет).

Предложения, задающие так называемые ключи и функциональные зависимости, естественно отнести к логической части R'' реляционной схемы R . Ключ — это набор атрибутов отношения, значения которых однозначно идентифицируют кортежи этого отношения (иначе, строки таблиц, представляющих отношение). В данном примере реляционной схемы имеем следующие предложения, задающие ключи:

НомерДетали ключ в Деталь, НомерСтанка ключ в Станок,

(НомерДетали, НомерСтанка) ключ в Операция.

Эти предложения просто выражаются в языке многосортного исчисления предикатов. Например, для третьего предложения имеем формулу

$$(\forall x, y \in \text{Операция}). (\text{№_дет_оп}(x) = \text{№_дет_оп}(y)) \wedge \\ (\text{№_ст}(x) = \text{№_ст}(y)) \rightarrow x = y.$$

Функциональная зависимость — это связь между набором атрибутов отношения и другим его атрибутом, определяющая ограничение на экземпляры схемы: в каждом кортеже значения атрибутов из набора однозначно определяют значение другого атрибута. Функциональная зависимость выражается предложением вида

$$(A_1, A_2, \dots, A_p) \text{ определяют } B \text{ в } C, \quad (21)$$

где A_1, A_2, \dots, A_p, B — атрибуты отношения C . Например, имеем

НомерСтанка определяет ТипОперации в Операция.

Функциональные зависимости также можно отнести к логической части R'' реляционной схемы. Они выражаются простыми формулами многосортного исчисления предикатов. Например, (21) представляется формулой

$$(\forall x, y) (\text{Операция}). (\text{№_ст}(x) = \text{№_ст}(y) \rightarrow (\text{тип_опер}(x) = \text{тип_опер}(y))).$$

Предложения реляционной схемы R , отличные от структурных предложений, ключей и функциональных зависимостей, выражают ограничения целостности. Например, знание Σ_2 следует отнести к ограничению целостности. В сигнатуре Σ Σ_2 выражено формулой (11). В сигнатуре Σ_1 это же знание выражается той же формулой, но с заменой имени Время на имя Начало или Конец (так как в Σ первое имя отсутствует, а второе и третье имена входят в Σ_1 и как сорта имеют ту же интерпретацию, что и первое имя).

Язык многосортного исчисления предикатов можно, таким образом, использовать в качестве формализма для записи ограничений целостности. В [Codd, 1972] предложено языки запросов к реляционным базам данных строить на основе исчисления предикатов.

Рассмотрим, например, следующий запрос к базе данных, записанной в табл. 1.5—1.7: «Вывести номера деталей с указанием их типа, которые к моменту времени 20 прошли токарную обработку». Его можно сформулировать как нахождение множества

$$\{(x, y) \mid (\exists z \in \text{Операция}). (\text{№_дет_оп}(z) = x) \wedge \\ (\text{тип_опер}(z) = \text{ток_обр}) \wedge \\ (\text{кон}(z) \geq 20) \wedge \\ (\exists u \in \text{Деталь}). (\text{тип_дет}(u) = y) \wedge \\ (\text{№_дет}(u) = x))\} \quad (22)$$

По методу Кодда этот запрос можно транслировать в последовательность операций над отношениями (таблицами), которая состоит из операций проекции и так называемого естественного соединения. Выполнив эту последовательность операций, получим ответ на заданный запрос.

Метод Кодда фактически основан на теоретико-множественной трактовке правил П8 и П9, при которой, в частности, конъюнкции соответствует операция естественного соединения, а квантору \exists — операция проекции.

Запросы к реляционной базе данных можно вычислить с помощью логического вывода. Для этого предварительно базу данных D нужно представить совокупностью Ω_D атомарных константных формул вида $f(c) = d$, где f — имя функции, соответствующее паре отношение — атрибут, c — имя строки таблицы, а d — значение атрибута, стоящее в этой строке. Например, четвертая строка табл. 1.5 вносит в Ω_D такие атомарные константные формулы: №_дет(опер4) = 2, №_ст(опер4) = 2, нач(опер4) = 10, кон(опер4) = 17.

Запрос (22) к базе данных D , представленной в табл. 1.5—1.7, вычисляется с помощью вывода из Ω_D формулы

$$(\forall x \in \text{НомерДетали}, \forall y \in \text{ТипДетали}) \{ \beta[x, y] \rightarrow \text{ответ}(x, y) \},$$

где $\beta[x, y]$ обозначает входящую в (22) формулу.

Метод вычисления запросов, основанный на логическом выводе, является менее эффективным, чем метод Кодда и другие подобные методы прямого вычисления запросов. Однако первый метод универсален и годится для вычисления запросов к неполным базам данных, другие же прямые методы оказываются непригодными.

Неполной базой данных называется такая база E , которая не удовлетворяет зависимостям и ограничениям из R'' , но может быть расширена до допустимой базы данных D (т. е. базы данных, удовлетворяющей всем зависимостям и ограничениям из R''). Запрос к неполной базе данных E может быть вычислен с помощью логического вывода запросной формулы из совокупности формул $\Omega_E \cup R''$.

Произвольную сигнатуру Σ в языке многосортного исчисления предикатов можно представить графически, как это показано на рис. 1.1 для сигнатуры $\Sigma = \{(1), (2), \dots, (9)\}$. Сигнатуры, интерпретирующие сигнатуру Σ , также могут быть представлены графически, как это видно из рис. 1.2, где показан фрагмент сети, представляющий информацию, записанную в табл. 1.2—1.4. Таким образом, рис. 1.1 играет роль схемы, а рис. 1.2 — роль экземпляра этой схемы.

Язык сетевых схем и сетей-экземпляров указанного типа называют *бинарной моделью данных*. Бинарную модель называют также функциональной моделью. Заметим, однако, что в бинарной модели используются только одно-

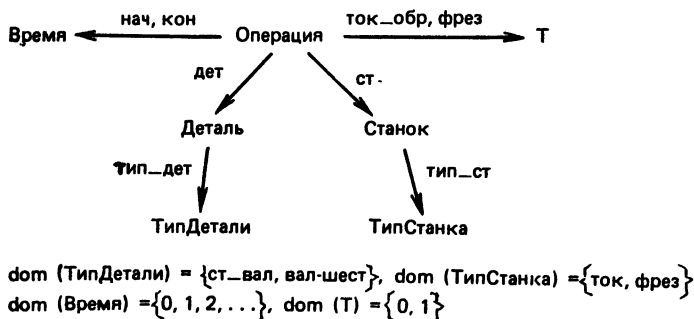


Рис. 1.1

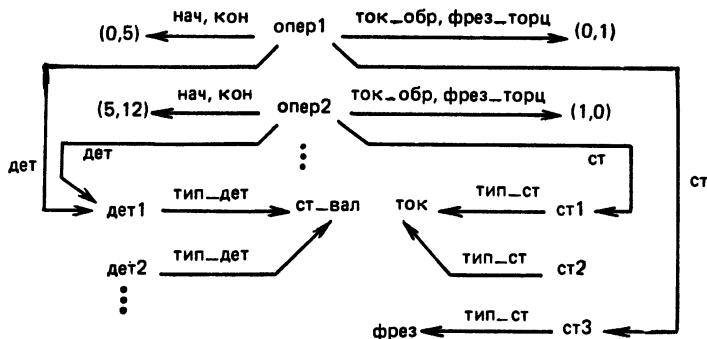


Рис. 1.2

местные функции и предикаты. Чтобы представить в бинарной модели много-местные функции или предикаты, нужно прибегать к искусственному приему, вводя дополнительные имена сортов (классов). Так, n -местную функцию $f: A_1 \times \dots \times A_n \rightarrow B$ можно заменить на $(2n+1)$ одноместных функций $f: C \rightarrow B$, $g_j: A_j \rightarrow C$, $h_j: C \rightarrow A_j$ ($1 \leq j \leq n$), где h_j — функция проекции декартова произведения на j -ю его компоненту, g_j — функция вложения j -й компоненты в декартово произведение.

Имеется тривиальный прием перевода n -местной функции в $(n+1)$ -местный предикат: $f(x_1, x_2, \dots, x_n) = y$ тогда и только тогда, когда предикат $f(x_1, \dots, x_n, y)$ истинен. Благодаря этому «чистое» (т. е. без функций) исчисление предикатов эквивалентно с точки зрения принципиальной выразительности многосортному исчислению предикатов.

Реляционная модель данных, естественно, переводится в «чистое» исчисление предикатов. Рассмотрим снова пример реляционной схемы R , который связан с предметной областью «механообработка». Получаемая «чисто» предикатная модель данных, обозначаемая Q , состоит из двух частей: структурной части Q' и логической части Q'' . Структурная часть соответствует «шапкам» таблиц-реляций:

$Q' = \{\text{операция}(\text{№_детали}, \text{№_станка}, \text{Начало}, \text{Конец}, \text{ТипОперации}),$
 $\text{деталь}(\text{№_детали}, \text{ТипДетали}), \text{станок}(\text{№_станка}, \text{ТипСтанка})\}.$

В логическую часть включаются предложения, выражающие ключи и функциональные зависимости. Например, в Q'' записываем предложения

$$U = V : \text{— операция}(X, Y, U, _, _), \text{ операция}(X, Y, V, _, _), \quad (23)$$

$$U = V : \text{— операция}(X, Y, _, U, _), \text{ операция}(X, Y, _, V, _), \quad (24)$$

$$U = V : \text{— операция}(X, Y, _, _, U), \text{ операция}(X, Y, _, _, V), \quad (25)$$

$$U = V : \text{— операция}(_, X, _, _, U), \text{ операция}(_, X, _, _, V). \quad (26)$$

Предложения (23)—(25), взятые вместе, выражают то, что (№_детали, №_станка) является ключом отношения «операция», а предложение (26) выражает то, что ТипОперации функционально зависит от №_ст в отношении «операция».

В предикатной схеме Q использовался синтаксис языка логического программирования Пролог [Клоксин и др., 1987], идеи которого были выдвинуты в [Kowalski, 1974]. Согласно этому синтаксису имена отношений, функций и констант пишутся малыми буквами, а переменные начинаются с заглавных букв. Знак $:-$ является символом импликации, но записанной справа налево (\leftarrow); запятая служит символом конъюнкции. Предложения (23)—(26) являются примерами так называемых правил. В правилах неявно предполагается,

что все переменные связаны квантором \forall . Таким образом, например, предложение (23) в обычной нотации выглядит так:

$(\forall X \in \mathbb{N}_0, \forall Y \in \mathbb{N}_0 \text{ — станка, } \forall U \in \text{Начало, } \forall V \in \text{Начало, } \forall Z_1 \in \text{Конец, } \forall Z_2 \in \text{Конец, } \forall Z_3 \in \text{ТипОперации, } \forall Z_4 \in \text{ТипОперации}).$

операция $(X, Y, U, Z_1, Z_2) \wedge \text{операция}(X, Y, V, Z_3, Z_4) \rightarrow U = V$.

Экземпляр предикатной схемы Q представляется списком так называемых фактов, т. е. атомарных константных формул. В данном примере имеем такой экземпляр (предикатную базу данных F):

операция(1, 3, 0, 5, фрез_торц).	деталь(1, ст_вал).
операция(1, 1, 5, 12, ток_обр).	деталь(2, ст_вал).
операция(2, 3, 5, 10, фрез_торц).	деталь(3, вал_шест).
операция(2, 2, 10, 17, ток_обр).	деталь(4, вал_шест).
операция(3, 3, 10, 16, фрез_торц).	станок(1, ток).
операция(3, 1, 16, 24, ток_обр).	станок(2, ток).
операция(4, 3, 16, 22, фрез_торц).	станок(3, фрез).
операция(4, 2, 22, 28, ток_обр).	

Рассмотрим снова запрос: «Выдать номера деталей с указанием их типов, которые к моменту времени 20 прошли токарную обработку», обратив его к предикатной базе данных F . Этот запрос вычисляется с помощью правила ответ(X, Y): — операция(X, V, Z, U , ток_обр), $U \geq 20$, деталь(X, Y). (27)

Вычисление запроса инициируется предложением ? — ответ(X, Y), которое формирует цель: «Найти факт, унифицируемый с ответ(X, Y)». Делая попытку достижения этой цели непосредственно, интерпретатор начинает просмотр базы фактов F , но, не найдя подходящего факта, обращается к базе правил, пытаясь найти такое, которое имеет «голову», унифицируемую с ответ(X, Y). Таким правилом является (27), из которого получается цель:

«Найти факт, унифицируемый с операция(X, V, Z, U , ток_обр)». (28)

На этот раз цель удовлетворяется при подстановке $X=1, V=1, Z=5, U=12$ (см. второй факт в F). Теперь формируется цель, состоящая в выяснении неравенства $12 \geq 20$, получаемого из второго атома «хвоста» (27) при найденной подстановке. Так как это неравенство неверно, подстановка отменяется и интерпретатор снова пытается удовлетворить цель (28), производя поиск в F с точки возврата. Цель удовлетворяется при подстановке $X=2, V=2, Z=10, U=17$ (см. четвертый факт в F). Однако опять получается неверное неравенство $17 \geq 20$, и процесс поиска возвращается с отменой подстановки и выдвижением цели (28). Шестой факт из F удовлетворяет этой цели с унифицирующей подстановкой $X=3, V=1, Z=16, U=24$. Так как неравенство $24 \geq 20$ верно, то формируется очередная цель: «Найти факт, унифицируемый с деталь(3, Y)», которая удовлетворяется при подстановке $Y=\text{вал_шест}$. Поскольку цели, ассоциированные с «хвостом» правила (27), удовлетворены при подстановке $X=3, V=1, Z=16, U=24, Y=\text{вал_шест}$, то получим вычисленный факт ответ(3, вал_шест). Процесс поиска продолжается по той же схеме. В результате имеем еще один вычисленный факт ответ(4, вал_шест).

Итак, основанная на «чистом» Прологе предикатная модель $Q \cup F$ характеризуется следующими чертами: (1) структурная часть Q' схемы Q составляется из предложений простейшего вида, задающих схему отношений; (2) логическая часть Q'' схемы Q составляется из правил, т. е. предложений вида $B: -A_1, A_2, \dots, A_n$, где B, A_i — предикатно-атомарные формулы; (3) кроме декларативной интерпретации («если одновременно истинны все A_i , то истинно B ») правило имеет простую процедурную интерпретацию («чтобы достичь цели B , нужно одновременно достичь все цели A_i »); (4) достижение нескольких целей осуществляется с помощью механизмов унификации и поиска с возвратом; (5) запрос к базе данных F записывается в виде правила ответ(X_1, X_2, \dots, X_p): — A_1, A_2, \dots, A_n , а вычисление запроса инициируется предложением ? — ответ(X_1, X_2, \dots, X_p).

Исчисление предикатов было создано усилиями Г. Фреге (1879 г.), К. Пирса (1885 г.), Дж. Пеано (1908 г.), А. Н. Уайтхеда и Б. Рассела (1922—1927 гг.) для формального описания математического знания в связи с попытками решения проблем основания математики. Формальная семантика исчисления предикатов, основанная на теоретико-множественных моделях, была предложена А. Тарским (1930 г.). Полная система дедукции разработана Д. Гильбертом, П. Бернайсом и В. Аккерманом (1922—1928 гг.), а ее полнота доказана К. Геделем (1930 г.).

Г. Саймон, А. Ньюэлл и Дж. Шоу написали в 1957 г. машинную программу «Логик-теоретик», выполняющую поиск доказательств тавтологий в аксиоматической системе Рассела — Уайтхеда для пропозиционального исчисления. Первая серьезная попытка построить машинную систему для поиска логического вывода в исчислении предикатов была предпринята в 1958 г. [Ван Хао, 1970]. Важные результаты для создания эффективных систем логического вывода резолюционного типа были получены в 1963 г. [Девис, 1970] (§ 2.4).

Первая вопросно-ответная система, использующая дедукцию резолюционного типа [Green, 1969], является прообразом современных «интеллектуальных» информационных систем (систем дедуктивных баз данных и баз знаний). Разработка и применение логических моделей для баз данных и баз знаний описаны в [Gallaire et al., 1978].

В [Hewitt, 1971; Kowalski, 1974] была замечена двойственность семантики импликации: импликация $p \wedge q \rightarrow r$ имеет, с одной стороны, логическую трактовку (так называемая «материальная импликация»), с другой стороны, операционную (или процедуральную) трактовку («если вы хотите установить r , попробуйте установить p и q »). Импликации с универсально квантифицированными переменными (связанными только кванторами \forall) называют также хорновскими предложениями. Благодаря указанной операционной трактовке последовательность хорновских предложений можно рассматривать как некоторую недетерминированную программу. Детерминизация процесса вычисления по такой программе достигается поиском с возвратом. Эти простые, но фундаментальные идеи были положены в основу языков логического программирования (Плэннер, Пролог и др.).

Логические модели, построенные с применением языков логического программирования, широко применяются в базах знаний, экспертных системах. Практическое использование логических моделей облегчается выбором удачного синтаксиса языка логического программирования, а также включением в язык различных абстракций, таких как агрегация, таксономия, родовидовое отношение, наследование свойств и т. п. Для усиления практической выразительности логических моделей важна также объектная направленность языков представления знаний, так как неформальное описание предметных областей эксперты выполняют в терминах объектов, их связей и динамики. Разработка объектно-ориентированных языков программирования представляется весьма перспективным направлением исследований.

1.3. Сетевые модели

В. С. Лозовский

Основные определения

Первые сетевые модели (СМ) появились в шестидесятые годы. Примерами их могут служить RX-коды и синтагматические цепи [Скороходько, 1962], простые семантические сети [Quillian, 1968]. Непосредственное отношение к СМ имеют исследования по реляционным, сетевым и иерархическим базам

данных (см. § 1.1). Разработка моделей типа «сущность — связь» [Цикритис и др., 1985] еще больше сближает эти два направления исследований.

Введем ряд определений. Под сущностью будем понимать объект произвольной природы. Этот объект может существовать в реальном мире. В этом случае он будет называться *П-сущностью*. В базе знаний ему соответствует некоторое описание, полнота которого определяется той информацией, которую имеет о П-сущности ИС. Такое представление в базе знаний называется *М-сущностью*. Отметим, что могут существовать М-сущности, для которых в окружающем ИС мире нет соответствующих П-сущностей. Такие М-сущности представляют собой абстрактные объекты, полученные в результате операций типа обобщения внутри базы знаний.

Разделение на два типа сущностей позволяет использовать в СМ идеи, впервые сформулированные в теории *семиотических моделей* и основанном на них *ситуационном управлении* [Поспелов Д., 1981]. Под *семиотическими моделями проблемных областей* будет пониматься комплекс процедур, позволяющих отображать в базе знаний П-сущности и их связи, фиксируемые в проблемной области инженером по знаниям, в совокупности связанных между собой М-сущностей. Способ интерпретации взаимосвязанных П-сущностей будет называться *денотативной семантикой*, а способ интерпретации взаимосвязанных М-сущностей — *коннотативной семантикой*.

П-сущность по отношению к соответствующей ей в базе знаний М-сущности называется *денотатом* или *референтом* этой М-сущности, а М-сущность по отношению к исходной П-сущности — ее *десигнатом*, *именем*, *меткой*, *идентификатором* и т. п. Десигнат — это простейший элемент в СМ. Он входит в класс терминальных объектов СМ. Терминальным объектом называется М-сущность, которая не может быть разложена на более простые сущности. Остальные М-сущности называются *производными объектами* или производными М-сущностями.

Перечень терминальных объектов, которые могут образовывать классы или типы, задается при проектировании ИС. Ими могут быть целые вещественные числа, идентификаторы, строки, списки и т. п. Семантика терминальных объектов определяется набором допустимых процедур, оперирующих с ними, например: арифметические действия над числами, сравнение между собой строк или идентификаторов, операции ввода-вывода, включающие необходимые трансформации представлений, и т. д. Итак, каждая СМ базируется на множестве терминальных типов сущностей: $T = \{D^0, T_1^0, \dots, T_i^0\}$. Принадлежность некоторого терминального объекта t_i типу T_j^0 обозначается $t_i \in T_j^0$; символ D^0 обозначает тип десигнатов.

Введем понятие фрейма в сетевой модели (*СМ-фрейма*), которое отличается от распространенного понятия фрейма (см. § 1.1). СМ-фрейм может быть задан в виде ассоциативного списка атрибутов (имя_атрибута_1 значение_1 ... имя_атрибута_N значение_N). Под атрибутом понимается пара «имя атрибута — значение». Имена атрибутов характеризуют роли объектов, стоящих в позициях значений атрибутов. (Далее будем говорить о фреймах, понимая под ними СМ-фреймы.)

Рассмотрим высказывание: «С 20 по 25 Иванов был командирован в Ленинград за счет заказчика». В виде СМ-фрейма оно может быть представлено так:

(командирован	Иванов
куда	Ленинград
с_какого_числа	20
по_какое_число	25
за_счет	заказчик).

Приведенный способ формализации, описанный для использования в СМ, недостаточен. Во-первых, при переходе к *сети фреймов* необходимо обеспечить уникальную идентификацию отдельных фреймов с помощью десигнатов, или *меток*. Метки фреймов становятся узлами ассоциации соответствующих набо-

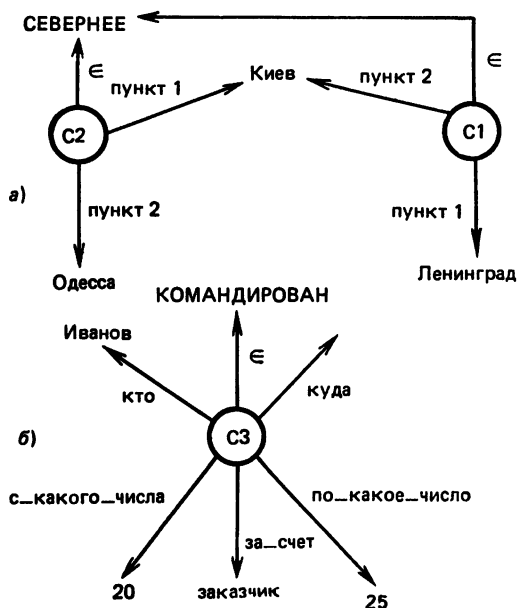


Рис. 1.3

символьной форме (для метки фрейма и его типа используется компактная позиционная запись):

(C1 СЕВЕРНЕЕ пункт_1 Ленинград пункт_2 Киев)
 (C2 СЕВЕРНЕЕ пункт_1 Киев пункт_2 Одесса)
 (C3 КОМАНДИРОВАН кто Иванов куда Ленинград
 с_какого_числа 20 по_какое_число 25 за_счет заказчик)

Представим рассмотренные СМ-фреймы в виде сетей (рис. 1.3, а, б). Порядок атрибутивных пар в символьной записи фреймов или в их графическом представлении значения не имеет.

В СМ атрибут — это указатель конкретной роли некоторого элемента во фрейме. Заметим, что вместо термина «атрибут» в литературе по ИС используются термины *глубинный падеж, актант, роль* и т. п. Благодаря свойству структуризации (см. § 1.1) атрибут также может быть представлен фреймом. Так, атрибут «кто» на рис. 1.3, б специфицирует фрейм типа «Субъектом командировки С3 является Иванов». Детализация этого атрибута в виде фрейма показана на рис. 1.4. Отсюда следует несколько замечаний. Во-первых, атрибут «субъект», расшифрованный на рис. 1.4, можно также развернуть во фрейм и т. д. На практике всегда решается вопрос об уровне детализации представлений. Другими словами, в отношении всех атрибутов устанавливается терминальный уровень $A^0: T = \{D^0 A^0 T_1^0, \dots, T_i^0\}$. Иногда атрибуты уровня A^0 называются *системными*, ибо их семантика должна быть заложена на этапе проектирования базы знаний.

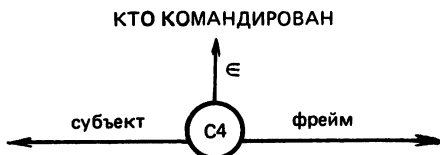


Рис. 1.4

ров атрибутов. Во-вторых, требует уточнения вопрос о коннотативном смысле фреймов. Можно сказать, что смысл задается перечнем имен (и значений) атрибутов. Если перечень имен постоянен и не совпадает с перечнем имен атрибутов фреймов другого вида, то возможно задание коннотативного смысла. Осложнения начинаются, если набор атрибутов переменен, в фреймах с разным смыслом приходится пользоваться одноименными атрибутами, а также используется упорядочивающая атрибутизация с именами 1, 2, 3, ... или ее сокращенный вариант — позиционная запись значений атрибутов без указания имен — порядковых номеров. Таким образом, целесообразно сопоставлять смысл фрейма с описанием его типа. Будем далее оперировать их именами.

С учетом сделанных уточнений сеть фреймов может быть представлена в следующей

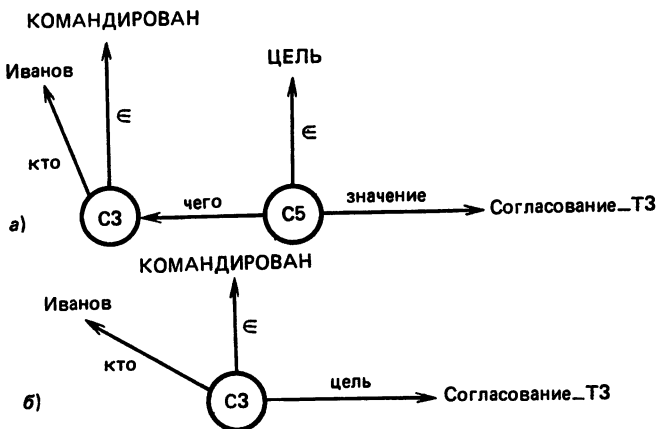


Рис. 1.5

Во-вторых, с введением в графовое представление десигнатов (C4 на рис. 1.4) появляется возможность уточнить семантику фрейма с помощью любого необходимого числа добавочных атрибутов.

Иногда можно реализовать процедуру обратного преобразования — *атрибутивной трансформации фрейма* (рис. 1.5). На рис. 1.5,а приведен фрейм, в котором описана цель командировки, а на рис. 1.5,б показано превращение этого фрейма в атрибут «цель». Фрейм, превращаемый в атрибут (C5 на рис. 1.5), должен иметь собственный атрибут, имеющий фрейм в качестве своего значения, в данном случае: *чего* (C5) = C3.

Структура М-объектов

Простейшими объектами СМ являются терминальные М-сущности, а также другие неразложимые представления. Назовем их *М-объектами*. Дальнейшее обогащение семантики объектных представлений осуществляется двумя путями.

1. Десигнат может стать меткой фрейма. Будем считать строку «Командировка Иванова для согласования ТЗ» (рис. 1.5,б) десигнатом соответствующего языкового описания. Как и в классическом исчислении высказываний, это неразложимое представление. Его можно обогатить атрибутами. На рис. 1.6 показана трансформация этой метки-десигната в соответствующий фрейм, аналогичный приведенному на рис. 1.5,б. В результате подобного обогащения получается фрейм, отражающий предикативную структуру, в явной форме специфицирующую роли составляющих ее объектов. Подобная трансформация порождает структуры с большими выразительными возможностями за счет дополнительных атрибутов. В реляционных базах данных подобный объект интерпретируется как строка соответствующего отношения, отражающая указанные свойства описанного объекта — конкретной командировки: кто, цель и т. д. При этом набор атрибутов-характеристик фиксирован. Именно они (и только они) трактуются как свойства рассматриваемого объекта, задавая в совокупности фиксированный объем понятия о соответствующей командировке.

2. Формализм, используемый в СМ представления знаний, позволяет реализовать иной путь обогащения системы представлений, обладающий большей гибкостью. Прежде всего десигнат «Командировка Иванова...» начинает выполнять функции *объектного ассоциатора*, относительно которого строится

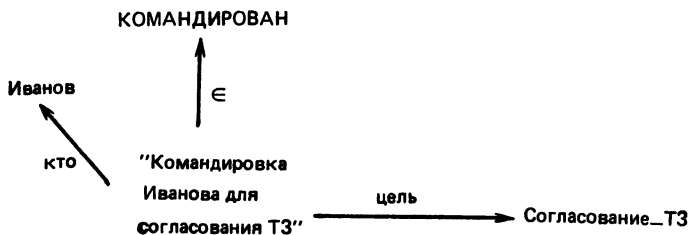


Рис. 1.6

нужный фрейм. На рис. 1.7 фрейм «Командировка Иванова» строится относительно С6, который выполняет роль объектного ассоциатора.

Уточним понятие факта. *Фактом (конкретным фреймом, фреймом-экземпляром)* называется фрейм, у которого значения всех атрибутов являются терминальными объектами.

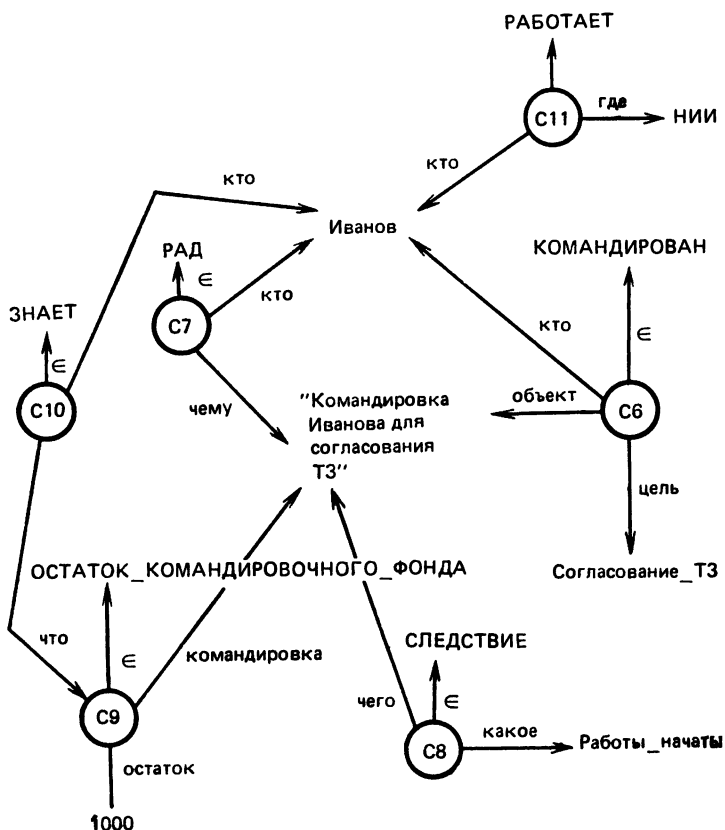


Рис. 1.7

Ситуацией будем называть выделенную в соответствии с определенным принципом совокупность фреймов. В частности, один фрейм представляет собой пример *элементарной ситуации*. Ситуация, все фреймы которой факты, называется *фактуальной, или экстенциональной*. Ситуация, включающая всю модель, может быть охарактеризована как *глобальная*, а любое ее собственное подмножество — как *локальная*. На рис. 1.7 зафиксирована экстенциональная ситуация: «Иванов, работающий в НИИ, был командирован с целью согласования ТЗ, чему он был рад. Остаток командировочного фонда после указанной командировки составил 1000 руб., что Иванову известно. Следствием рассматриваемой командировки явилось начало работ». Факты С7—С10, примыкающие к зафиксированному десигнату командирования Иванова, расширяют и уточняют коннотативную семантику конкретной командировки.

Формально ситуация на рис. 1.7 может быть трансформирована в один фрейм, являющийся расширением фрейма рис. 1.6 соответствующими атрибутами: следствие, остаток_фонда, рад_ли_командируемый, известен_ли_остаток_фонда. Обращает на себя внимание искусственность этого приема. Ясно, что такое расширение приведет либо к резкому возрастанию набора атрибутов фрейма КОМАНДИРОВАН, либо к появлению нескольких его вариантов. Оба решения непрактичны. Неудобства возрастают при сведении в один фрейм элементарных фреймов модального, статусного или референтного характера (например, С10 на рис. 1.7) и становятся непреодолимыми при дальнейшем усложнении ситуации. Это показывает, что ситуации — мощное изобразительное средство задания коннотативной семантики в СМ, значительно более богатое по изобразительным возможностям, чем фреймы.

Введем понятие *замкнутой ситуации для заданного объекта СМ*. Если ситуацию в СМ представить как предельно ассоциированную, то замкнутой ситуацией будем называть компонент связности сети, в который входит данный объект, т. е. полный набор фреймов, образующих этот компонент связности. *Глобальным свойством М-объекта* называется замкнутая относительно него ситуация. *Локальным свойством М-объекта* называется любая подситуация из ситуации, определяющая его глобальное свойство и включающая данный объект.

Приведенная на рис. 1.7 сеть представляет собой ситуацию, замкнутую относительно каждого из фигурирующих в ней термов. Таким образом, локальными свойствами «Иванова» является, в частности, то, что он работает в НИИ, был командирован с целью согласования ТЗ, что привело к началу работ, был рад своей командировке и т. д. Объединение всех этих локальных свойств будет глобальным свойством «Иванова». Никаких других свойств «Иванова» согласно данному состоянию сети не имеет. Приведенные определения являются исчерпывающими лишь для десигнатов. Другие терминальные объекты помимо этого, как правило, наделяются определенной операционной семантикой. Например, числам приписаны свойства, определяющие способ вычисления результата арифметических операций, сравнений и т. п.

В отдельных работах по искусственному интеллекту свойство трактуется как эквивалент одноместного предиката: (КРАСНЫЙ x), (БОЛЬШОЙ x), (ИСПРАВНЫЙ x). Подобная интерпретация является упрощенной формой записи представления

(<ИМЯ_ХАРАКТЕРИСТИКИ>

объект <десигнат объекта> | значение — характеристики <значение>).

Уточним представление о возможных объектах в СМ, формально квалифицируя их как значения атрибутов. Терминальные объекты (за исключением десигнатов) семантически наиболее просты несмотря на специфицированную операционную семантику. Таковы, например, числа, которые могут легко извлекаться из БД, подвергаться арифметическим преобразованиям и вновь помещаться в базу. Правила работы с десигнатами могут быть и более сложными, так как они — лишь ассоциаторы пучка свойств, определяющих модель некоторого П-объекта. Манипуляция подобными объектами зачастую бывает нетривиальной. Так, в отличие от классических реляционных моделей, где в ответ на запрос к базе данных по кадрам: «Выдай данные об Иванове» выдается

соответствующая запись, в которой заранее предусмотрен набор определенных полей, в СМ речь может идти о замыкании десигната Иванова либо о том или ином его подсобстве. Точно так же достаточно сложными могут оказаться правила исключения. Исключение Иванова из состава сотрудников конкретной организации не ограничивается локальным исправлением (например, Иванов должен рассчитаться с библиотекой, однако Иванов может сохранить за собой право на жилье, если оно не служебное, и т. д.).

Особые правила работы с объектами влекут за собой необходимость в иерархиях, строящихся на основе отношений типа «часть — целое». Так, атрибутивные пары являются частями фреймов, фреймы — частями ситуаций. Двигатель автомобиля состоит из узлов и механизмов; те, в свою очередь, из подузлов и деталей, причем связь между ними фиксируется с помощью разнообразных фреймов: ВХОДИТ В СОСТАВ, КРЕПИТСЯ К..., НАХОДИТСЯ ВНУТРИ, ЯВЛЯЕТСЯ ЧАСТЬЮ и т. п. Сложность описания возрастает, когда одни и те же объекты входят одновременно в несколько иерархических структур: тот же Иванов является сотрудником определенного отдела, членом коллектива некоторой комнаты и т. д. В то же время он — член семьи, множества пассажиров определенного автобуса и т. д. Поэтому на Иванова должны распространяться правила поведения, которым подчиняются члены упомянутых коллективов. При манипулировании объектами должны правильно обрабатываться последствия тех или иных действий.

Одним из способов задания иерархических структур является представление множеств в качестве объектов. Необходимость в этом возникает, в частности, если запретить, как это обычно и делается, фреймам иметь более одного атрибута с тождественными именами. Например: «Иванов был командирован в Ленинград и Таллинн». На граф-схемах условимся использовать однократно помеченное (здесь: «куда») разветвляющееся ребро, а в символических записях введем новый вид объекта — список с типизирующей пометкой :&. Приведенный факт при этом запишется так:

(С29 КОМАНДИРОВАН кто ИВАНОВ куда (:& Ленинград Таллинн)).

Семантика использования конъюнктивных множеств определяется их значением: по сути, это сокращенная форма записи конъюнкции соответствующих высказываний: «Командирован... в Ленинград» и «Командирован... в Таллинн». Повышение выразительных возможностей в СМ достигается также введением в формализм средств представления недоопределенной, неполной, неточной информации [Zadeh, 1973; Лозовский, 1983; Нариньяни, 1986a]. В тех случаях, когда текущие знания позволяют указать лишь множество объектов, из которого извлекается пока неопределенная выборка, естественно употребление объектов типа дизъюнктивного множества (OR-множества), например:

(С30 ОТПРАВИЛСЯ кто (:V Катя Маша Зина Нина) куда кино).

Подобная конструкция формализует ситуацию: «Подруги отправились в кино» в том случае, если круг подруг известен, но кто из них конкретно мог принять участие в данной мероприятии, не ясно. Аналогично вводятся XOR-множества («Исключающее ИЛИ»), когда речь идет о выборе одного объекта из совокупности. Множество всех указанных типов могут фигурировать в качестве самостоятельных объектов, задаваемых десигнатами. При этом фреймами могут специфицироваться их мощность, теоретико-множественные соотношения вхождения, пересечения, указания конкретных элементов.

Статусы и логическая структура модели предметной области

Использование фреймов требует фиксации их статусов в модели проблемной области. Простейшим статусом является статус истинности, а в нем тот статус, при котором фрейму, хранящемуся в базе, приписывается значение абсолютной истинности. Предполагается, что все фреймы, которые в данный момент имеются в базе, абсолютно истинны, а те, которых там сейчас нет, — абсолютно ложны. Такая база называется замкнутой, и ей соответствует зам-

жнутая модель представления знаний [Reiter, 1978]. Примером может явиться БД авиарейсов, включающая фреймы с атрибутами: номер-рейса, исх-пункт, пункт-назначения, время-вылета и т. д.

Конъюнктивность модели постулирует одновременную истинность всех составляющих ее утверждений. Под *утверждением* будем понимать фрейм в паре с истинным значением, определяющим его истинностный статус. В рассматриваемых моделях не возникает необходимости в логической операции отрицания. Оно реализуется исключением соответствующих утверждений из базы знаний. Бедность замкнутых моделей проявляется в том, что в них отсутствует статус фреймов, рассматриваемых в отрыве от своей истинной оценки, а также возможность работы с неопределенно-истинными утверждениями.

От этих недостатков избавлена *открытая модель*, в которой фреймам, отсутствующим в БД, присваивается истинностный статус «неопределен». Ложные факты в БД хранятся явно [Reiter, 1978]. Возможны и более сложные способы приписывания статусов истинности фреймам. Могут, например, использоваться статусы с набором градаций истинности, большим трех, или с бесконечным числом градаций, определенных на отрезке $[0, 1]$, когда значению 0 соответствует статус «ложь», а значению 1 — статус «истина». Статус истинности в СМ часто понимается шире, чем в моделях логического типа. Часто «истинность» интерпретируется в них как синоним «правильности». Приведем несколько примеров подобной интерпретации.

Синтаксическая правильность. Соответствие языковых выражений выбранному синтаксису языка представления знаний. Как правило, ИС требуют синтаксической правильности входных конструкций.

Семантическая правильность. В модели предметной области должны присутствовать специальные процедуры, которые называют *процедурами ассимиляции*. Процедуры ассимиляции соотносят синтаксически правильные фреймы с текущим состоянием дедуктивного замыкания модели без учета истинных значений. Успешное выполнение процедур ассимиляции похоже на человеческое понимание, когда говорят: «Я понял точку зрения собеседника», т. е. о чем идет речь безотносительно к тому, согласен ли слушающий с этим. Осмысленными будем считать фреймы, по отношению к которым выполнимы процедуры ассимиляции.

Вопрос конструирования процедур ассимиляции далеко не тривиален. Как, в частности, воспринимать высказывания, для которых нет соответствующих процедур, и оценивать степень достаточности, авторитетности процедур ассимиляции? В свое время известный советский лингвист И. Ревзин показал, что в множестве поэтических текстов фраза Хомского «Зеленые идеи яростно спят» вполне может оказаться осмысленной (например, в стихотворении «Зеленые идеи яростно спят, ворочаются во сне. Зеленые идеи чего-то ворчат, нашептывают мне»).

Возможны случаи частичной осмысленности, а также тактика, при которой система на время допускает неосмысленные фреймы «в надежде», что впоследствии процедуры ассимиляции смогут выполняться успешно. Или происходит пробное включение фрейма в модель, после чего дальнейшая работа продолжается в предположении, что ничего плохого не произойдет. Если не возникает противоречий, то фрейм сохраняется, иначе происходит возврат и исследуется другой вариант ассимиляции, если он есть. В противном случае данный фрейм считается неосмысленным.

Модельная истинность связана с понятием вывода на сети (см. гл. 2). Вывод в СМ есть последовательное применение правил вывода из заданной системы правил. Он позволяет получать утверждения, ранее в базе не зафиксированные (*виртуальные утверждения*). Вывод позволяет дать ответ на вопрос о существовании некоторого виртуального утверждения в данном состоянии базы либо определить значения атрибутов некоторого утверждения. Наличие вывода расширяет диапазон решаемых задач, не перегружая модель проблемной области множеством явных и легко выводимых утверждений.

Введем еще два понятия: *дедуктивное замыкание модели* и *модельная непротиворечивость*. Первое есть модель, пополненная всеми ее виртуальными

утверждениями, а второе — модель знаний, дедуктивное замыкание которой не содержит пары утверждений, отрицающих друг друга. Операции вывода должны быть такими, чтобы их применение для получения виртуальных утверждений не приводило к нарушению свойства модельной непротиворечивости.

Модельно-непротиворечивым утверждением называется такое, включение которого в модель не нарушает ее свойства модельной непротиворечивости. *Модельно-истинным* называется утверждение, входящее в дедуктивное замыкание и имеющее абсолютный истинный статус $t\text{-val} \neq \text{Undefined}$. *Абсолютно истинным (ложным)* считается модельно-истинное утверждение, присутствующее в модели с абсолютным истинным статусом $t\text{-val} = \text{True (False)}$. Подобные утверждения характеризуются абсолютным истинным статусом. *Экспlicitно модельно-истинным* называется множество утверждений, входящих в модели явно. Введенные определения позволяют уточнить требования к операциям пополнения модели: пополняющие утверждения не должны быть модельно-истинными. Требование модельной непротиворечивости для больших моделей на практике трудно выполнимо.

На рис. 1.8 приведена структура конъюнктивной и открытой модели проблемной области (она выделена штриховкой и обозначена МПОБ). При переходе к дедуктивному замыканию этой модели, т. е. к множеству модельно-истинных утверждений, необходимо отнести все утверждения, истинный статус которых неизвестен, к тем или иным областям, возникающим при дедуктивном замыкании отдельных утверждений. Несколько различных вариантов такого отнесения показаны на рис. 1.8 (для утверждений A_1 , A_2 и A_3). При замыкании модели необходимо избегать появления противоречий. В связи с этим вводится специальный статус утверждений в МПОБ, который называется *Д-статус*. Утверждения, имеющие такой статус истинности, называются *Д-истинными* или *денотационно истинными*. Их истинность в рамках данной модели считается достаточной для решения интересующих систему задач. Д-статус определяется той измерительной процедурой, которая порождает модельное утверждение, или теми ограничениями исходного описания, которые зафиксированы при вводе этого утверждения в базу.

Можно считать, что утверждения с Д-статусом являются утверждениями об эмпирических фактах, а утверждения с абсолютным истинным статусом являются утверждениями некоторой теории.

Так, в базах знаний возникают как бы две истинные системы: *теоретическая* и *эмпирическая*. Теоретически истинные утверждения, как правило, не могут быть эмпирически обоснованы. Например, утверждение «Все люди смертны», не может быть проверено на практике. А утверждение «Все женщины, которых я встретил в этом городе, одеты в белое и, следовательно, тут все женщины ходят в белой одежде» может иметь статус *эмпирической истины*, но нет никаких оснований приписывать ему статус абсолютной, т. е. теоретической, истины.

Истинный статус помимо абсолютного может иметь частный, или относительный, характер, отражать различные точки зрения на систему утверждений, составляющую конкретную МПОБ. В этом проявляется основное свойство систем представления знаний: многоаспектное моделирование действительности. Замкнутая конъюнктивная абсолютно истинная МПОБ является «наивной» одновариантной копией фрагмента реального мира. В общем случае полезно поддерживать множественный истинный статус утверждений. Множественность истинных оценок является частным случаем механизма множественных точек зрения.

Точка зрения есть способ формирования структуры утверждений о ситуации, при котором вводятся особые утверждения — высказывания точек зрения, несущие в себе редактирующую систему атрибутов для этой ситуации. Результат редактирования ситуации с помощью таких утверждений и представляет собой точку зрения. При этом соблюдаются следующие правила доступа к утверждениям ситуации через утверждения точек зрения:

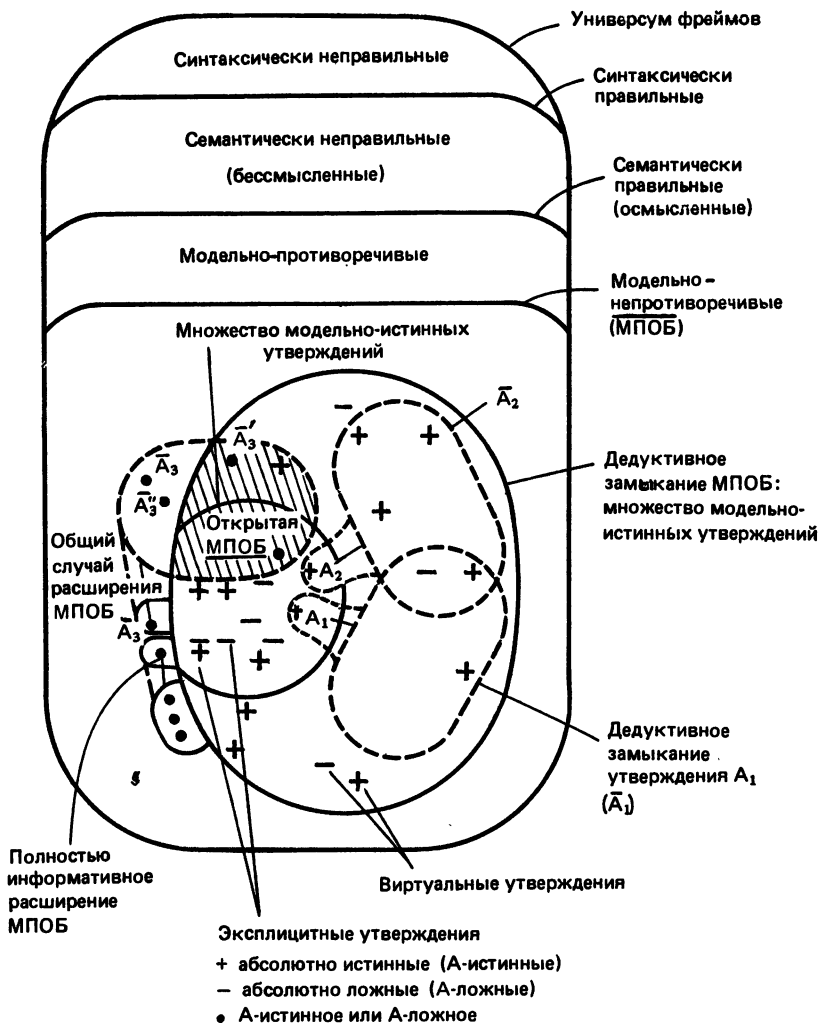


Рис. 1.8

атрибутивные пары утверждений точек зрения, атрибуты которых отсутствуют в списках атрибутов утверждений ситуации, виртуально к ним добавляются;

значения атрибутов утверждений точек зрения, имена атрибутов которых совпадают с именами атрибутов утверждений ситуации, замещают ранее имевшиеся значения;

с помощью специального значения (например, NIL) в редактирующем атрибуте можно исключить (экранировать) любое значение соответствующего атрибута утверждения.

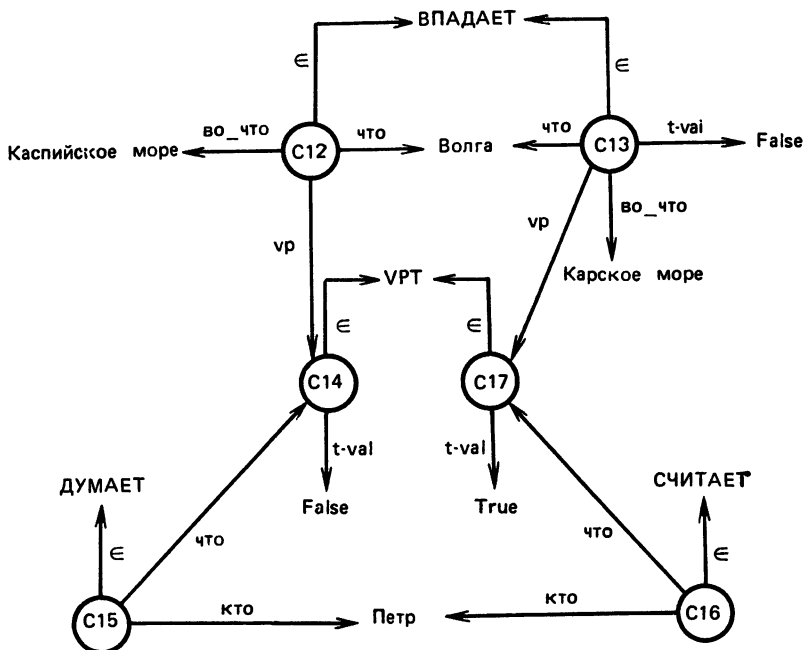


Рис. 1.9

Примем, что утверждения точек зрения будут иметь тип VPT и что утверждения точек зрения будут ссылаться на утверждения ситуации с помощью атрибута *vp*. При этом атрибуты, отображаемые на графе входящими дугами, будут помечаться знаком «—», а исходящие дуги маркироваться не будут. Абсолютный истинный статус утверждения будет задаваться с помощью атрибута *t-val* со значениями True, False и Undefined. По умолчанию всегда подразумевается истинный статус *t-val* ≠ True.

Утверждения, ссылающиеся на ситуации, в том числе с помощью точек зрения, будут называться *референциальными*. Они служат для спецификации, уточнения ментальных состояний и интенций субъектов либо иных активных действующих лиц, включенных в предметную область. Абсолютный истинный статус можно считать «точкой зрения» природы на состояние модели. К референциальным утверждениям относятся, в частности, ЗНАЕТ, СКАЗАЛ, СТРЕМИТСЯ К, МОЖЕТ, НРАВИТСЯ и г. д.

На рис. 1.9 приведена СМ, соответствующая тексту: «Волга впадает в Каспийское море, однако Петр думает, что это не так. Он считает, что Волга впадает в Карское море, что неверно». Факт C12 («Волга впадает в Каспийское море»), а также факты C15 и C16 являются (по умолчанию) истинными. Факту C13 («Волга впадает в Карское море») придан абсолютный истинный статус «Ложно». Факт C14 является точкой зрения на факт C12, приписывающей ему истинный статус «Ложно». Факт C15 («Петр думает, что...») является референциальным, замыкающимся на точку зрения C14. Аналогично факт C17 задает точку зрения на C13 как на истинный факт (так считает Петр). С точки зрения C17 кажется, что C13 имеет истинное значение «Истина».

Многоальтернативные представления в базе знаний

Простейшей логической структурой обладает замкнутая конъюнктивная база знаний — *псевдофизическая модель*. Все зафиксированные в ней факты считаются истинными; необходимости в отрицаниях нет. В открытых МПОБ возникает потребность в хранении и манипулировании фактами с Д-статусами. Утверждения о точках зрения позволяют реализовать множественный референциальный статус. Возможности МПОБ увеличиваются, если предусмотреть средства отображения многоальтернативных представлений, в частности OR и XOR множеств утверждений.

Рассмотрим один возможный способ реализации этих представлений. С этой целью вводится еще один тип вершин: логические связки BOOL. Специфицирующий атрибут *sr* будет определять характер связки, а связываемые множества утверждений будут указываться с помощью атрибута *s*.

Введем понятие базового логического уровня МПОБ, под которым будем понимать уровень псевдофизической модели с подразумеваемой конъюнктивностью по отношению к остальным утверждениям этого уровня. Статус Undefined исключает соответствующее утверждение из базового уровня. Приписывание атрибута *t-val* ≠ Undefined логической связке привязывает соответствующую структуру утверждений к базовому логическому уровню. Обратимся к примеру на рис. 1.10. Основу базового логического уровня изображенного фрагмента МПОБ составляет истинное (по умолчанию) утверждение C18 и ложное C19. В базовый уровень включена также XOR-выборка двух истинных утверждений

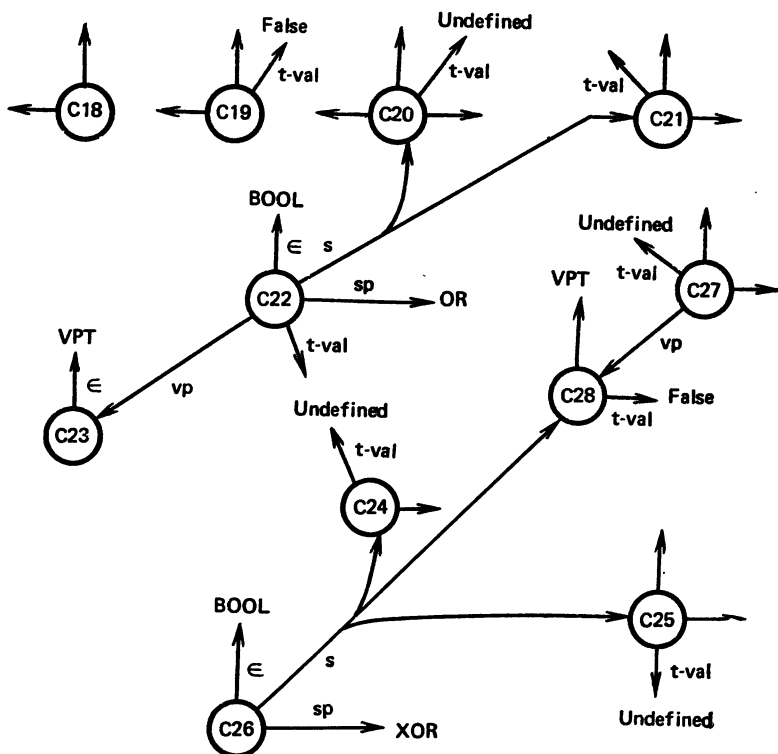


Рис. 1.10

C24 и C25 и одного ложного C27. Ложность последнего задается с помощью утверждения точки зрения C28. Смысл ситуации C24—C28 заключается в том, что с ее помощью зафиксирован текущий уровень знания в МПОБ о денотативной истинности утверждений C24, C25 и C27 относительно проблемной области: имеется в виду, что имеет место либо C24, либо C25, либо не имеет места C27. Ситуация C20—C23 не включена в базовый логический уровень МПОБ: истинность C20—C22 не определена, а истинная точка зрения C23 не является референтной ни по одному утверждению. Поэтому семантика ситуации C20—C23 с учетом характера логической связки C22 (OR) может быть передана следующей фразой: «Существует точка зрения (C23), согласно которой C20 или C21 или они вместе истинны».

Представление времени в сетевых моделях

Известны различные варианты построения логики времени [Prior, 1957; Ивин, 1970; Поспелов Д., 1975; Караваев, 1983]. Взаимосвязь временных понятий в БЗ с терминами, используемыми при коммуникации, проанализирована в [Литвицева и др., 1980; Кандрашина, 1983; Поспелов Д., 1986].

Существуют два основных типа *темпоральных фреймов*, т. е. фреймов, привязанных ко времени: *события и состояния*. Под событием понимается мгновенное (с точки зрения целей моделирования) изменение, переход в новый режим, включение либо отключение чего-либо, изменение значения параметра и т. п. Событие фиксируется в базе знаний в виде факта, имеющего единственный темпоральный атрибут датирования, определяющий момент его реализации. Например: «Германия напала на Советский Союз 22 июня 1941 года».

В качестве значения атрибута датирования могут выступать и недоопределенные значения, когда момент свершения события известен приблизительно (от ... до).

Состояние — это утверждение, привязанное к шкале времени и определенное на некотором *интервале существования*, задаваемом с помощью двух специальных маркеров — атрибутов начала (tb) и конца (te). Например: «Великая Отечественная война продолжалась с 1941 по 1945 год». Если отсутствует te, то считается, что $te = \infty$, если отсутствует tb, то $tb = -\infty$. Отсутствие tb и te одновременно ведет к распространению интервала существования на всю временную ось, т. е. рассматриваемый фрейм становится вневременным.

Рассмотрим пример: «Решив приготовить кофе, Николай налил в чайник воды и, пока он нагревался, смолот кофе. Потом залил его кипятком». Граф данной ситуации (рис. 1.11) включает события: C39 — «Решил приготовить кофе», C40 — «Николай налил воды (во что-то)» со своими атрибутами датирования: $t(C39) = C53$ и $t(C40) = C54$. C53 и C54 — десигнаты соответствующих моментов времени. Факты-состояния: C41 — «Вода греется», C44 — «Николай мелет кофе», C45 — «Залил водой кофе», привязаны к шкале времени с помощью соответствующих атрибутов для интервалов существования tb и te. Так же, как и в случае событий, вместо конкретных значений моментов времени, отсутствующих в исходном описании ситуации, здесь фигурируют их десигнаты: C42, C43, C47. При этом появляется возможность корректно привязать упоминаемые события и состояния к специфицированным моментам времени, которые могут впоследствии обрести и конкретные значения, задаваемые с помощью отдельных фактов. На рис. 1.11 использованы псевдоатрибуты VL, дающие значения десигнатов моментов времени C42 и C43.

Псевдоатрибутом называется сокращенная запись утверждения, лишенная метки и формально имеющая вид объектного атрибута. На схемах псевдоатрибуты выделяют двойными стрелками.

Нетривиальный вопрос, возникающий при работе с темпоральными фреймами, заключается в том, как отслеживать «судьбу» объектов. Принципы работы, использовавшиеся, в частности, в СМ типа «мультисеть» [Лозовский, 1978; Горячук и др., 1982], базировались на следующем постулате: «Объект существует с того момента и до того момента, пока он участвует хотя бы

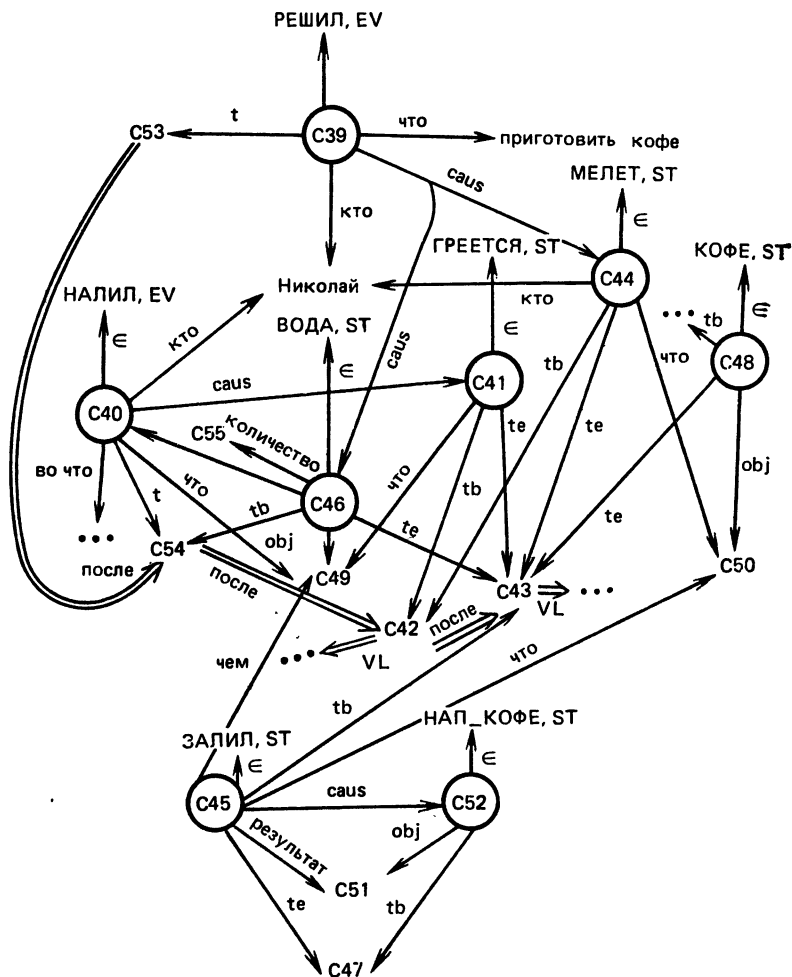


Рис. 1.11

в одном фрейме, упоминающем данный объект в качестве значения своих атрибутов». Другими словами, объект без свойств в этой системе существовать не может.

Работа с объектами-ресурсами требует особого внимания, если ресурс является частично используемым и (или) восполнимым (жидкости, газы, сыпучие тела, множества однородных предметов). Такими объектами на рис. 1.11 являются вода C49 и кофе C50. Существование этих объектов, их количества, объемы и прочие атрибуты фиксируются с помощью специальных фактов-состояний: C46 — Вода, C48 — КОФЕ (сухой) и C52 — НАПИТОК_КОФЕ. Объект является конкретным объемом воды, который с момента $tb(C46) = C54$ до момента $te(C46) = C43$ обладал оценкой количества C55. В указанный момент

он был полностью использован (C45) для приготовления кофе и прекратил свое существование в качестве самостоятельной сущности. После окончания заливки кипятка согласно диаграмме рис. 1.11 возникает НАПИТОК_КОФЕ (C52). Последовательность изменения состояния какого-либо М-объекта или величины представляется множеством фактов состояний, в которых непременно указывается десигнат нужного объекта или величины и необходимый набор атрибутов, уточняющих (количественно и (или) качественно) состояние данного М-объекта.

Для учета причинно-следственных зависимостей в фреймах приходится предусматривать *атрибуты каузации*. Так, на рис. 1.11 (атрибуты каузации представлены лишь частично, чтобы не загромождать диаграмму) следствием решения C39 приготовить кофе является появление определенной величины (C49) воды и помолка кофе. Наличие воды (C46) позволило ее налить во что-то и т. д.

Особенности сетевых моделей

Средствами баз знаний удобно реализовать механизм обобщения, или абстракции. Первый шаг на пути к абстракции — фиксация *типа фреймов*. Имя типа, например P_i , объединяет факты данного типа $F_{ik} \in P_i$, $k=1, \dots, E_i$.

Экстенсионалом фрейма EXT(P_i) будем называть множество всех фактов $\{F_{ik} \in P_i\}$, $k=1, \dots, E_i$, зафиксированных в базе данных МПОБ. Здесь E_i — мощность EXT(P_i). Экстенсионалы фреймов в процессе функционирования ИС могут меняться. Поэтому, строго говоря, следовало бы писать EXT(P_i, t).

Множество экстенсионалов образует *суперэкстенсионал* SEXT(P_i). При этом для любого состояния МПОБ $EXT(P_i) \subseteq SEXT(P_i)$.

Под *интенсионалом фрейма* P_i будем понимать функцию INT(p_i), которая вырабатывает множество фактов $\{F_j\} \subseteq SEXT(P_i)$, являющихся конкретизацией прототипа p_i фрейма P_i , либо пустое множество, если пересечение $\{F_j\}$ и SEXT(P_i) пусто (т. е. p_i на самом деле не является прототипом P_i).

Запись вида

$$SCH(P_i) = (P_i \dots \langle a_j : DOM(P_i, a_j) \rangle \dots), j=1, \dots, N_i,$$

будем называть *схемой фрейма* $P_i \in \{P_i, \dots, P_T\}$, где T — количество типов фреймов в МПОБ; a_j — имя j -атрибута фрейма P_i ; $DOM(P_i, a_j)$ — множество, называемое доменом, допустимых значений атрибута a_j фрейма P_i ; N_i — местность, или арность, фрейма P_i .

Факт $F_{ik} = (\langle \text{метка-факта} \rangle P_i \dots \langle a_j V_{ijk} \rangle \dots)$ является *конкретом схемы* SCH(P_i), если $V_j V_{ijk} \in DOM(P_i, a_j)$.

Прототипом p_i фрейма P_i будем называть фрейм, имеющий структуру факта F_{ik} , но в котором помимо терминальных $V_{ijk} \in DOM(P_i, a_j)$ допускаются V_{ijk} вида $(\langle \text{имя переменной} \rangle)$, которые будем называть *объявлением переменной*.

Вернемся к обсуждению понятия интенционала. Рассмотрим пример. Пусть $P_i = \langle \text{«сложить»} \rangle$, или $\langle + \rangle$, понимаемое в обычном арифметическом смысле. Тогда

$$SCH(+) = (+ a_1 : \text{INTEGER } a_2 : \text{INTEGER } a_3 : \text{INTEGER t-val : BOOLEAN}).$$

Пусть МПОБ в некотором своем состоянии содержит факты:

$$SEXT(+) = \{ (C67 + a_1 5 a_2 3 \quad a_3 8 \text{ t-val TRUE}) \\ (C68 + a_1 5 a_2 8 \quad a_3 11 \text{ t-val FALSE}) \\ (C69 + a_1 1 a_2 -9 \quad a_3 -8 \text{ t-val TRUE}) \}.$$

SEXT(+) представляет собой бесконечное множество, в котором каждой тройке целых чисел ставится в соответствие значение TRUE, если третье число равно сумме первого и второго, и наоборот. Ясно, что хранение EXT(+) большого смысла не имеет, а SEXT(+) хранить вообще невозможно. В то же

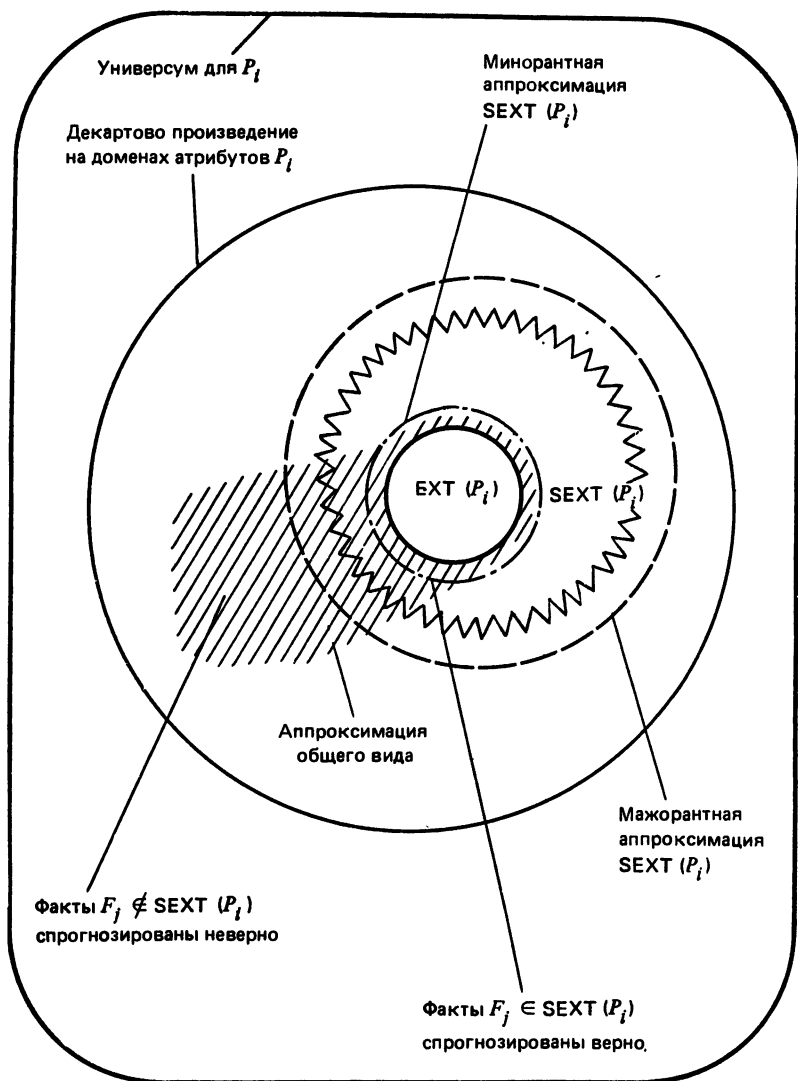


Рис. 1.12

время построить функцию $INT(+)$, используя обычные арифметические операции сложения, вычитания и механизм конкретизации переменных, объявленных в прототипе p_i , не составляет труда. Использование интенциональных описаний позволяет полностью виртуализовать соответствующие утверждения. Для фреймов, имеющих интенциональное описание, бессмысленно вводить в базу данных МПОБ конкретизирующие их факты: они либо не несут новой информации, либо противоречат модели. К сожалению, лишь для немногих представляющих практическую ценность типов фреймов удастся выписать

«чистый» интенционал. Однако полезной оказывается даже его аппроксимация. На рис. 1.12 показано соотношение теоретико-множественных диаграмм экстенционала, суперэкстенционала фрейма P_i и их различных аппроксимаций.

При описании схем отношений в базах знаний можно использовать декларацию доменов, что равносильно аппроксимации интенционала декларативным производением на доменах. Для этого в базу знаний помещаются описания фреймов, включающих декларации доменов. При этом несложно организовать синтаксический контроль информации, однако отличить, например, наборы атрибутов фактов фрейма сложения от фактов фрейма умножения невозможно. Учет помимо доменного взаимных соотношений значений атрибутов позволяет достичь лучших аппроксимаций. Из трех типов аппроксимаций (рис. 1.12): *мажорантной*, *минорантной* и *общего вида*, наиболее предпочтительна первая, не дающая ложных отказов в ассимиляции контролируемых фактов.

Сказанное требует включения в базу знаний особых объектов — *деклараций фреймов*, дающих концептуальное описание соответствующих понятий. Из анализа интенциональных аспектов деклараций фреймов вытекает полезность наличия декларации доменов для значений атрибутов и важная роль ограничительных выражений на совместные значения атрибутов, которые обеспечивают мажорирование суперэкстенционала фрейма. При декларациях целочисленных доменов часто достаточно иметь лишь интервальные типы. Но при этом операции поиска в СМ, опирающейся на значения, становятся непростыми, возникает большое число переборных с малым количеством релевантных результатов. Поэтому домены типа амперы, киловатты, километры или градусы полезно специфицировать даже в том случае, когда их численные диапазоны совпадают. Подобные домены иногда называют *интерпретируемыми*. С каждым из них могут быть связаны свои единицы измерения и процедуры перевода единиц измерения. В частном случае система интерпретируемых доменов может образовывать иерархию.

Ограничения на мощность множества значений атрибутов несут достаточно большую семантическую нагрузку. Наиболее ярко этот аспект проявляется в бинарных утверждениях. Рассмотрим фрейм (СУПРУГИ муж ... жена ...). Если ограничения на мощность множества значений атрибутов не предусмотрены, ничто не мешает появиться в базе данных, например, следующим фактам:

(C70 муж Иванов жена (:& Нина Марина Светлана))

(C71 муж (:& Петр Сидор Федор) жена Ольга).

Требования моногамии должны найти отражение во взаимно однозначном соотношении между множеством мужей и множеством жен. Описывая фрейм, например, стола, легко зафиксировать, что ножка может принадлежать лишь одному столу (или никакому): (0..1), а стол может иметь от одной до, скажем, шести ножек: (1..6).

Наличие *функциональных зависимостей* между атрибутами при построении баз знаний может быть использовано как синтаксический фильтр для контроля вводимой информации и частичной виртуализации соответствующего фрейма.

Упрощенным вариантом вычисляемых атрибутов являются значения по умолчанию, или *дефолты*, предусматривающие в декларациях фреймов наиболее типичные значения атрибутов, которые при вводе соответствующих фактов можно не задавать. Например, дефолтным цветом ворон естественно считать черный. При отклонениях от нормы допускается явное задание значения, которое в этом конкретном случае заместит дефолтное (ворона-альбинос).

В декларациях фреймов несложно указать подмножество атрибутов, обязательных для задания в фактах, а также указать, является ли фрейм замкнутым или открытым, т. е. допускающим иные атрибуты в фактах сверх указанных в декларации.

Если допустить работу с фактами недеklarированных фреймов (чисто экстенциональное задание коннотативной семантики, полностью альтернативное интенциональной спецификации), то такая система вряд ли будет удобной для пользователя-непрофессионала, но вполне приемлема для системных програм-

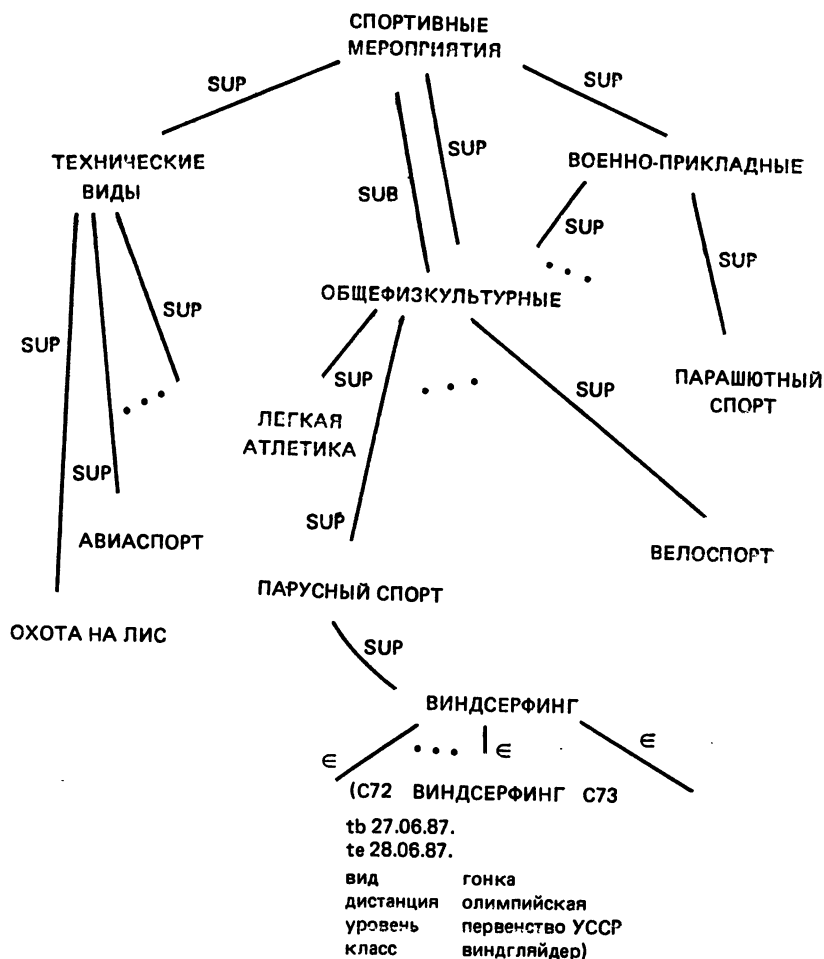


Рис. 1.13

мистов, которые готовы за максимальную гибкость системы терпеть последствия от ошибочности своего вмешательства в ее работу.

В работах по СМ значительное внимание уделяется *таксономическим структурам* — иерархиям абстрактных понятий, имеющим структуру дерева, корень которого представляет наиболее общее понятие, а остальные варианты — более частные понятия (подпонятия). На рис. 1.13 приведена таксономия понятия «спортивные мероприятия». Факты базы данных оказываются связанными, как правило, с наиболее детальными абстрактными понятиями таксономии отношением \in , ISA, «конкретизирует» и т. п. В примере конкретная фактуальная информация об одной из гонок зафиксирована в факте C72. Каждое абстрактное понятие таксономии, за исключением наиболее общего, наследует все свойства непосредственного надпонятия и добавляет к ним свои уточнения.

Отношения SUP (быть надпонятием) и ему обратное SUB (быть подпонятием) являются транзитивными.

Использование таксономических структур в СМ имеет как положительные, так и отрицательные стороны [Лозовский, 1982]. К положительным относятся: организация деклараций абстрактных понятий, в частности фреймов в виде таксономической структуры, существенно облегчает доступ к экстенционалам соответствующих понятий для любого уровня общности, а также дедукцию при ответах на запросы к системе, в которых фигурируют фреймы, включенные в таксономию (так, для примера на рис. 1.13 легко может быть найден ответ на требование указать все общефизкультурные мероприятия);

наличие таксономической структуры может служить основой для рассуждений по аналогии, выполнения индуктивных умозаключений; так, подпонятия некоторого абстрактного подпонятия в некотором смысле более близки, чем подпонятия разных понятий; эта особенность может использоваться в предположениях о формально не заданных свойствах некоторых объектов либо служить основанием для подсказок и уточнений при диалоге с пользователем в процессе задания деклараций понятий.

Недостатки таксономии заключаются в следующем. Не существует единых принципов построения классифицирующих структур. Вопрос о выделении подклассов решается для каждого уровня таксономического разбиения отдельно. Любые другие аспекты работы с МПОБ классификацией поддерживаться не будут. В подобной ситуации естественно использование мультииерархических таксономий, в которых отдельное абстрактное понятие может быть подпонятием не одного, а нескольких более общих понятий. Так, с одной стороны, квадрат — это равносторонний прямоугольник, а с другой — ромб, содержащий прямой угол. Однако при этом возрастает сложность работы: приходится иметь дело с графом классификации, а не с деревом.

Удобным практическим инструментом классификации понятий является *фасетная таксономия*, использованная, например, в системе H-P/REX [Абрахманов и др., 1990]. Идентификаторы (имена фреймов, значения атрибутов) в таких системах помечаются набором элементарных свойств из некоторого множества. Каждое такое свойство называется *фасетом*, или *элементарным таксоном*. Фасет — это символический атом, имеющий смысл пометки. Если имя помечено набором фасетов (F_1, \dots, F_N), то говорят, что оно входит в элементарные таксоны: (F_1, \dots, F_N). Другими словами, все имена, помеченные фасетом F_j , составляют его экстенционал.

На множестве фасетов задается *таксономическая структура*, т. е. вводятся определения основных таксонов, базирующиеся на определениях других таксонов и (или) фасетов. Определение таксона заключается в соотнесении с его именем набора имен включаемых в него таксонов (фасетов). Включение в данном случае происходит и интенционально, и экстенционально. На рис. 1.14 задан набор (R_1, \dots, R_7) имен понятий, помеченных фасетами F_1, \dots, F_5 . Экстенционал фасета F_1 включает единственное понятие R_1 , а фасета F_4 — множество (R_1, R_3, R_5). Таксон T_5 представляет собой объединение фасета F_2 и таксона T_2 или, что то же самое, фасетов (F_1, F_2, F_3, F_4). *Экстенционал таксона* представляет собой теоретико-множественное объединение экстенционалов составляющих его фасетов. Так, экстенционал таксона T_6 — это ($R_1, R_3, R_4, R_5, R_6, R_7$).

Возникает вопрос о целесообразности декларации фреймов, описывающих объекты. Из определения глобального свойства объекта следует, что отразить такое свойство в декларациях вряд ли легко. К тому же свойства объекта, как правило, ситуативны и динамичны. Если проблемная область ограничена и, следовательно, речь идет не о глобальном свойстве объекта вообще, а о его глобальном свойстве в рамках данной проблемной области, то это свойство может описываться достаточно просто. В ряде случаев оказывается полезным зафиксировать наибольшее число характерных из них в *объектной декларации*. Подобная декларация имеет два аспекта: интенциональный и экстенциональный. Интенциональный аспект проявляется при анализе состояния базы данных

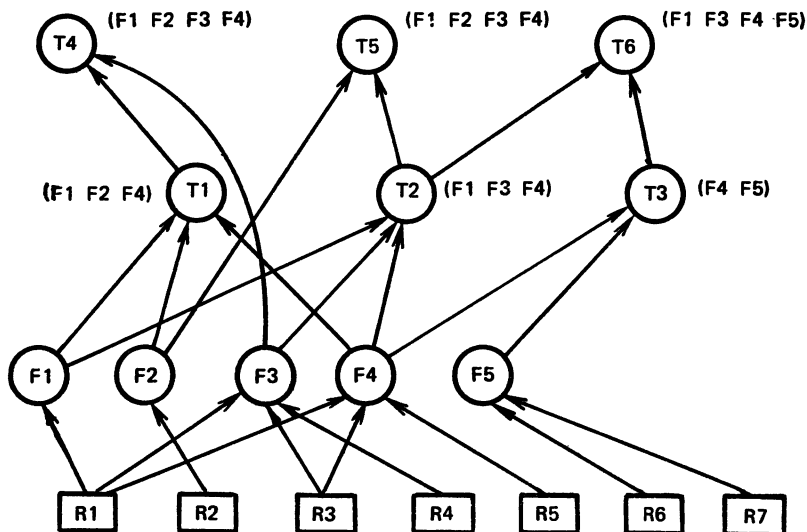


Рис. 1.14

МПОБ. Если обнаруживается факт, например, свидетельствующий, что некто X являлся участником профессионального спектакля, то можно утверждать, что X — артист, и включить X в экстенционал понятия АРТИСТ со всеми логическими следствиями, вытекающими из этого обстоятельства (получает зарплату в театре, может выезжать на гастроли и т. п.).

Экстенциональный аспект при работе с МПОБ проявляется при инициированном извне, отнесении некоторого десигната к определенному классу (или классам). При этом объективных оснований в самой МПОБ для такого отнесения в этот момент нет.

От деклараций объектов перейдем к *декларациям ситуаций*. Декларация ситуаций осуществляется двумя способами: *информационно-структурным* (ИС-метод) и *процедурным*. ИС-метод заключается в том, что декларация встраивается в структуру модели как ее составная часть. При этом используются ассоциативные свойства СМ, стимулирующие быстрый доступ к релевантной информации. Распознанные (либо экстенционально активированные) связи фактуальных структур с декларацией ситуации реализуются явно. Достоинством ИС-метода является высокое быстродействие и легкость организации режимов многоаспектного функционирования. Недостаток — повышенный расход памяти и статическая сложность результирующей сетевой модели.

Процедурный метод, как правило, реализуется в виде продукционных систем в рамках гетерархических программных архитектур [Ленат, 1975; Davis et al., 1975; PDIS, 1978; Лозовский, 1979 а, б; Нариньяни и др., 1984]. При этом система операционных модулей-продукций оснащается прототипами своей применимости (декларациями ситуаций). Прототипы применимости функционируют в интенциональном режиме, осуществляя распознавание фактуальных ситуаций. Обнаружение нужного соответствия позволяет одновременно извлечь из базы данных значения всех нужных атрибутов и использовать их в операционных частях продукционных правил для реализации логического вывода, трансформации МПОБ, выдачи сообщения пользователю. Связь ситуаций с их прототипами (декларациями) осуществляется в динамике, что сокращает расход памяти по сравнению с ИС-методом (но снижает быстродействие системы).

Используются и комбинированные системы, позволяющие получить в среднем лучшие результаты [Абдрахманов и др., 1990].

Подобные механизмы в СМ позволяют решать вопросы сохранения целостности модели в процессе ее функционирования. Под *сохранением целостности МПОБ*, понимается обеспечение входного контроля, исключающего попадание в систему модельно-противоречивой информации, а также обработка причинно-следственных трансформаций модели как реакции на модификацию модели извне (ввод, исключение или изменение фактов). При этом цепочки причинно-следственных изменений (*иммедиат-эффектов*) могут быть сколь угодно длинными.

Процедуры в сетевых моделях

Процедуры, реализуемые в базе, определяют, что и как надо делать для достижения нужных целей. В языках моделирования [Grinberger et al., 1968], в Плэнере [Hewitt, 1971; Пильщиков, 1983], Коннайвере [Mc Dermott et al., 1972] выделялись два основных типа условных запусков процедур (в терминологии Коннайвера): в ответ на запрос IF—NEEDED и как следствие фактуальных изменений в базе знаний (IF—ADDED и IF—DELETED). Часто механизмы такого запуска называют *демонами*. Первый тип условных запусков процедур — методы вывода, средства работы с виртуальными утверждениями, включая вычисление функционально зависимых атрибутов значений. В ряде языков (например, в Прологе или Плэнере) управление выбором правил вывода на каждом шаге не является детерминированным. Для организации выбора используются списки, в которых задаются статические или динамические приоритеты правил.

Второй тип запуска опирается на реализацию причинно-следственных (иммедиат-эффектных) трансформаций баз знаний. Любые изменения МПОБ вызываются реализацией отдельных событий, возникновением (IF—ADDED) и (или) завершением (IF—DELETED) тех или иных состояний. Тип факта, вызывающий определенный иммедиат-эффект, задается обычно в виде прототипа с использованием переменных. Прототип может включаться в декларации соответствующих фреймов и сопровождаться необходимым количеством проверок дополнительных условий, которые также фиксируются в виде прототипов, по которым осуществляется поиск в базе данных или базе знаний. Если поиск неудачен, иммедиат-эффект не выполняется; в противном случае он дает побочно значения дополнительных термов, которые наряду с атрибутами факта-инициатора могут использоваться в процессе вычисления нужной реакции. Например, в системах H-P/83 и H-P/REX [Лозовский, 1984; Абдрахманов и др., 1990] используются *логико-трансформационные правила* (ЛТП) (продукции, реализующие процедуры, снабженные двойным механизмом запуска: активатором и прототипом применимости). Активатор — это прототипное утверждение, включаемое в декларацию фрейма. Одновременно специфицируется тип иммедиат-эффекта: В — начало состояния; Е — конец состояния, ВС — начало текущее и ЕС — конец текущий. Два последних типа используются при планировании реакции на будущие события или изменения состояний. Фактическая инициация активатора произойдет, когда специфицированный момент в будущем станет текущим. Инициация активатора приводит к проверке условий применимости соответствующего ЛТП. Если условия выполняются, происходит запуск. В процессе проверок одновременно из БД извлекаются и сохраняются значения переменных — параметров процедурной части ЛТП, которые составляют фактический контекст его текущей активации.

Отметим, что СМ пока не имеют общей теории. В них много эвристик и вариантов решения. Интересные результаты, касающиеся СМ, связаны с работами по растущим пирамидальным сетям [Гладун, 1970, 1977]. Сетевые модели лежат в основе системы ПРИЗ [Тыгу, 1984]. Вычислительная модель системы ПРИЗ представляет собой результат автоматизированного синтеза сценария, у которого исходными данными являются множества входных и результирующих атрибутов. При синтезе устанавливаются такие связи между

фреймами отдельных зависимостей, при которых (при полноте исходной базы знаний) последовательно-параллельное выполнение синтезированной системы фреймов приводит к решению требуемой задачи. Плодотворные исследования ведутся по СМ специального вида — М-сетям и ансамблевым сетям — для решения задач управления и распознавания образов с ориентацией на высокоэффективную аппаратную реализацию [Амосов и др., 1975].

1.4. Продукционные модели

Д. А. Поспелов

Основные определения

Продукции наряду с фреймами являются наиболее популярными средствами представления знаний в ИС. Продукции, с одной стороны, близки к логическим моделям, что позволяет организовывать на них эффективные процедуры вывода, а с другой стороны, более наглядно отражают знания, чем классические логические модели. В них отсутствуют жесткие ограничения, характерные для логических исчислений (см. § 1.2), что дает возможность изменять интерпретацию элементов продукции.

В общем виде под *продукцией* понимается выражение следующего вида:

(*i*); Q ; P ; $A \Rightarrow B$; N .

Здесь *i* — *имя продукции*, с помощью которого данная продукция выделяется из всего множества продукций. В качестве имени может выступать некоторая лексема, отражающая суть данной продукции (например, «покупка книги» или «набор кода замка»), или порядковый номер продукции в их множестве, хранящемся в памяти системы.

Элемент Q характеризует *сферу применения продукции*. Такие сферы легко выделяются в когнитивных структурах человека. Наши знания как бы «разложены по полочкам». На одной «полочке» хранятся знания о том, как надо готовить пищу, на другой — как добраться до работы и т. п. Разделение знаний на отдельные сферы позволяет экономить время на поиск нужных знаний. Такое же разделение на сферы в базе знаний ИС целесообразно и при использовании для представления знаний продукционных моделей.

Основным элементом продукции является ее *ядро*: $A \Rightarrow B$. Интерпретация ядра продукции может быть различной и зависит от того, что стоит слева и справа от знака секвенции \Rightarrow . Обычное прочтение ядра продукции выглядит так: ЕСЛИ A , ТО B , более сложные конструкции ядра допускают в правой части альтернативный выбор, например, ЕСЛИ A , ТО B_1 , ИНАЧЕ B_2 . Секвенция может истолковываться в обычном логическом смысле как знак логического следования B из истинного A (если A не является истинным выражением, то о B ничего сказать нельзя). Возможны и другие интерпретации ядра продукции, например A описывает некоторое условие, необходимое для того, чтобы можно было совершить действие B .

Элемент P есть *условие применимости ядра продукции*. Обычно P представляет собой логическое выражение (как правило, предикат). Когда P принимает значение «истина», ядро продукции активизируется. Если P ложно, то ядро продукции не может быть использовано. Например, если в продукции «НАЛИЧИЕ ДЕНЕГ; ЕСЛИ ХОЧЕШЬ КУПИТЬ ВЕЩЬ X , ТО ЗАПЛАТИ В КАССУ ЕЕ СТОИМОСТЬ И ОТДАЙ ЧЕК ПРОДАВЦУ» условие применимости ядра продукции ложно, т. е. денег нет, то применить ядро продукции невозможно.

Элемент N описывает *постусловия продукции*. Они актуализируются только в том случае, если ядро продукции реализовалось. Постусловия продукции описывают действия и процедуры, которые необходимо выполнить после реализации B . Например, после покупки некоторой вещи в магазине необходимо

в описи товаров, имеющихся в этом магазине, уменьшить количество вещей такого типа на единицу. Выполнение N может происходить не сразу после реализации ядра продукции.

Если в памяти системы хранится некоторый набор продукции, то они образуют *систему продукции*. В системе продукции должны быть заданы специальные процедуры управления продуктами, с помощью которых происходит актуализация продукции и выбор для выполнения той или иной продукции из числа актуализированных.

В ряде ИС используются комбинации сетевых и продукционных моделей представления знаний. В таких моделях декларативные знания описываются в сетевом компоненте модели, а процедурные знания — в продукционном. В этом случае говорят о работе продукционной системы над семантической сетью [Нариньяни и др., 1984].

Классификация ядер продукции

Ядра продукции можно классифицировать по различным основаниям. Прежде всего все ядра делятся на два больших типа: *детерминированные* и *недетерминированные*. В детерминированных ядрах при актуализации ядра и при выполнении A правая часть ядра выполняется обязательно; в недетерминированных ядрах B может выполняться и не выполняться. Таким образом, секвенция \Rightarrow в детерминированных ядрах реализуется с необходимостью, а в недетерминированных — с возможностью. Интерпретация ядра в этом случае может, например, выглядеть так: ЕСЛИ A , ТО ВОЗМОЖНО B .

Возможность может определяться некоторыми *оценками реализации ядра*. Например, если задана вероятность выполнения B при актуализации A , то продукция (в простейшем случае продукция состоит лишь из ядра) может быть такой: ЕСЛИ A , ТО С ВЕРОЯТНОСТЬЮ p РЕАЛИЗОВАТЬ B . Оценка реализации ядра может быть лингвистической, связанной с понятием терм-множества лингвистической переменной, например: ЕСЛИ A , ТО С БОЛЬШОЙ ДОЛЕЙ УВЕРЕННОСТИ B . Возможны иные способы задания оценки реализации ядра.

Детерминированные продукции могут быть *однозначными* и *альтернативными*. Во втором случае в правой части ядра указываются альтернативные возможности выбора, которые оцениваются специальными *весами выбора*. В качестве таких весов могут использоваться вероятностные оценки, лингвистические оценки, экспертные оценки и т. п. (например, ЕСЛИ A , ТО ЧАЩЕ ВСЕГО НАДО ДЕЛАТЬ B_1 , РЕЖЕ B_2).

Особым типом являются *прогнозирующие продукции*, в которых описываются последствия, ожидаемые при актуализации A , например: ЕСЛИ A , ТО С ВЕРОЯТНОСТЬЮ p МОЖНО ОЖИДАТЬ B .

Дальнейшую классификацию ядер продукции можно провести, опираясь на типовую схему ИС (рис. 1.15). Если x и y обозначают любой из блоков рисунка (О, Д, З, Л), то ядро $A_x \Rightarrow B_y$ означает, что информация об A берется из блока x , а результат срабатывания продукции B посылается в блок y . Комбинации x и y , осмысленные с точки зрения ИС, отмечены в табл. 1.8 знаком «+».

Таблица 1.8

A \ B	О	Д	З	Л
	О	Д	З	Л
О				
Д	+	+	+	+
З	+		+	+
Л	+		+	+

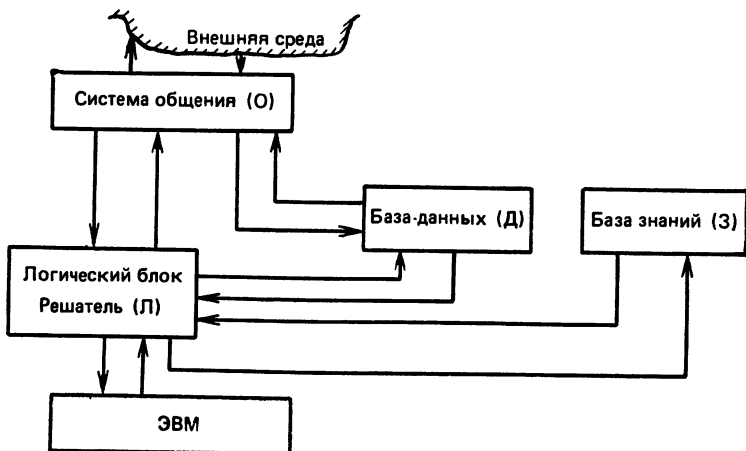


Рис. 1.15

Рассмотрим часто встречающийся тип продукции $A_3 \Rightarrow B_3$. В этом случае A_3 и B_3 представляют собой некоторые фрагменты информации, хранящейся в базе знаний. При сетевом представлении это могут быть фрагменты семантической сети, при логических моделях — формулы того или иного исчисления. Тогда смысл продукции $A_3 \Rightarrow B_3$ состоит в замене одного фрагмента базы знаний другим. Для актуализации этой продукции необходимо, чтобы в базе знаний существовал фрагмент, совпадающий с A . При поиске в базе знаний A играет роль образца, а процедура такого поиска называется *поиском по образцу*.

Для иллюстрации поиска по образцу предположим, что в базе знаний для представления знаний используется семантическая сеть (рис. 1.16,а) и продукция (рис. 1.16,б). Поиск A в базе знаний организуется различными способами. Можно, например, сначала искать вершину a . Если в базе знаний такой вершины нет, то поиск по образцу заканчивается неудачей. Если вершина a найдена, то ищутся все выходящие из нее дуги, помеченные отношением R_3 , так как в образце справа от этой дуги стоит вершина x , на месте которой в базе знаний может находиться любая вершина. Если из a не выходит ни одной дуги, помеченной отношением R_3 , то поиск по образцу заканчивается неудачей. Но если такие дуги есть, то происходит переход во все вершины, с которыми вершину a связывает отношение R_3 , т. е. возникает параллельный процесс поиска. В примере произойдет переход от вершины a к вершинам b и f , из которых начнется поиск выходящих из них дуг, помеченных отношением R_1 , ведущих в любую вершину, так как в образце далее стоит вершина, которой соответствует свободная переменная y . Далее процесс продолжается аналогичным образом. В примере поиск по образцу оказывается успешным. После нахождения A в семантической сети происходит замена, которая определяется правой частью образца. В результате возникает трансформированная сеть (рис. 1.16,в).

Продукция $A_d \Rightarrow B_3$ может соответствовать процедуре нахождения закономерностей по эмпирическим данным. Логический блок на основании просмотра и анализа данных выдвигает гипотезы о наличии закономерностей и, убедившись в их приемлемости и достаточной обоснованности, записывает их в базу знаний. Аналогично можно интерпретировать и иные типы продукций из табл. 1.8.

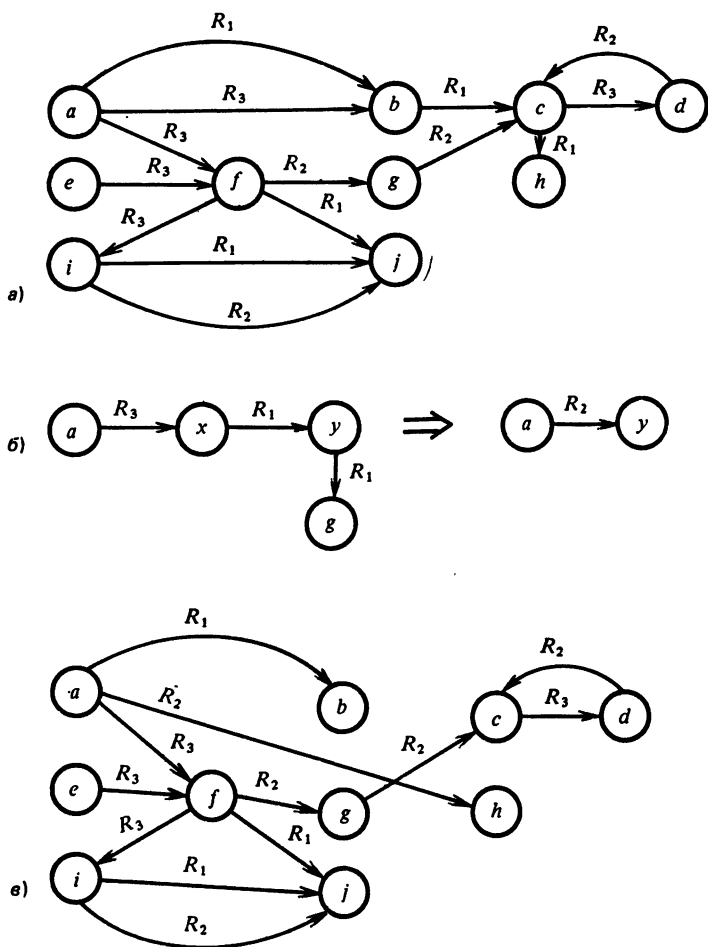


Рис. 1.16

Краткая история

Термин «продукция» принадлежит американскому логик Э. Посту [Post, 1943]. В понимании Поста в качестве продукции выступала только та ее часть, которую теперь называют ядром. Запись «ЕСЛИ A , ТО B » трактовалась как оператор замены цепочки A цепочкой B в некотором входном слове, т. е. продукции были тем, что позже стали называть подстановками и использовать при описании различных уточнений понятия алгоритма. Например, подстановки являются основным оператором в нормальных алгоритмах Маркова. Поэтому в языке программирования Рефал, основанном на идее такой модели уточнения понятия алгоритма, оператор подстановки играет основную роль. Идея продукции используется в языках логического программирования, нашед-

ших применение в ИС. Среди языков подобного типа наиболее известен Пролог. Продукционный подход стал стилем нового этапа программирования.

Системы продукции широко распространены в экспертных системах [Алексеева и др., 1984; Микучич, 1985].

Популярность продукционных моделей определяется несколькими факторами.

1. Подавляющая часть человеческих знаний может быть записана в виде продукции.

2. Системы продукции являются модульными. За небольшим исключением удаление или добавление продукции не приводит к изменениям в остальных продукциях.

3. При необходимости системы продукции могут реализовать любые алгоритмы и, следовательно, способны отражать любое процедурное знание, доступное ЭВМ.

4. Наличие в продукциях указателей на сферу применения продукции позволяет эффективно организовать память, сократив время поиска в ней необходимой информации. Классификация сфер может быть многоуровневой, что еще более повышает эффективность поиска знаний, так как позволяет наследовать информацию в базе знаний.

5. При объединении систем продукции и сетевых представлений получают средства, обладающие большой вычислительной мощностью (см. § 1.3).

6. Естественный параллелизм в системе продукции, асинхронность их реализации делают продукционные системы удобной моделью вычислений для ЭВМ новой архитектуры, в которой идея параллельности и асинхронности является центральной.

Продукционные модели имеют по крайней мере два недостатка. При большом числе продукции становится сложной проверка непротиворечивости системы продукции. Это заставляет при добавлении новых продукции тратить много времени на проверку непротиворечивости новой системы. Из-за присущей системе недетерминированности (неоднозначного выбора выполняемой продукции из фронта активизированных продукции) возникают принципиальные трудности при проверке корректности работы системы. Считается, что если в ИС число продукции достигает тысячи, то мало шансов, что система продукции во всех случаях будет правильно функционировать. Именно поэтому число продукции, с которым, как правило, работают современные ИС, не превышает тысячи.

Управление системой продукции

При выполнении условия применимости ядер продукции для группы продукции возникает дилемма выбора той продукции, которая в данной ситуации будет активизирована. Решение этой задачи возлагается на систему управления системой продукции. Если, например, ИС реализована на ЭВМ с параллельной архитектурой, то из *фронта готовых продукции* может выбираться не одна продукция, а столько, сколько параллельных ветвей может одновременно в данной ситуации выполнять ЭВМ. Но независимо от количества актуализированных продукции задача альтернативного выбора остается.

Задача управления характерна не только для систем продукции, но и для всех систем, где необходимо выполнение параллельных процессов с учетом имеющегося технического ресурса [Поспелов Д., 1972; Варшавский и др., 1984]. Возможны два пути ее решения: *централизованный* и *децентрализованный*. При централизованном управлении выполнением продукции решение об актуализации принимается специальной системой управления, а при децентрализованном определяется складывающейся в этот момент ситуацией. Если порядок выполнения продукции произволен, то решение определяется текущей ситуацией или системой управления; если порядок важен, то в продукциях должна содержаться информация о требованиях к этому порядку. Если в условиях продукции указывается имя продукции, которая должна выполняться после данной, система продукции превращается в обычную программу для ЭВМ, т. е. реализует некоторый алгоритм.

Опишем несколько стратегий управления выполнением продукций.

1. *Принцип «стопки книг».* Основан на идее, что наиболее часто используемая продукция является наиболее полезной. Готовые продукции как бы образуют «стопку», в которой порядок определяется накопленной частотой использования продукций в прошлом. На самом верху «стопки» находится продукция, которая использовалась чаще всех. При актуализации некоторого фронта готовых продукций для исполнения выбирается та продукция (или те продукции при наличии параллельных технических устройств), у которой частота использования максимальна. Подобный принцип управления особенно хорош, когда частота исполнения подсчитывается с учетом некоторой ситуации, в которой ранее исполнялась продукция, и это исполнение имело положительную оценку. При такой обратной связи метод стопки книг может превратиться в обучающуюся процедуру, адаптирующуюся к тем задачам, которые возникают во внешней среде (см. гл. 6 и 7). Управление по принципу стопки книг целесообразно применять, если продукции относительно независимы друг от друга, например, когда каждая из них есть правило вида «ситуация (А) \Rightarrow действие (В)». Именно такой случай имеет место в планирующих системах для роботов (см. § 7.1).

2. *Принцип наиболее длинного условия.* Заключается в выборе из фронта готовых продукций той, у которой стало истинным наиболее «длинное» условие полезности ядра. Этот принцип опирается на соображение «здорового смысла», что частные правила, относящиеся к узкому классу ситуаций, важнее общих правил, относящихся к широкому классу ситуаций, так как первые учитывают больше информации о ситуации, чем вторые. Трудность использования данного принципа состоит в том, что надо заранее упорядочить условия по вхождению друг в друга по отношению «частное — общее». Исследование систем продукций, в которых продукции упорядочены подобным образом, привело к появлению понятия К-систем, оказавшихся более богатой моделью процессов, чем классические алгоритмические модели [Кузнецов, 1986], однако не гарантирующих однозначного результата, получаемого за конечное число шагов, который является обязательным для алгоритмических систем. Управление по принципу наиболее длинного условия в продукциях, образующих фронт готовых продукций, целесообразно применять в тех случаях, когда знания и сами продукции хорошо структурированы привязкой к типовым ситуациям, на которых задано отношение типа «частное — общее».

3. *Принцип метапродукций.* Основан на идее ввода в систему продукций специальных *метапродукций*, задачей которых является организация управления в системе продукций при возможности неоднозначного выбора из фронта готовых продукций. Приведем пример использования метапродукций, заимствованный из американской экспертной системы MYCIN-TEIRESTAS, предназначенной для диагностики инфекционных заболеваний.

ЕСЛИ	инфекция есть pelvic-abscess и имеются продукции, входящие в состав фронта, в которых в условии А упоминается grampos-rods
ТО	продукции, у которых в А имеется enterobacteriaceae, следует активизировать раньше, чем продукции, содержащие в А grampos-rods

В примере метапродукция опирается на факт вхождения определенных продукций во фронт готовых продукций. Конечно, условием, записанным в метапродукции, может выступать и некоторое утверждение о невхождении определенных продукций во фронт готовых продукций.

4. *Принцип «классной доски».* Основан на идее спусковых функций [Поспелов Д., 1972]. При реализации принципа «классной доски» в ИС выделяется специальная рабочая область памяти — аналог классной доски, на которой мелом пишут объявления и стирают их при необходимости тряпкой. На этой «доске» параллельно выполняющиеся процессы находят информацию, инициирующую их запуск, на нее же они и выносят информацию о своей работе, которая может оказаться полезной для других процессов.

Как правило, на «классной доске» выделены специальные поля для формирования условий применимости ядер продукции, различные для разных сфер применения продукции, специальные поля для записи результатов срабатывания продукции и для записи постусловий, если они адресованы другим продукциям. С принципом «классной доски» может комбинироваться принцип управления с помощью метапродукций, ибо он требует проверки некоторых условий, которые фиксируются в рабочем поле памяти, а также другие принципы управления.

Если система продукции работает над некоторой сетевой моделью в базе знаний, то необходимо (особенно при альтернативных продукциях) принимать специальные меры защиты от «порчи знаний» работающими продукциями. Для этого также используется рабочее поле памяти, куда временно переносится фрагмент знаний, которым оперируют продукции. Этот фрагмент фиксируется на специальном поле «классной доски».

5. *Принцип приоритетного выбора.* Связан с введением статических или динамических приоритетов на продукции. Статические приоритеты могут формироваться априори на основании сведений о важности продукционных правил в данной проблемной области. Эти сведения, как правило, представляют собой информацию, извлекаемую из эксперта (см. § 2.1). Динамические приоритеты вырабатываются в процессе функционирования системы продукции и могут отражать, например, такой параметр, как время нахождения продукции во фронте готовых продукции. К данному типу управления относится задание последовательности приоритетов с помощью специальной каузальной семантической сети [Feigenbaum et al., 1971]. В этом случае задается некоторый каузальный сценарий (см. § 1.5), движение по которому определяется складывающимися ситуациями и в каждой вершине которого задана функция выбора очередной продукции из фронта готовых продукции.

Если продукции работают над семантической сетью, приоритеты могут формироваться не в системе продукции, а в сети [Нильсон, 1985].

Как и при логическом выводе (см. § 2.5), существуют два типа выполнения систем продукции: *прямой и обратный*. В первом случае поиск идет от левых частей продукции, т. е. проверяются условия A и актуализируются те продукции, для которых A имеет место; во втором — от изначально заданных B , по которым определяются необходимые для B значения A , которые, в свою очередь, отождествляются с правыми частями ядер продукции в системе. Два этих метода часто называют *восходящим и нисходящим* методами реализации системы продукции.

6. *Управление по именам.* Основано на задании для имен продукции, входящих в некоторую систему, некоторой формальной грамматики или другой процедуры, обеспечивающей сужение фронта готовых продукции и выбор из него очередной продукции для выполнения.

Например, пусть система продукции представлена четырьмя простейшими продуктами, состоящими лишь из ядерных частей: (а) $A \Rightarrow B$; (б) $B \& D \Rightarrow A$; (в) $A \vee B \Rightarrow D$; (г) $D \Rightarrow C$. В таком виде система продукции явно недетерминированна. Если выполняется A , то фронт готовых продукции включает продукцию с именами (а) и (в), а если выполняются B и D , то три последние продукции. Для устранения подобной недетерминированности может быть введена некоторая грамматика для имен продукции: (а) \Rightarrow (б); (в) \Rightarrow (б); (а) \Rightarrow (г). Тогда если в некоторый момент была выполнена продукция с именем (б), то новые продукции выполняться не будут. Если же в некоторый момент выполнялась продукция с именем (а), то после нее смогут выполняться продукции с именем (б) или (г). Но левые части ядер в этих продукциях таковы, что неоднозначность возникнет лишь в случае, когда B и D одновременно выполнены. С помощью грамматики для имен продукции можно задать однозначный алгоритмический процесс.

Продукционным моделям не хватает строгой теории. Пока в них царит эвристика. При задании модели проблемной области в виде совокупности продукции нельзя быть уверенным в ее полноте и непротиворечивости. Причина неудач создания теории кроется в расплывчатости понятия продукции, в той

интерпретации, которая приписывается ядру, а также в различных способах управления системой продуктов. Переход к алгоритмической схеме мало что дает, так как при этом исключается основное свойство продукционных систем — их модульность и происходящая из этого асинхронность и параллельность выполнения продуктов в системе.

Следует ожидать, что теория продукционных систем появится не ранее, чем будут созданы эффективные теории параллельных информационных процессов [Карпов, 1987]. Но некоторые результаты можно получить на основе существующих методов оптимального управления [Поспелов И. и др., 1987], введя ряд ограничений на понимание продуктов и систем продуктов.

Другая проблема практического применения продукционных моделей — переход от статических систем к динамическим, меняющим состав продуктов в системе или перестраивающих алгоритм управления выбором продуктов из фронта готовых продуктов в зависимости от текущих ситуаций. Такие системы продуктов называются *адаптивными*, или системами продуктов *реального времени*. Во многих приложениях быстрота срабатывания продуктов системой имеет решающее значение [Clema et al., 1985; Stefic, 1985]. Достичь этого можно только за счет гибкого набора стратегий управления, среди которых должны быть стратегии различного уровня сложности и быстроты срабатывания [Sorgels, 1985]. Ряд авторов считает, что прототипом такого управления может стать управление, которое было использовано в программе игры в шашки [Samuel, 1959, 1967]. Ее идеи были реализованы в проекте CONSIM (моделирование конфликтов), который ставил своей целью разработку системы продукционных правил для принятия решений при управлении воздушным боем. Для этого проекта была создана система EMULAR, реализующая ряд функций динамической продукционной системы. Для повышения скорости работы таких систем разрабатывались специальные средства аппаратной поддержки [Stolfo et al., 1982; Masnio et al., 1983; Shaw, 1985].

1.5. Сценарии

Л. В. Литвинцева

Основные определения

Особую роль в системах представления знаний играют стереотипные знания, описывающие известные стандартные ситуации реального мира. Такие знания позволяют восстанавливать информацию, пропущенную в описании ситуации, предсказывать появление новых фактов, которых можно ожидать в данной ситуации, устанавливать смысл происхождения ситуации с точки зрения более общего ситуативного контекста.

Для описания стереотипного знания используются различные модели. Среди них наиболее распространенными являются сценарии. *Сценарием* называется формализованное описание стандартной последовательности взаимосвязанных фактов, определяющих типичную ситуацию предметной области. Это могут быть последовательности действий или процедур, описывающие способы достижения целей действующих лиц сценария (например, обед в ресторане, командировка, полет самолета, поступление в вуз). В ИС сценарии используются в процедурах понимания естественно-языковых текстов, планирования поведения, обучения, принятия решений, управления изменениями среды и др.

Краткая история

Впервые понятие сценария было введено Р. Шенком и Р. Абельсоном [Schank et al., 1975] при разработке новых средств понимания истории. Сценарии в их системе понимания представлялись в виде фреймоподобных структур и использовались для связывания событий истории. Каждый сценарий состоял из набора

словотворения и их значений, описывающих роли, причины и последовательности сцен, которые, в свою очередь, являлись последовательностью определенных действий. Слот «роль» задавал исполнителей сценария, слот «цель» — мотивировку предпринимаемых действий.

Приведем пример представления сценария.

⟨сценарий: ресторан

роли: посетитель, официант, кассир

цель: принятие пищи, чтобы насытиться и получить удовольствие

сцена 1: вход в ресторан

войти в ресторан

осмотреть места

мысленно выбрать свободное место

пройти к свободному столику

сесть

сцена 2: заказ

взять меню

прочитать меню

решить, что заказать

заказ меню официанту

сцена 3: прием пищи

получение пищи

съедение пищи

сцена 4: уход

просьба рассчитать

получение чека

движение к кассиру

передача денег кассиру

выход из ресторана⟩.

Каждая последовательность действий в сценах обладает свойством каузальных цепочек: всякое предшествующее действие создает условия для совершения последующего действия. Такие сценарии были использованы в программе вывода SAM по связыванию частей текста в процессе понимания истории [Schank et al., 1975].

В [Поспелов Д., 1981, 1986; Литвинцева, 1985] сценарии используются для пополнения знаний о ситуации. Так называется процедура обогащения входной информации сведениями, хранящимися в памяти системы. В [Поспелов Д., 1986] сценарий представляется некоторой сетью, вершинам которой соответствуют факты, а дугам — связи, описывающие отношения специального типа, например «причина — следствие», «цель — подцель», «часть — целое».

На рис. 1.17 приведен пример сценария, в котором в качестве связей между вершинами сети выступает причинно-следственное отношение. Вершины сети задают следующие факты: Ф1 — станок простаивает; Ф2 — на рабочем месте нет рабочего; Ф3 — станок неисправен; Ф4 — в цехе нет заготовок; Ф5 — обеденный перерыв; Ф6 — рабочий покинул станок в рабочее время; Ф7 — рабочий находится в столовой.

В [Литвинцева, 1985] сценарии рассматриваются как средство представления проблемно-зависимых каузальных знаний и задаются в виде фреймподобных списочных структур. Значения слотов в таких сценариях отражают следующую семантику: роли сценария (деятели и участники сценария), цели и мотивы поведения деятеля и участников, ключевое событие сценария, послышки и следствия ключевого события, место, время, средства проведения сценария, побочные действия, закономерности.

В [Литвак и др., 1981] используются сходные со сценариями структуры знаний, называемые *гиперсобытиями*. Гиперсобытия описывают типичные ситуации из области юриспруденции (кража, грабеж, продажа и т. п.). С их помощью в системе ТАРЛЮС происходит анализ связанных текстов, в результате которого выявляются соответствующие тексту гиперсобытие, его участники и их роли.

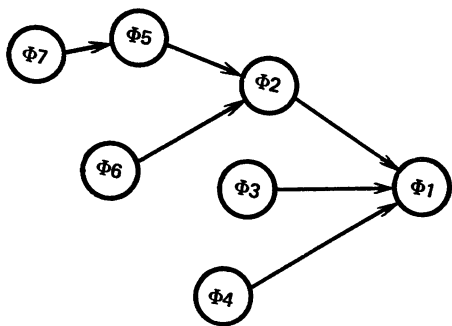


Рис. 1.17

Кроме гиперсобытий в ИС используются такие структуры высокого уровня представления, как MOP, TOP, TAU, PLOTUNIT, STORYPOINTS, KCI.

MOP (Memory Organization Packets) — пакет организации памяти, в котором связываются знания из разных фрагментов (эпизодов) реальной картины мира [Schank, 1982]. Функции MOP: помощь в организации сценариев; помощь в организации вывода и нахождения шаблона; управление поиском соответствующей информации. MOP отличаются от сценариев двумя основными параметрами: количествен-

ном — охватывают гораздо больший объем знаний; уровнем обобщения знаний — включают такие категории, как цели и мотивы. Кроме того, сценарии ограничены последовательностью действий, происходящих в одном глобальном контексте. MOP могут содержать информацию относительно нескольких глобальных ситуативных контекстов. С каждым MOP ассоциируется набор MOP-связей, которые определяют отношения между ними. Примером использования таких структур знаний является система BORIS [Lehnert et al., 1981], понимающая короткие связанные тексты.

TAU (Thematic Abstraction Unit) абстрактные тематические структуры, разработанные для понимания различных пословиц и поговорок [Dyer, 1982]. В TAU содержится информация о ситуациях, встречающихся в пословицах и поговорках; о возможных эмоциональных реакциях их героев; о «морали» или «соли» поговорок.

TOP (Thematic Organization Point) — точки тематической организации, являющиеся высокоуровневыми независимыми от предметной области структурами знаний, описывающими сходство между ситуациями в различных областях [Schank, 1982]. Назначение TOP: предсказание последствий встретившейся ситуации; выявление информации об одной ситуации, применимой к другой; распознавание известного сюжета в новой форме; обоснование взаимоотношений участников TOP.

PLOTUNIT — единицы сюжета, описывающие знания об эмоциональных реакциях и состояниях (так называемых состояниях аффекта) действующих лиц [Lehnert et al., 1982]. Событиям в PLOTUNIT приписываются две оценки: положительная и отрицательная в зависимости от эмоциональной реакции действующего лица на события. В PLOTUNIT содержатся правила, которые указывают, как можно связывать между собой разные аффектные состояния. Рассматриваются четыре типа связей. Одна из них — каузальные отношения между отрицательными событиями и аффектными состояниями — описывает мотивацию поведения действующего лица.

STORYPOINTS — точки истории, представляющие собой структуры, в которых в обобщенном виде описываются основные компоненты истории: содержание истории (совокупность событий) и ее «интерес» или общий смысл (совокупность точек истории) [Wilensky, 1983]. Каждая «точка истории» задает общий контекст (один или несколько), в котором происходит восприятие сюжета истории. С их помощью можно предсказывать появление новых событий, которых следует ожидать в данном контексте, производить умозаключения относительно взаимосвязи событий, описывать основные моменты (резюме) текста истории. Представление знаний в виде STORYPOINTS использовалось в программах PAM (Plan Applier Mechanism) и FAUSTUS (Frame Activated Unified Story Understanding System), понимающих рассказы.

Каузальные сценарии

Каузальные сценарии (КСЦ) разработаны для представления проблемно-зависимых каузальных знаний о событиях, действиях и процедурах [Кандрашина и др., 1989]. КСЦ задает в обобщенном и структурированном виде типичную последовательность действий (или процедур) в заданной предметной области и описывается в виде фрейма следующего вида:

(КСЦ имя:

имя слота₁ (значение слота₁);

имя слота₂ (значение слота₂);

...
имя слота_n (значение слота_n)).

Имена слотов (сценария) отражают следующие понятия: деятель и участники сценария, цели и мотивы деятеля и участников, время, место, средства реализации сценария, ключ, посылки, следствия, побочные действия закономерности, системное имя сценария.

Формально «значение слота» описывается в нотации Бэкуса — Науэра следующим образом:

<значение слота> :: = <спецификация значения слота> :

<значение> | <спецификация значения> : <последовательность значений> | NIL

<спецификация значений слота> :: = $n|s|p|C|f|o|low|sys$

<значение> :: = «имя»

<последовательность значений> :: = (<значение>, <значение>, ..., <значение>) | (<значение> R_1 <значение> R_2 ...)

Спецификация значения слота указывает класс значений данного слота. Символы n , s , p , C , f , o , low , sys обозначают следующие классы значений: числа, субъекты действий, события, сценарии, процедуры, объекты, закономерности, системные имена. Символы R_1 , R_2 , ..., R_k задают временные или каузальные отношения. Значение слота NIL говорит о его неопределенности.

Рассмотрим пример сценария

(КСЦ «посадка на рейс»:

№ рейса (n : 861);

деятель (s : дежурная);

участники (s : пассажиры);

цель деятеля (p : нахождение пассажиров в самолете);

цель участника (p : нахождение пассажиров в конечном пункте полета);

время (f : расчетное время проведения посадки);

посылки (CЦ: (регистрация пассажиров R_1 сбор на посадку R_1 прохождение контроля R_1 подача автобуса R_1 подача трапа));

ключ (p : посадка в самолет);

следствия (p : нахождение пассажиров на самолете);

побочные действия (CЦ: (формирование багажа R_1 погрузка багажа на рейс));

закономерности (low : (схема 1, схема 2, схема 3));

системное имя (sys : CЦ 3)).

Здесь R_1 — отношение «быть раньше». Слот «ключ» задает основное событие, определяющее тип ситуации. Реализация ключевого события обеспечивает достижение цели деятеля и участников сценария. Слот «посылки» описывает необходимые условия реализации сценария. В посылках содержится последовательность действий, которые надо выполнить, чтобы создать необходимые условия для осуществления ключевого события. Слот «следствия» задает результаты его выполнения. Слот «побочные действия» описывает действия, реализующиеся параллельно с выполнением действий в посылках сценария. Сценарий считается завершенным, если реализовано ключевое событие и достигнута цель деятеля.

Кроме КСЦ встречаются и иные типы сценариев. Наиболее распространенными являются сценарии в виде *деревя целей и классифицирующие сценарии*. В сценариях первого типа описывается, как некоторая цель может быть декомпозирована в систему подцелей. Такие сценарии применяются при планировании решений (см. § 7.1). Классифицирующие сценарии используются при обобщении знаний (см. § 2.3) и представляют собой сети, между вершинами которых имеются отношения типа «часть — целое», «элемент — класс» или «род — вид».

1.6. Ленымы

Е. Ю. Кандрашина

Основные определения

В искусственном интеллекте принято выделять три парадигмы представления и обработки знаний: логическую, структурную и процедурную. К первой относятся языки логического программирования и системы продукций, ко второй — семантические сети и фреймы, последняя представлена так называемыми функциональными сетями [Гуцу, 1984; Нариньяни, 1986а], а также частично отражена в языках фреймового типа.

Опыт разработки ИС убеждает в том, что языки представления знаний (ЯПЗ), разрабатываемые в рамках какой-либо одной парадигмы, т. е. базирующиеся на одной концепции обработки знаний, хороши для решения задач в простых, однородных предметных областях. При переходе к сложным моделям, включающим знания различных типов, возникает необходимость совмещения в одном ЯПЗ различных концепций. В то же время на характер ЯПЗ при его разработке влияет конфликтность двух точек зрения: разработчика и пользователя (инженера знаний). В первом случае возникает требование максимальной простоты и однородности языка, во втором — удобства (в частности, наличие развитых средств поддержки методологии создания моделей предметных областей).

Рассмотрим модель представления знаний, в которой этот конфликт в какой-то мере устранен. В основу модели положена двухуровневая схема, состоящая из *языка спецификации знаний (L-язык)* и *базовой формальной системы (БФС)*. БФС представляет собой самостоятельный ЯПЗ, который объединяет все три концепции обработки знаний.

Семантика L-языка полностью описывается в терминах БФС. Как и всякий ЯПЗ, он включает средства для формирования модели предметной области (МПОБ) и подязык для спецификации конкретных фактов, сигнатура которого формируется в процессе ввода МПОБ. Специфической единицей L-языка является *ленема* — конструкция, задающая схему описания понятия. Спецификации МПОБ и конкретных фактов в форме выражений L-языка существуют только на внешнем, пользовательском уровне, а собственно хранение и обработка информации осуществляются на уровне БФС.

Базовая формальная система

Базовая формальная система состоит из трех компонентов: *библиотеки понятий, функционально-семантической сети (фс-сеть)* и продукционной системы над фс-сетью.

Библиотека содержит описания классов (сортов) объектов, отношений и функций. Описания понятий включают два уровня: декларативный и интерпретационный (последний может отсутствовать). На первом вводится обозначение понятия, на втором — интерпретация; для класса объектов — носитель (множество значений), возможно, неполный; для отношений и функций — способ (процедура) их «вычисления». Описание функции дополнительно несет информацию о существовании и единственности ее значения при всяком фиксированном на-

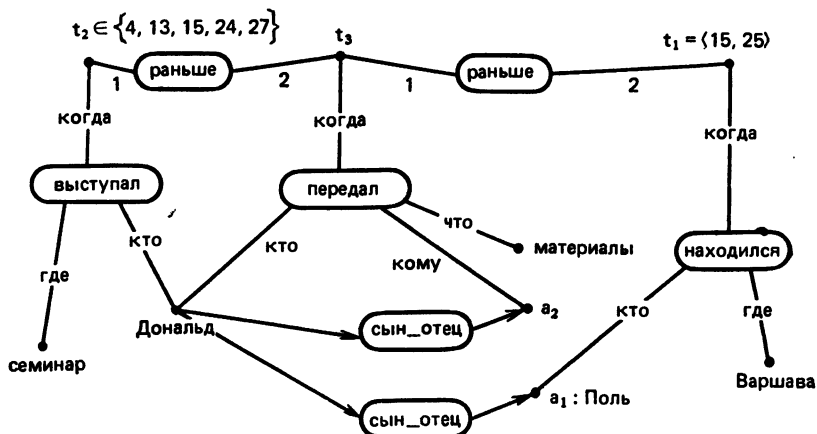


Рис. 1.18

боре аргументов. Эта информация порождает структурный уровень фс-сети.

Элементами носителя могут быть числа, строки и структуры, построенные на их основе. Так, интерпретация может быть сопоставлена с классом **ЧИСЛО** — целые положительные числа, классом **МАРКА-МАШИНЫ** — множество соответствующих наименований, а в некоторых случаях и с классом **ЧЕЛОВЕК** — (чел 1, чел 2, ..., чел 10), если в описываемой предметной области наполнение этого класса зафиксировано.

Фс-сеть объединяет в себе возможности функциональной и семантической сетей. В ней выделяется три типа вершин: объекты, функции и отношения. Дуги отражают связи функций и отношений с их аргументами.

Каждая объектная вершина, как и библиотека понятий, имеет два уровня представления: декларативный и интерпретационный. На первом с ней связывается сорт (возможно, несколько) и имя, на втором — значение данного объекта или множество возможных значений. Например, вершина, соответствующая числу g ($g \in \text{ЧИСЛО}$) или цвету M ($M \in \text{ЦВЕТ}$), кроме своего имени и указания сорта может содержать значение ($g=4$; $M=\text{«красный»}$) или множество возможных значений ($g \in \{3, 7, 8, 15, 40\}$, $M \in \{\text{«зеленый»}, \text{«светло-голубой»}, \text{«белый»}\}$).

Предполагается, что БФС изначально содержит определение сорта **МНОЖЕСТВО**, семантика которого считается встроенной. Тем самым объектная вершина может представлять как единичный объект, так и множество объектов.

Функциональные вершины, а также отношения, имеющие интерпретацию, в рамках фс-сети обладают собственной активностью. Отношения и функции, имеющие интерпретацию, пытаются вычислить или скорректировать значения своих аргументов подобно тому, как это происходит в обобщенных вычислительных моделях [Нариньяни, 1986а, б]. Функции независимо от наличия интерпретации стремятся «склеить» объектные вершины, соответствующие их результатам, при условии тождественности аргументов. Такие локальные процессы коррекции активируются при всяком изменении объектных вершин (в частности, при появлении новых) и исполняются вплоть до полной стабилизации фс-сети.

Продemonстрируем процесс коррекции фс-сети на примере простой ситуации: «Поля с 15 по 25 мая был в Варшаве».

Дональд — сын Поля на мае выступал на одном из заседаний семинара.

Вскоре после своего выступления он передал материалы отцу».

Ее несколько стилизованное изображение в виде фс-сети представлено на рис. 1.18, где использована дополнительная информация о том, что семинары

в мае проводились по следующим числам: 4, 13, 15, 24, 27. Такое состояние фс-сети вызовет следующие процессы коррекции:

вершины a_1 и a_2 склеятся;

отношение **раньше** (t_3, t_1) сформирует значение $t_3 = \langle ?, 14 \rangle$;

отношение **раньше** (t_2, t_3) скорректирует значения обеих переменных следующим образом: $t_3 = \langle 5, 14 \rangle, t_2 \in \{4, 13\}$.

Наряду с перечисленными средствами допускается структуризация фс-сети, т. е. выделение фрагментов (подсетей), которые рассматриваются как самостоятельные вершины класса ПОДСЕТИ. Различаются *открытые* и *закрытые подсети*. В первом случае наполнение подсети считается включенным в объемлющую подсеть и доступно наравне с остальными ее элементами, во втором — содержащаяся в подсети информация считается изолированной и добраться до нее можно только через обращение к соответствующей подсети.

Открытые сети могут использоваться для отображения сообщений типа: «содержание указа гласило...», «в газете сообщалось, что...» или для проблемной структуризации фс-сети. Закрытые сети удобны при введении альтернативных фактов или отображении сообщений, достоверность которых вызывает сомнения, например, таких, как «Мэри считает, что...», «ей приснилось, что...» и т. п.

Система продукции в БФС вводится как некоторое расширение продукционного языка, реализованного в составе технологического комплекса для разработки семантических процессоров [Нариньяни, 1986б]. Наряду со средствами динамического управления исполнением продукции, заимствованными из упомянутого языка, в БФС существует возможность активации продукции из фс-сети. Это обеспечивается следующим образом. Со всякой объектной вершиной фс-сети (в частности, подсетью) может быть связана продукция или множество продукции, которые активируются при изменении данной вершины. Таким образом, например, можно осуществлять локализацию исполнения группы продукции (их привязку к конкретному фрагменту фс-сети).

Существуют некоторые отличия от традиционного (§ 1.4) исполнения отдельной продукции. Продемонстрируем это на примере продукции, отражающей свойство отношения порядка между точками на прямой (x_1, x_2, x_3, x_4 — точки, R — функция расстояния):

$$(x_1 < x_2 < x_4) (x_1 < x_3 < x_4) (R(x_1, x_4) < R(x_1, x_2) + R(x_3, x_4)) \Rightarrow x_3 < x_2.$$

В терминах БФС данное утверждение будет выглядеть следующим образом:

$$\begin{aligned} (x_1, x_2, x_3, x_4 \in \text{ТОЧКА}) (x_1 < x_2) (x_2 < x_4) (x_1 < x_3) \\ (x_3 < x_4) (y_1 = R(x_1, x_2)) (y_2 = R(x_3, x_4)) (y_3 = R(x_1, x_4)) \\ (y_4 < y_1 + y_2) (y_3 < y_4) \Rightarrow x_3 < x_2, \end{aligned}$$

где $y_1, y_2, y_3, y_4 \in \text{ЧИСЛО}$.

Пусть с классом ЧИСЛО, функцией «+» и отношением «<», определенными над числами, соотносена интерпретация, а ТОЧКА и функция расстояния R интерпретации не имеют. При этом левую часть продукции можно разбить на две части:

$$\begin{aligned} \text{декларативную: } P_d = (x_1, x_2, x_3, x_4 \in \text{ТОЧКА}) (x_1 < x_2) (x_2 < x_3) (x_1 < x_3) \\ (x_3 < x_4) (y_1 = R(x_1, x_2)) (y_2 = R(x_1, x_4)) (y_3 = \\ = R(x_1, x_4)); \end{aligned}$$

$$\text{интерпретируемую: } P_{int} = (y_4 = y_1 + y_2) (y_3 < y_4).$$

Проверка условий выполнимости данной продукции будет включать два этапа (результаты поиска объединяются): обычный поиск по образцу для всей левой части продукции и частичный поиск, состоящий в поиске образца P_d и вычислении P_{int} .

L-язык и ленема

L-язык как бы надстраивается на БФС за счет ввода терминальных выражений и специальных конструкций, обеспечивающих компактное и удобное представление знаний. Специфической единицей L-языка является ленема. С первого

взгляда она напоминает фрейм, однако сходство это чисто внешнее. Ленема предназначена для структурного комплексного описания понятий предметной области. При «трансляции» выражений *L*-языка в БФС она трансформируется в множество определений библиотеки понятий и группу продукций.

В *L*-языке выделяется три типа леном: для описания функций, отношений и классов объектов, незначительно отличающихся друг от друга составом и семантикой своих полей. Наиболее богата по содержанию *объектная ленема* (или *о-ленема*), которая включает следующие компоненты (поля): имя, иерархический контекст, определитель, отрицательный определитель, интерпретацию, оболочку, функциональную схему оболочки, модель.

Описание конкретного класса объектов не обязательно включает полный набор полей, любое из них может отсутствовать (конечно, с учетом взаимообусловленности их компонентов). Рассмотрим поля более подробно.

Поле **имя** вводит обозначение описываемого класса и переменную, используемую в остальных полях как произвольный элемент данного класса, например:

имя: ЧИСЛО *x*;

имя: КНИГА *y*.

Поле **иерархический контекст** содержит перечень понятий, являющихся более общими и (или) более частными по отношению к данному, например:

иер-контекст:

родители ДОМАШНЕЕ ЖИВОТНОЕ, ПЛОТОЯДНОЕ;

дети ДВОРНЯГА, СЛУЖЕБНАЯ, ОХОТНИЧЬЯ, ДЕКОРАТИВНАЯ;

Существует возможность помечать полные альтернативно-исключающие разбиения класса.

Поле **определитель** включает достаточные условия принадлежности элементов классу и (или) его точное определение. В рамках данного поля можно использовать определения трех типов:

предикатно-сужающее — накладывает ограничения на класс или множество классов, например, ленема, описывающая класс РЕБЕНОК, может содержать определение

п-опред: ЧЕЛОВЕК *x*; возраста (*x*) < 12 лет;

интерпретируемое следующим образом:

$(x \in \text{ЧЕЛОВЕК}) \ \& \ (\text{возраста} \ (x) < 12 \text{ лет}) \Rightarrow x \in \text{РЕБЕНОК};$

предикатно-конструирующее — описывает «способ построения» объектов, вводимых данной леномой, например:

к-опред: $(x_1, x_2, x_3 \in \text{ТОЧКА}) \ \& \ P_1(x_1, x_2, x_3) \Rightarrow (\exists x) \ (x \neq \text{ТРЕУГ} \ (x_1, x_2, x_3)),$

где P_1 — предикат «не лежат на одной прямой», x — переменная из поля **имя** данной леномы, описывающей класс ТРЕУГ — треугольник;

конструктивное — задает способ построения описываемого класса, например:

опред: <ЧИСЛО *x*, ЧИСЛО *y*> ($x < y$),

данная запись описывает сорт, элементами носителя которого являются пары упорядоченных чисел.

Поле **отрицательный определитель** содержит условия непринадлежности объектов данному классу. Например, для класса СОБАКА

не-опред: $(x \in \text{ДОМАШНЕЕ ЖИВОТНОЕ}) \ \& \ \text{мяукает} \ (x).$

В частности, это поле может использоваться для вывода «от противного».

Поле **интерпретация** вводит носитель (множество значений) описываемого сорта.

Поле **оболочка** является аналогом системы слотов во фрейме. Оболочка выделяет семантическую окрестность (когнитивное пространство) объекта и представляет синтаксические возможности для прямого доступа к своим элементам.

В *L*-языке допускаются элементы оболочки (их также именуют слотами), порождаемые унарными функциями, бинарными и тернарными отношениями: (предикатами), соответственно выделяется три типа слотов: *функциональные*, *предикатные* и *параметрические*. Функциональный слот соответствует унарной функции, определенной на описываемом классе. Например: год рождения человека, стоимость вещи, название книги и т. п.

В общем случае значением предикатного слота является корректируемое множество объектов. Его описание наряду с указанием имени и сортности включает функцию его коррекции. В качестве примера таких слотов можно указать: *сторона-треугольника* (значение—множество из трех отрезков), *последнее-место-работы* (значение меняется при поступлении новой, более свежей информации).

Параметрические слоты индуцируются тернарными отношениями. При этом один из аргументов должен принадлежать описываемому классу, второй выделяется как параметр и выносится в заголовок поля оболочки, третий выступает в качестве слота. Примерами таких слотов служат: *возраст*, *дети*, *должность*, *вес*, *рост* человека, где в роли параметра выступает время. Вообще говоря, с объектом может связываться несколько оболочек, соответствующих различным сферам (подмоделям) знаний, выделяемым в предметной области. Например, с понятием ЧЕЛОВЕК могут быть связаны оболочки, соответствующие следующим «граням» данного понятия: физиология, психология, профессиональная деятельность, этические и (или) моральные «свойства характера», бытовое поведение и т. п.

Функциональная схема оболочки задает условия однозначности определения объекта описываемого класса и (или) значений его слотов. **Модель оболочки** — это совокупность отношений и продукций, связывающих элементы оболочки. Модель отражает внутренние свойства описываемого понятия и при переходе к БФС трансформируется в фрагмент системы продукций. Модель оболочки и ее функциональная схема соотносятся не столько с классом, сколько с приписываемой ему оболочкой.

Семантику двух последних полей продемонстрируем на примере описания понятия ИНТервал (отрезок прямой) — основное понятие формальной модели времени, реализация которой была осуществлена в рамках экспертной системы ВРЕМЯ-1 [Kandrashina, 1985]:

имя: ИНТ *t*

оболочка:

начало; ТОЧКА *x*

конец; ТОЧКА *y*

длина; ЧИСЛО *g*

f-схема:

$x, y; x, g; y, g \rightarrow t$

$g = R(x, y)$

Данная ленама вводит класс объектов ИНТервал, а также три функции: *начало*, *конец*, *длина*, определенных на этом классе. В функциональной схеме (*f*-схема) указано, что значения любой пары слотов однозначно определяют интервал. Наполнение модели эквивалентно утверждению:

$t \in \text{ИНТ} \Rightarrow (\text{начало}(x) < \text{конец}(y)) \text{ (длина}(t) = R(\text{начало}(t), \text{конец}(t)))$. Ленымы, задающие схемы описания функций и отношений, имеют практически тот же набор полей, лишь немного модифицируется их семантика.

Описанный подход к представлению и обработке знаний имеет три особенности: 1) двухуровневость рассматриваемого ЯПЗ (*L*-язык — для пользователя, БФС — для разработчика ядра интеллектуальной системы); 2) объединение трех существующих в настоящее время парадигм представления знаний по крайней мере на пользовательском уровне; 3) введение в ЯПЗ (в образе ленымы) средств поддержки некоторой методологии описания предметной области. По изобразительным возможностям ленымы превосходят такие устоявшиеся традиционные средства представления знаний, как семантическая сеть, фрейм, система продук-

ций, функциональная сеть. Конечно, всегда найдутся понятия, для которых данные средства будут неудобны либо неприемлемы. В частности, понятия, в описании которых существенную роль играет внутренняя динамика (традиционно для их спецификации используются сценарии).

Глава 2

Методы работы со знаниями

2.1. Приобретение и формализация знаний

А. Н. Аверкин, А. Ф. Блишун, Т. А. Гаврилова, Г. С. Осипов

Основные определения

Приобретением знаний называется выявление знаний из источников и преобразование их в нужную форму, а также перенос в базу знаний ИС. Источниками знаний могут быть книги, архивные документы, содержимое других баз знаний и т. п., т. е. некоторые *объективизированные знания*, переведенные в форму, которая делает их доступными для потребителя. Другим типом знаний являются *экспертные знания*, которые имеются у специалистов, но не зафиксированы во внешних по отношению к нему хранилищах. Экспертные знания являются *субъективными*. Еще одним видом субъективных знаний являются *эмпирические знания*. Такие знания могут добываться ИС путем наблюдения за окружающей средой (если у ИС есть средства наблюдения).

Ввод в базу знаний объективизированных знаний не представляет особой проблемы, выявление и ввод субъективных и особенно экспертных знаний достаточно трудны. Чтобы разработать методологию приобретения субъективных знаний, получаемых от эксперта, надо четко различать две формы репрезентации знаний. Одна форма связана с тем, как и в каких моделях хранятся эти знания у человека-эксперта. При этом эксперт не всегда осознает полностью, как репрезентированы у него знания. Другая форма связана с тем, как инженер по знаниям, проектирующий ИС, собирает их описывать и представлять. От степени согласованности этих двух форм репрезентации между собой зависит эффективность работы инженера по знаниям.

В когнитивной психологии изучаются формы репрезентации знаний (*когнитивные структуры знаний*) характерные для человека. Примерами могут служить [Хафман, 1986]: представление класса понятий через его элементы (например, понятие «птица» репрезентируется рядом «чайка, воробей, скворец, ...»); представление понятий класса с помощью базового прототипа, отражающего наиболее типичные свойства объектов класса (например, понятие «птица» репрезентируется прототипом «нечто с крыльями, клювом, летает ...»); представление с помощью признаков (для понятия «птица», например, наличие крыльев, клюва, двух лап, перьев ...).

Кроме понятий репрезентируются и отношения между ними. Как правило, отношения между понятиями определяются процедурным способом, а отношения между составляющими понятий (определяющими структуру понятия) — декларативным способом. Наличие двух видов описаний заставляет в моделях представления знаний одновременно иметь оба компонента, например семантическую сеть и продукционную систему, как это представлено в когнитивной модели [Anderson, 1983].

При приобретении знаний важную роль играют так называемое *поле знаний*, в котором содержатся основные понятия, используемые при описании предметной области, и свойства всех отношений, используемых для установления связей

между понятиями. Поле знаний связано с концептуальной моделью проблемной области, в которой еще не учтены ограничения, которые неизбежно возникают при формальном представлении знаний в базе знаний. Переход от описания некоторой области в поле знаний к описанию в базе знаний аналогичен переходу от концептуальной модели базы данных к ее логической схеме, когда уже зафиксирована система управления базой данных. Важно отметить, что переход непосредственно к формальным представлениям в базе знаний без этапа концептуального описания в поле знаний приводит к многочисленным ошибкам, что замедляет процесс формирования базы знаний ИС.

Возможны три режима взаимодействия инженера по знаниям с экспертом-специалистом: *протокольный анализ, интервью и игровая имитация профессиональной деятельности*. Протокольный анализ заключается в фиксации (например, путем записи на магнитную ленту) «мыслей вслух» эксперта во время решения проблемы и в последующем анализе полученной информации. В режиме интервью инженер по знаниям ведет с экспертом активный диалог, направляя его в нужную сторону. При игровой имитации эксперт помещается в ситуации, похожие на те, в которых протекает его профессиональная деятельность. Наблюдая за его действиями в различных ситуациях, инженер по знаниям, формирует свои соображения об экспертных знаниях, которые впоследствии могут быть уточнены с экспертом в режиме интервью. Принципы игровой имитации нашли применение в разнообразных деловых играх, специальных тренажерах.

Каждый из упомянутых способов извлечения знаний имеет свои преимущества и недостатки. Так, при анализе протоколов инженеру по знаниям нелегко отделить понятия, важные для включения в словарь предметной области, от тех, которые при «мыслях вслух» появляются случайно. Кроме того, в протоколах обнаруживаются пробелы, когда рассуждение эксперта как бы прерывается и продолжается уже на основе пропущенных шагов вывода. Заполнение подобных лакун возможно лишь в режиме интервью. Таким образом, во всех трех подходах к извлечению знаний из экспертов необходим этап интервью, что делает его одним из важнейших методов приобретения знаний.

Существует не менее двух десятков стратегий интервьюирования. Наиболее известны три: разбиение на ступени, репертуарная решетка и подтверждение сходства.

При разбиении на ступени эксперту предлагается назвать наиболее важные, по его мнению, понятия предметной области и указать между ними отношения структуризации, т. е. отношения типа «род — вид», «элемент — класс», «целое — часть» и т. п. Эти понятия используются на следующем шаге опроса как базовые. Стратегия нацелена на создание иерархии понятий предметной области, выделение в понятиях тесно связанных между собой групп — *таксонов* (классов).

Стратегия репертуарной решетки направлена на выявление характеристических свойств понятий, позволяющих отделять одни понятия от других. Методика состоит в предъявлении эксперту троек понятий с предложением назвать признаки для каждого из понятий, которые отделяли бы их от третьего. Так как каждое понятие входит в несколько троек, то на основании такой процедуры происходит уточнение объемов понятий и формируются «симптокомплексы» понятий, с помощью которых эти понятия могут идентифицироваться в базе знаний.

Стратегия подтверждения сходства состоит в том, что эксперту предлагается установить принадлежность каждой пары понятий из предметной области к некоторому отношению сходства (толерантности). Для этого эксперту задается последовательность достаточно простых вопросов, цель которых заключается в уточнении того понимания сходства, которое вкладывает эксперт в утверждение о сходстве двух понятий предметной области.

Процесс взаимодействия инженера по знаниям (аналитика) с экспертом-специалистом включает три основных этапа.

1. Подготовительный этап. Для успеха общения оба участника должны тщательно подготовиться к диалогу или игре. Желательно, чтобы эксперт был

не только компетентным специалистом, но и заинтересованным (морально или материально) лицом в достижении конечной цели — построении ИС. Он должен быть доброжелателен к аналитику и уметь объяснять свои знания (наилучший случай, когда эксперт имеет опыт преподавательской работы).

Аналитику необходимо: глубоко познакомиться со специальной литературой по предметной области, чтобы не задавать очень «глупых» вопросов (просто «глупые» вопросы бывают чрезвычайно полезны), а также увеличить количество «пакетов ожиданий» [Шенк и др., 1987]; уметь слушать и грамотно задавать вопросы; настроиться на роль «ученика», а не «экзаменатора»; разбираться в моделях когнитивной психологии, а также в моделях представления знаний, чтобы из знаний эксперта выделять четкие структуры.

В любой совместной деятельности большое значение имеют психологические качества исследователей, такие как личность, манера поведения, стиль научного мышления. Существуют различные классификации научных работников. В качестве примера приведем следующую: инициатор — быстро реагирует на перспективные проблемы, т. е. один из первых ощущает необходимость решения проблемы с элементами неопределенности; диагност — способен к быстрой оценке сильных и слабых сторон решения задачи; эрудит — наделен исключительной памятью, отличается повышенным вниманием к деталям и стремлением к упорядоченности; ремесленник — способен воплощать в жизнь плохо оформленные идеи других; эстет — стремится исследовать проблемы, приводящие к изящным решениям, не склонен к кропотливому труду; методолог — заинтересован методологическими аспектами исследований; независимый — стремится к индивидуальному решению проблем; фанатик — самоотверженно увлечен своей научной проблемой, того же требует и от окружающих.

Принадлежность научного работника к тому или иному типу определяется с помощью косвенных методик (тестов личности, интеллекта, когнитивных стилей, проектных методик). Автоматизация опроса и получения психологического портрета испытуемого реализована, например, в системе АВТАНТЕСТ [Гаврилова, 1984].

Для роли эксперта наиболее предпочтительны инициатор, эрудит, диагност и ремесленник (в паре с аналитиком-эрудитом), а для роли аналитика — диагност, методолог, эрудит, инициатор. При этом наилучшее сочетание дают сочетания разных типов. Благодаря различиям в подходах к решению задачи, в точках зрения, стиле мышления, восприятия, памяти и т. п. участники в такой паре с разных сторон подходят к поставленной цели, в результате увеличивается общее количество гипотез, идей, альтернативных вариантов, а следовательно, обогащается поле знаний. Однако не все сочетания даже из приемлемых типов улучшают взаимодействие, а некоторые типы (например, фанатик, эстет, независимый, ремесленник) часто слабо приспособлены для творческого взаимодействия, что приводит к возникновению скрытых и явных конфликтов, которые усложняют процесс продуктивного общения.

Важное значение имеет также лидерство в паре. В ходе любого диалога одна сторона обычно занимает позицию ведущей, чаще эту роль берет интервьюер, т. е. аналитик. Роль лидера в диалоге позволяет аналитику направлять и систематизировать процесс создания поля знания, не давая эксперту «размыться» или излишне детализовать процесс. С другой стороны, догматизм и настойчивость могут привести к неадекватному полю. Имеет место также эффект «фасада», т. е. желание эксперта не ударить «в грязь лицом» перед аналитиком, и отсюда генерирование неподтвержденных гипотез.

2. Установление «общего кода». Для создания лингвистического альянса взаимодействия участники взаимодействия должны пытаться сократить «расстояние» между объектом (т. е. исследуемой предметной областью) и аналитиком. Необходимо определить главные понятия, т. е. выработать словарную основу базы знаний; уровень детализации; взаимосвязи между понятиями.

3. Гносеологический этап. На этом этапе происходит выяснение закономерностей, присущих предметной области, условий достоверности и истинности утверждений, структурирование за счет введения отношений и т. п. Этот этап

является определяющим во взаимодействии аналитика и эксперта. В процессе анализа игры или диалога вербализуется и формализуется знание эксперта и зачастую для него самого порождается новое знание. Репрезентация внешнего мира в его памяти получает материальное воплощение в форме поля знаний.

В процессе извлечения знаний сначала желательно получить от эксперта поверхностные знания (такие, например, как репрезентация признаков), постепенно переходя к глубинным структурам и более абстрактным понятиям (таким, например, как прототипы).

При формировании поля знаний учитываются особенности эмпирического знания: модальность, противоречивость, неполнота и т. д.

Аналитик должен за частным видеть общее, т. е. строить цепочки «факт — обобщенный факт — эмпирический закон — теоретический закон». Центральное звено цепочки — формализация эмпирики. При этом иногда основным на этапе формализации становится не извлечение «слепых» непонятных связей, а понимание внутренней структурной связи понятий предметной области. Искусство аналитика состоит в стремлении к созданию ясной и понятной модели проблемной области.

Следует также учитывать, что эксперты в проблемной области не всегда опираются на логические рассуждения. В их представлениях о проблемной области и методах решения задач, характерных для нее, широкое применение находят ассоциативные рассуждения и рассуждения правдоподобия (см. § 2.5). Опишем примерную методику работы с экспертом по формированию поля знаний.

Подготовительный этап

1. Четкое определение задач проектируемой системы (сужение поля знаний): определение, что на входе и выходе; определение режима работ — консультации, обучение и др.

2. Выбор экспертов: определение количества экспертов; выбор уровня компетентности (не всегда хорошо выбирать самый высокий уровень сразу); определение способов и возможности заинтересовать экспертов в работе; тестирование экспертов.

3. Знакомство аналитика со специальной литературой в предметной области

4. Знакомство аналитика и экспертов (в дальнейшем для простоты будем считать, что эксперт один).

5. Знакомство эксперта с популярной литературой по искусственному интеллекту (желательно, но необязательно).

6. Попытка аналитика создать поле знаний первого приближения априорным знаниям из литературы (прототип поля знаний).

Основной этап

1. «Накачка» поля знаний: а) в зависимости от предметной области выбор способа интервьюирования; б) протоколирование мыслей вслух или запись на магнитофон рассуждений эксперта (аналитик по возможности не должен пока вмешиваться в рассуждения).

2. «Домашняя работа». Попытка аналитика выделить некоторые причинно-следственные связи в рассуждениях эксперта; построение словаря предметной области (возможно, на карточках) и подготовка вопросов к эксперту.

3. «Подкачка» поля зрения. Обсуждение с экспертом прототипа поля знаний и домашней работы, а также ответы на вопросы аналитика.

4. Формализация концептуальной модели.

5. Построение поля знаний второго приближения.

6. При необходимости повторение пп. 1б, 2, 3, 4 и 5.

Системы приобретения знаний от экспертов

Одно из первых рассмотрений интервью как метода инженерии знаний проведено в [Newel, 1972]. Проблемы, возникающие при извлечении экспертных знаний, некоторые психологи связывают с так называемой когнитивной защитой. В [Kelly, 1985] была развита теория человеческого познания, основанная на понятии «персональных конструкторов», которые человек создает и пытается приспособить к реалиям мира. В [Bose, 1984] теория «персональных конструкторов» использована для создания системы извлечения экспертных знаний и показала свою способность успешно преодолевать когнитивную защиту, т. е. нежелание экспертов достичь четкого и осознанного ими истолкования основных понятий, отношений между понятиями и приемов решения задач в интересующей инженера по знаниям проблемной области.

Методы интервьюирования эксперта предметной области знаний с использованием нескольких различных стратегий применены при создании системы TEIRESIAS [Davis, 1982]. В [Kahn et al., 1984] выделено восемь различных стратегий интервью, в [Kahn et al., 1985] на основе этих стратегий исследуется возможность автоматического интервьюирования. Автоматизации метода протокольного анализа посвящены работы [Waterman, 1971, 1973; Krippendorf, 1980].

В [Kahn et al., 1985] на примере диагностической системы MORE; описана техника интервьюирования, направленная на выяснение следующих сущностей: гипотез, симптомов, условий, связей и путей. Гипотеза — событие, идентификация которого имеет своим результатом диагноз. Симптом — событие, являющееся следствием существования гипотезы, наблюдение которого приближает последующее принятие гипотезы. Условие — событие или некоторое множество событий, которое не является непосредственно симптоматическим для какой-либо гипотезы, но которое может иметь диагностическое значение для некоторых других событий. Связи — соединения сущностей (в том числе, других связей). Путь — выделенный тип связи, который соединяет гипотезы с симптомами. В соответствии с этим используются следующие стратегии интервью: дифференциация гипотез, различение симптомов, симптомная обусловленность, деление пути и др.

Дифференциация гипотез направлена на поиск симптомов, которые обеспечивают более точное различение гипотез. Наиболее мощными в этом смысле являются те симптомы, которые происходят из одного диагностируемого события. Различение симптомов выявляет специфические характеристики симптома, которые, с одной стороны, идентифицируют его как следствие некоторой гипотезы, с другой — противопоставляют другим. Симптомная обусловленность направлена на выявление негативных симптомов, т. е. симптомов, отсутствие которых имеет больший диагностический вес, чем их присутствие. Деление пути обеспечивает нахождение симптоматических событий, которые лежат на пути к уже найденному симптому. Если такой симптом существует, то он имеет большее диагностическое значение, чем уже найденный.

Аналогичные стратегии интервьюирования эксперта использованы при создании инструментальной диагностической системы ИДИС [Голубев и др., 1987].

В системе KRITON [Diederich et al., 1987] для приобретения знаний используются два источника: эксперт с его знаниями, полученными на практике (эти знания, как правило, неполны, отрывочны, плохо структурированы); книжные знания, документы, описания инструкции (эти знания хорошо структурированы и фиксированы традиционными средствами). Для извлечения знаний из первого источника в KRITON применена техника интервью, использующая стратегии репертуарной решетки и разбиения на ступени. При этом применяется прием переклассификации стратегий: если при предъявлении тройки семантически связанных понятий эксперт не в состоянии назвать признак, отличающий два из них от третьего, система запускает стратегию разбиения на ступени и предпринимает попытку выяснения таксономической структуры этих понятий с целью выявления признаков, их различающих.

Для выявления процедурных знаний эксперта в KRITON применен метод протокольного анализа. Он осуществляется в пять шагов. На первом шаге протокол делится на сегменты на основании пауз, которые делает эксперт в про-

цессе записи. Второй шаг — семантический анализ сегментов, формирование высказываний для каждого сегмента. На третьем шаге из текста выделяются операторы и аргументы. Далее делается попытка поиска по образцу в базе знаний для обнаружения переменных в высказываниях (переменная вставляется в высказывание, если соответствующая ссылка в тексте не обнаружена). На последнем шаге утверждения упорядочиваются в соответствии с их появлением в протоколе.

Анализ текста используется в KRITON для выявления хорошо структурированных знаний из книг, документов, описаний, инструкций.

В [Morik, 1987] описан метод выявления модели предметной области. Первая фаза — формирование инженером знаний грубой модели предметной области путем определения предикатов и сортов их возможных аргументов и сообщения системе фактов об области, выражимых этими предикатами. Система выявляет свойства предикатов и устанавливает отношения между ними, структурируя таким образом предметную область. На второй фазе с помощью метазнаний (общих структур), отражающих особенности человеческого мышления, осуществляется проверка соответствия фактов предикатам, индуктивный вывод правил из фактов, вывод правил из других правил.

В системах SIMER и ДИАПС [Осипов, 1987; Osipov et al., 1987] основным методом приобретения знаний является автоматизированное интервьюирование эксперта, которое управляется знаниями, приобретенными системой. В системах SIMER и ДИАПС не выявляется предварительная модель области. Все объекты (события) и их атрибуты определяются в режиме прямого интервьюирования эксперта. Предполагается только, что на множестве объектов могут быть заданы ряд отношений из известного (конечного) множества: «элемент — множество», «часть — целое», «пример — прототип», отношения структурного сходства объектов, структурной иерархии и некоторые другие. Все отношения попарно различаются формальными свойствами. Так, отношение структурного сходства не обладает транзитивностью, но симметрично. Отношение структурной иерархии, напротив, не обладает симметричностью, однако транзитивно. На выяснение этих и ряда других свойств отношений и объектов направлено интервью.

В частности, для установления структурного сходства на первой фазе интервью для каждого вновь вводимого понятия эксперту предлагается указать (с помощью меню) те понятия предметной области, с которыми может быть связано данное (без спецификации отношения). Затем в процессе интервью для каждой пары понятий (из выделенных на первой фазе) связь специфицируется, устанавливаются свойства и тип отношения, в число элементов которого включается исследуемая пара. Так, для включения некоторой пары понятий X и Y , о которых эксперт сообщил, что X влияет на Y (например X увеличивает возможность Y), в число элементов некоторого отношения R , обладающего среди прочих свойств симметричностью, необходимо задать эксперту вопрос: «Увеличивает ли Y возможность X ?». При положительном ответе на этот вопрос (и если прочие свойства уже установлены и удовлетворяют определению отношения R) пара (X, Y) включается в R . Для установления структурного сходства и структурной иерархии понятий используются стратегии подтверждения сходства и разбегания на ступени.

В модели имеются метапроцедуры и метаправила, которые проверяют корректность модели, используют формальные свойства отношений для пополнения модели и генерируют правила.

Сформулируем основные этапы реализации системы приобретения знаний.

1. Интервью для определения актуальной области, в которой происходит процесс решения интересующей проблемы, и расчленение ее на автономные области.

2. Автоматизированное интервью для выявления и формирования декларативной модели предметной области.

3. Протокольный анализ к выявленным на предыдущем этапе понятиям и отношениям предметной области для пополнения модели процедурными знаниями

(этапы 2 и 3 можно использовать попеременно до тех пор, пока модель не достигнет нужной полноты).

4. Протокольный анализ для попечения декларативных знаний модели.

5. Проверка полноты модели. Обычно протокольный анализ выявляет пустоты в модели. Имеется в виду случаи, когда понятия, использованные в «мыслях вслух», недостаточно описаны. В этом случае интервью и протокольный анализ повторяются.

Формализация качественных знаний

При формализации качественных знаний может быть использована теория нечетких множеств [Заде, 1974], особенно те ее аспекты, которые связаны с лингвистической неопределенностью, наиболее часто возникающей при работе с экспертами на естественном языке. Под лингвистической неопределенностью подразумевается не полиморфизм слов естественного языка, который может быть преодолен на уровне понимания смысла высказываний в рамках байесовской модели [Налимов, 1974], а качественные оценки естественного языка для длины, времени, интенсивности, для целей логического вывода, принятия решений, планирования.

Лингвистическая неопределенность в системах представления знаний задается с помощью лингвистических моделей, основанных на теории лингвистических переменных и теории приближенных рассуждений [Kikert, 1978]. Эти теории опираются на понятие нечеткого множества, систему операций над нечеткими множествами и методы построения функций принадлежности.

Одним из основных понятий, используемых в лингвистических моделях, является понятие лингвистической переменной. Значениями лингвистических переменных являются не числа, а слова или предложения некоторого искусственного либо естественного языка. Например, числовая переменная «возраст» принимает дискретные значения между нулем и сотней, а целое число является значением переменной. Лингвистическая переменная «возраст» может принимать значения: молодой, старый, довольно старый, очень молодой и т. д. Эти термы — лингвистические значения переменной. На это множество (как и на числа) также налагаются ограничения. Множество допустимых значений лингвистической переменной называется терм-множеством.

При вводе в ЭВМ информации о лингвистических переменных и терм-множестве ее необходимо представить в форме, пригодной для работы на ЭВМ. Лингвистическая переменная задается набором из пяти компонентов: $\langle A, T(A), U, G, M \rangle$, где A — имя лингвистической переменной; $T(A)$ — ее терм-множество; U — область, на которой определены значения лингвистической переменной; G описывает операции по порождению производных значений лингвистической переменной на основе тех значений, которые входят в терм-множество. С помощью правил из G можно расширить число значений лингвистической переменной, т. е. расширить ее терм-множество. Каждому значению a лингвистической переменной A соответствует нечеткое множество X_a , являющееся подмножеством U . По аналогии с формальными системами (см. § 1.1) правила из G часто называют синтаксическими. Наконец, компонент M образует набор семантических правил. С их помощью происходит отображение значений лингвистической переменной a в нечеткие множества X_a и выполняются обратные преобразования. Именно эти правила обеспечивают формализацию качественных утверждений экспертов при формировании проблемной области в памяти ИС.

На рис. 2.1 показаны все компоненты, определяющие лингвистическую переменную «возраст». В качестве терм-множества использовано множество, состоящее из трех значений: очень молодой (μ_m), пожилой (μ_p) и старый (μ_s), задаваемых функциями принадлежности на области U , которую называют носителем лингвистических значений. В примере область U — года жизни от 0 до 150 лет. В качестве семантических правил выступают отображения, задаваемые функциями принадлежности $0 < \mu_a(u) < 1$ к нечетким множествам X_m , X_p , X_s . Как видно из рис. 2.1, человек, возраст которого равен 60 годам, принадлежит

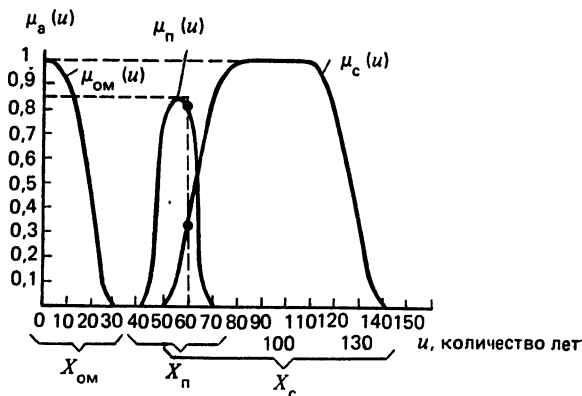


Рис. 2.1

к $X_{ом}$ со значением 0 (т. е. человек в 60 лет не является очень молодым), к $X_{п}$ со значением 0,8 и к $X_{с}$ со значением 0,4.

Для перехода от качественных описаний к формализованным необходимо построить отображения, входящие в M , т. е. построить функции принадлежности. В таком виде подобная задача была исследована в [Блишун, 1987].

При получении от экспертов информации о виде функций принадлежности необходимо учитывать характер измерений (первичные и производные измерения) и тип шкалы, на которую проецируются измерения и на которой будут определяться функции принадлежности [Готов и др., 1976]. На этой шкале задается вид допустимых операторов и операций, т. е. некоторая алгебра для функций принадлежности. Кроме того, следует различать характеристики, которые можно измерять непосредственно, и характеристики, которые являются качественными и требуют попарного сравнения объектов, обладающих этими характеристиками, чтобы определить их отношение к исследуемому понятию.

Можно выделить две группы методов построения функций принадлежности: прямые и косвенные. В прямых методах эксперт непосредственно задает правила определения значений функции принадлежности $\mu_a(u)$. Эти значения согласуются с его предпочтениями на множестве объектов следующим образом: для любых $u_1, u_2 \in U$ имеет место $\mu_a(u_1) < \mu_a(u_2)$ тогда и только тогда, когда u_2 предпочтительнее u_1 , т. е. в большей степени определяется понятием a ; для любых $u_1, u_2 \in U$ имеет место $\mu_a(u_1) = \mu_a(u_2)$ тогда и только тогда, когда u_2 и u_1 не различаются по отношению к понятию a . К прямым методам относится непосредственное задание функции принадлежности таблицей, формулой или примером [Zadeh, 1975; Ragade et al., 1977; Thole et al., 1979].

В косвенных методах значения функции принадлежности выбираются таким образом, чтобы удовлетворялись заранее сформулированные условия. Экспертная информация является только исходной для дальнейшей обработки. Дополнительные условия могут налагаться как на вид получаемой информации, так и на процедуру обработки. Примерами дополнительных условий могут служить следующие: функция принадлежности должна отражать близость к заранее выделенному эталону, объекты множества являются точками в параметрическом пространстве [Scala, 1978]; результатом процедуры обработки должна быть функция принадлежности, удовлетворяющая условиям интервальной шкалы [Жуковин и др., 1983]; при попарном сравнении объектов, если один объект оценивается в k раз сильнее, чем другой, то второй объект оценивается в $1/k$ раз сильнее, чем первый объект [Saaty, 1974], и т. д.

Как правило, прямые методы используются для описания понятий, которые характеризуются измеримыми признаками (высотой, ростом, массой, объемом).

В этом случае удобно непосредственное задание функции принадлежности. К прямым методам можно отнести методы, основанные на вероятностной трактовке функций принадлежности: $\mu_a(u) = P(a/u)$, т. е. вероятность того, что объект $u \in U$ будет принадлежать к множеству, которое характеризует понятие a . Так как люди часто искажают оценки, например сдвигают их в направлении концов оценочной шкалы [Thole et al., 1979], то прямые измерения, основанные на непосредственном определении значений функций принадлежности, могут быть использованы только в том случае, когда такие искажения незначительны или маловероятны. Косвенные методы более трудоемки, чем прямые, но обладают стойкостью к искажениям в ответе. Результатом применения косвенных методов является интервальная шкала. В [Thole et al., 1979] выдвигается для косвенных методов «условие безоговорочного экстремума»: при определении степени принадлежности множество исследуемых объектов должно содержать по крайней мере два объекта, численные представления которых на интервале $[0, 1]$ — 0 и 1 соответственно.

Функции принадлежности могут отражать мнение как некоторой группы экспертов, так и одного уникального эксперта. Комбинируя возможные два метода построения функций принадлежности с двумя типами экспертов (коллективным и уникальным), можно получить четыре типа экспертизы [Блишун, 1988].

Пример формализации качественных знаний

При анализе ситуации эксперт рассуждает в семантическом пространстве (пространстве шкал), в котором ситуации соответствует оцененный образ. Семантическое пространство аналогично субъективному пространству ощущений,

Таблица 2.1

Тип шкалы	Аксиомы	Примечания
Номинальная	$a \sim b$ или $a \not\sim b$, $a \sim a$ Если $a \sim b$, $b \sim c$, то $a \sim c$ Если $a \sim b$, то $b \sim a$	Знак: читается как «такое, что» D — четырехместное отношение P, R, I — двухместные отношения
Порядковая	Не существует $a \in U : a > a$ Если $a > b$, $b > c$, то $a > c$	$P : aPb \Rightarrow abDaa$ $I : aIb \Leftrightarrow abDbc, baDab$
Интервальная	Если $abDcd$, $cdDef$, то $abDef$ $abDcd$ или $cdDab$ Если $abDcd$, то $acDbd$ Если $abDcd$, то $dcDba$ Существует $c \in U : acDcb, cbDac$ Если aPb и \overline{abDcd} , то существует $e \in U : aPe, ePb, cdDae$ Если aPb , $abDcd$, то существуют $e, f \in U$ и $n : ceM^nf d$	$M : abMcd \Leftrightarrow \{abDcd, cdDab, bIc\}$ $M^n : abM^1cd \Leftrightarrow abMcd$ $abM^{n+1}cd \Leftrightarrow \{\exists e, f \in U : abM^nef, efMcd\}$ o — операция: $1oa = a$ $noa = (n-1)oa$
Отношений	Если aRb , bRc , то aRc (aob) o cRa o (boc) Если aRb , то $aocRcob$ Если \overline{aRb} , то существует $a \in U : aRboc, aobRa$ Если aRb , то существует $n : bRna$	

Таблица 2.2

Метод формализации нечеткости	Метод измерения	Число экспертов	Процедура получения исходных данных	Тип шкалы	Интерпретация	Тип измерения	Источник
Назначение принадлежности с упорядоченным носителем	П	1	ОЗ	П	Д	Ф	[Zadeh, 1975]
Назначение принадлежности с упорядочением носителя	П	1	ОЗ	П	Д	Ф	[Ragade et al., 1977]
Семантических дифференциалов	П	1	ОЗ	А	Д	П	[Ragade et al., 1977]
Равноделения	П	1	ПС	А	ВС	Ф	[Борисов и др., 1982]
Удельного пересечения множеств	П	1	ОФ	А	Д	П	[Ragade et al., 1977; Борисов и др., 1982; Sanchez, 1979]
Отклонение от экстремума	П	1	ОФ	А	Д	Ф	[Ragade et al., 1977]
Стандартного набора графиков	П	1	ОФ	А	Д	Ф	[Барисов и др., 1982]
Оценки по унифицированной шкале	П	1	ОЗ	П	Д	Ф	[Батыршин, 1986]
Прямого оценивания	П	1	ОЗ	И	Д	Ф	[Norwich et al., 1984]
α -срезов	П	1	ОЗ	И	Д	Ф	[Yager et al., 1982]
Отношение интенсивности различий	К	1	ПС	О	Д	П	[Snaty, 1974]
Варьирования прототипов	К	1	ОФ	А	Д	Ф	[Scala, 1978]
Факторного пространства	К	1	ОПО	А	Д	Ф	[Блишун, 1988]
Коэффициент доверия	К	$\overline{1, N}$	ОФ	А	В	Ф	[Civanlar, et al., 1986]
Условной вероятности	П	$\overline{1, N}$	ОДН	А	ВЧ	Ф	[Осис, 1968]
Полной системы классов	П	$\overline{1, N}$	ОЗ	А	ВС	Ф	[Борисов и др., 1970]
Голосования	П	N	ОДН	А	ВЧ	Ф	[Zadeh, 1975]
Поиска характеристической функции	П	N	ОДН	А	ВЧ	Ф	[Шер, 1978]

Метод формализации нечеткости	Метод измерения	Число экспертов	Процедура получения исходных данных	Тип шкалы	Интерпретация	Тип измерения	Источник
Упорядоченных последовательностей	К	$\overline{1, N}$	Р	И	Д	Ф	[Киквидзе и др., 1979]
Прямого оценивания с условием безоговорочного экстремума	К	N	ОЗ	А	Д	Ф	[Thole et al., 1979]
Расширения метода Саати на интервальные шкалы	К	N	ОДН	И	ВЧ	Ф	[Жуковин, 1983]
Частотных оценок	К	N	ОДН	А	ВЧ	Ф	[Ежкова и др., 1977]
Отношения моделирования	П	$\overline{1, N}$	ОЗ	И	Д	Ф	[Averkin et al., 1987]

в котором формируется внутренний образ внешних сигналов и возникают субъективные связи между свойствами (признаками, параметрами). В зависимости от индивидуального восприятия одно и то же значение признака может быть оценено по-разному. Однако для конкретного индивидуума оцененная ситуация является инвариантом относительно определенного класса ситуаций. Следовательно, при отождествлении реальных значений признаков с семантическим образом существенной является форма нечеткого отображения пространства признаков в семантическое пространство.

Отображение любой ситуации на единичный интервал происходит таким образом, что точка интервала характеризует степень проявления некоторого свойства (0 соответствует отсутствию свойства, 1 — интересующему нас максимальному проявлению свойства). При построении функции принадлежности используется *модель измерений*, которая определяется двумя параметрами: типом шкалы принадлежности, на которую отображается информация от эксперта, и типом измерения (прямой или косвенный). Шкала называется фундаментальной, если она допускает прямое взаимодействие множества U и того нечеткого свойства, которое нас интересует. Такая шкала дает возможность прямого измерения субъективного восприятия нечетких множеств на U со свойствами понятия a [Yager, 1982; Norwich et al., 1984]. В табл. 2.1 приведены наиболее часто встречающиеся типы шкал и связанные с ними аксиомы.

Процесс формализации знаний, полученных у эксперта, состоит из следующих шагов: выбор метода измерения нечеткости, получение исходных данных посредством опроса эксперта, реализация алгоритма построения функции принадлежности. Известные методы формализации нечеткости систематизированы в табл. 2.2. В процессе реализации метода используются следующие характеристики: тип метода измерения (П — прямой, К — косвенный); интерпретация принадлежности (ВЧ — вероятность частотная, ВС — вероятность субъективная, В — возможность, Д — детерминированная); процедура получения исходных данных (ОФ — определение функции принадлежности в виде формул, ОЗ — назначение значений принадлежности, ОДН — оценивание типа «да — нет»; ОПО — оценивание пар объектов; Р — ранжирование, РП — ранжирование пар объектов, ПС — попарное сравнение); тип измерений (Ф — фундаментальное, П — производное), тип шкалы (Н — номинальная, П — порядковая, И — интервальная, О — отношений, А — абсолютная).

Таблица 2.3

Номер оценки	Качественная оценка сходства	Вычисленная оценка	Предполагаемая оценка
1	Очень большое сходство, различия незаметно	1	1
2	Практическое сходство, существуют некоторые различия	0,85	0,8
3	Сходства больше, чем различий	0,7	0,6
4	Не ясно, чего больше: сходства или различий	0,35	0,5
5	Различий больше, чем сходства	0,23	0,4
6	Практическое различие, существует некоторое сходство	0,13	0,2
7	Очень большие различия, сходство незаметно	0	0

Таблица 2.4

Ранг	Пары оценок сходства
1	(1, 2), (2, 3), (6, 7)
2	(1, 3), (3, 4), (4, 5), (5, 6) (5, 7)
3	(1, 4), (2, 4), (2, 5), (4, 6)
4	(1, 5), (2, 6), (3, 5), (3, 6), (3, 7), (4, 8)
5	(1, 6)
6	(1, 7), (2, 7)

Таблица 2.5

*	1	2	3	4	5	6	7
1	*	1	2	3	4	5	6
2	1	*	1	3	3	4	6
3	2	1	*	2	4	4	6
4	3	3	2	*	2	3	4
5	4	3	4	2	*	2	2
6	5	4	4	3	2	*	1
7	6	6	4	4	2	1	*

Приведем пример измерения нечеткости. Множество оценок сходства приведено в табл. 2.3. В [Горячев и др., 1984] предполагается, что при оценке сходства используются числовые значения из табл. 2.3. Процедура формирования значений функции принадлежности следующая: 1) фиксация понятия «Сходство»; 2) ранжирование пар оценок сходства из табл. 2.3 по сходству в парах (чем больше сходство, тем меньше ранг); матрица сравнения пар оценок сходства приводится в табл. 2.4, 2.5 соответственно в строчной и матричной форме; 3) вычисление методом факторного пространства. Вычисленные значения функции принадлежности приведены в табл. 2.3.

2.2. Пополнение знаний

Л. В. Литвинцева, Д. А. Поспелов

Основные определения

Проблема пополнения знаний возникла при решении задач понимания естественного языка, обучения, поиска ответов на вопросы к базе знаний, анализа ситуаций, сцен и др. Знания могут быть представлены в виде фактов, хранящихся в базе знаний, или в виде описаний ситуаций, поступающих на вход ИС.

При описании ситуации человек обычно не стремится сообщить все подробности, относящиеся к этой ситуации. Однако большинство опущенных деталей необходимо для того, чтобы полностью понять ситуацию. В связи с этим требуются механизмы восстановления пропущенной информации. Например, если че-

ловек воспринимает сообщение «Воронам же, гнездящимся на ветках, все нипочем», то он скорее всего не будет думать, что ветки с воронами висят в пространстве. Он представит дерево, на ветвях которого расположены гнезда, а в них вороны. Однако даже такой тривиальный вывод требует достаточно большого числа рассуждений: что ветка является частью дерева, что птицы вьют гнезда на деревьях, что деревья растут на земле и т. д. Этот простой пример предполагает использование сложной процедуры вывода на основе знаний, хранящихся в памяти.

Можно указать несколько подходов к пополнению знаний: модели «здорового смысла» [Шенк, 1980], сценарии (см. § 1.5) [Литвинцева, 1986], подход, опирающийся на идею о том, что физические закономерности внешнего мира могут быть описаны в рамках специальных *псевдофизических логик* [Поспелов Д., 1986]. Все эти подходы в той или иной мере используют идею продукционных правил (см. § 1.4).

Система продукций отражает закономерности, присущие заданной проблемной области. Например, для ситуаций, описывающих статические пространственные отношения между объектами, можно выбрать следующие продукции:

x <содержится в> y , y <содержится в> $z \Rightarrow x$ <содержится в> z (свойство вложенности, которое выполняется в «матрешках»);

x <находится на> y , z <находится на> $x \Rightarrow z$ <находится над> y .

Пополняя постепенно базу знаний продукциями такого типа, можно в ИС создать достаточно мощные средства для порождения новых фактов из тех, которые в ней имеются. Процедуры пополнения знаний отличаются друг от друга структурой продукционных правил и стратегиями управления в системе продукций.

Псевдофизические логики

При определенных стратегиях управления система продукций превращается в дедуктивную логическую систему. Широкий класс таких систем называется *псевдофизическими логиками* (ПФЛ). В их аксиомах и правилах вывода используются не только реальные физические свойства окружающего мира, но и особенности его восприятия человеком. ПФЛ обладают рядом особенностей.

1. ПФЛ есть логики отношений: *временных, пространственных, каузальных, действий*.

2. Часть рассуждений в ПФЛ связана со *шкалами: метрическими (абсолютными и относительными) и топологическими*. На абсолютной метрической шкале задан некоторый масштаб и выбрана точка отсчета; на относительной указывается лишь расстояние между точками, а начало отсчета может «плавать». Топологическая шкала представляет собой порядковую шкалу. На ней указываются отношения порядка (строгого, нестрогого, размытого) между упорядочиваемыми объектами. Различные абсолютные шкалы преобразуются друг в друга. Для перехода от относительной шкалы к абсолютной требуется определить абсолютное значение начальной точки отсчета, после чего легко переводятся остальные точки шкалы. Изоморфизма между топологическими шкалами не существует.

3. ПФЛ содержат в качестве аксиом утверждения, вытекающие из восприятия мира человеком, которые подтверждаются результатами соответствующих психологических экспериментов.

4. Совокупность ПФЛ характеризуется связями между отдельными частями — логиками (например, логика времени тесно связана с логиками пространства и действий).

Существуют модели временных логик [Литвинцева и др., 1986], статической пространственной логики [Варосян и др., 1982], фрагменты логики действий и каузальной логики [Кандрашина и др., 1989].

Каждая ПФЛ является системой, включающей модель предметной области в виде понятий и отношений, существующих в ней, и модель вывода. Первая модель задает формальный язык описания знаний (ситуаций), вторая позволяет получать пополнение этих знаний.

Рассмотрим примеры конкретных ПФЛ и пополнение знаний на их основе.

Логика времени. Введем множество $T = \{t_1, t_2, t_3, \dots\}$, элементы которого назовем моментами времени, и множество $P = \{p_1, p_2, \dots, p_n\}$, элементы которого назовем событиями. Будем говорить, что событие p_i происходит в момент времени t_j , если ему можно приписать момент времени t_j . Если для p_j существует лишь один момент времени, в который оно происходит, то событие p_j называется *точечным*; если оно сопоставляется с некоторой непрерывной последовательностью моментов, то оно называется *интервальным*.

За исходные свойства времени примем односторонность, линейность, непрерывность, бесконечность, гомогенность. Заметим, что из свойств односторонности и линейности вытекает свойство транзитивности времени.

На множестве $s \subset P_1$ точечных событий зададим шесть временных отношений трех типов: неметрические (r_0 — одновременно, r_1 — быть раньше, r_2 — быть позже); метрические и частотные ($r_3^{n,L}$ — быть раньше на n единиц ($n=1, 2, 3, \dots$) по шкале L , $r_4^L t$ — происходить в момент t на шкале L , $r_5^L \pi$ — происходить с частотой π на шкале L , где $L \in \mathcal{L}$ — множество шкал).

На множестве $s \subset P_2$ интервальных событий зададим 14 временных отношений: R_1 — строго предшествовать во времени; R_2 — строго следовать; R_3 — пересекаться и др. [Литвинцева и др., 1986].

Выбранная базовая система временных отношений не является минимальной и единственно возможной. Стремление к минимальности не всегда оправданно. Чем меньше отношений, тем более громоздкое представление ситуаций и тем больше длина вывода.

Зададим множество M правил построения ППФ в логике времени, которыми являются комбинации троек типа $(p_i R p_j)$, где $p_i, p_j \in P$, $R \in \{r_0, r_1, \dots, r_5, R_1, R_2, \dots, R_{14}\}$. Множество аксиом Γ составляют продукции вида $\alpha, \beta \vdash \gamma$. Имеется также три правила вывода:

$$\frac{\Phi_1, \Phi_2 \vdash \Phi_3}{\Phi_3}, \quad \frac{\Phi_1 \wedge \Phi_2}{\Phi_1}, \quad \frac{\Phi_1 \wedge \Phi_2}{\Phi_2}.$$

Приведем примеры схем аксиом в логике времени:

$$(p_i r_3^{n,L} p_j), (p_i r_3^{m,L} p_k) \vdash (p_i r_3^{n+m,L} p_k), \quad (1)$$

$$(p_i r_4^L t), (p_i r_3^{n,L} p_j) \vdash p_j r_4^L (t \oplus n), \quad (2)$$

где \oplus — операция сложения времен на шкалах,

$$(p_i r_3^{n,L} p_j) \vdash (p_i r_1 p_j), \quad (3)$$

$$(p_i r_1 p_j), (p_j r_1 p_k) \vdash (p_i r_1 p_k). \quad (4)$$

Посмотрим, как работают эти схемы в задаче пополнения знаний. Пусть на вход ИС поступает следующее описание ситуации в виде текста на естественном языке: «Самолет, следовавший рейсом Баку — Москва, совершил посадку в аэропорту в 15 ч 20 мин. Через 10 мин был подан трап. Затем прибыл автобус. Пассажиры покинули самолет. Через 3 мин автобус доставил их в здание аэровокзала». Для простоты будем считать, что события данного текста точечные. Выделим их: p_1 — «посадка самолета», p_2 — «подача трапа», p_3 — «прибытие автобуса», p_4 — «выход пассажиров из самолета», p_5 — «доставка пассажиров в здание аэровокзала». Описание ситуации на языке M выглядит так:

$$TS = (p_1 r_4^L t) \wedge (p_1 r_3^{10,L} p_2) \wedge (p_2 r_1 p_3) \wedge (p_4 r_3^{3,L} p_5),$$

где $t=15$ ч 20 мин на шкале L с единицей измерения «минута».

Формальное представление событий текста обрабатывается процедурой пополнения, которая выведет следующие факты:

$$(p_1 r_4^L t), (p_1 r_3^{10,L} p_2) \vdash (p_2 r_4^L t_*).$$

где $t_*=15$ ч 30 мин на шкале L (по схеме 2);

$(p_1 r_3^{10,L} p_2) \vdash (p_1 r_1 p_2)$ (по схеме 3);

$(p_1 r_1 p_2), (p_2 r_1 p_3) \vdash (p_1 r_1 p_3)$ (по схеме 4);

$(p_4 r_s^{3,L} p_5) \vdash (p_4 r_1 p_5)$ (по схеме 3).

Полученные в результате вывода дополнительные факты служат для расширения ситуации, описанной исходным текстом.

Логика действий. Псевдофизическая логика действий (ПФЛД) представляет собой логическую систему, описывающую семантические и прагматические свойства физических действий, условия их протекания во времени и пространстве, их взаимодействие, а также цели и мотивы действующих лиц. Действия могут быть связаны между собой различными способами: во времени, в пространстве, посредством одного и того же исполнителя, одного объекта, на которое направлено действие. Наряду с этими простейшими связями существуют и более сложные зависимости. Одна из них — каузальная.

Различаются следующие типы каузальных отношений между действиями:

π_1 — необходимая и достаточная причина. Действия d_1, d_2 связаны отношением π_1 , если реализация d_1 всегда вызывает реализацию d_2 , и, наоборот, появление d_2 всегда вызывается d_1 ; чаще всего π_1 отражает различные физические законы реального мира (например, сверкнула молния — грянул гром).

π_2 — достаточная причина, которая означает, что реализация d_1 всегда вызывает d_2 , однако из появления d_2 не всегда следует появление d_1 (например, «нырнул в реку, оказался в воде»).

π_3 — обуславливающая причина. Действия d_1, d_2 связаны отношением π_3 , если реализация d_1 обеспечивает необходимые условия для реализации d_2 , которое, однако, может и не произойти (например, «вошел в комнату, включил свет»).

Опишем действия в виде предикатов $d_i (S_k, O_j, I_m, t_i, l_p)$, где в качестве аргументов выступают глубинные падежи данного действия: субъект, объект, инструмент, время реализации действия и его локализация. Введем следующие множества: множество действий $D = \{d_1, d_2, \dots, d_n\}$; множество объектов $O = \{O_1, O_2, \dots, O_m\}$; множество инструментов $I = \{I_1, I_2, \dots, I_l\}$; множество времен $T = \{t_1, t_2, \dots, t_i\}$; множество локализаций $L = \{l_1, l_2, \dots, l_p\}$.

На множестве действий рассмотрим множество из восьми отношений, характеризующих их взаимосвязь: R_1^D — действие d_i является частью действия d_j ; R_2^D — действие d_i приводит к результату d_j ; R_3^D — действие d_i сопровождается действием d_j ; R_4^D — субъект действия $d_i(S_k)$ имеет цель d_j ; R_5^D — субъект действия $d_i(S_k)$ имеет мотив d_j (считается, что в качестве результатов, целей и мотивов выступают другие действия). Сюда же относятся три каузальных отношения π_i между действиями. Выделенная группа отношений составляет базовую систему отношений ПФЛД.

Схемы аксиом ПФЛД описывают семантические и прагматические свойства физических действий. Приведем несколько схем правил, описывающих действия:

- 1) $d_i(t_i) R d_j(t_j) \vdash t_i < \text{быть раньше} > t_j$, где $R = \{\pi_1, \pi_2, \pi_3\}$;
- 2) $d_i(t_i) R_3^D d_j(t_j) \vdash t_i < \text{пересекается} > t_j$;
- 3) $d_i(S_i, O_j, I_z, t_m) \vdash d'(O_j, I_k, t_m)$, где $d' = \{\text{находиться около}\}$, $d_i \in D' \subset D$;
- 4) $d_i(S_i, O_j, t_k) \vdash d'(S_i, O_j, t_k)$, где $d' = \{\text{находиться около}\}$, $d_i \in D' \subset D$;
- 5) $d_i R_2^D d_j, d_j R_2^D d_k \vdash d_i R_2^D d_k$; 6) $d_i(S_j) \vdash d_i R_4^D d_k$;
- 7) $d'(S_i, I_j, t_k) \vdash d' R_2^D d''(S_i, I_j, t_m)$, где $t_m = t_k + \Delta t$, $d' = \{\text{переместиться внутрь; войти; ... и т. п.}\}$, $d'' = \{\text{находиться в}\}$;
- 8) $d'(S_i, I_j, t_k) \vdash d' R_2^D d''(S_i, I_j, t_m)$, где $t_m = t_k + \Delta t$, $d' = \{\text{переместиться изнутри; выйти; ... и т. п.}\}$, $d'' = \{\text{находиться вне}\}$;
- 9) $d'(S_i, I_p, t_k), (l_p < \text{часть} > l_m) \vdash d'(S_i, I_m, t_k)$, где $d' = \{\text{войти; ... и т. п.}\}$;
- 10) $d'(O_i, I_m, t_k), d''(S_p, O_i, t_k) \vdash d''(S_p, I_m, t_k)$, где $d' = \{\text{двигаться к; мчаться; помчаться; ... и т. п.}\}$, $d'' = \{\text{находиться в; сидеть; ... и т. п.}\}$.

Схемы правил позволяют анализировать и пополнять динамические ситуации. Рассмотрим пример такого пополнения. Пусть на вход ИС поступил следующий текст: «На маленькой станции он вошел в вагон, и поезд помчался

на юг вдоль проселочной дороги. Промелькнуло несколько деревень. Он вышел, когда поезд остановился у берега». Выпишем действия, упомянутые в рассказе; d_1 — «вошел», d_2 — «помчался», d_3 — «промелькнуло», d_4 — «вышел», d_5 — «остановился». Используя схемы правил, получаем:

$$\begin{aligned} d_1(\text{ОН, вагон, } t_1), (\text{вагон} < \text{часть} > \text{поезд}) | - d_1(\text{ОН, поезд, } t_1); \\ d_1(\text{ОН, поезд, } t_1) | - d'(\text{ОН, поезд, } t_1 + \Delta t), \text{ где } d' = \{\text{находиться в}\}; \\ d'(\text{ОН, поезд, } t_1 + \Delta t), d_2(\text{поезд, юг, } t_1 + \Delta t), d_2(\text{ОН, юг, } t_1 + \Delta t); \\ d_4(\text{ОН, вагон, } t_2) | - d''(\text{ОН, вагон, } t_2 + \Delta t), \text{ где } d'' = \{\text{находиться вне}\}. \end{aligned}$$

Описанный вариант ПФЛД представляет собой попытку смоделировать рассуждения человека о физических действиях в рамках здравого смысла. Концептуальное ядро ПФЛД составляет система отношений на действиях и схемы правил (аксиом), описывающие их свойства.

Пополнение знаний на основе сценариев

Пополнение знаний можно выполнить с помощью сценариев (см. § 1.5). Опишем основные схемы рассуждений на сценариях, используемых при выводе. Часть схем является интерпретацией значений слотов сценария (СЦ).

Схема 1. Пусть в СЦ значением слота «ключ» является имя события p_* , а значением слота «посылки» — список имен событий (p_1, p_2, \dots, p_k) . В этом случае имеют место соотношения $(p_i R_1 p_*)$, $(p_i \pi_3 p_*)$, $i = 1, 2, \dots, k$, где R_1 — отношение предшествования во времени, π_3 — каузальное отношение.

Схема 2. Пусть в СЦ значением слота «ключ» является имя события p_* , а значением слота «следствия» — список имен событий (p_1, p_2, \dots, p_m) . Тогда имеют место соотношения $(p_i R_1 p_*)$, $(p_i \pi_3 p_*)$, $i = 1, 2, \dots, k$, где R_1 — отношение «следовать за».

Схема 3. Обозначим множество значений слотов «посылки» через X , а множество значений слота «следствие» через Y . Имеют место следующие утверждения: для $\forall p_i \in X$ и $\forall p_j \in Y$ верно $(p_i \ll \text{предшествует} \gg p_j)$.

Схема 4. Пусть π — значение слота «цель деятеля» в СЦ, p_* — имя ключевого события. Верно следующее: $(p_* R_1 \pi)$, $(p_* \pi_3 \pi)$.

Схема 5. Пусть p_1^* — значение слота «ключ» в СЦ1, p_2^* — значение аналогичного слота в СЦ2. Известно, что $p_1^* \pi_3 p_2^*$. Обозначим множество значений слота «посылки» в СЦ1 через X_1 , а в СЦ2 — через X_2 ; множество значений слота «следствие» в СЦ1 — через Y_1 , а в СЦ2 — через Y_2 . Имеют место следующие отношения: из $\forall p_i \in X_1$ и $\forall p_j \in X_2$ следует $(p_i R_1 p_j)$; из $\forall p_k \in Y_1$ и $\forall p_m \in Y_2$ следует $(p_k R_1 p_m)$.

Схема 6. Рассмотрим случай схемы 5 — $p_1^* \pi_3 p_2^*$. Имеет место следующее: из $\forall p_i \in X$ следует $(p_i \pi_3 p_2^*)$, где $X = X_1 \cup X_2$.

Рассмотрим использование каузальных сценариев (КСЦ) для планирования действий во времени. Представим ситуацию «пожар», в которой находится робот. Цель его поведения — устранение пожара. Для нахождения последовательности действий, удовлетворяющей поставленной цели, планирующая система робота обращается к базе знаний, в которой находятся следующие сценарии:

- (КСЦ «тушение пожара»:
 - деятель (s);
 - цель деятеля (p : «устранение пожара»);
 - местоположение объекта пожара (l : *loc.*);
 - посылки (СЦ («поиск средств тушения» R_1 , «транспортировка средств тушения к объекту пожара»);
 - ключ (f : «контакт средства тушения с огнем до полного прекращения огня»);
 - следствие (p : «прекращение огня»);
 - системное имя (sys : СЦ*1)).
- (КСЦ «поиск средств тушения»,
 - деятель (s);

цель деятеля (p : «нахождение средств тушения»);
 средства тушения (p : «вода»);
 посылки (f («определение местонахождения средства тушения» R_1 ,
 «движение к местонахождению средства тушения»));
 ключ (f : «схватывание средства тушения»);
 следствие (p : «нахождение в месте расположения объекта средства тушения»);
 системное имя (sys : $CЦ*2$)).

3. (КЦИ «транспортировка средства тушения к объекту пожара»:

деятель (s):
 цель деятеля (p : «наличие средства тушения в месте пожара»);
 посылки (f («наличие средства тушения у деятеля» R_1 , «наличие координат местоположения объекта пожара»));
 ключ (f : «передвижение к объекту пожара»);
 следствие (p : «нахождение у объекта пожара»);
 системное имя (sys : $CЦ*3$)).

Спецификация значения слота f означает, что далее идет название процедуры, которая должна быть выполнена при реализации данного сценария. Отношение R_1 представляет временное отношение «предшествовать на n единиц по шкале L ».

Общая схема построения плана достижения цели по сценариям такова:

1. Из описания проблемной области выбирается сценарий, значение слота «цель» которого соответствует поставленной цели (для рассматриваемого случая — первый сценарий).

2. Анализируются значения всех слотов указанного сценария и строится последовательность Π_1 , являющаяся значением слотов «посылки» и «ключ» с их спецификациями. Для примера $\Pi_1 = CЦ$ («поиск средств тушения») R_1 ($CЦ$ «транспортировка средства тушения к объекту пожара») R_1 (f : «контакт средства тушения с огнем до полного прекращения огня»). Заметим, что второе отношение R_1 получено с использованием схемы 1.

3. Выполняется последовательное обращение к сценариям, указанным в Π_1 . При этом фрагмент последовательности Π_1 , имеющий вид ($CЦ$ <имя>), заменяется соответствующей последовательностью Π_2 , построенной аналогично Π_1 :

$\Pi_1 = (f$ «определение местонахождения средства тушения») R_1 (f «движение к местонахождению средства тушения») R_1 (f «схватывание средства тушения»).

В цепочке Π_2 отсутствуют спецификации типа $CЦ$, которые должны быть заменены. Следовательно, построение Π_2 закончено, и происходит возврат к обработке следующего фрагмента Π_1 .

Второй фрагмент Π_1 со спецификацией $CЦ$ заменяется на следующий:

$\Pi_3 = (f$ «наличие средства тушения у робота») R_1 (f «наличие координат местоположения объекта пожара») R_1 (f «передвижение к объекту пожара»).

Последовательность Π_3 также не содержит спецификаций $CЦ$. На этом ее построение заканчивается.

Полученная расширенная последовательность Π_1 представляет собой план действий во времени, состоящий из имен конкретных процедур, реализация которых приводит к достижению поставленной перед роботом цели устранения пожара.

$\Pi_1 = (f$ «определение местонахождения средства тушения») R_1 (f «движение к местонахождению средства тушения») R_1 (f «схватывание средства тушения») R_1 (f «наличие средства тушения у деятеля») R_1 (f «наличие координат местоположения объекта тушения») R_1 (f «передвижение к объекту пожара») R_1 (f «контакт средства тушения с огнем до полного прекращения огня»).

2.3. Обобщение и классификация знаний

В. Н. Вагин, Н. П. Викторова

Суть проблемы

Способность человеческого мышления к обобщению лежит в основе любого научного исследования. Именно в результате обобщения получаются новые знания, т. е. знания, не следующие непосредственно из ранее известных. В системах, моделирующих мышление, *обобщение* понимают как процесс получения знаний, объясняющих имеющиеся факты и способных объяснять, классифицировать или предсказывать новые. В общем виде задача обобщения формулируется следующим образом [Michalski et al., 1983]: по совокупности наблюдений (фактов) F , совокупности требований и допущений к виду результирующей гипотезы H и совокупности базовых знаний и предположений, включающих знания об особенностях предметной области, выбранном способе представления знаний, наборе допустимых операторов, эвристик и др., сформировать гипотезу $H: H \Rightarrow F$ (H «объясняет» F).

Форма представления и общий вид гипотезы H зависят от цели обобщения и выбранного способа представления знаний. Модели обобщения, включающие модели классификации, формирования понятий, распознавания образов, обнаружения закономерностей, определяются целями обобщения, способами представления знаний, общими характеристиками фактов, критериями оценки гипотез.

Согласно [Michalski et al., 1983] можно выделить модели *обобщения по выборкам* и модели *обобщения по данным*. В первом случае совокупность фактов F имеет вид обучающей выборки — множества объектов e_{ij} , $i \in I$, каждый из которых сопоставляется с именем класса K_j , $j \in J$. Целью обобщения в этом случае может являться:

формирование понятий: построение по данным обучающей выборки для каждого класса K_j , $j \in J$, максимальной совокупности его общих характеристик [Бонгард, 1967; Cohen, 1977; Уинстон, 1978; Гитлина и др., 1981а, б];

классификация: построение по данным обучающей выборки для каждого класса K_j , $j \in J$, минимальной совокупности характеристик, которая отличала бы элементы K_j от элементов других классов [Quinlan, 1979; Michalski, 1980; Харалик, 1982; Вагин и др., 1985];

определение закономерности последовательного появления событий [Dietterich, 1979].

К моделям обобщения по выборкам относятся лингвистические модели [Фу, 1977], методы автоматического синтеза алгоритмов и программ по примерам [Biermann, 1978; Бардэинь, 1982] и др.

В моделях обобщения по данным априорное разделение фактов по классам отсутствует. Здесь могут ставиться следующие цели:

формулирование гипотезы, обобщающей данные факты [Morgan, 1971; Plotkin, 1971; Финн, 1983];

выделение образов на множестве наблюдаемых данных, группировка данных по признакам [Бонгард, 1967; Александров и др., 1983] (задача формирования понятий, определенная в модели обобщения по выборкам, также часто ставится без априорного разбиения обучающей выборки по классам);

установление закономерностей, характеризующих совокупность наблюдаемых данных [Загоруйко, 1981; Гаек и др., 1984].

С точки зрения способа представления знаний и допущений на общий вид объектов наблюдений e_i методы обобщения делятся на методы *обобщения по признакам* и *структурно-логические* (или *концептуальные*) методы. В первом случае объекты $e_i \in F$, $i \in I$, представляются в виде совокупности значений косвенных признаков π_k , $k \in Q$ [Хант и др., 1970; Харалик, 1982]. Методы обобщения и распознавания по признакам различаются для качественных и количественных (измеримых) значений признаков.

В формально-логических системах, использующих структурно-логические методы обобщения, вывод общих следствий из данных фактов называют *индуктивным выводом*. Правило вывода $F|H$ гипотезы H из фактов F называют индуктивным, если из истинности H следует истинность F , а обратное неверно [Забейайло и др., 1982; Финн, 1983; Гаек и др., 1984; Michalski et al., 1983]. В [Plotkin, 1971] сформулированы основные вопросы, на которые должны давать ответы индуктивные логики и логики выдвижения гипотез: 1. Является гипотеза H обоснованной данным знанием? 2. Существуют методы обоснования H при данном знании? 3. Каковы условия для H при данном знании, такие, что H дает наиболее разумное и интересное объяснение? 4. Существуют методы для выдвижения гипотез на основании данного знания, дающих наиболее разумное и интересное объяснение изучаемого явления?

В ИС для представления знаний об объектах используются обычно модификации языка исчисления предикатов без ограничения аристотелевской предикатных символов или адекватные им по выразительной мощности семантические сети.

Моделям обобщения на семантических сетях [Гитлина и др., 1981б; Вагин и др., 1985] свойственны черты как алгоритмов обобщения по признакам, так и индуктивной логики. Здесь также определяется набор операторов, используемых при формировании обобщенного представления (гипотезы) H и выдвигаются критерии оценки «интересности» и обоснованности гипотез. Кроме того, в этих моделях широко используется характерный для обобщения по качественным признакам [Гуревич и др., 1974; Харалик, 1982] аппарат теории покрытий (см. гл. 4) и устанавливаются отношения на множестве значений признаков объектов — элементов сети. Методами структурного обобщения решаются обычно задачи классификации, формирования понятий, анализа сцен [Уинстон, 1978].

Для задачи обобщения по признакам известен следующий результат: каков бы ни был реальный вид разделяющей функции ψ (в общем случае индуктивной гипотезы H) и алгоритм ее формирования по обучающей выборке, всегда найдется такая (непустая) обучающая выборка, что сформированная функция ψ' (гипотеза H') явится некорректной (ложной).

В связи с этим гипотезы принято оценивать с точки зрения их «разумности», «рациональности», «интересности». Так, в [Гаек и др., 1984] рациональность ответа на вопрос 1 индуктивного вывода понимается следующим образом. Пусть Φ — истинные утверждения, ϕ — эмпирические данные. Тогда для

каждой Σ -зависимой структуры U , если $\Phi, \phi|=\psi$, то при условии ложности ψ на U вероятностная мера наблюдения $M_\phi (M \subseteq U, \sigma \in \Sigma)$, такого, что ϕ истинно на M_ϕ , должна быть мала (например, меньше 0,05).

Для оценки обоснованности гипотезы в ДСМ-методе [Забейайло и др., 1982; Финн, 1983] используется квантор $J_m, m \in \{0, 1/(n-1), 2/(n-1), \dots, 1\}$. Значение $m=1/(n-1)$ соответствует гипотезам, достоверность которых неизвестна, значение $m=1$ — истинным гипотезам. Если применяемое правило вывода подтверждает гипотезу, то значение m возрастает, если не подтверждает, то уменьшается.

В алгоритмах обобщения, использующих аппарат покрытий [Гитлина и др., 1981а, б; Вагин и др., 1982; Харалик, 1982], принято оценивать гипотезы с точки зрения мощностей подмножеств покрываемых ими элементов обучающей выборки.

В ряде исследований для подтверждения или отрицания выдвигаемой гипотезы используются методы автоматического порождения новых элементов обучающей выборки, которые выдаются для классификации человеку-эксперту. Решающее правило переопределяется, пока не будет достигнута равновесная ситуация.

Методы обобщения находят применение в экспертных системах, системах распознавания образов и анализа сцен, в задачах автоматизированного исследования закономерностей явлений физики, химии, биологии, медицины [Buchanan et al., 1978; Гитлина и др., 1981а, б; Hájek et al., 1982]. На-

личные подсистемы, решающих задачи обобщения, обязательно для систем ситуационного управления [Поспелов Д., 1982, 1986], в которых должны решаться задачи формирования обобщенных описаний классов ситуаций системы, соответствующих различным управляющим воздействиям, задачи структуризации ситуаций, получения по текущей ситуации ее иерархической модели — «слоеного пирога». Формирование иерархической модели требует решения как задач формирования понятий на этапе обучения, так и непосредственно задачи структуризации [Фомина, 1985].

Рассмотрим кратко связь между задачами обобщения и классификации и задачами, решаемыми в рамках теории вероятностей и математической статистики. Фактически в математической статистике также ставятся и решаются задачи вывода новых знаний на основании анализа совокупности наблюдений. В статистике устанавливаются частотные закономерности появления событий, т. е. определяются общий вид и параметры функций распределения вероятностей событий по данным наблюдений, делаются выводы о степени статистической зависимости наблюдаемых случайных величин, принимаются решения о выборе гипотезы о характеристиках случайного события на основании применения байесовской процедуры последовательного анализа наблюдений [Вальд, 1962; Барабаш и др., 1967; Хазен, 1968]. Общая задача формирования гипотез по данным наблюдений, рассматриваемая как одно из направлений развития ИС, не ограничивается установлением статистических закономерностей. Известны формально-логические модели выдвижения статистических гипотез [Гаек и др., 1984]. Действительно, ставя перед собой задачу формализации представления и вывода знаний о реальном мире, нельзя не учитывать наличия статистических закономерностей в его проявлениях.

Методы обобщения по признакам

На признаковых структурах решаются обычно задачи классификации и формирования понятий. Обзор и анализ методов обобщения по признакам можно найти в [Журавлев, 1978а, б; Ту и др., 1978; Хант, 1978; Поспелов Д., 1986], а также в гл. 4.

Задача классификации по признакам ставится следующим образом [Журавлев, 1978а, б; Поспелов Д., 1986]. Пусть имеется некоторое множество Z объектов и множество $\Pi = \{\pi_i\}$, $i \in \overline{1, n}$, признаков. Для каждого $\pi_i \in \Pi$, $i \in \overline{1, n}$, задана область его определения D_i . Каждый объект $s \in Z$ однозначно определяется вектором Π , значений его признаков: $\Pi_s \in D_1 \times \dots \times D_n$. Известно, что классы K_1, K_2, \dots, K_m являются подмножествами Z . Для каждого K_j , $j \in \overline{1, m}$, задана информация I_j в виде обучающей выборки $\langle v_j^+, v_j^- \rangle$ множеств положительных и отрицательных примеров: $v_j^+ \in K_j$; $v_j^- \cap K_j = \emptyset$. Требуется построить *решающие правила* ψ_j , $j \in \overline{1, m}$, позволяющие определить принадлежность произвольного объекта $s \in Z$ классам K_j .

В задаче формирования понятий по обучающей выборке $v \in Z$ требуется по тем или иным критериям выделить совокупность классов $\{K_j\}$ и построить для них решающие правила ψ_j . В случае, когда области D_i , $i \in \overline{1, n}$, обладают метрикой, каждый объект обучающей выборки можно представить точкой n -мерного признакового пространства; тогда классы будут являться областями этого пространства и решающие правила ψ_j , $j \in \overline{1, m}$, можно представить в виде разделяющих поверхностей в пространстве признаков. В процессе обобщения происходит настройка коэффициентов разделяющих функций, которые, как правило, выбираются линейными, квадратичными или кусочно-линейными. Описанный метод называется методом разделения.

Метод потенциальных функций [Айзерман и др., 1970] также применим лишь в случае, когда признаки являются измеримыми величинами. Потенциальная функция ψ_K для класса K строится так, чтобы ее значение на множестве объектов $s \in K$ было максимальным.

В [Журавлев и др., 1974; Журавлев, 1978а, б] предлагаются способы классификации по признакам, основанные на вычислении оценок (методы голосования). Вначале определяется совокупность \mathcal{Z} решающих подмножеств на

множестве признаков (обычно это множества тупиковых тестов или подмножества выбранной мощности), а затем определяется мера близости объекта s каждому из классов K_j на основании сравнения значений признаков объекта s из заданных подмножеств с соответствующими значениями признаков эталонных объектов. Данные модели пригодны также для обработки качественных (неизмеримых) признаков и используют методы покрытий (см. гл. 4).

Для решения задачи классификации по качественным признакам чаще всего применяется *техника покрытий*. В [Харалик, 1982] эта задача формулируется следующим образом. Пусть C — множество классов: $C = \{K_1, K_2, \dots, K_m\}$, Z — множество образов, v — множество примеров. Тогда $T \subseteq v \times C$ — обучающая выборка, представляющая собой отношение принадлежности примеров классам. Решающее правило ψ является отношением $T \subseteq \psi \subseteq Z \times C$. В результате обобщения формируется совокупность \mathcal{P} подмножеств, покрывающих v и обладающих следующими свойствами при $L \in \mathcal{P}$: 1) $L \cap v \neq \emptyset$; 2) $\exists K \in C: s \in L \cap v \Rightarrow (s, K) \in T$.

Классическим методом формирования решающих правил на множестве булевых переменных является метод минимизации частично определенных булевых функций [Mc Cluskey, 1956]. Вариант этого метода для небулевых переменных предложен в [Michalski, 1973]. В [Haralick, 1977; Харалик, 1982] описан метод покрытия, предназначенный для распознавания как n -мерных векторов, так и цепочек символов. В основе метода лежит идея покрытия обучающей выборки цилиндрическими множествами, где под цилиндрическим понимается такое подмножество $L \subseteq D_1 \times D_2 \times \dots \times D_n$, что для выделенной совокупности индексов j_1, j_2, \dots, j_k значения признаков фиксированы, а для остальных совпадают с областями их определения.

Отметим, что методы разделения применимы только в том случае, когда реальный вид классов K_j , $j \in \overline{1, m}$, допускает разделение функциями выбранного вида. Так, невозможно получить методом разделения характеристические функции множеств четных или нечетных чисел. В [Бонгард, 1967] для данной области применения выбирается богатый набор логических функций, определенных на множестве признаков, значения которых могут вычисляться процедурно. Указанные функции могут отображать логические, арифметические или иные отношения на признаках. Решающее правило строится в виде булевой функции в процессе анализа элементов обучающей выборки.

Перспективным с точки зрения использования в ИС является метод обобщения, называемый методом растущих пирамидальных сетей (РПС) [Гладун, 1970; Гладун и др., 1975; Гладун, 1977; Поспелов Д., 1986]. Решающее правило формируется в виде РПС путем многократного просмотра обучающей выборки. На нулевом уровне РПС находятся вершины-рецепторы, соответствующие множеству всех возможных значений признаков. Метод применим, в частности, для формирования понятий, зависящих от времени или имеющих динамическую природу.

Методы формирования понятий по обучающей выборке $v \subseteq Z$ без априорного разбиения на классы рассматриваются в кластерном анализе [Классификация, 1980; Александров и др., 1983]. В класс (кластер) группируются объекты, близкие друг к другу в признаковом пространстве. В [Бонгард, 1967] предложен метод формирования понятий, называемый «развал на кучи» и состоящий в группировании объектов вокруг некоторого «ядра».

Для решения задачи обобщения по признакам необходимо первоначально выделить совокупность «информативных» признаков, характеризующих объекты $s \in Z$. Методы определения словаря признаков предложены в [Горелик, 1973; Горелик и др., 1977].

В синтаксических методах распознавания [Нарасимхан, 1969; Фу, 1977; Фи, 1978] образы (элементы $s \in Z$) представляются в виде цепочек символов — слов некоторого языка. Решающее правило ищется в виде грамматики G , такой, что цепочки символов из v^+ (множества положительных примеров) порождаются G , а цепочки из v^- (множества контрпримеров) не порождаются. Для этой цели также часто используют аппарат покрытий.

Структурно-логические методы обобщения

Структурно-логические методы в отличие от признаковых предназначены для решения задачи обобщения на множестве объектов, имеющих внутреннюю логическую структуру (последовательности событий, иерархически организованные сети, характеризуемые как признаками и свойствами объектов — элементов сети, так и отношениями между ними, алгоритмические и прогаммные схемы) [Кайберг, 1978; Michalski et al., 1983; Финн, 1984; Поспелов Д., 1986]. Сначала рассмотрим методы индуктивного вывода в формальных исчислениях, а затем — методы обобщения на семантических сетях.

Основой каждой модели индуктивного вывода является набор правил (схем) индуктивного вывода. В [Plotkin, 1970; Plotkin, 1971] за основную схему вывода была выбрана схема индуктивной резолюции (обозначения по [Michalski et al., 1983])

$$\& \quad \frac{P \& F_1 \Rightarrow K}{\neg P \& F_2 \Rightarrow K} \Big| = F_1 \vee F_2 \Rightarrow K$$

или в другой форме

$$\vee \quad \frac{P \& F_1 \Rightarrow K}{\neg P \& F_2 \Rightarrow K} \Big| = F_1 \& F_2 \Rightarrow K$$

(знак $|=$ обозначает отношение индуктивного следования).

Эта схема применяется в классическом исчислении предикатов первого порядка вместе с алгоритмом антиунификации. Результатом применения правила индуктивной резолюции к двум предложениям является их «наименее общее обобщение».

В [Michalski et al., 1983] строится система индуктивного вывода в рамках модификации языка исчисления предикатов первого порядка, включающей обобщенные кванторы и префиксную запись отношений равенства и неравенства. Помимо принципа индуктивной резолюции используются следующие схемы вывода:

$$\begin{aligned} \text{CTX} \& S \Rightarrow K &|= \text{CTX} \Rightarrow K && \text{(селекция)}, \\ \text{CTX} \& [L = R_1] \Rightarrow K &|= \text{CTX} \& [L = R_2] \Rightarrow K, && \text{(расширение)}, \end{aligned}$$

где $R_1 \subseteq R_2 \subseteq \text{DOM}(L)$

$$\begin{aligned} \text{CTX}_1 \Rightarrow K &|= \text{CTX}_1 \vee \text{CTX}_2 \Rightarrow K && \text{(добавление);} \\ \text{CTX} \& [L = a] \Rightarrow K && \\ \vdots && \\ \text{CTX} \& [L = Z] \Rightarrow K && \Big| = \text{CTX} \& [L = [a, z]] \Rightarrow K && \text{(переход к интервалу);} \\ \begin{matrix} F[a] \\ \vdots \\ F[i] \end{matrix} && \Big| = \forall v F(v) && \text{(классическая индукция)} \end{aligned}$$

и др.

Схема формирования гипотез по обучающей выборке (метод «звезды») заключается в применении к множеству фактов и полученных утверждений всех возможных правил вывода (как индуктивных, так и дедуктивных). На каждом этапе поиска выбирается определенное количество наиболее «перспективных» согласно выбранным эвристикам гипотез. Процесс заканчивается, когда достигается цель обобщения.

Правила индуктивного вывода, используемые в ДСМ-методе [Финн и др., 1981; Забейало и др., 1982; Финн, 1983] основываются на схемах, предложенных в [Милль, 1900].

Получивший широкую известность GUHA-метод [Гаек и др., 1984] позволяет формировать множество «всех интересных гипотез» о наблюдаемых дан-

ных. Гипотезы, порождаемые GUHA-методом, имеют вид $\psi \sim \varphi$, где ψ и φ — утверждения, имеющие единственную свободную переменную x , а \sim — связывающий x ассоциативный или индуктивный квантор: $(\sim x)(\psi, \varphi)$. Выражение $\psi \sim \varphi$ в зависимости от выбранного квантора может служить для представления широкого диапазона зависимостей между ψ и φ (от логического следования до статистической корреляции). Для утверждений специального класса метод позволяет сформировать такое подмножество гипотез, что все остальные следуют из них в соответствии с принятым набором правил вывода.

В качестве языка представления знаний (отдельно рассматриваются языки представления эмпирических и теоретических утверждений) используется модификация языка исчисления предикатов первого порядка с обобщенными кванторами в смысле [Mostovski, 1957]. Каждому квантору q ставится в соответствие функция Asf_q , вычисляющая значение его истинности на каждой модели $\langle M, f_1, f_2, \dots, f_n \rangle$, где M — непустое множество из некоторого семейства \mathcal{M} , а f_1, f_2, \dots, f_n — набор логических функций, определенных на этом множестве.

Квантор q , определенный на моделях $\langle M, \psi, \varphi \rangle$ типа $\langle 1, 1 \rangle$ (кортеж арностей функций), называют ассоциативным в следующем случае. Пусть a_M — количество элементов M , на которых ψ и φ одновременно истинны, b_M — количество элементов, на которых ψ истинно, а φ ложно; c_M : φ истинно, ψ ложно; d_M : как ψ , так и φ ложно.

Тогда если $a_{M_1} \geq a_{M_2}$, $b_{M_1} \leq b_{M_2}$, $c_{M_1} \leq c_{M_2}$ и $d_{M_1} \geq d_{M_2}$, то для ассоциативного квантора условие $Asf_q \langle M_2, \psi, \varphi \rangle = 1$ влечет $Asf_q \langle M_1, \psi, \varphi \rangle = 1$. Определение импликативного квантора аналогично для условий $a_{M_1} \geq a_{M_2}$ и $b_{M_1} \leq b_{M_2}$.

Для задания GUHA-метода требуется определить ассоциативный или импликативный квантор \sim , квантор эквивалентности \Leftrightarrow и еще ряд характеристик (например, запрет или разрешение использовать дедуктивные правила вывода). Особое внимание уделяется методам формирования гипотез $\psi \sim \varphi$, таких, что ψ представляет собой элементарную конъюнкцию, а φ — элементарную дизъюнкцию одноместных предикатов. С помощью языка, использующего понятие случайной модели $\langle M, f_1, f_2, \dots, f_n \rangle$, где $\sigma \in \Sigma$ — случайный параметр, вводятся вероятностная мера достоверности гипотез и рассматриваются статистические гипотезы. Важный класс ассоциативных (и импликативных) кванторов образуют кванторы, функции Asf которых являются тестами для проверки статистических гипотез. GUHA-метод нашел применение в медицине [Тума, 1970], социологии [Navranek et al., 1977], лингвистике, строительстве,

В [Ефимов, 1979] предложен алгоритм индуктивного формирования понятий вида $\bigwedge_1 x_{i_1}, \dots, \bigwedge_n x_{i_n} C(x_{i_1}, \dots, x_{i_n})$, где каждый \bigwedge_i — квантор всеобщности или существования, а $C(x_{i_1}, \dots, x_{i_n})$ — выражение в конъюнктивной нормальной форме.

Понятие формируется из определения конкретного набора кванторов и определения содержимого понятия $C(x_{i_1}, \dots, x_{i_n})$ (см. § 4.3). В [Погосян, 1977] процедура индуктивного обобщения формализована для задачи формирования описаний (см. гл. 6).

Вопросы взаимосвязи теории вероятностей и индуктивной логики рассматриваются в [Carnap, 1950; Hintikka, 1965], где разработана концепция индуктивной вероятности, а также установлена связь между отношением логического следования и понятием индуктивной вероятности.

Применение аппарата семантических сетей для решения задач классификации и формирования понятий позволяет получать обобщенные представления множеств объектов, имеющие наглядную семантическую трактовку, и использовать в процессе обобщения семантику таких понятий, как «признак», «имя», «класс», «отношение». Роль языка представления наблюдений и классов (по-

нений) играет в данном случае выбираемый формализм семантической сети.

Рассмотрим методы *обобщения на сетях* [Вагин и др., 1982; 1985]. Пусть Z — множество объектов. Каждый объект $s \in Z$ представляется сетью, называемой семантическим графом (СГ), который включает вершины двух типов: объектные и предикатные. Объектной вершине приписывается имя (объекта), имя базового класса объекта и вектор его признаков; предикатной вершине — имя отношения, выполняющегося между объектами (возможно, с отрицанием). Семантический граф, предназначенный для представления объекта $s \in Z$, распадается на иерархически упорядоченное множество двухуровневых р-подграфов, служащих для представления объекта s , его «частей» (р-потомков), частей его частей и т. д.; р-подграф содержит на первом уровне объектную вершину v , связанную отношением p (целое — часть) с объектными вершинами второго уровня, которые связаны между собой отношениями через предикатные вершины.

Обобщенный семантический граф (ОСГ) также представляется в виде набора обобщенных р-подграфов (ор-подграфов). Каждый ор-подграф \tilde{g} предназначен для представления множества $M = m(\tilde{g})$ объектов. Объектной вершине первого уровня ор-подграфа приписывается имя множества $M: M = m(\tilde{g})$, имя базового класса $T: M \subseteq T$ и совокупность μ_π ограничений на изменения значений признаков объектов из M . Объектная вершина v второго уровня взвешена именем множества M_v , которое может быть получено из множеств, представимых на ор-подграфах применением к ним конечного числа операций объединения и пересечения. Предикатным вершинам ор-подграфов приписаны выражения логики исчисления высказываний, в которых роль высказываний играют имена отношений между объектами. На л-предикатных вершинах ОСГ отображаются отношения, выполняющиеся на множестве значений признаков различных объектов.

Каждый ор-подграф \tilde{g} определяет множество $m(\tilde{g})$ объектов g , таких, что \tilde{g} изоморфно наложим на некоторый частичный подграф g и выполняются заданные условия наложимости вершин (в соответствии с трактовкой их весов).

Обобщенное представление искомого класса K ищется по обучающей выборке $\langle v^+, v^- \rangle$ в виде покрывающей совокупности ОСГ $\tilde{g}_i, i \in I_K$, таких, что

$$v^+ \subseteq \bigcup_{i \in I_K} m(\tilde{g}_i); \quad v^- \cap \left(\bigcup_{i \in I_K} m(\tilde{g}_i) \right) = \emptyset.$$

В [Вагин и др., 1982] обобщения \tilde{g} получаются применением к элементам $g \in v^+$ операторов обобщения, таких, например, как удаление вершины из графа, замена значения признака множеством значений, замена имени отношения Q на предикатной вершине выражением $L = Q \vee R$. В [Вагин и др., 1985] обобщения \tilde{g} формируются «от противного». За исходный принимается «наиболее общий» граф \tilde{g}_0 , покрывающий все элементы базового класса T . Все остальные ОСГ получаются в результате применения к \tilde{g}_0 конечного числа ограничивающих операторов, таких, например, как ввод предикатной вершины в ОСГ, ввод конъюнктивного члена в предикатную вершину, удаление дизъюнктивного члена из предикатной вершины, сужение допустимого множества изменения значений признаков, ввод отношения (л-предикатной вершины) на множестве значений признаков. В последнем случае, следуя [Бонгард, 1967], изначально выбирается достаточно богатый набор допустимых отношений. Процесс формирования обобщенных представлений классов ведется по уровням ОСГ (согласно иерархии р-подграфов), начиная с нижних, методом ветвей и границ.

В [Гитлина и др., 1981а, б] предложен алгоритм структурного обобщения, в котором обобщенное описание класса K также ищется в виде множества подграфов $\tilde{g}_i, i \in I_K$, в совокупности покрывающих элементы обучающей выборки. Подграф \tilde{g} покрывает граф g , если он изоморфен одному из частичных подграфов g в классическом смысле. Метод успешно применялся для распознавания структуры химических соединений.

В [Уинстон, 1978] описан алгоритм структурного обобщения для решения задач анализа сцен. Эта задача подробно рассматривается в [Дуда и др., 1976].

Особое место занимают задачи «алгоритмической структурной индукции». Целью является автоматический синтез алгоритмов и программ по конкретным (числовым) примерам их работы [Вьетманн, 1978; Бардзинь, 1982].

2.4. Дедуктивный вывод на знаниях

В. Н. Вагин

Основные определения

В дедуктивных моделях представления и обработки знаний решаемая проблема записывается в виде утверждений формальной системы, Цель — в виде утверждения, справедливость которого следует установить или опровергнуть на основании аксиом (общих законов) и правил вывода формальной системы. В качестве формальной системы используют исчисление предикатов первого порядка.

В соответствии с правилами, установленными в формальной системе, заключительному утверждению — теореме, полученной из начальной системы утверждений (аксиом, посылок), приписывается значение ИСТИНА, если каждой посылке, аксиоме также приписано значение ИСТИНА. Формула B является логическим следствием формул F_1, F_2, \dots, F_n тогда и только тогда, когда для любой интерпретации, в которой конъюнкция $F_1 \& F_2 \& \dots \& F_n$ истинна, формула B также истинна.

Доказательством теоремы называется поиск ответа на вопрос: следует ли логически формула B из заданного множества формул F_1, F_2, \dots, F_n , что равносильно доказательству общезначимости формулы $F_1 \& F_2 \& \dots \& F_n \rightarrow B$ или противоречивости (невыполнимости) формулы $F_1 \& F_2 \& \dots \& F_n \& \neg B$. Из практических соображений удобнее доказывать противоречивость формулы, причем процедура установления невыполнимости формулы $F_1 \& F_2 \& \dots \& F_n \& \neg B$ называется *процедурой опровержения*.

В логике было доказано, что не существует эффективной разрешающей процедуры для исчисления предикатов первого порядка, позволяющей узнать по данной формуле, является она теоремой или нет [Church, 1936]. Неразрешимость исчисления предикатов первого порядка не закрывает путь для автоматического доказательства теорем, если воспользоваться ограничением, связанным с априорным предположением об общезначимости формулы, которую следует доказать.

Рассматриваемые далее методы поиска доказательств подтверждают, что формула общезначима (т. е. является теоремой), если она на самом деле таковой является. Для необщезначимых формул алгоритмы доказательства теорем работают бесконечно долго. Принимая во внимание результат А. Черча и А. Тьюринга, это лучшее, что можно ожидать от процедур доказательства теорем.

Краткая история

Стремление найти общую разрешающую процедуру доказательства теорем наблюдается у Лейбница, Пеано, в школе Гильберта. Автоматическое доказательство теорем берет начало от работы [Herbrand, 1930]. Процедура вывода Эрбрана основывается на его теореме, которая гласит: множество дизъюнктов (т. е. букв или их отрицаний, соединенных знаком \vee) невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество фундаментальных примеров (т. е. константных случаев) дизъюнктов в S . Таким образом, для установления невыполнимости (противоречивости) множе-

ва дизъюнктов необходимо образовывать множества $S_1, S_2, \dots, S_m, \dots$ константных случаев дизъюнктов и последовательно проверять их на ложность. Теорема Эрбрана гарантирует, что если исходное множество дизъюнктов S невыполнимо, то данная процедура обнаружит такое m , что S_m ложно.

С помощью эрбрановской процедуры вывода, реализованной на ЭВМ, был доказан ряд простых теорем их логики высказываний, но программа столкнулась с непреодолимыми трудностями при доказательстве теорем логики предикатов [Giltmore, 1960]. Основным недостатком процедуры Эрбрана состоит в экспоненциальном росте конечных множеств фундаментальных примеров дизъюнктов S_i при увеличении i .

Ненамного эффективнее был метод Девиса и Патнема, с помощью которого были доказаны некоторые теоремы из логики предикатов первого порядка, оказавшиеся недоказуемыми у Гилмора [Davis et al., 1960].

Наконец, в 1964—1965 гг. были созданы обратный метод вывода С. Маслова [Маслов, 1964] и принцип резолюций Дж. Робинсона [Robinson, 1965].

Обратный метод вывода

В этом методе поиск вывода идет от целевой формулы к аксиомам или постулатам, истинность которых априорно известна. Чтобы определить выводимость формулы B из посылок F_1, F_2, \dots, F_n , необходимо найти формулы-предшественники, из которых B может быть выведена одним применением правила вывода. Затем по каждой из получившихся формул-предшественников, не являющейся аксиомой исчисления, определяется множество непосредственных формул-предшественников и т. д. вплоть до построения окончательного вывода. Дерево, возникающее в ходе такого процесса, называется деревом поиска вывода, оно превратится в *дерево вывода* в тот момент, когда все его «листья» окажутся аксиомами исчисления.

Поясним работу обратного метода на следующем примере. Имеем две схемы вывода:

$$1. F_1(a, x_i) \& F_2(x_i, a) \rightarrow B_1(a, x_i), i = \overline{1,5}.$$

$$2. F_3(y_j^i) \& F_4(z_j^i) \& B_1(a, x_i) \rightarrow B_2(a, u_j), j = \overline{1,3}.$$

Априорно истинными считаются следующие предикаты: $F_1(a, x_1), F_1(a, x_2), F_1(a, x_4), F_2(x_3, a), F_2(x_4, a), B_1(a, x_1), F_3(y_1^1), F_3(y_2^1), F_3(y_1^4), F_3(y_1^5), F_3(y_2^2), F_3(y_3^4), F_4(z_2^1), F_4(z_1^3), F_4(z_3^4)$ (начальная ситуация). Необходимо получить истинное значение для предиката $B_2(a, u_3)$.

Обратный вывод идет от нужного результата. Предположим, что $B_2(a, u_3)$ является истинным, и определим, нет ли среди априорно истинных предикатов предиката $B_2(a, u_3)$. Если он находится в этом списке, то вывод заканчивается. В примере это не так. Тогда просматриваются все схемы вывода и в правых частях схем отыскивается предикат вида $B_2(a, u_j)$. В примере такая схема одна. Заменяем в ней всюду j на 3 и анализируем истинность выражения $F_3(y_3^i) \& F_4(z_3^i) \& B_1(a, x_i)$. Так как это конъюнкция, то истинными должны быть все три предиката, входящих в нее. Но их истинность зависит еще от значения i . Положим $i=1$. Проверим, используя описание начальной ситуации, истинность предикатов $F_3(y_3^1), F_4(z_3^1), B_1(a, x_1)$. Видим, что $F_3(y_3^1)$ не является истинным, поэтому берем $i=2$. По той же причине это значение i отвергается. При $i=3$ наблюдается аналогичная картина. При $i=4$ предикат $F_3(y_3^4)$ является истинным. Предикат $F_4(z_3^4)$ также истинен. Предикат $B_1(a, x_4)$ ложен. Определим условия, при которых $B_1(a, x_4)$ истинно. Для этого просматриваются правые части схем вывода и находятся те схемы, в правых частях которых стоит предикат $B_1(a, x_i)$. В примере такая схема одна. Полагаем в ней $i=4$ и анализируем левую часть схемы: $F_1(a, x_4) \& F_2(x_4, a)$. Из описания начальной ситуации видно, что данное выражение является истинным. Вывод окончен.

Если бы на некотором шаге поиска вывода было получено несколько схем вывода с нужными правыми частями, то были бы альтернативные варианты и

возникла бы задача выбора из них более приемлемого. Если бы на некотором шаге не нашлось схем вывода, в которых были бы нужные правые части, или в левых частях всюду были бы ложные предикаты, для которых нет средств сделать их истинными, то это свидетельствовало бы о недостижимости поставленной цели.

На основе обратного метода вывода были разработаны машинный алгоритм вывода [Давыдов и др., 1969] и несколько версий алгоритма машинного поиска естественного логического вывода в исчислении высказываний (АЛПЕВ) [Шанин и др., 1965]. Разработка версий систем АЛПЕВ — ЛОМИ и алгоритма вывода на основе обратного метода носила поисковый характер и показала принципиальную возможность машинных доказательств теорем.

Принцип резолюций

Метод резолюции (см. § 1.2) требует, чтобы исходная логическая формула была приведена к специальному виду, называемому *преנקской нормальной формой* (ПНФ), имеющей вид $\bigwedge_1 x_1 \bigwedge_2 x_2 \dots \bigwedge_n x_n M$, где $\bigwedge_i \in \{\forall, \exists\}$ ($1 \leq i \leq n$). M — бескванторная формула (матрица), а префикс есть последовательность кванторов. В свою очередь, M представляется в виде конъюнктивной нормальной формы (КНФ) $F_1 \& F_2 \& \dots \& F_n$, где каждая из F_1, F_2, \dots, F_n есть дизъюнкция литер (т. е. букв или их отрицаний). Сохраняя противоречивость (общезначимость) формулы, в ней можно устранить кванторы существования, используя сколемовские функции. В итоге получим стандартную форму.

Например, для определения стандартной формы формулы

$$\forall x (\exists y (P(x, y) \vee \neg Q(y, z)) \rightarrow \exists z R(x, y, z)).$$

преобразуем ее в ПНФ:

$$\begin{aligned} \forall x (\neg \exists y (P(x, y) \vee \neg Q(y, z)) \vee \exists z R(x, y, z)) &= \\ = \forall x (\forall y (\neg P(x, y) \& Q(y, z)) \vee \exists z R(x, y, z)) &= \\ = \forall x \forall y \exists z (\neg P(x, y) \& Q(y, z) \vee R(x, y, z)). \end{aligned}$$

Матрицу сведем к КНФ:

$$\forall x \forall y \exists z ((\neg P(x, y) \vee R(x, y, f(x, y))) \& (Q(y, f(x, y)) \vee R(x, y, f(x, y)))).$$

Устраним $\exists z$ введением сколемовской функции $z = f(x, y)$. Имеем следующую стандартную форму:

$$\forall x \forall y ((\neg P(x, y) \vee R(x, y, f(x, y))) \& (Q(y, f(x, y)) \vee R(x, y, f(x, y)))).$$

В дальнейшем кванторы общности будем опускать, предполагая, что все переменные, ими связанные, являются универсально квантифицированными.

В итоге любая формула может быть представлена множеством дизъюнктов, т. е. дизъюнкцией литер. В примере множество дизъюнктов $S = \{\neg P(x, y) \vee R(x, y, f(x, y)), Q(y, f(x, y)) \vee R(x, y, f(x, y))\}$. Здесь запятая между дизъюнктами заменяет знак $\&$. Более подробное изложение сведения любой формулы к множеству дизъюнктов можно найти в [Чень и др., 1983].

Основная идея принципа резолюции заключается в проверке, содержит ли множество дизъюнктов S пустой дизъюнкт (см. § 1.2). Когда дизъюнкт не содержит никаких литер, то он называется пустым. Так как пустой дизъюнкт не содержит литер, которые могли бы быть истинными при любых интерпретациях, то он всегда ложен. Обозначим его через \square . Если S содержит \square , то S противоречиво (невыполнимо). Если S не содержит \square , то следующие шаги заключаются в выводе новых дизъюнктов до тех пор, пока не будет получен \square (что всегда будет иметь место для невыполнимого S). Таким образом, принцип резолюций рассматривается как правило вывода, с помощью которого порождаются новые дизъюнкты из S .

По существу принцип резолюций является расширением однолитерного правила Девиса и Патнема, идея которого заключается в следующем. В двух дизъюнктах, один из которых состоит из одной литеры, а второй содержит произвольное число литер (в том числе и одну), находится *контрарная пара литер* (например, P и $\neg P$), которая вычеркивается, и из оставшихся частей дизъюнктов формируется новый дизъюнкт (например, Q из P и $\neg P \vee Q$). Отметим, что ничего нового по сравнению с известным правилом *modus ponens* здесь нет, так как, например, $\neg P \vee Q$ равносильно $P \rightarrow Q$ и из выводимости P и $P \rightarrow Q$ следует выводимость Q .

Дж. Робинсон расширил однолитерное правило Девиса и Патнема на случай произвольных дизъюнктов с любым числом литер. Если в любых двух дизъюнктах C_1 и C_2 имеется контрарная пара литер, то при вычеркивании ее формируется новый дизъюнкт из оставшихся частей дизъюнктов. Этот вновь сформированный дизъюнкт называется *резольвентой дизъюнктов* C_1 и C_2 . Например, пусть

$$C_1: P \vee \neg Q \vee \neg R,$$

$$C_2: Q \vee P.$$

Тогда резольвента

$$C: P \vee \neg P.$$

Обоснованность получения таким образом резольвент вытекает из теоремы, гласящей, что резольвента C , полученная из двух дизъюнктов C_1 и C_2 , является логическим следствием этих дизъюнктов.

Если в процессе вывода новых дизъюнктов получены два однолитерных дизъюнкта, образующих контрарную пару, то резольвентой этих двух дизъюнктов будет пустой дизъюнкт \square . Таким образом, выводом пустого дизъюнкта \square из множества дизъюнктов S называется конечная последовательность дизъюнктов C_1, C_2, \dots, C_k , такая, что любой C_i ($1 \leq i \leq k$) является либо дизъюнктом из S , либо резольвентой, полученной принципом резолюций, и $C_k = \square$.

В логике предикатов первого порядка нахождение контрарных пар затруднено. Действительно, если имеются дизъюнкты типа

$$C_1: P(x) \vee \neg R(x),$$

$$C_2: \neg P(f(x)) \vee Q(y),$$

то резольвента может быть получена только после применения к C_1 подстановки $f(x)$ вместо x . Имеем

$$C_1': P(f(x)) \vee \neg R(f(x)),$$

$$C_2: \neg P(f(x)) \vee Q(y),$$

резольвента

$$C: \neg R(f(x)) \vee Q(y).$$

Отсюда подстановка θ называется *унификатором* для множества выражений W_1, W_2, \dots, W_k тогда и только тогда, когда $W_1\theta = W_2\theta = \dots = W_k\theta$ (см. § 1.2).

Унификатор σ для множества выражений W_1, W_2, \dots, W_k называется *наиболее общим унификатором (НОУ)* тогда и только тогда, когда для каждого унификатора θ для этого множества существует такая подстановка λ , что $\theta = \sigma \circ \lambda$, где $\sigma \circ \lambda$ — композиция подстановок. В свою очередь композиция подстановок $\sigma = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$ и $\lambda = \{u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$ есть подстановка вида $\{t_1\lambda/x_1, t_2\lambda/x_2, \dots, t_n\lambda/x_n, u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$, из которой вычеркиваются все $t_j\lambda/x_j$, если $t_j\lambda = x_j$, и все u_i/y_i , такие, что $y_i \in \{x_1, x_2, \dots, x_n\}$. В примере унификатор $\{f(x)/x\}$ являлся НОУ.

Дж. Робинсон предложил алгоритм унификации, который находил НОУ, если множество выражений W_1, W_2, \dots, W_k было унифицируемо, и сообщал о неудаче, если это не так.

Если две или более литеры (с одинаковым знаком) дизъюнкта C имеют НОУ σ , $C\sigma$ называется *фактор-дизъюнктом*. Например, пусть $C = P(x) \vee P(f(y)) \vee Q(x)$. Тогда $\sigma = \{f(y)/x\}$ и $C\sigma = P(f(y)) \vee Q(f(y))$ есть фактор-дизъюнкт.

В логике предикатов первого порядка резольвента для исходных дизъюнктов C_1 и C_2 может быть получена одним из четырех способов: 1) из C_1 и C_2 ; 2) из C_1 и $C_2\sigma$; 3) из $C_1\sigma$ и C_2 ; 4) из $C_1\sigma$ и $C_2\sigma$.

Принцип резолюций обладает важным свойством — полнотой, теорема о которой гласит: множество дизъюнктов S невыполнимо тогда и только тогда, когда существует вывод из S пустого дизъюнкта. В силу неразрешимости логики предикатов первого порядка для выполнимого множества дизъюнктов S процедура, основанная на принципе резолюций, будет работать бесконечно долго.

Так, имеем две посылки и заключение: некоторые студенты любят всех преподавателей, ни один из студентов не любит невежд, следовательно, ни один из преподавателей не является невеждой. Докажем это заключение из двух посылок принципом резолюций. Для этого введем следующие предикаты: $C(x) — x — студент$, $P(x) — x — преподаватель$, $H(x) — x — невежда$ и $L(x, y) — x$ любит y . Тогда две посылки на языке логики предикатов первого порядка будут иметь вид

$$\exists x(C(x) \& \forall y(P(y) \rightarrow L(x, y))); \\ \forall x(C(x) \rightarrow \forall y(H(y) \rightarrow \neg L(x, y))).$$

Заключение будет выражено как $\forall y(P(y) \rightarrow \neg H(y))$.

На языке дизъюнктов посылки и заключение (его необходимо взять с отрицанием) будут иметь вид

- 1) $C(a)$;
- 2) $\neg P(y) \vee L(a, y)$;
- 3) $\neg C(x) \vee \neg H(y) \vee \neg L(x, y)$;
- 4) $P(b)$;
- 5) $H(b)$.

Докажем, что это множество дизъюнктов невыполнимо. Имеем

- | | | |
|----------------------------------|--------------|----------------------|
| 6) $\neg H(y) \vee \neg L(a, y)$ | из 1) и 3); | $\sigma = \{a/x\}$, |
| 7) $L(a, b)$ | из 2) и 4); | $\sigma = \{b/y\}$, |
| 8) $\neg H(b)$ | из 6) и 7); | $\sigma = \{b/y\}$, |
| 9) \square | из (5) и 8). | |

Указанный вывод представляется деревом вывода (рис. 2.2).

Принцип резолюций эффективнее процедуры Эрбрана. Но и он страдает существенным недостатком, заключающимся в формировании всевозможных резольвент, большинство из которых оказываются ненужными. Многочисленные модификации принципа резолюций направлены на нахождение более эффективных стратегий поиска нужных дизъюнктов. Остановимся на тех, которые нашли применение в моделях обработки знаний, в основе которых лежит исчисление предикатов первого порядка.

Отметим, что основными способами устранения причин «экспоненциального взрыва», имеющего место при доказательстве теорем практической степени сложности, являются использование семантики и встраивание в правила вывода и алгоритмы унификации спецификации конкретной области применения.

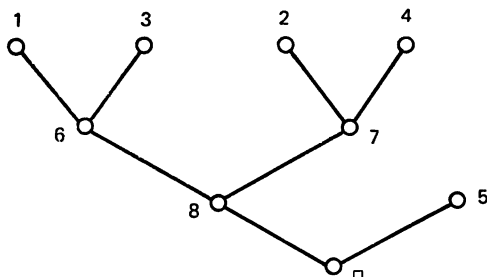


Рис. 2.2

За некоторым исключением, которое специально оговаривается, будем удалять из рассмотрения порождаемые дизъюнкты типа тавтологий, так как невыполнимое множество дизъюнктов, содержащее дизъюнкт-тавтологию, останется снова невыполнимым после его удаления. Аналогично будем вычеркивать *поглощенные дизъюнкты*, которые также не меняют невыполнимости множества дизъюнктов. Дизъюнкт C_1 поглощает дизъюнкт C_2 тогда и только тогда, когда

да имеется подстановка δ , такая, что $C_1\delta \subseteq C_2$. Дизъюнкт C_2 называется поглощенным дизъюнктом. Например, пусть $C_1 = P(x)$ и $C_2 = P(a) \vee R(y)$. Так как $\delta = \{a/x\}$, то $C_1\delta = P(a)$. Тогда $C_1\delta \subseteq C_2$ и C_1 поглощает C_2 , т. е. C_2 можно без ущерба для вывода удалить из множества дизъюнктов.

Семантическая резолюция

Одной из эффективных модификаций принципа резолюций является семантическая резолюция [Slagle, 1967]. В ней используется интерпретация для разделения множества дизъюнктов S на два класса: S_1 — непустое множество дизъюнктов, которое выполняется (т. е. принимает значение $и$) этой интерпретацией, S_2 — непустое множество дизъюнктов, которое не выполняется (т. е. принимает значение $л$) этой интерпретацией; $S_1 \cup S_2 = S$. Разрешается резольвирование дизъюнктов, принадлежащих только разным множествам, и запрещается образование резольвент от дизъюнктов, входящих в одно и то же множество. Тем самым сокращается образование ненужных дизъюнктов, так как только резольвированием из разных множеств можно получить пустой дизъюнкт.

Другим способом ограничения количества порождаемых дизъюнктов является упорядочение предикатных букв. Если имеется упорядочение предикатных букв типа $P_1 > P_2 > \dots > P_k$, то разрешается резольвирование литеры, обладающей наибольшим порядком, т. е. P_1 .

Определим формально семантическую резолюцию. Пусть I — интерпретация и P — упорядочение предикатных букв. Конечное множество дизъюнктов $\{E_1, E_2, \dots, E_r, N\}$, $r \geq 1$, называется *семантическим клашем* (clash) относительно P и I (PI -клашем) тогда и только тогда, когда E_1, E_2, \dots, E_r (называемые электронами) и N (называемое ядром) удовлетворяют следующим условиям: 1) E_1, E_2, \dots, E_r ложны в интерпретации I ; 2) при $R_i = N$ для каждого $i = 1, 2, \dots, r$ существует резольвента R_{i+1} , образованная из R_i и E_i ; 3) резольвируемая литера в E_i является наибольшей предикатной буквой в E_i , $i = 1, 2, \dots, r$; 4) R_{r+1} ложна в интерпретации I ; R_{r+1} называется PI -резольвентой, полученной из PI -клаша $\{E_1, E_2, \dots, E_r, N\}$.

Вывод из множества дизъюнктов S называется PI -выводом тогда и только тогда, когда каждый дизъюнкт в выводе является или дизъюнктом из S , или PI -резольвентой.

Проиллюстрируем семантическую резолюцию на рис. 2.2, задав интерпретацию и упорядочение предикатных букв следующим образом:

$$I = \{ \neg C(a), \neg P(b), \neg H(b), L(ab) \},$$

что значит $C(a)$, $P(b)$, $H(b)$ ложны в этой интерпретации, а $L(a, b)$ истинен и $P > C > H > L$. Дерево вывода показано на рис. 2.3. Здесь $\{E_1, E_2, N\}$ удовлетворяет всем четырем условиям, следовательно, является PI -клашем, и

$\neg L(ab)$ — PI -резольвента. Аналогично $\{E_3, E_4, N\}$ является PI -клашем и \square — PI -резольвентой.

В семантической резолюции можно использовать любую интерпретацию и любое упорядочение предикатных букв. Семантическая резолюция полна. Теорема о полноте семантической резолюции гласит: если P — упорядочение предикатных букв в конечном невыполнимом множестве дизъюнктов S , а I — интерпретация множества дизъюнктов S , то существует PI -вывод пустого дизъюнкта из S .

Специальными случаями семантической резолюции являются положительная и отрицательная гиперрезолюции и стратегия множества поддержки.

Дизъюнкт называется *положительным*, если он не содержит знаков отрицания, и *отрицательным*, если каждая его литера содержит знак отрицания.

Положительной гиперрезолюцией называется специальный случай PI -резолюции, в которой каждая литера в интерпретации I содержит знак отрицания. Положительной она называется в силу того, что все электроны и PI -резольвенты здесь положительны. Аналогично определяется *отрицательная гиперрезолюция*, у которой каждая литера в интерпретации I не содержит знака отрицания.

Стратегия множества поддержки была предложена в [Wos et al., 1965]. Для доказательства того, что формула B логически следует из формул F_1, F_2, \dots, F_n , доказывалась невыполнимость формулы $F_1 \& F_2 \& \dots \& F_n \& \neg B$. Так как посылки $F_1 \& F_2 \& \dots \& F_n$ всегда выполнимы (истинны), то естественно запретить резольвирование литер в $F_1 \& F_2 \& \dots \& F_n$. Пусть S — невыполнимое множество дизъюнктов и S_1 — подмножество из S , такое, что $S - S_1$ выполнимо. Тогда подмножество S_1 называется множеством поддержки. Резолюция с множеством поддержки — это применение принципа резолюции к паре дизъюнктов, не принадлежащих одновременно $S - S_1$.

Все специальные случаи семантической резолюции полны.

Лок-резолюция

Идея лок-резолюции [Boyer, 1971] состоит в использовании индексов для упорядочения литер в дизъюнктах из множества дизъюнктов S . Иными словами, она включает индексацию каждого вхождения литеры с S некоторым целым числом; разные вхождения одной и той же литеры в S могут быть индексированы по-разному. После этого резольвировать разрешается только литеры с наименьшим индексом в каждом из дизъюнктов. Литеры в резольвентах наследуют свои индексы из посылок. Если литера в резольvente может унаследовать более одного индекса, то ей ставится в соответствие наименьший индекс.

Например, рассмотрим следующие два дизъюнкта: 1) ${}_1P \vee {}_2R$, 2) ${}_3\neg P \vee {}_4R$. Целое число, помещенное под литерой слева, — это индекс, поставленный в соответствие этой литере. Так как индекс ${}_1P$ меньше индекса ${}_2R$, то разрешается

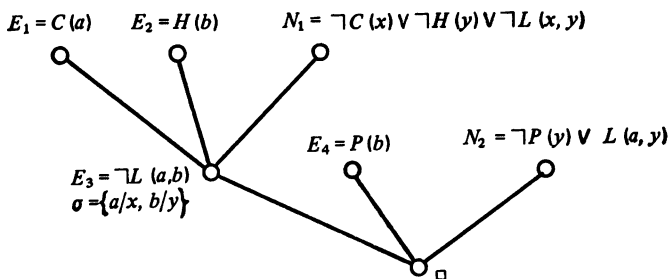


Рис. 2.3

резольвировать ${}_1P$. Аналогично, так как индекс ${}_3\overline{P}$ меньше индекса ${}_4R$, то можно резольвировать ${}_3\overline{P}$. Таким образом, резольвируя дизъюнкты 1) и 2) по ${}_1P$ и ${}_3\overline{P}$, получаем дизъюнкт 3) ${}_2R \vee {}_4R$. Так как одна и та же литера R наследует два индекса (2 и 4), то ей ставится в соответствие наименьший индекс, т. е. 4) ${}_2R$. Дизъюнкт ${}_2R$ называется лок-резольвентой исходных дизъюнктов 1) и 2).

Отметим, что если бы литеры в дизъюнкте 2) были индексированы иначе, а именно ${}_2\overline{P} \vee {}_4R$, то резольвируемой литерой в этом дизъюнкте была бы ${}_3R$. Однако она не резольвируется с литерой из дизъюнкта 1). Поэтому в этом случае не существует лок-резольвенты для дизъюнктов 1) и 2').

Лок-резольюция полна.

Линейная резолюция

Линейная резолюция была независимо предложена в [Loveland, 1968a, б] и [Luskham, 1968]. *Линейным выводом* из множества дизъюнктов S называется последовательность дизъюнктов (C_1, C_2, \dots, C_n) , в которой верхний дизъюнкт $C_i \in S$, а каждый член C_{i+1} , $i=1, 2, \dots, n-1$, является резольвентой дизъюнкта C_i (называемого центральным) и дизъюнкта B_j (называемого боковым), который удовлетворяет одному из двух условий: 1) $B_i \in S$, $i=1, 2, \dots, n-1$; 2) B_j является некоторым дизъюнктом C_j , предшествующим в выводе дизъюнкта C_i , т. е. $j < i$. Линейный вывод имеет вид дерева (рис. 2.4).

Линейная резолюция полна: множество дизъюнктов S невыполнимо тогда и только тогда, когда существует линейный вывод пустого дизъюнкта.

Линейная резолюция может быть усилена введением понятия упорядоченного дизъюнкта и использованием информации о резольвированных литерах. *Упорядоченным дизъюнктом* называется дизъюнкт с определенной последовательностью литер. Говорят, что литера L_2 старше литеры L_1 в упорядоченном дизъюнкте тогда и только тогда, когда L_2 следует за L_1 в последовательности, определенной упорядоченным дизъюнктом. Старшая (наибольшая) литера дизъюнкта является последней литерой дизъюнкта, а младшая литера — первой.

Если две или более литер (с одинаковыми знаками) упорядоченного дизъюнкта S имеют НОУ σ , то упорядоченный дизъюнкт, полученный из последовательности $S\sigma$ вычеркиванием любой литеры, идентичной младшей литере, называется *упорядоченным фактором дизъюнкта S* . Например, для упорядоченного дизъюнкта $S = P(x) \vee Q(x) \vee P(a)$ имеем $\sigma = \{a/x\}$ и $S\sigma = P(a) \vee Q(a) \vee P(a)$. Вычеркивая последнюю литеру, получаем упорядоченный фактор дизъюнкта $P(a) \vee Q(a)$.

Другим усилением линейной резолюции является использование информации о резольвированных литерах. Обычно при выполнении резолюции образование резольвент происходит путем удаления резольвированных литер, однако эти литеры несут полезную информацию, которая может быть использована для усиления линейной резолюции. Например, если $C_1 = P(x) \vee Q(x) \vee \overline{P}(a)$ и $C_2 = \overline{P}(a) \vee Q(a)$ — упорядоченные дизъюнкты, то конкатенация $C_1(\sigma)$ и $C_2(\sigma)$, где $\sigma = \{a/x\}$, дает последовательность $P(a) \vee Q(a) \vee \overline{P}(a) \vee \overline{P}(a) \vee Q(a)$.

Теперь вместо удаления $P(a)$ и $\overline{P}(a)$ будем оставлять в резольвенте первую из них, пометить ее (двумя вертикальными чертами) и называть обрамленной литерой. Если за обрамленной литерой не следует никакая другая литера, то ее будем вычеркивать. Продолжая пример, получаем упорядоченную резольвенту:

$$|P(a)| \vee Q(a) \vee \overline{P}(a).$$

Пусть $S = \{P \vee Q, \overline{P} \vee Q, P \vee \overline{Q}, \overline{P} \vee \overline{Q}\}$. Линейный вывод, использующий информацию о резольвированных литерах и понятие упорядоченного дизъюнкта, изображен на рис. 2.5. Видно, что из четырех боковых дизъюнктов три принадлежат S и один P является центральным. Рассмотрим

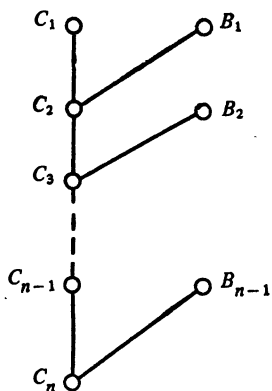


Рис. 2.4

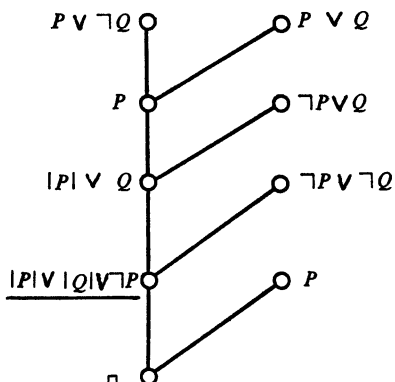


Рис. 2.5

подчеркнутую резольвенту, в которую входит литера $\neg P$, являющаяся дополнением к обрамленной литере $|P|$. Данное обстоятельство указывает на необходимость использования центрального дизъюнкта P в качестве бокового. Резольвируя $|P| \vee |Q| \vee \neg P$ с P , получаем $|P| \vee |Q| \vee \neg P$. Так как за этими обрамленными литерами не следует никакой другой литеры, то, вычеркивая их, получаем пустой дизъюнкт \square .

Упорядоченный дизъюнкт S называется *приведенным упорядоченным дизъюнктом* тогда и только тогда, когда последняя литера в S унифицируется с отрицанием некоторой обрамленной литеры из S . При получении приведенного упорядоченного дизъюнкта нет необходимости искать, с каким из полученных ранее дизъюнктов он образует линейную резольвцию, можно вычеркнуть последнюю литеру в этом упорядоченном дизъюнкте.

Алгоритм линейного вывода, применяющий как понятие упорядоченного дизъюнкта, так и информацию о резольвируемых литерях, называется *OL-выводом* (Ordered Linear Deduction). *OL-вывод* — это по существу то же самое, что в [Loveland, 1968] называется методом элиминации моделей. *OL-вывод* является разновидностью линейной резольвции с функцией выбора (*SL-вывод*) [Kowalski, 1971]. *OL-вывод* полон.

Возвращаясь к рис. 2.5, видим, что невозможно построить линейный вывод пустого дизъюнкта, если в качестве боковых дизъюнктов брать только дизъюнкты из исходного множества S . Назовем дизъюнкты множества S *входными дизъюнктами*. Тогда резольвция, у которой хотя бы один из двух дизъюнктов является входным, называется *входной резольвцией*. Входная резольвция проста и эффективна, но, как видно из рис. 2.5, не полна. Однако существует специальный класс так называемых *хорновских дизъюнктов* (см. § 1.2), для которых входная резольвция полна. Дизъюнкт называется хорновским, если он содержит не более одной положительной литеры. Входная резольвция является логической основой для языка логического программирования Пролог [Clocksin et al., 1981].

Резольвция с абстракцией

Методы доказательства теорем, основанные на абстракции, были предложены в [Plaisted, 1981], где использовалась идея аналогии, заключающаяся в том, что данная проблема A преобразуется в более простую проблему B . Далее, если A имеет решения, то и B имеет решения, причем структура одного из решений для B аналогична структуре решений для A . Поэтому решения для

В можно использовать как руководство в поиске решений для A . Конечно, может случиться, что B имеет решения, даже если A их не имеет.

Как и раньше, полагаем, что дизъюнкт C_1 поглощает дизъюнкт C_2 , если имеется подстановка θ , такая, что $C_1\theta \in C_2$ ($C_1\theta$ называется *примером дизъюнкта* C_1 ; если θ — константная подстановка, то $C_1\theta$ называется фундаментальным примером дизъюнкта C_1).

Абстракцией называется соотнесение множества дизъюнктов $f(C)$ с любым дизъюнктом C , таким, что f обладает следующими свойствами: 1) если дизъюнкт C_3 является резольвентой дизъюнктов C_1 и C_2 и $D_3 \in f(C_3)$, то существуют $D_1 \in f(C_1)$ и $D_2 \in f(C_2)$, такие, что некоторая резольвента дизъюнктов D_1 и D_2 поглощает D_3 ; 2) $f(\Box) = \{\Box\}$; 3) если C_1 поглощает C_2 , то для любой абстракции D_2 дизъюнкта C_2 имеется абстракция D_1 дизъюнкта C_1 , такая, что D_1 поглощает D_2 . Если f является отображением, обладающим этими свойствами, то оно называется отображением абстракций. Если $D \in f(C)$, то D называется абстракцией дизъюнкта C . Абстракции обычно удовлетворяют свойству, что любой D в $f(C)$ является тавтологией, если дизъюнкт C — тавтология.

Для конструирования абстракций в [Plaisted, 1981] доказана следующая теорема. Пусть F — множество отображений литер на литеры. Предположим, что для всех $\Phi \in F$ и для всех литер L имеет место $\Phi(\bigwedge L) = \bigwedge \Phi(L)$. Если C — дизъюнкт, то $\Phi(C) = \{\Phi(L) : L \in C\}$. Предположим также, что если дизъюнкт D является примером дизъюнкта C , то для всех $\Phi_2 \in F$ существует $\Phi_1 \in F$, такое, что $\Phi_2(D)$ является примером $\Phi_1(C)$. Определим f через $f(C) = \{\Phi(C) : \Phi \in F\}$. Тогда f является отображением абстракций.

Имеются *синтаксические* и *семантические абстракции*. Примерами синтаксических абстракций служат фундаментальная абстракция, пропозициональная абстракция, абстракции, связанные с переименованием предикатных и функциональных букв, с изменением знака литер, с перестановкой и вычеркиванием некоторых аргументов у предикатов или функций. Если C — дизъюнкт, то $f(C) = \{C' : C' \text{ является фундаментальным примером для } C\}$ есть фундаментальная абстракция.

Пропозициональная абстракция определяется следующим образом. Пусть C — дизъюнкт вида $L_1 \vee L_2 \vee \dots \vee L_k$. Тогда $f(C)$ есть $\{C'\}$, где C' — дизъюнкт вида $L_1' \vee L_2' \vee \dots \vee L_k'$ и $L_i' (1 \leq i \leq k)$ есть P , если L_i есть $P(t_1, t_2, \dots, t_n)$, и $\neg P$, если L_i есть $\neg P(t_1, t_2, \dots, t_n)$, т. е. получаем дизъюнкт для исчисления высказываний. Специальным случаем этой абстракции является абстракция, связанная с вычеркиванием переменных (необязательно всех).

Если в дизъюнктах переименовать некоторым систематическим образом предикатные и функциональные буквы, то получится абстракция с переименованием.

Аналогично, меняя знаки литер или переставляя аргументы у функциональных и предикатных букв, имеем соответствующий тип синтаксической абстракции.

Семантическая абстракция связана с интерпретацией множества дизъюнктов, в котором означиваются все переменные функций и предикатов.

Покажем на примерах, как используются абстракции при доказательстве. Рассмотрим доказательство $R(a)$ на дереве вывода (рис. 2.6,а) и с помощью пропозициональной абстракции (рис. 2.6,б). Однако, применяя эту же пропозициональную абстракцию к доказательству другого дизъюнкта, получаем доказательство, неадекватное исходному, как это будет видно из следующего примера.

Для дерева вывода (рис. 2.7,а) применяем пропозициональную абстракцию. Получаем дерево вывода на рис. 2.7,б. Здесь имеет место потеря литеры $\neg P(b)$. Во избежание таких потерь используем понятие *мультидизъюнкта*, в котором одна и та же литера может встречаться более одного раза, т. е., например, для дизъюнкта $\neg P(a) \vee \neg P(b) \vee Q(c)$ мультидизъюнкт, полученный пропозициональной абстракцией, будет иметь вид $\neg P \vee \neg P \vee Q$. Тогда, продолжая пример и используя понятие мультидизъюнкта, нетрудно получить вывод $Q \vee R$.

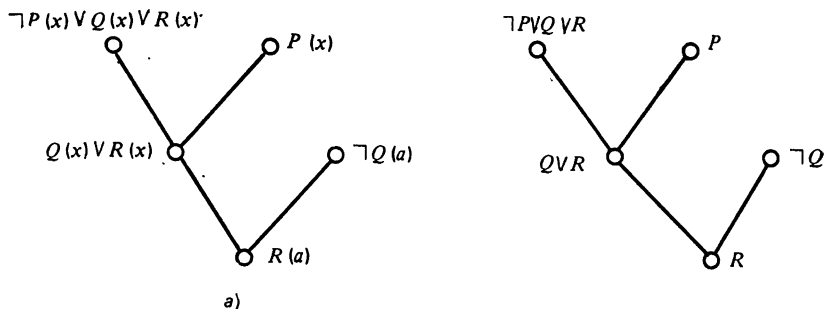


Рис. 2.6

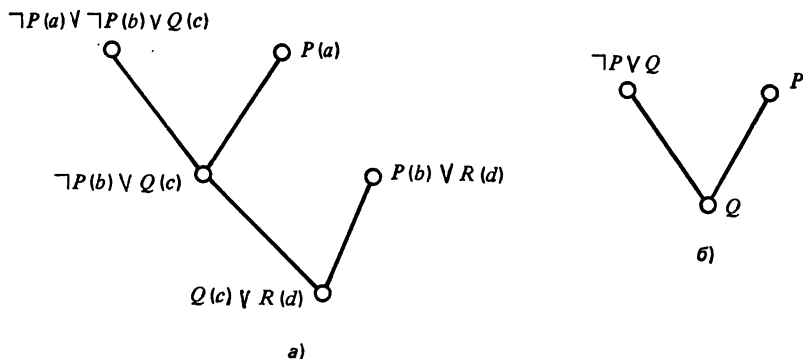


Рис. 2.7

Таким образом, доказательство с абстракциями дает как бы путь для доказательства теорем, идя по которому мы восстанавливаем исходные дизъюнкты и, резольвируя их, получаем доказательство теоремы. Естественно, что не все пути, построенные таким образом, будут приводить к нужному результату.

В [Plaisted, 1981] было разработано несколько вариантов полных алгоритмов доказательства теорем, основанных на абстракциях.

Вывод для формул, представленных в матричном виде

Методы дедуктивного вывода для формул, представленных в матричном виде, были предложены в [Andrews, 1981]. Особенностью этих методов вывода является преобразование формулы в *негативную нормальную форму* (ННФ), которая затем представляется в матричном виде. Будем говорить, что формула F находится в ННФ тогда и только тогда, когда каждое вхождение знака отрицания в F атомарно, т. е. отрицание стоит перед атомом.

Так как при доказательстве может потребоваться несколько примеров одной и той же подформулы, введем понятие *квантифицированного удвоения*. Формула R' получается из формулы R посредством квантифицированного удвоения тогда и только тогда, когда R' является результатом замены некоторой подформулы в R вида $\forall x M$ на $\forall x M \& \forall x M$. Все переменные, стоящие под знаком квантора, должны быть отмечены различными буквами, т. е. $\forall x M \& \forall y M$.

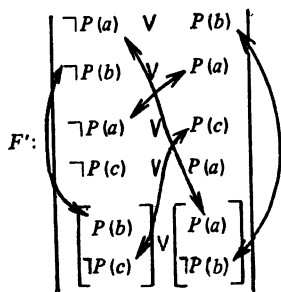


Рис. 2.8

Формулу F будем называть расширением исходной формулы R , если она получена из R последовательностью квантифицированных удвоений, переименованием различными буквами всех переменных, стоящих под знаками кванторов, и удалением кванторов из R (кванторы существования удаляются сколемизацией).

Бескванторная формула необязательно должна быть представлена в КНФ. Она может иметь произвольный вид, приведенный к базису $\neg, \vee, \&$.

Для удобства представим формулу в виде матрицы таким образом, что дизъюнкты расположены по горизонтали, а конъюнкты (т. е. наборы литер, соединенных конъюнкциями) — по вертикали. Например, формула

$$F = ((\neg P(a) \vee P(y)) \& (\neg P(y) \vee P(a))) \& ((\neg P(a) \vee P(w)) \& (\neg P(w) \vee P(a))) \& ((P(b) \& \neg P(c)) \vee (P(z) \& \neg P(x)))$$

в матричной форме имеет вид

$$F: \begin{vmatrix} \neg P(a) \vee P(y) \\ \neg P(y) \vee P(a) \\ \neg P(a) \vee P(w) \\ \neg P(w) \vee P(a) \\ [P(b)] \vee [\neg P(c)] \\ [\neg P(x)] \end{vmatrix}. \quad (1)$$

Сцепкой M для формулы F называется бинарное отношение на множестве входящих литер в F , такое, что имеется подстановка θ , такая, что $L_1\theta = \neg L_2\theta$ всякий раз, когда L_1ML_2 , где L_1 и L_2 — сцепленные литеры. Сцепка M является сцепкой опровержения для F тогда и только тогда, когда F ложна относительно любого приписывания истинностных значений атомам, т. е. когда F не имеет модели.

Для матрицы (1) найдем такие подстановки θ , которые делают формулу F невыполнимой (противоречивой). В итоге получим матрицу (для удобства сцепку изобразим в виде линий, соединяющих сцепленные литеры), показанную на рис. 2.8.

Вертикальный путь через матрицу — это последовательность литер формулы, которая содержит все литеры конъюнкции, если они встретились на этом пути, и одну литеру из дизъюнкции, если путь проходит через дизъюнкцию. Для матрицы (1) одним из таких вертикальных путей будет путь $\neg P(a), \neg P(b), \neg P(a), \neg P(c), P(b), \neg P(c)$.

Сцепка M является допустимой некоторым путем сцепкой (p -допустимой) тогда и только тогда, когда каждый вертикальный путь через матрицу содержит сцепленную пару литер.

Основная идея этого метода состоит в следующем: для доказательства невыполнимости универсально квантифицируемой формулы, представленной в виде матрицы, необходимо доказать, что каждый вертикальный путь содержит контрарные литеры и для них существует подстановка, унифицирующая соответствующие термы. Метод базируется на теореме П. Эндрюса. Пусть R — универсально квантифицированная формула, представленная в ННФ. Она не выполнима тогда и только тогда, когда некоторое расширение R имеет сцепку опровержения и, как следствие, p -допустимую сцепку.

Возвращаясь к матрице (1), видим, что для F имеется p -допустимая сцепка $(\neg P(y), P(b))$ для пути $\neg P(a), \neg P(y), \neg P(a), \neg P(w), P(b), \neg P(c)$; сцепка $(P(y),$

$\neg P(x)$) для пути $P(y), P(a), P(w), P(a), P(z), \neg P(x)$; сцепка $(\neg P(a), P(z))$ для пути $\neg P(a), P(a), P(w), P(a), P(z), \neg P(x)$ и т. д. — всего 32 пути. Отсюда очевидно, что F противоречива.

Поскольку число вертикальных путей через матрицу может быть значительным, то особое внимание при реализации этого метода должно уделяться эффективности проверки этих путей. В [Andrews, 1981] предлагаются некоторые пути решения этой проблемы. Так, нет необходимости рассматривать каждый вертикальный путь отдельно в полной мере. Как только найдена пара сцепленных литер на некоторой части пути, следует исключить из дальнейшего рассмотрения все расширения этой части.

Представление формул в виде матриц и методы доказательства теорем, основанные на таком представлении, были также рассмотрены в [Bibel, 1983; 1986].

Дедукция на семантических сетях

Одним из формализмов представления знаний являются семантические сети (см. § 1.3). Семантическая сеть в общем виде представляет собой информационную модель предметной области, включая факты и общие закономерности (аксиомы, посылки). В вершинах сети помещаются объекты или понятия, а дуги задают отношения между ними. С каждой семантической сетью сопоставлена совокупность дизъюнктов вида $B_1 \& B_2 \& \dots \& B_m \rightarrow A_1 \vee A_2 \vee \dots \vee A_n$ (или $\neg B_1 \vee \neg B_2 \vee \dots \vee \neg B_m \vee A_1 \vee \dots \vee A_n$), где B_i — условия, A_j — заключения дизъюнкта.

Дизъюнкты, имеющие как условия, так и заключения, представляют собой общие законы (аксиомы), что соответствует интенциональному представлению знаний. Дизъюнкты без условий — это факты, соответствующие экстенциональному представлению. И, наконец, дизъюнкты без заключений — это отрицания фактов или цели, которые нужно доказать. Как обычно, используем процедуру опровержения при доказательстве теорем.

В семантической сети присутствуют два вида вершин: предикатные, соответствующие предикатным буквам, входящим в дизъюнкты, и отвечающие дизъюнктам. Каждая дуга, идущая от дизъюнктовой вершины в предикатную, взвешена атомом из этого дизъюнкта. Предикатные вершины связаны между собой дугами только через дизъюнктные вершины.

Для уменьшения перебора при поиске нужной информации, а также более быстрого и эффективного извлечения информации из сети введем операцию «раскраски» сети [Вагин и др., 1984]. Правило «раскраски» следующее: в дизъюнкте $B \rightarrow A$ условие B будет «раскрашено» цветом Ц1 (на рис. 2.9, 2.10 соответствующие дуги изображены непрерывной линией), а заключение A — цветом Ц2 (соответствующие дуги — штриховой линией). Это правило распространяется на множество условий и заключений и на его частные случаи. Например, для дизъюнкта $g: B_1 \& B_2 \rightarrow A_1 \vee A_2$ имеем дизъюнктовую вершину g , четыре предикатные вершины B_1, B_2, A_1, A_2 и четыре дуги: две непрерывные, идущие от вершины g к B_1 и B_2 , и две штриховые, идущие от g к A_1 и A_2 .

Формально сеть задается четверкой вида $L = \langle G, P, F_1, F_2 \rangle$, где G — множество дизъюнктивных вершин, P — множество предикатных вершин, F_1 — отображение G в P на дугах, раскрашенных цветом Ц1, и F_2 — отображение G в P на дугах, раскрашенных цветом Ц2.

Задача дедуктивного вывода формулируется следующим образом. Задана логико-лингвистическая модель в виде семантической сети, с которой сопоставлен набор дизъюнктов указанного типа, и на ее вход подается ситуация, представленная совокупностью фактов A_1, A_2, \dots, A_r . Эта ситуация, накладываемая на семантическую сеть, представляет собой текущее состояние системы обработки знаний, отражающее динамику ее изменения. Кроме того, на вход модели подается запрос, также приведенный к виду дизъюнктов.

Решение задачи дедукции состоит в получении противоречий в семантической сети или пустой сети по аналогии с получением пустого дизъюнкта в прин-

ципе резолюции. Отметим, что процедура дедуктивного вывода на сети основана на использовании многосортной логики предикатов первого порядка, что позволяет существенно ограничить область поиска необходимых утверждений, хотя сортность переменных может быть исключена введением соответствующих одноместных предикатов.

Существующие методы вывода на семантических сетях, использующие идею пересечения путей на сетях или наложения одного фрагмента сети на другой, имеют существенный недостаток. Основным недостатком, заключающийся в ограниченности дедуктивных возможностей вывода, связан с комбинаторным ростом числа наложений или пересечений и негибкостью этих методов при переходе с одной предметной области на другую.

Рассмотрим три алгоритма дедукции на семантических сетях: для хорновских дизъюнктов, связанный с трансформацией сети и нахождения пути на сети [Вагин и др., 1984].

Алгоритм дедукции на раскрашенных семантических сетях. Для предикатной вершины P выбирается любая контрарная пара литер, которая затем резольвируется. Если вершина содержит другие контрарные пары, то они рассматриваются как альтернативные и их выбор происходит только после неудачи с унификацией текущей контрарной пары. Если в вершину входят только дуги одного цвета (так называемые «чистые» дизъюнкты), то они не влияют на процесс вывода пустой сети и удаляются из рассмотрения. Процесс продолжается до тех пор, пока не будет выведена пустая сеть. Характерной чертой данного алгоритма является то, что в процессе вывода семантическая сеть не изменяется.

Алгоритм позволяет обрабатывать сети большой размерности и быстрее и эффективнее извлекать нужную информацию, чем другие методы вывода. Однако этот алгоритм неполон, т. е. не гарантирует в любых случаях получение пустой сети для исходного невыполнимого множества дизъюнктов. Алгоритм эффективен и полон для применения в таких системах обработки знаний, которые описываются наборами хорновских дизъюнктов.

Алгоритм дедукции с использованием операторов удаления и расщепления вершин семантических сетей. Основан на преобразовании семантических сетей, для чего вводятся операторы трансформации сетей: *оператор удаления вершины*; *оператор расщепления вершины*.

Предикатная вершина считается свободной от мультидуг, если число дуг, выходящих из одной и той же какой-нибудь другой вершины и входящих в данную вершину, не больше единицы.

Правило удаления вершины следующее: если в сети имеется вершина P , свободная от мультидуг, и этой вершиной связаны дизъюнкты g_1, g_2, \dots, g_n , то после всевозможных резольвирований этих дизъюнктов по литере P из сети эти дизъюнкты вместе с вершиной P удаляются и добавляются новые дизъюнкты, полученные резольвированием g_1, g_2, \dots, g_n . Вершина будет удалена из сети и в случае «чистого» дизъюнкта.

Пронлюстрируем оператор удаления вершины на примере рис. 2.2, задав следующую нотацию (см. набор дизъюнктов):

- 1) $\rightarrow C(a);$
- 2) $P(y) \rightarrow L(a, y);$
- 3) $C(x) \& H(y) \& L(x, y) \rightarrow$
- 4) $\rightarrow P(b);$
- 5) $\rightarrow H(b).$

Здесь для фактов $C(a)$, $P(b)$, $H(b)$ в условии дизъюнкта буква «и» (истина) для краткости опущена. Аналогично опущена буква «л» (ложь) в заключении дизъюнкта 3.

Раскрашенная сеть, соответствующая этому набору дизъюнктов, показана на рис. 2.9,а. Применяв оператор удаления к вершине C , получим сеть, изобра-

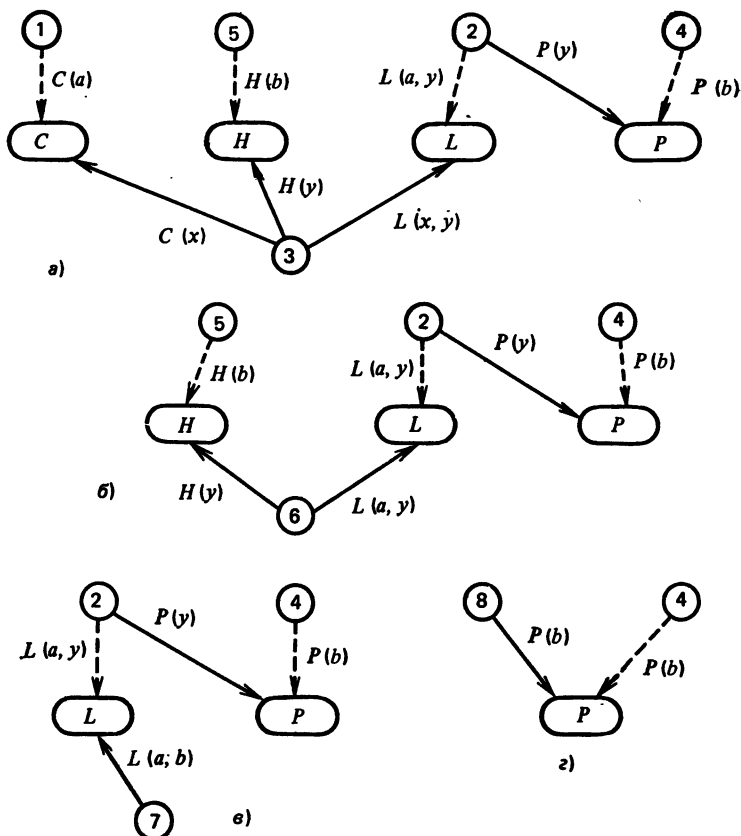


Рис. 2.9

женную на рис. 2.9,б, а к вершинам H, L — противоречие в вершине P или пустую сеть (рис. 2.9,а, в).

Оператор расщепления вершины применяется при наличии в предикатной вершине мультидуг. Пусть имеется множество дизъюнктов $S = \{P \vee \Gamma, \Phi\}$, где $P \vee \Gamma$ — дизъюнкт, в котором компонент дизъюнкта Γ также содержит литеру P , а Φ — любое подмножество дизъюнктов. После применения оператора расщепления множество S примет вид $S' = \{P_1 \vee \Gamma, [\Phi]^{P_1|P}, \Phi\}$,

где $[\Phi]^{P_1|P}$ означает подстановку литеры P_1 вместо P в дизъюнктах Φ , имеющих P . Проиллюстрируем оператор расщепления на примере. Имеем следующий набор дизъюнктов:

- 1) $P(x) \& P(y) \rightarrow P(a);$
- 2) $\rightarrow P(b);$
- 3) $P(a) \rightarrow$

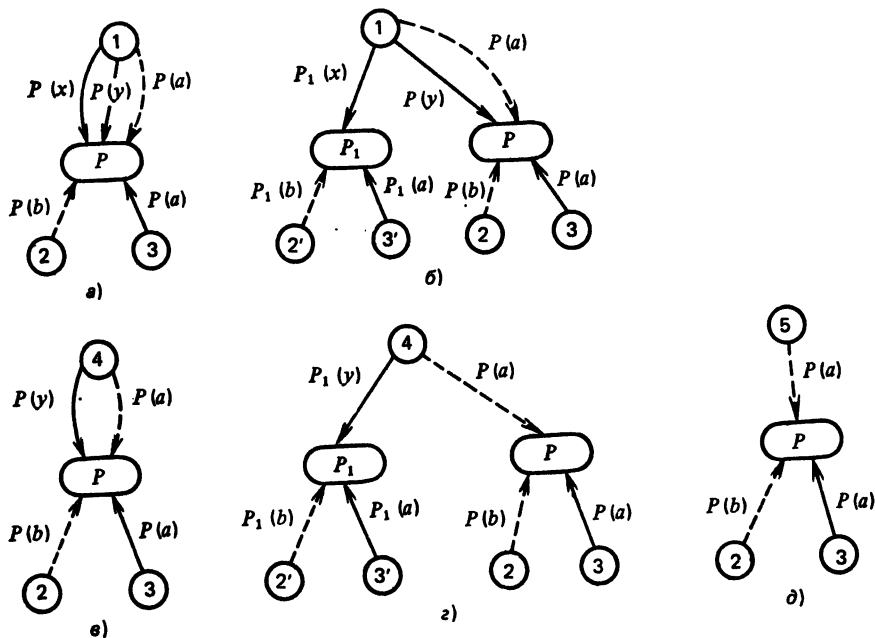


Рис. 2.10

Ему соответствует сеть, представленная на рис. 2.10,а. Применяя оператор расщепления, получаем сеть на рис. 2.10,б. Затем последовательно применяя оператор удаления к вершине P_1 и оператор расщепления к вершине P , получаем противоречивую сеть (рис. 2.10,в—д).

Алгоритм вывода пустой сети с использованием операторов удаления и расщепления состоит в следующем: 1) если в сети имеются вершины, к которым применим оператор удаления, то после его применения эти вершины удаляются; 2) если в сети имеются вершины с мультидугами, то в результате применения оператора расщепления появляются вершины, свободные от мультидуг; затем снова применяем оператор удаления и т. д. до получения противоречия в сети.

Алгоритм выделения путей. Выделим из каждого дизъюнкта невыполнимого множества дизъюнктов S какую-нибудь литеру. Получим последовательность литер L_1, L_2, \dots, L_k . Любая последовательность литер называется *путем на множестве дизъюнктов S* .

Например, для $S = \{P(a) \vee Q(b), \neg Q(x), \neg P(x) \vee C(d), \neg C(x)\}$ путем будет последовательность литер $P(a), \neg Q(x), \neg P(x), \neg C(x)$. Алгоритм вывода пустого дизъюнкта из невыполнимого множества дизъюнктов S состоит в следующем: 1) выделяем какой-нибудь путь Π из исходного множества дизъюнктов S ; 2) выводим всевозможные дизъюнкты на этом пути, т. е. если на пути Π имеются контрарные пары, то они резольвируются и их оставшихся частей дизъюнктов, к которым принадлежат эти пары, формируются новые дизъюнкты; 3) вновь сформированные дизъюнкты добавляются в S и в выбранный путь Π добавляем новые литеры из новых дизъюнктов; процесс продолжается до получения пустого дизъюнкта.

Выделение литеры на семантической сети соответствует выделению соответствующей вершины по дуге, которая взвешена этой литерой. Тогда путем на множестве дизъюнктов будем называть множество выделенных соответствующими литерами вершин на семантической сети. Для сетей этот алгоритм состоит в следующем: после выделения предикатных вершин получаем множество выделенных вершин, дизъюнкты которых или резольвируются, или нет. После резольвирования литеры вновь образованных дизъюнктов добавляются к выделенным вершинам, и процесс продолжается до получения противоречия в сети. Алгоритм полон.

Известно, что автоматическое доказательство теорем в силу вычислительной сложности относится к труднорешаемым задачам [Rabin, 1974; Galil, 1975]. Поэтому возникает необходимость построения параллельных процедур дедуктивного вывода, в которых возможна независимая обработка порций знаний внутри одного обрабатываемого элемента, соединенного с другими аналогичными элементами сетью сообщений. Такой возможностью обладает алгоритм дедукции, связанный с трансформацией семантической сети. В этом алгоритме имеется пять видов параллелизма [Вагин, 1986]:

1. Параллелизм при унификации, осуществляемый внутри одной вершины, свободной от мультидуг. Относящиеся к предикатной вершине дизъюнкты резольвируются по литере, принадлежащей этой вершине, т. е. осуществляется параллельная унификация аргументов литеры цвета 1 (2) с аргументами литер-кандидатов цвета 2 (1) и параллельная генерация новых дизъюнктов.

2. Параллельное удаление вершин, свободных от мультидуг, не имеющих общих дизъюнктов, т. е. параллельное применение оператора удаления вершин.

3. Параллельная стяжка мультидуг одного цвета (если это возможно) как внутри одной вершины, так и среди множества вершин.

4. Параллельное расщепление внутри одной вершины, имеющей несколько мультидуг.

5. Параллельное расщепление нескольких вершин, имеющих мультидуги.

2.5. Неточный вывод на знаниях

А. Р. Хачатрян

Основные определения

Данные и знания, с которыми приходится иметь дело в ИС, редко бывают абсолютно точными и достоверными. Присущая знаниям неопределенность может иметь разнообразный характер, и для ее описания используется широкий спектр формализмов [Bonissone et al., 1985; Prade, 1985; Kanal et al., 1986]. Рассмотрим один из типов неопределенности в данных и знаниях — их неточность. Будем называть высказывание *неточным*, если его истинность (или ложность) не может быть установлена с определенностью. Основопологающим понятием при построении моделей *неточного вывода* является понятие вероятности, поэтому все описываемые далее методы связаны с вероятностной концепцией.

Модель оперирования с неточными данными и знаниями включает две составляющие: язык представления неточности и механизм вывода на неточных знаниях. Для построения языка необходимо выбрать форму представления неточности (например, скаляр, интервал, распределение, лингвистическое выражение, множество) и предусмотреть возможность приписывания меры неточности всем высказываниям.

Механизмы оперирования с неточными высказываниями можно разделить на два типа. К первому относятся механизмы, носящие «присоединенный» характер: пересчет мер неточности как бы сопровождает процесс вывода, ведущийся на точных высказываниях. Для разработки присоединенной модели неточного вывода в основанной на правилах вывода системе необходимо задать

функции пересчета, позволяющие вычислять: а) меру неточности x antecedenta правила (его левой части) по мерам неточности x_i составляющих его высказываний: $x = f(x_1, \dots, x_n)$; б) меру неточности y консеквента правила (его правой части) по мерам неточности правила (r) и посылки правила (x): $y = h(r, x)$; в) объединенную меру неточности высказывания A по мерам, полученным из правил, консеквентом которых является A : $y^0 = g(y_1, \dots, y_m)$.

Для возможности пересчета значений неопределенности при дедуктивном выводе достаточно функций $f(\cdot)$ и $h(\cdot)$. Введение меры неточности позволит привнести в процесс вывода нечто принципиально новое — возможность объединения силы нескольких свидетельств, подтверждающих или опровергающих одну и ту же гипотезу. Другими словами, при использовании мер неточности целесообразно выводить одно и то же утверждение различными путями (с последующим объединением значений неточности), что совершенно бессмысленно в традиционной дедуктивной логике. Для объединения свидетельств требуется функция пересчета $g(\cdot)$, занимающая центральное место в пересчете. Заметим, что, несмотря на «присоединенность» механизмов вывода этого типа, их реализация в базах знаний оказывает влияние на общую стратегию вывода: с одной стороны, необходимо выводить гипотезу всеми возможными путями для того, чтобы учесть все релевантные этой гипотезе свидетельства, с другой — предотвратить многократное влияние силы одних и тех же свидетельств.

Для механизмов оперирования с неточными высказываниями второго типа характерно наличие схем вывода, специально ориентированных на используемый язык представления неточности. Как правило, каждому шагу вывода соответствует пересчет мер неточности, обусловленный соотношением на множестве высказываний (соотношением может быть элементарная логическая связь, например $A = B \& C$, безотносительно к тому, является ли это отношение фрагментом какого-либо правила). Таким образом, механизмы второго типа применимы не только к знаниям, выраженным в форме правил. Вместе с тем для них, как и для механизмов «присоединенного» типа, одной из главных является проблема объединения свидетельств.

Краткая история

Начало развития проблематики неточного вывода связано с работами над одними из первых экспертных систем — MYCIN и PROSPECTOR [Buchanan et al., 1975; Duda et al., 1976]. Оба механизма вывода носили «присоединенный» характер и опирались на простейшую (скалярную) форму представления неточности. Недостатком первых методов неточного вывода является необходимость введения слабо или совсем не обоснованных предположений о независимости высказываний. Эти методы в целом являются эвристическими.

Стремление разработать формализованные механизмы неточного вывода в рамках вероятностного подхода привело к появлению ряда новых методов [Shafer, 1976; Kim et al., 1983; Quinlon, 1983, 1985; Appelbaum et al., 1985; Konolige, 1985; Bundy, 1985, 1986; Nilsson, 1986]. Жесткость скалярной формы представления неточности приводит к отказу от концепции «присоединенности» и (или) к необходимости использования в представлении более сложных, интервальных, форм.

Более радикальное преобразование понятия «вероятностный вывод» осуществлено в исчислении инцидентов [Bundy, 1985, 1986], в котором для описания неточности высказываний последним ставятся в соответствие подмножества специального множества. Ясно прослеживается тенденция к разработке методов, позволяющих обнаруживать и устранять противоречия в знаниях [Quinlon, 1983, 1985; Bundy, 1985; Konolige, 1985].

Типы неточного вывода

PROSPECTOR. Вероятностные методы принятия решений основаны на байесовском подходе. Для задач вывода в ИС этот подход был развит при работе над системой PROSPECTOR [Duda et al., 1976]. В системе PROSPEC-

ГОР в качестве весов, приписанных высказываниям, выступают вероятности соответствующих событий. Множества событий организованы в сети вывода, по которым производится пересчет априорных вероятностей высказываний в апостериорные. Правила в системе имеют вид «если E , то H » и «если E , то \bar{H} », причем в сети вывода для каждой пары высказываний $\{E, H\}$ эти правила присутствуют или отсутствуют одновременно. Вид функции пересчета $f(\cdot)$ постулируется, т. е. для конъюнкции и дизъюнкции используются формулы нечеткой логики: $P(A \vee B) = \max(P(A), P(B))$, $P(A \& B) = \min(P(A), P(B))$. Попытки формального обоснования функций $h(\cdot)$ и $g(\cdot)$ связаны с принятием предположений о независимости: а) $P(H|E \& E') = P(H|E)$ и $P(H|\bar{E} \& E') = P(H|\bar{E})$, где E' — свидетельство, «опосредованно» (через E) связанное правилами с H ; б) $P(E_1 \& E_2|H) = P(E_1|H)P(E_2|H)$ и $P(E_1 \& E_2|\bar{H}) = P(E_1|\bar{H})P(E_2|\bar{H})$. Однако предположения б) [Хачатрян, 1987] являются слишком сильными, фактически сводя объединение свидетельств к выбору одного из них. Таким образом, механизм неточного вывода в системе PROSPECTOR несмотря на свой вероятностный характер является, по сути, эвристическим.

Схема Пиэрла. Байесовские методы неточного вывода получили более глубокое и формально обоснованное развитие в [Kim et al., 1983; Pearl, 1986].

Согласно этим работам знания людей о проблемной области (проблеме) можно представить в виде совместного распределения вероятностей $P(x_1, \dots, x_n)$ на множестве пропозиционных переменных (т. е. переменных, значениями которых являются высказывания) x_1, \dots, x_n . Интерес представляет вычисление условных вероятностей $P(H|E)$, где H (гипотеза) и E (свидетельство) — некоторые композиционные высказывания, составленные из элементарных (т. е. типа {значение x_i есть V_j^i }). Все условные вероятности такого типа можно получить из совместного распределения вероятностей $P(x_1, \dots, x_n)$, однако такое направление преобразования информации не соответствует поведению человека, который не оперирует полным совместным распределением или маргинальными вероятностями высокого порядка (например, $P(x_1, \dots, x_{n-1})$). Элементарными блоками, из которых строится общее знание человека, являются маргинальные вероятности низкого порядка (например, $P(x_i)$, $P(x_i, x_j)$) и условные вероятности, определенные на небольших множествах высказываний.

Пусть фиксирован некоторый произвольный порядок d переменных x_1, \dots, x_n , каждой из которых соответствует вершина в графе. Тогда $P(x_1, \dots, x_n) = P(x_n|x_{n-1}, \dots, x_1) \dots P(x_2|x_1)P(x_1)$. Пусть $S_i \subseteq \{x_1, \dots, x_{i-1}\}$ — минимальное подмножество вершин множества $\{x_1, \dots, x_{i-1}\}$, удовлетворяющее условию $P(x_i|S_i) = P(x_i|x_{i-1}, \dots, x_1)$. Можно показать, что такое подмножество единственно. Проведем от вершин множества S_i направленные дуги к вершине x_i . Осуществив эту операцию для всех вершин x_i , получим ориентированный ациклический граф, в котором отражены многие соотношения независимости, неявно содержащиеся в $P(x_1, \dots, x_n)$. Наоборот, порядок d переменных и ориентированный ациклический граф с совокупностью условных распределений вероятностей однозначно определяют распределение $P(x_1, \dots, x_n)$. Такой «локальный» способ определения $P(x_1, \dots, x_n)$ соответствует заданию системы вероятностных суждений, которыми наиболее естественно оперируют эксперты. Графы (а точнее, гиперграфы), построенные описанным способом, носят название *сетей доверия* или *байесовских сетей*.

Процесс рассуждения (вывода) в системе знаний сопровождается распространением по сети влияния вновь поступивших свидетельств. Введение в систему знаний новых данных приводит к возникновению переходного процесса распространения по сети влияния вновь поступившего свидетельства. После завершения переходного процесса каждому высказыванию (ассоциированному с вершинами графа) оказывается приписана апостериорная вероятность

$\text{Bel}(V_j^i) \triangleq P(V_j^i|D)$, где D — объединение всех поступивших на данный момент в систему данных. Механизм пересчета вероятностей имеет следующую структуру. С каждой вершиной в сети ассоциирован процессор, который получает сообщение от соседних (связанных с ним дугами) процессоров, осуществ-

влет пересчет апостериорных вероятностей $Bel(V_j^i)$ для всех возможных значений V_j^i данной переменной x_i и посылает соседним вершинам ответные сообщения. Деятельность процессора инициируется нарушением условий согласованности с состояниями соседних процессоров и продолжается до восстановления этих условий. Правомочность данной процедуры строго обоснована для одноязычных сетей, т. е. сетей, в которых от одной вершины к другой ведет не более одного ненаправленного пути. Обобщение метода пересчета на многосвязные сети осуществляется преобразованием их структуры в древовидную и требует выполнения ряда ограничительных условий [Pearl, 1986].

MYCIN (EMYCIN). К байесовским методам неточного вывода примыкают механизмы пересчета весов, разработанные в системах MYCIN и EMYCIN [Shortliffe et al., 1975; Buchanan et al., 1984]. Любое высказывание (а не только правило) в этих системах полагается условным по форме, причем в качестве условия высказывания, не являющегося правилом, принимается вся имеющаяся на данный момент в базе данных информация. Высказываниям приписываются веса, называемые *факторами уверенности*. Фактор уверенности (CF) определяется в виде разности $CF[H, E] = MB[H, E] - MD[H, E]$, где $MB[H, E]$ ($MD[H, E]$) — мера возросшей веры (возросшего неверия) в гипотезу H на основе свидетельства E . Меры MB и MD определяются через вероятности $P(H|E)$ и $P(H)$, причем $0 \leq MB, MD \leq 1$, $-1 \leq CF \leq 1$. В зависимости от того, является ли вновь поступившее свидетельство подтверждающим или опровергающим гипотезу, производится пересчет приписанной гипотезе меры MB или MD . Вид функции пересчета $f(\cdot)$ для конъюнкции и дизъюнкции аналогичен используемой в системе PROSPECTOR, а вера в отрицание гипотезы определяется из соотношения $MB[\bar{H}, E] = MD[H, E]$. Функция $h(\cdot)$ определяется произведением фактора уверенности, приписанного правила и веры в истинность свидетельства. Объединение свидетельств посредством функции $g(\cdot)$ осуществляется по формулам, неявно опирающимся на предположения условной независимости свидетельств либо по гипотезе, либо по ее отрицанию [Хачатрян, 1987]. В системе EMYCIN разделение фактора уверенности на две меры MB и MD устранено, что приводит к изменению функций пересчета.

Механизмы неточных рассуждений систем MYCIN и EMYCIN не требуют: 1) априорной жесткой фиксации сетей вывода: правила вместе с приписанными им факторами уверенности можно модифицировать, удалять или добавлять в базу знаний, не заботясь об обеспечении необходимой для пересчета информации; 2) задания априорных вероятностей, так как меры CF , MB и MD аккумулируют как априорную, так и апостериорную информацию; однако отказ от явного использования в процессе пересчета априорных вероятностей может приводить к таким нежелательным последствиям, как принятие гипотезы с меньшей апостериорной вероятностью.

Теория Демпстера — Шефера (ТДШ). Эта теория [Shafer, 1976] была разработана с целью обобщения вероятностного подхода к описанию неопределенности и связана с попыткой освободиться от догмата аксиом теории вероятностей при описании субъективной веры людей.

Рассмотрим конечное множество θ взаимно исключающих друг друга возможностей, именуемое в ТДШ *фреймом различия*. На множестве всех подмножеств θ как на множестве элементарных событий зададим базисное распределение вероятностей $m: 2^\theta \rightarrow [0, 1]$, $m(\emptyset) = 0$, $\sum_{A \subseteq \theta} m(A) = 1$. Величина $m(A)$

понимается как мера доверия, приписанная A . Мера общего доверия, приписанная A , определяется соотношением $Bel(A) = \sum_{B \subseteq A} m(B)$, где $Bel: 2 \rightarrow [0, 1]$ —

функция доверия, характеризующая веру субъекта в истинность события $A \subseteq \theta$. Наряду с величиной $Bel(A)$ существенную информацию о событии A несет значение $1 - Bel(\bar{A})$, определяющее его *меру правдоподобия* ($Pl(A)$). Меры $Bel(A)$ и $Pl(A)$ называют соответственно *нижними и верхними вероятностями*,

что оправдано, в частности, неравенством $\text{Bel}(A) \leq \text{Pl}(A)$. Таким образом, неопределенность (неточность) знаний субъекта об A выражается интервалом $[\text{Bel}(A), \text{Pl}(A)]$.

Важнейшим элементом ТДШ является операция объединения свидетельств (т. е. функция $g(\cdot)$), задаваемая правилом Демпстера [Dempster, 1967]. Правильно является эвристическим и основывается на интуитивных соображениях о разумном способе объединения не связанных друг с другом (независимых) свидетельств.

Несмотря на интуитивную привлекательность многих аспектов ТДШ, она до последнего времени не находила широкого применения в процедурах неточного вывода из-за вычислительной сложности операции объединения, экспоненциальной по мощности θ . Намечившиеся в последнее время возможности по преодолению этих трудностей [Barnett, 1981; Gordon et al., 1985; Shenoy et al., 1986] позволяют рассчитывать на более широкое использование ТДШ в рассуждающих системах.

INFERNO. Механизм неточного вывода INFERNO [Quinlan, 1983] разрабатывался исходя из следующих критериев: 1) система вывода не должна зависеть от каких-либо предположений о вероятностном распределении на множестве высказываний; 2) имеющаяся об этом распределении информация (например, данные о независимости событий) должна быть легко вводимы в систему; 3) высказывания не должны быть жестко разделены на свидетельства и гипотезы — должна иметься возможность гибкого изменения направления вывода; 4) при наличии противоречий в исходных данных система должна их обнаруживать и предлагать пути устранения.

В INFERNO каждому высказыванию A ставится в соответствие вероятностный интервал $[t(A), 1-f(A)]$, где $t(A)$ — оценка снизу вероятности $P(A)$, а $f(A)$ — оценка вероятности $P(A)$. Таким образом, $t(A) \leq P(A) \leq 1-f(A)$. В процессе пересчета оценка $t(A)$ аккумулирует подтверждающую A информацию, а $f(A)$ — опровергающую. Релевантные высказыванию A данные считаются противоречивыми, если $t(A)+f(A) > 1$. Изначально каждому высказыванию A приписаны значения $f(A)=f(A)=0$. Процесс пересчета управляет ограничениями на $t(A)$ и $f(A)$ типа неравенства \gg , полученные хотя и из слабого, но справедливого во всех случаях неравенства $\max_i P(S_i) \leq P(S_1 \vee S_2 \vee \dots \vee S_n) \leq \sum_i P(S_i)$. Например, с отношением « A влечет B с силой S » ($P(B|A)=S$)

связано два ограничения: $t(B) \geq t(A)S$ и $f(A) \geq 1-(1-f(B))/S$, которые приводят к увеличению $t(B)$ и $f(A)$ с ростом $t(A)$ и $f(B)$ соответственно. Пересчет продолжается до тех пор, пока все ограничения не будут удовлетворены. Если по окончании пересчета для каких-либо высказываний A_i имеют место неравенства $t(A_i)+f(A_i) > 1$, т. е. исходные условия противоречивы, то INFERNO с помощью специальных ограничений осуществляет обратный пересчет и формирует множества возможных исправлений, внесение которых в исходные данные делает их совместными. Выбор конкретного из этих множеств осуществляет пользователь, обращаясь при необходимости к интегральному показателю «существования» носимых исправлений.

Исчисление инцидентов. В данном случае [Bundy, 1985; 1986] каждому высказыванию A ставится в соответствие подмножество $i(A)$ множества ω ; $i(A)$ называется инцидентом A и интерпретируется как соответствующее высказыванию A событие из вероятностного пространства элементарных событий ω . Инцидент $i(A)$ принимается в качестве меры неопределенности, приписанной A . При таком введении меры неопределенности логические связки оказываются функционально-истинностными по отношению к инцидентам, например $i(A \& B) = i(A) \cap i(B)$. Если для всех высказываний A известны их инциденты, то, как следствие, известны не только вероятности всех A (элементарные высказывания из ω обычно полагают равновероятными), но и зависимости между ними. При этом инциденты (а значит, и вероятности) сложных высказываний могут быть вычислены по инцидентам высказываний, их составляющих.

Процессом вывода в исчислении инцидентов управляет механизм обнаружения допустимых присвоений (МОДП). На вход МОДП поступает специфицированное пользователем исходное присвоение F , т. е. совокупность пар оценок сверху и снизу $\{\sup_F(A), \inf_F(A)\}$ инцидентов высказываний A . Пользуясь специальными правилами вывода, МОДП строит все допустимые сужения $\{C_K\}$ присвоения F ($\inf_F(A_j) \subseteq \inf_{G_K}(A_j), \sup_{G_K}(A_j) \subseteq \sup_F(A_j)$),

определенные на всем множестве интересующих пользователя высказываний. Если допустимых сужений присвоения F не существует, т. е. исходное присвоение содержит противоречия, то результатом МОДП является установление факта наличия противоречия. Основные трудности применения исчисления инцидентов сводятся к спецификации исходного присвоения (в частности, по вероятностям высказываний). Возможный подход к решению этой проблемы предлагается в [Corlett et al., 1986].

Проблематика неточного вывода вместе с теорией возможностей и приближенными рассуждениями объединены в направлении «Рассуждения в условиях неопределенности», которое интенсивно развивается [Cohen, 1985; Gupta et al., 1985; Kopolige, 1985; Nguyen et al., 1985; Kanal et al., 1986]. Однако на сегодняшний день имеется набор (постоянно пополняющийся) взаимовлияющих методов и опыт их применения в ряде задач [Gordon et al., 1985; Shenoy et al., 1986], причем в одних задачах наблюдается объединение механизмов неточных и приближенных рассуждений [Prade, 1985], а в других — противоборство [Hisdal, 1986; Zadeh, 1986]. В перспективе можно надеяться на сближение подходов [Kyburg, 1987] и разработку теории рассуждений в условиях неопределенности при участии когнитивных психологов [Shafer, 1985].

Вторым перспективным направлением является разработка методики выбора (синтеза) механизмов вывода в условиях неопределенности исходя из специфики задачи и характера мышления эксперта [Tong et al., 1985].

По-видимому, весьма плодотворным окажется сближение методов немонотонного и индуктивного (правдоподобного) вывода [Bobrow, 1980] (см. § 2.6) с методами рассуждения в условиях неопределенности.

Глава 3 Общие

3.1. Уровни понимания

Д. А. Поспелов

Основные определения

В средние века существовали специальные науки — герменевтика, экзогетика и гомелетика, связанные с пониманием и толкованием текстов. Для постижения смысла текста и значения содержащейся в нем информации герменевтика использует только текст; экзогетика — не только текст, но и все сведения о том контексте, в рамках которого этот текст создавался; гомелетика — общие сведения о морально-этических нормах времени создания текста, мотивы и цели автора, его личностные характеристики. Создать ИС, не вложив в них средства понимания, а следовательно, и того опыта, который накопили эти науки, а также литературоведение, невозможно. Именно поэтому в работах по искусственному интеллекту стали мелькать термины «семантическая герменевтика» или «герменевтика текста».

Введем ряд обозначений: T — текст на естественном языке, предъявленный для ввода в систему; E — расширенный текст, включающий условия его порождения в некоторой среде; P — расширенный текст, включающий все невербальные компоненты, относящиеся к субъекту, порождающему текст; TR — прави-

ла пополнения текста, опирающиеся на его структуру; *ER* — правила пополнения текста, опирающиеся на знание о среде; *PR* — правила пополнения текста, опирающиеся на знание о коммуникации и психологии личности; *A* — ответ, формируемый системой; *K* — знания, хранящиеся в базе знаний; *F* — факты, хранящиеся в базе фактов; *FRK* — правила порождения знаний из фактов; *KM* — метафорические знания; *KRM* — правила порождения знаний метафорического уровня.

Предположим, что на вход ИС (рис. 3.1) поступает текст $B \in \{T, E, P\}$. Будем говорить, что ИС понимает текст *B*, если она дает ответы, правильные с точки зрения человека, на любые вопросы, относящиеся к тому, о чем говорится в тексте. Под «человеком» понимается конкретный человек-эксперт, которому поручено оценить способности системы к пониманию. Это вносит долю субъективизма, ибо разные люди могут по-разному понимать одни и те же тексты.

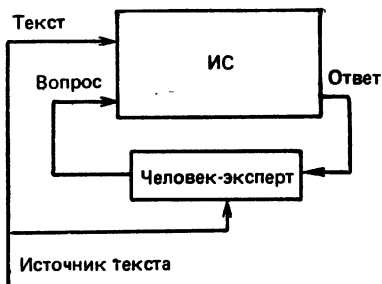


Рис. 3.1

Классификация уровней понимания

В существующих ИС можно выделить пять основных уровней понимания и два уровня метапонимания.

Первый уровень характеризуется схемой $T \rightarrow A$, показывающей, что любые ответы на вопросы система формирует только на основе прямого содержания, введенного из текста *T*. Если, например, в систему введен текст: «В восемь утра, после завтрака, Петя ушел в школу. В два часа он вернулся домой. После обеда он ушел гулять», то на первом уровне понимания система обязана уметь отвечать правильно на вопросы типа: «Когда Петя ушел в школу?» или «Что сделал Петя после обеда?». ИС, способная отвечать на такие вопросы, должна иметь средства, показанные на рис. 3.2. В лингвистическом процессоре происходит морфологический, синтаксический и семантический анализ текста и вопросов, относящихся к нему (см. § 3.2). На выходе лингвистического процессора получается внутреннее представление текста и вопросов, с которыми может работать блок вывода. Используя специальные процедуры (см. § 1.2, 2.4), этот блок формирует ответы. Другими словами, уже понимание на пер-

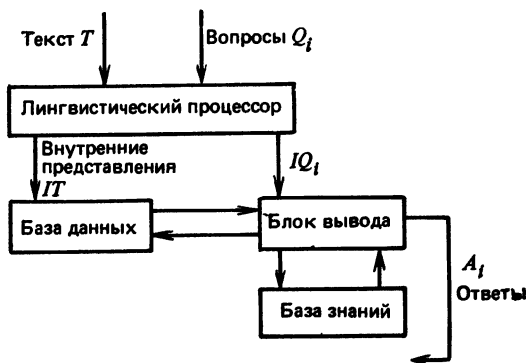


Рис. 3.2

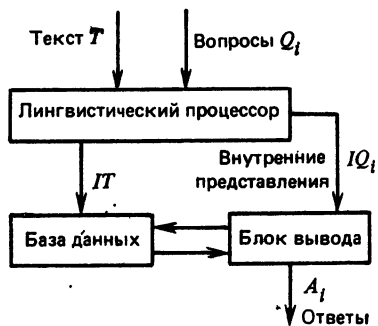


Рис. 3.3

Схема ИС, с помощью которой может быть реализован второй уровень понимания (рис. 3.3), имеет еще одну базу знаний. В ней хранятся закономерности, относящиеся к временной структуре событий, возможной их пространственной организации, каузальной зависимости и т. п., а логический блок обладает всеми необходимыми средствами для работы с псевдофизическими логиками (см. § 2.2).

Третий уровень характеризуется схемой $(T, TR, ER) \Rightarrow A$. К средствам второго уровня добавляются правила пополнения текста T знаниями системы о среде. Эти знания в ИС, как правило, носят логический характер и фиксируются в виде сценариев или процедур иного типа. На третьем уровне понимания ИС должна дать правильные ответы на вопросы типа: «Где был Петя в десять утра?» или «Откуда Петя вернулся в два часа дня?» Для этого надо знать, что означает процесс «пребывание в школе» и, в частности, что этот процесс является непрерывным и что субъект, участвующий в нем, все время находится «в школе».

Схема ИС, в которой реализуется понимание третьего уровня, внешне не отличается от схемы, показанной на рис. 3.3. Однако в логическом блоке должны быть предусмотрены средства не только для чисто дедуктивного вывода (см. § 1.2, 2.4), но и для вывода по сценариям (см. § 1.3, 1.4).

Три перечисленных уровня понимания реализованы во всех практически работающих ИС. Первый уровень и частично второй входят в разнообразные системы общения на естественном языке [Преображенский, 1990].

Следующие два уровня понимания реализованы в существующих ИС лишь частично.

Четвертый уровень характеризуется схемой $(E, TR, ER) \Rightarrow A$. Вместо текста T в ней используется расширенный текст E , который порождается лишь при наличии двух каналов получения информации. По одному в систему передается текст T , по другому — дополнительная информация, отсутствующая в T . При человеческой коммуникации роль второго канала, как правило, играет зрение. Более одного канала коммуникации имеют интеллектуальные роботы, обладающие зрением.

Зрительный канал коммуникации позволяет фиксировать состояние среды «здесь и сейчас» и вводить в текст T наблюдаемую информацию. Система становится способной к пониманию текстов, в которые введены слова, прямо связанные с той ситуацией, в которой порождается текст. На более низких уровнях понимания нельзя понять, например, текст: «Посмотрите, что сделал Петя! Он не должен был брать это!» При наличии зрительного канала процесс понимания становится возможным.

При наличии четвертого уровня понимания ИС способна отвечать на вопросы типа: «Почему Петя не должен был брать это?» или «Что сделал Пе-

вом уровне требует от ИС определенных средств представления данных и вывода на этих данных.

Второй уровень характеризуется схемой $(T, TR) \Rightarrow A$. На втором уровне добавляются средства логического вывода, основанные на информации, содержащейся в T . Это разнообразные логики текста (временная, пространственная, каузальная и т. п. (см. § 3.2)), которые способны породить информацию, явно отсутствующую в тексте. Для нашего примера на втором уровне возможно формирование правильных ответов на вопросы типа: «Что было раньше: уход Пети в школу или его обед?» или «Гулял Петя после возвращения из школы?» Только построив временную структуру текста T , ИС сможет ответить на подобные вопросы.

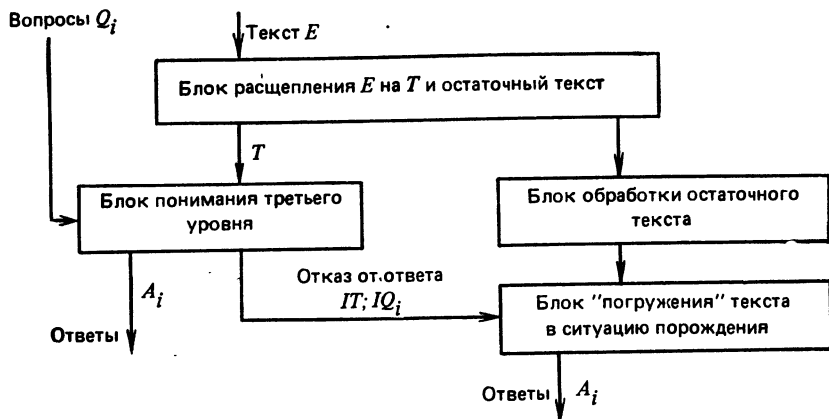


Рис. 3.4

тя?» На рис. 3.4 показана схема такой ИС. Блок понимания третьего уровня соответствует схеме рис. 3.3. Если вопрос, поступивший в систему, соответствует третьему уровню, то система выдает нужный ответ. Если для ответа необходимо привлечь дополнительную («экзогетическую») информацию, то внутреннее представление текста и вопроса передается в блок, который осуществляет соотношение текста с той реальной ситуацией его порождения, которая доступна ИС по зрительному или какому-нибудь иному каналу фиксации ситуации внешнего мира.

Пятый уровень характеризуется схемой $(P, TR, PR) \Rightarrow A$. Для ответа на этом уровне ИС кроме текста T использует информацию о конкретном субъекте, являющемся источником T , и хранящуюся в памяти системы общую информацию, относящуюся к коммуникации (знания об организации общения, о целях участников общения, о нормах участия в общении). Теория, соответствующая пятому уровню, — это так называемая теория речевых актов [Wierzbicka, 1969; Grice, 1973; Searle, 1973].

В [Austin, 1956] было обращено внимание на то, что любая фраза не только обозначает некое явление действительности, но и объединяет в себе три действия: *локуцию*, *иллокуцию* и *перлокуцию*. Локуция — это говорение как таковое, т. е. те действия, которые говорящий произвел, чтобы высказать свою мысль. Иллокуция — это действие при помощи говорения: вопрос, побуждение (приказ или просьба) и утверждение. Перлокуция — это действие, которым говорящий пытается осуществить некоторое воздействие на слушающего: «лестить», «удивлять», «угаваривать» и т. д. *Речевой акт* можно определить как минимальную осмысленную (или целесообразную) единицу речевого поведения. Каждый речевой акт состоит из локутивного, иллокутивного и перлокутивного акта [Брунер, 1984; Моделирование, 1987].

Для четвертого и пятого уровней понимания интересны результаты по невербальным (несловесным) компонентам общения и психологическим принципам, лежащим в основе общения [Горелов, 1980, 1987]. Кроме того, в PR входят правила вывода, опирающиеся на знания о данном конкретном субъекте общения, если такие знания у системы есть. Например, система может доверять данному субъекту, считая, что порождаемый им текст T истинен. Но может не доверять ему и понимать T , корректируя его в соответствии со своими знаниями о субъекте, породившем T . Знания такого типа должны опираться на психологические теории общения, которые пока развиты недостаточно.

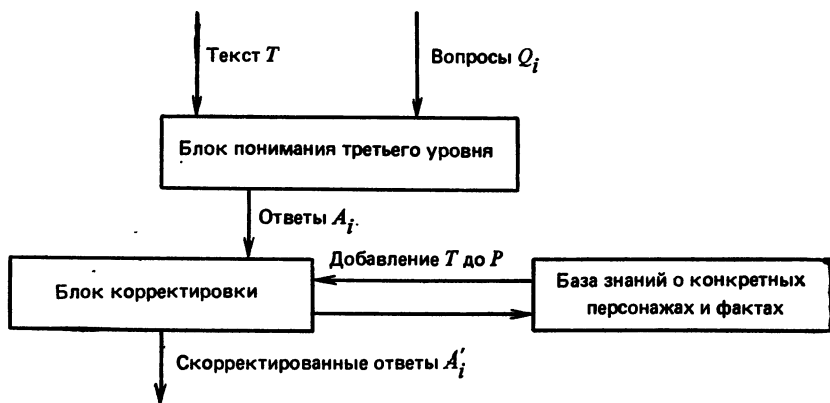


Рис. 3.5

Например, на вход системы поступает текст: «Нина обещала скоро прийти». Если о Нине у системы нет никакой информации, она может обратиться к базе знаний и использовать для оценки временного указателя «скоро» некоторую нормативную информацию (см. гл. 2). Из этой информации можно узнать, что с большой долей уверенности «скоро» не превышает полчаса. Но у системы может иметься специальная информация о той Нине, о которой идет речь во входном тексте. В этом случае система, получив нужную информацию из базы знаний, может приготовиться, например, к тому, что Нина скорее всего придет не ранее чем через час.

Структура ИС, способной осуществить понимание на пятом уровне, показана на рис. 3.5. Блок корректировки ответа использует информацию из базы знаний, хранящей информацию о конкретных людях, городах, автомобилях и т. п., которая отличает их от абстрактных понятий: человек, город, автомобиль. База знаний может быть совмещена с базой данных, так как эти сведения представляют собой некоторые константные факты.

Если в системе реализуются четвертый и пятый уровни, то на ее вход поступает текст E , а текст, расширяющий T до P , формируется так, как показано на рис. 3.5.

Первый метауровень характеризуется схемой $(F, FRK) \Rightarrow K$. На этом уровне происходит изменение содержимого базы знаний. Она пополняется фактами, известными системе и содержащимися в тех текстах, которые в систему введены. Разные ИС отличаются друг от друга характером правил FRK . Например, в системе ДСМ (см. § 2.6) или в системе [Розенблит, 1983], предназначенных для экспертизы в области фармакологии, эти правила опираются на методы индуктивного вывода и распознавания образов (см. гл. 4). Правила FRK могут быть основаны на принципах вероятностей, размытых выводов и т. п. Но во всех случаях база знаний оказывается априорно неполной и в таких ИС возникают трудности с поиском ответов на запросы. В частности, в базах знаний становится необходимым немонотонный вывод (см. § 2.6).

Второй метауровень характеризуется схемой $(K, KRM) \Rightarrow KM$. На этом уровне происходит порождение метафорического знания. Правила KRM , используемые для этих целей, представляют собой специальные процедуры, опирающиеся на вывод по аналогии и ассоциации (см. § 2.6). Известные в настоящее время схемы вывода по аналогии используют, как правило, диаграмму Лейбница, которая отражает лишь частный случай рассуждений по аналогии. Еще более бедны схемы ассоциативных рассуждений.

Если рассматривать уровни и метауровни понимания с точки зрения архитектуры ИС (рис. 3.2—3.5), то можно наблюдать последовательное наращи-

вание новых блоков и усложнение реализуемых ими процедур. На первом уровне достаточно лингвистического процессора с базой знаний, относящихся только к самому тексту. На втором уровне в этом процессоре возникает процедура логического вывода. На третьем уровне необходима база знаний. Появление нового канала информации, который работает независимо от исходного, характеризует четвертый уровень. Кроме процедур, связанных с работой этого канала, появляются процедуры, увязывающие между собой результаты работы двух каналов, интегрирующие информацию, получаемую по каждому из них. На пятом уровне развитие получают разнообразные способы вывода из знания и данных (см. гл. 2). На этом уровне становятся важными модели индивидуального и группового поведения. На метауровнях возникают новые процедуры для манипулирования знаниями, которых не было на более низких уровнях понимания. И этот процесс носит открытый характер. Понимание в полном объеме — некоторая, по-видимому, недостижимая мечта. Но понимание на уровне «бытового понимания» людей в ИС вполне достижимо.

Существуют и другие интерпретации феномена понимания. Можно, например, оценивать уровень понимания по способности системы к объяснению полученного результата. Здесь возможен не только уровень *объяснения*, когда система объясняет, что она сделала, например, на основе введенного в нее текста, но и уровень *обоснования (аргументации)*, когда система обосновывает свой результат, показывая, что он не противоречит той системе знаний и данных, которыми она располагает. В отличие от объяснения обоснование всегда связано с суммой фактов и знаний, которые определяют текущим моментом существования системы. И вводимый для понимания текст в одних состояниях может быть воспринят системой как истинный, а в других — как ложный. Кроме объяснения и обоснования возможна еще одна функция, связанная с пониманием текстов, — *оправдание*. Оправдать нечто означает, что выводимые утверждения не противоречат той системе норм и ценностей, которые заложены в ИС [Лурья, 1979]. Существующие ИС типа экспертных систем, как правило, способны давать объяснения и лишь частично обоснования. В полном объеме процедуры обоснования и оправдания еще не реализованы.

3.2. Понимание текстов на естественном языке

И. В. Совпель

Основные определения

При восприятии текста сначала формируется его промежуточное когнитивное представление [Новиков, 1983]. Это представление является не конечным результатом, а лишь средством, позволяющим соотносить информацию, содержащуюся в сообщении, с системой знаний, на основе чего происходит ее понимание. Далее формируется целостное представление о содержании текста как о фрагменте действительности, описываемом этим текстом.

Можно считать, что система понимает текст, если она может, во-первых, отвечать на все прямые вопросы по этому тексту и пополнять информацию, содержащуюся в тексте, теми знаниями о действительности, которые хранятся в ее базе знаний, и, во-вторых, «представлять» то множество ситуаций, которые могли бы соответствовать введенному в нее тексту в реальном мире.

Понимание текстов на естественном языке (ТЕЯ) включает три уровня интерпретации: синтаксический, семантический и прагматический. Синтаксические процедуры на основе информации, заложенной в словаре, осуществляют грамматический разбор предложений: выделяется подлежащее, сказуемое, дополнение и т. п., между которыми устанавливается связь по управлению в виде *дерева зависимости*. Семантика связана с выводом значений различных синтаксических составляющих. Для этого часто используют специальные структуры — *ролевые фреймы*, которые описывают то необходимое окружение, с ко-

торым всегда связано данное слово на этапе его понимания. На семантическом этапе строится *семантический граф предложения*, от которого затем может быть осуществлен переход к тем соответствующим данному предложению знаниям, которые хранятся в базе предметных знаний. Наконец, прагматика пытается соотнести отдельные предложения друг с другом и с непосредственным контекстом в действительности.

Первые исследования по пониманию ТЕЯ с помощью ЭВМ были связаны с машинным переводом (МП), основными этапами которого являются анализ и синтез ТЕЯ. Первоначальный подход состоял в создании огромных по объему двуязычных словарей и организации пословного перевода. На следующем этапе разработчики пришли к мысли, что необходимо привлекать синтаксис естественного языка (ЕЯ). Приобрела популярность теория трансформационных грамматик (ТГ) Хомского [Chomsky, 1957]. Однако синтаксические методы также не смогли обеспечить хорошего качества МП. Стало очевидным, что только грамматический разбор является недостаточным для моделирования процесса понимания ТЕЯ.

Общие с МП проблемы анализа ТЕЯ возникли и в других задачах обработки текстов на ЕЯ: при разработке вопросно-ответных систем [Winograd, 1972; Woods, 1973], систем общения с ЭВМ на ЕЯ [Попов, 1982], решателей задач, сформулированных на ЕЯ, и т. д. Между системами такого типа (обозначим их СЕЯ) и системами МП (СМП) существует ряд существенных различий:

в СМП основная роль принадлежит лингвистическому анализу, а СЕЯ, кроме того, должны использовать общие и конкретные знания о действительности и речи;

входные тексты для СЕЯ, как правило, сильно ограничены как по тематике, так и по разнообразию допустимых структур, что в сочетании с наличием диалога пользователя с системой значительно упрощает задачу понимания текстов;

входной и выходной языки СЕЯ обычно совпадают, поэтому в таких системах не возникает проблемы грамматической и семантической несоизмеримости языков;

в СМП предъявляются более жесткие требования к временным и стоимостным характеристикам, поэтому здесь особенно важными являются вопросы разработки эффективных с указанной точки зрения алгоритмов и программ анализа и синтеза ТЕЯ [Совпель, 1980].

Система понимания ТЕЯ должна как минимум выделять из текста отдельные термины, обозначающие объекты и их свойства, а также использовать отношения между терминами, уточняющие их роль в том контексте, в котором они встретились. При этом наиболее важными являются два типа отношений [Montgomery, 1972]: выражающие связи, постоянно существующие между языковыми единицами, и выражающие связи между языковыми единицами, возникающие в конкретном тексте.

Методы описания языковых объектов и создания правил анализа текстов должны использовать следующие процедуры [Fillmore, 1971; Walker, 1973]:

составление грамматической характеристики для каждой лексической единицы: определение набора грамматических конструкций, в которых употребляется данная лексическая единица, а также описание грамматических процессов, характеризующих ее употребление в тексте;

грамматическое (структурное) описание каждого предложения на «глубинном» (или абстрактном) уровне, предполагающем подробную синтаксическую и семантическую характеристику всех элементов предложения — как присутствующих в явном виде, так и имплицитных;

прагматическое и семантическое описание предложения, которое составляется после проведения структурного анализа и получения по каждой лексической единице словаря необходимой грамматической информации;

преобразование глубинной структуры предложения, получаемой при анализе содержания текста, в «поверхностную» структуру, непосредственно используемую в языке для передачи смысла;

использование теории речевого поведения, характеризующей употребление предложений в различных социальных и речевых ситуациях, а также логического метода, позволяющего судить о приемлемости данного предложения и его элементов в речевом контексте.

Система понимания ТЕЯ, претендующая на нечто большее, чем выделение из текста отдельных терминов, должна состоять по крайней мере из трех частей [Солтон, 1979]:

описания данной предметной области в терминах важнейших базисных объектов или понятий этой области, в том числе основных, связывающих их логико-семантических отношений;

лингвистической теории, опирающейся на соответствующие характеристики лексических единиц и на грамматические и семантические правила, которые лежат в основе системы анализа языка;

набора правил, позволяющих перейти от каждой приемлемой входной цепочки к глубинной структуре, выражающей лингвосемантические отношения между объектами, полученными в результате лингвистического анализа, а также логико-семантические отношения.

Понимание ТЕЯ на лингвистическом уровне

В большинстве случаев системы понимания ТЕЯ на лингвистическом уровне состоят из двух частей. Первая представляет собой подсистему морфологического и синтаксического анализа (СинтА) — анализатор, который для каждого предложения входного ТЕЯ строит дерево, где указывается грамматическая функция слов и определяется тип синтаксической связи между ними. Вторая представляет собой подсистему семантического анализа (СемА) — интерпретатор, который использует результаты работы анализатора, а также семантическую информацию, получаемую из словарей или справочников, и выдает формализованное представление ТЕЯ.

Морфологический анализ является наиболее освоенным ЭВМ этапом понимания ТЕЯ, не вызывающим в настоящее время трудностей. При этом используются различные методы построения формальной морфологии [Братчиков и др., 1968], методы, основанные на построении строгих систем окончаний и приписаний каждой основе или словоформе морфологического типа [Марчук и др., 1975; Совпель, 1980] и др.

Более сложным считается ограниченный или полный СинтА ТЕЯ. Системы ограниченного СинтА не могут дать идеальных результатов, но многие из них довольно просты и работают с относительной ошибкой ниже 10%.

Количество алгоритмов, дающих синтаксическую структуру в виде дерева зависимостей, очень велико. Большинство из них осуществляли СинтА в соответствии с КС-грамматикой, например известная система «предсказуемого анализа» [Kuno et al., 1963]. Меньшая часть — в соответствии с НС-грамматикой: Q — преобразователь [Rosen, 1967] и др. Основные трудности использования формальных грамматик Хомского связаны с недостаточной эффективностью алгоритмов из-за чрезмерных расходов либо времени, либо памяти ЭВМ, неадекватностью грамматик.

С целью устранения неадекватности КС- и НС-грамматик были введены трансформационные грамматики (ТГ), в основу которых положена идея трансформационных преобразований языковых выражений, при которых сохраняются семантические инварианты и обеспечиваются новые возможности учета некоторых явлений ЕЯ. В системах, основанных на ТГ, каждому предложению входного ТЕЯ ставятся в соответствие две различные структуры: глубинная структура, отражающая семантическую интерпретацию, которая производится в базовом компоненте ТГ; поверхностная структура, отражающая фонетическое представление предложения, порождаемое из глубинной структуры трансформационным компонентом грамматики. При этом используются операции, изме-

няющие структуру предложения: соединение, элиминация и перестановка компонентов структуры [Bobrow et al., 1969; Woods, 1970]. Эти операции могут применяться для сведения синтаксически различных, но семантически тождественных высказываний к единому формату.

В настоящее время ТГ-подход разделился на несколько направлений, в основу которых легли работы [Chomsky, 1970; Fillmore, 1977; Bresnan, 1978; Gazdar, 1982]. Для большинства из этих направлений характерен постепенный отход трансформационных правил на второй план, сокращение их количества. Этот процесс не затрагивает трансформаций движения, которые используются в связи с так называемой теорией следов [Chomsky, 1970]. Оказалось, что определенные ТГ-правила могут быть заменены правилами базисного КС-компонента [Gazdar, 1983]. Основными проблемами использования существующих версий ТГ (см., например, довольно известную систему [Petrick, 1981]) являются организация поиска правильных трансформаций и разработка систем инверсных трансформаций.

Для сокращения в ТГ-алгоритмах количества переборов был разработан синтаксический анализатор расширенной сети переходов (РСП) [Woods, 1970], который работает на сети переходов, состоящей из узлов и дуг. Узлы представляют состояния гипотетической машины, дуги — переходы из одного состояния в другое, которые осуществляются, если выполняются определенные условия, указанные при дугах. В основе анализатора лежит граф переходов конечного автомата, используемый для распознавания регулярных языков в теории автоматов. В сети предусмотрены два вида дуг: лексические дуги, которые соответствуют переходам, разрешенным отдельными словами анализируемого предложения, и рекурсивные дуги, обеспечивающие рекурсивные операции в сети.

Преимуществом РСП является то, что она позволяет получить компактное представление и эффективные алгоритмы для КС-грамматик, а также то, что ее можно расширять до более мощной модели, которая допускает НС-грамматики. Но построение словаря, с помощью которого определяется синтаксическая категория каждой словарной статьи и самой сети переходов, представляет собой сложную задачу, для решения которой нет простых и надежных методов.

Впервые РСП была применена при создании лингвистического процессора (ЛП) системы LUNAR [Woods, 1970]. Тесно связан с РСП и ЛП системы SHRDLU [Winograd, 1972]. В рамках системы SOPHIE [Burton et al., 1979] была продемонстрирована возможность применения РСП для реализации семантически ориентированных ЛП.

Еще одна широко известная система СинтА основана на теории цепочного анализа, согласно которой в каждом предложении можно выделить центральную цепочку (или ядро) с возможными левыми и правыми адьюнктами, которые могут определять либо отдельные фрагменты предложения, либо целые предложения [Sager, 1967, 1973; Grishman, 1973]. Анализ предложения, однако, приводит к двум типам многозначности: одна и та же цепочка адьюнктов приписывается различным элементам предложения; один компонент предложения может выступать в нескольких синтаксических функциях, что приводит к существованию разных вариантов анализа предложений.

Дальнейшее развитие метод цепочного анализа получил в [Sager, 1981], где грамматика ЕЯ представлена НС-правилами, расширенными за счет множества булевых условий — ограничений, оперирующих значениями свойств лексем, заданных в словаре.

Эксперименты, отраженные в [Winograd, 1972], основывались на системной грамматике, которая учитывает большой объем семантической информации об анализируемых объектах. Частью этой грамматики является системная сеть, отображающая взаимодействие и взаимозависимость различных признаков составляющих. Эти значимые признаки используются для контроля системы анализа подобно ограничениям в цепочной грамматике или операциям с реги-

страми в РСР. Грамматика реализована как набор процедур, выполняющих операции типа тех, которые задаются сетью переходов в РСР.

Последний класс систем понимания ТЕЯ на лингвистическом уровне представляется системами анализа, включающими семантические процедуры, которые используются в дополнение к синтаксическим методам или вместо них. В этих системах учитываются семантические характеристики слов, зависящие как от характера анализируемых текстов, так и от более широкого социального контекста, в котором эти тексты употребляются.

Понимание ТЕЯ на семантическом уровне

Семантическими принято считать системы, в которых в процессе анализа содержания текста делаются попытки учесть не только лингвистические, но и логико-семантические отношения между языковыми объектами. Кроме того, контекст, определяющий лингвистические отношения и в обычных системах СинтА не выходящий за пределы предложения, в семантических системах распространяется на уровни дискурса и текста. Наконец, предполагается, что система семантического анализа должна учитывать как сведения о данной предметной области (ПО), так и ее связи с внешним миром в целом. Таким образом, в семантических системах делаются попытки осуществить глобальное понимание ТЕЯ.

Первой важной работой по семантике считается [Katz et al., 1964]. Большой вклад в данную область внесли работы [Fillmore, 1968; 1977; Апресян, 1974]. В [Montague, 1970 a, b] по формализации семантики и частично грамматики с использованием средств интенциональной логики и алгебры были высказаны исходные предпосылки этого направления, которые получили дальнейшее свое развитие в [Мартынов, 1984; Montague, 1974; Bigelow, 1978; Thomason, 1980].

Переход к уровню семантического представления влечет за собой введение новых формализмов. В отличие от синтаксических структур, имеющих вид дерева, семантическое представление имеет вид графа или квазиграфа.

Существует много подходов к построению систем семантического анализа (СемаА). В большинстве из них анализ основывается на особенностях рассматриваемых текстов и в то же время используется некоторое количество «сведений о внешнем мире», содержащихся в семантических словарях и таблицах. Последние строятся на основе универсального семантического алфавита, набора элементарных объектов и связывающих их отношений, так что значения языковых единиц можно представить с помощью абстрактных отношений между комбинациями объектов.

Среди семантических систем, в основе которых лежит прагматическая точка зрения на язык как вид человеческой деятельности, можно выделить [Winograd, 1971] категоризационную, ассоциативную и процедурную модели. Каждая из них предполагает возможность разумного постулирования понятийной организации человеческого знания и соотнесения его с языковыми способами выражения мысли.

Значительная часть моделей СемаА относится к моделям типа «Смысл — Текст». Наиболее известны из них следующие.

Модель семантик представления (СП) [Wilks, 1972] базируется на использовании семантических представлений и принципа предпочтения без промежуточных морфологической и синтаксической интерпретаций. При возникновении неоднозначности на любой стадии выбирается вариант, которому можно отдать предпочтение по определенным критериям. СемаА в рассматриваемой системе ведется в соответствии со следующей схемой. Осуществляется фрагментация ТЕЯ, затем словам приписываются все их значения из словаря, и на анализируемый фрагмент текста поочередно накладываются простые шаблоны, известные системе. Далее применяются специальные правила расширения, преобразующие простой образец в полный добавлением тех слов, которые не вошли в первый. После получения полных образцов для всего фрагмента начинается работа процедура по установлению их семантической близости. В результате

формируется окончательное представление входного ТЕЯ. Модель СП получила дальнейшее развитие благодаря использованию сценариев и фреймов [Wilks, 1977].

Модель концептуальной зависимости (КЗ) [Schank, 1975] в качестве основы семантического представления использует сеть концептуализаций (пар семантических единиц, связанных соответствующими отношениями), являющуюся квазиграфом. Отличие от графа состоит в том, что кроме бинарных рассматриваются тернарные и кватернарные отношения и дуги квазиграфа связывают не только вершины, но и другие дуги. Вводятся основные (примитивные) семантические категории — концепты и правила, на основе которых могут строиться новые концепты произвольной сложности, а также множество примитивных действий (актов), к которым в ходе анализа можно свести все действия, описывающие поведение человека, его намерения, взгляды, умозаключения, ощущения и т. п. Предлагается механизм вывода новых умозаключений (см. § 2.2, 3.3).

Существенное отличие модели КЗ от других состоит в том, что синтаксис играет небольшую роль в понимании ТЕЯ и не делается явного различия между лингвистическим и нелингвистическим знаниями. Базовым механизмом понимания ТЕЯ являются знания о внешнем мире в виде так называемых ожиданий — описаний ситуаций, наиболее вероятных в ближайшем будущем. При этом с каждой ситуацией связывается множество действий, уместных в ожидаемой ситуации. Процесс понимания определяется как извлечение и идентификация концептуализаций, заключенных во входном ТЕЯ.

В рамках идеи КЗ реализованы ЛП таких систем, как MARGIE [Riesbeck, 1975], ДИЛОС [Брябрин, 1978], BORIS [Dyer, 1981].

Модель «Смысл — Текст» [Жолковский, 1974; Мельчук, 1974]) рассматривает семантическое представление связного фрагмента текста как состоящее из двух компонентов: семантического графа и сведений о коммуникативной организации смысла. Для описания существа и степени семантических сходств лексически различных слов предложен язык семантических множителей, отличающийся высокой структурированностью значений.

Среди современных систем, построенных на базе модели «Смысл — Текст», можно выделить ЛП СМП ЭТАП-2 [Апресян и др., 1978] и ЛП системы ПО-ЭТ [Попов, 1982], которую следует отнести к модели типа «Смысл — Текст — Действительность».

В области грамматики наиболее сильные результаты связаны с РСР-механизмом [Woods, 1980; Gazdar, 1983]. Перспективными являются различные схемы для объяснения семантики ЕЯ, которые включают интенциональную логику [Montague, 1974] и основаны на получившей широкую поддержку теории ситуационной семантики [Barwise et al., 1981]. Значительное число исследований в области понимания ТЕЯ сконцентрировано на моделировании связного дискурса и диалога, особенно на прагматических проблемах, таких как структура события [Wilensky, 1978], фокус [Grosz, 1980], анафора [Hirst, 1981] и т. д. Еще одно направление связано с построением практических ЕЯ-систем и прежде всего ЕЯ-интерфейсов для систем баз данных, подобных системам PLANES [Waltz, 1978], ЗАПСИБ [Нариньяни, 1979], DIID [Obermeier et al., 1985].

Современные системы понимания ТЕЯ, как правило, ориентированы на определенную ПО и в качестве средств понимания используют лингвистические знания в виде словарей и грамматик и знания о ПО в виде структуры семантической сети, в которой фиксированы типовые ситуации. Причем ПО ограничивается одним или несколькими сюжетами, а элементы знаний о ПО образуют большую гамму переходов — от атомов смысла (словесные, текстовые и т. п. атомы) до крупных концептуальных единиц (гиперфреймы, планы, сценарии). В основе понимания ТЕЯ в одних случаях используется синтаксическая модель одного из рассмотренных ранее типов или его модификация, а схема ПО представляет собой набор фильтров, уменьшающих неоднозначность синтаксического анализа, в других — основным средством понимания является семантиче-

ская сеть, а грамматические признаки и структуры привлекаются в качестве правил локального контекстного разбора [Леонтьева, 1986]. В связи с этим для построения конкретных практических систем понимания ТЕЯ (индексирование, реферирование, перевод, общение с ЭВМ и т. д.) определенный интерес представляет информационная модель понимания текста [Леонтьева, 1982], которая вводит понятие полного информационного представления текста в виде сложной структуры, включающей простые структуры,— отображения разных семантических подсистем (событийной, модельно-оценочной, временной, денотативно-реферативной и др.). В таких системах происходит выборочный анализ и вследствие отказа от установки на полное понимание задача последнего становится разрешимой.

В связи с работой [Gazdar, 1982] резко возрос интерес к грамматическому анализу ТЕЯ (см. также [Barton, 1985]) и одновременно к решению вопроса о том, является ли ЕЯ КС-языком. Доказательства того, что ЕЯ не является КС-языком [Higginbotham, 1984; Postal et al., 1984], подвергаются сомнению в [Pullum, 1984]. В связи с этим развиваются исследования по комбинированным семантико-синтаксическим системам (проблемы построения таких систем и пример интегрированного семантико-синтаксического ЛП обсуждаются в [Kaplan et al., 1982; Barwise et al., 1983; Cappelli et al., 1984]).

Реализованные методы анализа ТЕЯ продемонстрировали неадекватность моделей ЕЯ, ограниченных в основном структурной изолированного предложения. Основным источником неадекватности является их ориентация на языковой материал, семантику ЕЯ. Человек при формировании когнитивных структур ориентируется прежде всего на прагматику, отражающую его функционирование в окружающей среде. Эта идея реализована в экспериментальных системах [Файн, 1982; Сильдмяз, 1983].

Понимание ТЕЯ—это воссоздание реальной ситуации действительности, о которой говорит текст. Описание всякой ситуации действительности потребует построения некоторой познавательной структуры, в которой основным является выделение отношений между элементами действительности и указание тех *познавательных ролей*, в которых эти элементы находятся в связи с этими отношениями [Поспелов Д. и др., 1985]. Совокупность ролей составляет остов, на котором строится полная познавательная структура, обеспечивающая понимание ТЕЯ. Так что можно говорить о новой тенденции в рассматриваемой области, связанной с отходом от чисто лингвистического понимания проблемы и переходом к модели «Текст—Действительность» вместо традиционной «Текст—Смысл». (Интерес в этом плане представляет система SUSY [Fum et al., 1984] с ее BLR- и ELR-формализмами, основанными на пропозициональном исчислении и учитывающими проблемы сылок, времени, места, манеры для всего входного ТЕЯ; здесь же строится «взвешенное» представление текста в виде иерархической пропозициональной сети, которое учитывает значимость составляющих текста в зависимости от целей.)

В отличие от моделей, основанных на построении отображения между текстом и его некоторым абстрактным представлением, новые модели сфокусированы на коммуникативных функциях языка [Nagin'yani, 1984].

3.3. Синтез связанных текстов

А. В. Зубов, Н. В. Руссова

Основные определения

Проблема синтеза связанного текста возникает в таких ИС, как системы машинного перевода, реферирования и аннотирования текстов, диалоговые системы, отвечающие на естественном языке (ЕЯ), экспертные системы. В идеале задача сводится к тому, чтобы некоторое исходное содержание, хранящееся в памяти ЭВМ в закодированном виде, передать в виде взаимосвязанной це-

почки предложений на ЕЯ. Трудность этой задачи заключается в том, что текст является сложным образованием, в котором неизвестным способом пересекаются логические, семантические, синтаксические, морфологические и другие правила. Причем среди этих правил существуют как строго определяемые (детерминированные), зависящие от исходного замысла, так и случайные, вносимые в алгоритм порождения индивидуальными особенностями текстообразования, автора текста, модальностью текста, его прагматикой и др.

Синтез текста состоит из трех этапов: определение содержания текста и последовательности развертывания его во времени; запись будущего текста на внутреннем языке системы; перевод текста на ЕЯ.

Сочинение любого текста предполагает адресата: автор текста (или собеседник в диалоге) старается учесть возможности понимания его читателем (или слушающим). Это отражается прежде всего на общей организации текста, закономерности построения которого исследует теория дискурса.

Более локальной является теория фокуса внимания [Grosz, 1977], которая исследует условия, переключающие внимание адресата с одного объекта на другой, и приемы автора текста для управления вниманием адресата.

Способы соотнесения единиц языка с понятиями и объектами действительности изучает теория референции. Проблема здесь заключается в том, что единицы языка имеют обобщенный характер, но используются для обозначения и конкретных вещей («Этот дом»), и обобщенных понятий («Дом — жилье человека»), и образов («Земля — наш дом»). В связи с этим изучаются функции указателей («этот», «тот»), местоимений, а также моделируются способы правильной референции с помощью нелингвистического контекста.

Теория коммуникативных неудач исследует случаи неправильного понимания текста и их причины [Городецкий, 1985; Селиванова, 1989; Goodman, 1986]. Это направление основывается на более общих фактах и закономерностях, но имеет практическую задачу — предупреждать, выявлять и устранять коммуникативные неудачи в диалоге с ЕЯ-системами.

Первые эксперименты по синтезу на ЭВМ осмысленных предложений относятся к началу 60-х годов, когда были разработаны первые программы синтеза отдельных английских и русских предложений. Детальный анализ программ синтеза предложений дан в [Карпилович, 1977].

Алгоритм синтеза русского абзаца описан в [Su et al., 1969]. Исходной лингвистической базой являлись словарь (550 слов), грамматика и некоторые данные о совместной встречаемости слов. Слова разделены на 74 семантических класса. Используется упрощенная грамматика зависимостей. Вначале синтезируется отдельное предложение. С учетом структуры первого предложения выбирается структура второго и т. д. При отборе слова в предложения используется датчик случайных чисел. Число предложений в абзаце задается в начале работы программы.

Первые программы синтеза связного текста [Gunzenhauser, 1963] строили последовательность осмысленных фраз заданной синтаксической структуры. В единый текст они объединялись благодаря семантической близости слов, входящих в словарь системы порождения. В [Stickel, 1967; Моль, 1975] описаны программы, порождающие «автопоэмы» или «автостихотворения». В первой работе процедура порождения опиралась на словарь, хранящийся в памяти ЭВМ. В словаре каждое слово снабжалось набором грамматических индексов, представляющих окончания слова и ограничения на сочетаемость его с другими. По датчику случайных чисел строилась последовательность индексов, удовлетворяющая некоторым условиям. Затем вместо индексов с помощью датчика случайных чисел подбирались слова из словаря, удовлетворяющие каждому набору признаков.

Ранние системы скорее имитировали, чем порождали тексты. Они даже не имели заданной темы и заранее определенных действующих лиц. Связными их делали читающие, улавливая связи между значениями слов, входящих в соседние предложения текста.

Система порождения прозаического текста [Ильин и др., 1982] синтезирует описание некоторой трехмерной сцены (уличной сцены с 30 объектами). Составляющие сцены представляются в виде глубинной структуры с помощью словаря объектов и отношений. Словарь объектов — это сопоставление каждого эталона («заместителя» объекта) с его именем на русском языке. В словаре указывается также морфологическая информация о соответствующих именах. Элементарные отношения представлены пространственными отношениями, выражаемыми предлогами («над», «под», «перед» и др.). Причем используются не только предлоги, но и глагольные выражения пространственных отношений.

В [Кузнецов, 1975] представлен алгоритм порождения описания с помощью рекуррентной процедуры, осуществляющей анализ окрестностей элементов (т. е. выявление в памяти ЭВМ элементов, связанных с заданными) и сопоставляющей с каждой окрестностью определенный текст описания. В качестве элементов выступают слова и словосочетания. Для начального заполнения памяти берутся предложения связного текста и каждое предложение представляется графом, использующим различные типы отношений «быть подсоединенным», «принадлежать», «быть элементом» и т. п.).

В [Шереметьева, 1984; Буторов и др., 1985] описан алгоритм синтеза формулы изобретения (краткая характеристика сущности изобретения). Предварительным анализом формул изобретения определенного типа выявляется набор признаков, необходимый для передачи идеи изобретения. Этот набор признаков фиксируется в виде простых предложений на ЕЯ, причем одно предложение репрезентирует один признак. Содержание синтезируемой формулы изобретения представляется в виде семантико-синтаксической структуры, которая объединяет все простые предложения, так как по нормативным требованиям изобретение должно описываться одним, пусть сложным, предложением. На последнем этапе семантико-синтаксическая структура формулы изобретения преобразуется в текст на ЕЯ с применением трансформаций свертывания.

В [Зубов, 1985] представлена система синтеза связного текста, построенная на анализе русских научных, публицистических и поэтических текстов. Задание на порождение (тема текста) представляется в виде таблицы. В ней перечислены конкретные имена, которые репрезентируют главных и второстепенных действующих лиц (предметов, понятий) будущего текста. Имена разделены на имена-субъекты, имена-объекты, места времени. Процесс порождения начинается с порождения семантико-синтаксической формулы текста, которая строится из семантико-синтаксических формул абзацев, хранящихся в памяти ЭВМ, с помощью вероятностных правил простой цепи Маркова. Семантико-синтаксические формулы абзацев представлены в памяти с помощью специального языка Семсинт, в основе которого лежат 20 семантических функций, подобных семантическим падежам Филлмора. Заполнение позиций построенной семантико-синтаксической формулы текста конкретной лексикой из словаря основ, хранящегося в памяти ЭВМ, вначале осуществляется из таблицы основного статического содержания, а затем — из специальной таблицы слов-конкретизаторов. Такие таблицы сопровождают в памяти ЭВМ каждую семантико-синтаксическую формулу абзаца. Прилагательные к существительным и наречия к глаголам в формулу текста подбираются с помощью датчика случайных чисел из некоторого множества, типичного для соответствующего существительного или глагола.

Схемы дискурса и синтез текстов

Теория дискурса отражает законы построения устных и письменных текстов и строится на базе трех фундаментальных предпосылок.

1. Форма, в которой информация появляется в тексте, не обязательно совпадает с формой, в которой она хранится в памяти человека (в базе знаний). Эксперименты [Cheif, 1976] показали, что в различных ситуациях люди, рассказывая об одном и том же, значительным образом варьируют свои рассказы при одном и том же внутреннем представлении текста. Однако и внутреннее

семантическое представление текста, которое порождается из знаний, может быть неоднозначно, так как может отображать знания не в полном объеме, вычлняя из них те или иные фрагменты.

2. Наряду с общечеловеческими законами построения текстов имеются законы, характерные для отдельных социокультур, для носителей данного языка или для данной группы людей.

3. При построении текстов необходимо учитывать их жанровые особенности. При описании многих моделей дискурса в качестве базовых структурных единиц, которыми говорящий пользуется для описания действительности, используются так называемые *риторические предикаты*. Риторическим предикатом называется семантико-синтаксический инвариант всех предложений, имеющих единую риторическую (или, в иной терминологии, коммуникативную) функцию. Их число колеблется от 10 до 25. Наиболее часто используются следующие риторические предикаты.

1. Атрибуция (У меня есть новый справочник).
2. Эквивалентность (Этот метод аналогичен классическому).
3. Уточнение общего факта (Таня толстая, Она весит 80 кг).
4. Объяснение некоторого логического заключения (Она ничего не успевает, потому что берется за несколько дел сразу).
5. Доказательство факта (Публика любит этого актера: фильмы с его участием подолгу не сходят с экрана).
6. Аналогия (Коктейль с белым вином готовится точно так же, как и с красным вином).
7. Отличительный признак в классе (В жирафе нет ничего особенного. Пожалуй, лишь шея у него слишком длинная).
8. Состав чего-либо (Всякий рассказ состоит из завязки, развития сюжета, кульминации и эпилога).
9. Возможность (Если Иван пойдет в кино, то он потом нам все расскажет).
10. Альтернатива (В театр я хотел бы пойти или сегодня, или завтра).
11. Причинно-следственная связь (Добавление спирта в период ферментации замедляет ее развитие).
12. Противопоставление с учетом закона исключенного третьего (Или — пан, или — пропал).
13. Вывод (Значит, нам стоит попробовать сделать по-другому).
14. Идентификация (Москва — столица СССР).
15. Переименование (Схемы дискурса — это то же, что схемы Проппа).
16. Упоминание (Вернемся теперь к тому моменту, когда Петр родился).
17. Пример из класса (Писатели-натуралисты, например, такие как Дарелл, пробуждают в людях доброту и любовь к животным).

Считается, что достаточно иметь такой список риторических предикатов, чтобы строить любые тексты, состоящие из одного абзаца. В абзаце последовательность риторических предикатов не является случайной. Можно задать определенные правила, по которым сочетаются риторические предикаты, описать сочетания в виде систем, формальной моделью которых могут быть семантические сети, фреймы, продукции и сценарии (см. гл. 1). Схемы возможных сочетаний риторических предикатов называются *схемами дискурса*. В них находят отражение жанровые особенности генерируемых текстов: основная лексика, наиболее частые синтаксические конструкции и специфика сочетания риторических предикатов в тексте.

В качестве примера приведем схемы дискурса, выделенные из текстов, которые по жанру принадлежат к словарным статьям [McKeown, 1985]. Схемы реализованы в программе TEXT, разработанной для генерации ответов справочной системы — энциклопедических справок. В схемах используются следующие обозначения: {} — необязательный элемент; / — или; + — элемент может повторяться произвольное число раз; * — необязательный, но повторяющийся элемент; , — множественная интерпретация элемента.

Опишем четыре схемы, наиболее часто встречающиеся в текстах этого жанра:

1. Схема текста с общим атрибутивным значением:
{подробности, сужение содержания первого предложения}
конкретный пример *
{отличительный признак в классе}
{вопрос, проблема и ответ}/
{сравнение, простое противопоставление, противопоставление по закону исключенного третьего}
подробности/объяснение/вывод/сравнение
2. Схема с общим значением идентификации:
идентификация (класс и принадлежность к классу/функция)
{аналогия/состав/атрибуция/подробности}
конкретный пример/доказательство+
{подробности/аналогия/атрибуция}
{конкретный пример/доказательство}
3. Схема с общим значением состава:
причина — следствие * / атрибуция * / {идентификация нижестоящих элементов / атрибуция нижестоящих элементов {конкретный пример / доказательство}
{сравнение / аналогия}}+
{подробности / объяснение / атрибуция / аналогия}
4. Схема сравнения и противопоставления:
упоминание / атрибуция ($\neg A$)
{атрибуция (A) / конкретный пример / доказательство (A) /
подробности (A) / вывод (A) / объяснение (A)}+
{сравнение (A и $\neg A$) / вывод (A и $\neg A$) / обобщение (A и $\neg A$)}+

Выбор схемы дискурса при порождении текстов определяется целями, которыми руководствуется генератор текстов. Эти цели могут формулироваться в терминах типа «доказать», «сравнить», «описать», «объяснить» и т. п. Если выбор схемы неоднозначен, то дальнейшее уточнение определяется информацией, которая имеется у ИС (или у человека) в базе знаний. В вопросно-ответных системах схема дискурса чаще всего определяется типом заданного системы вопроса. Например, при ответе на вопрос: «Что есть X?», система отбирает две схемы дискурса: основную — идентификации и вспомогательную — состава класса. Если оказывается, что X является в иерархической базе знаний единицей высокого уровня общности (см. § 2.3), то объем информации, вызываемой системой на вопрос, будет невелик, так как чем шире класс, тем меньшим количеством общих свойств он обладает. Поэтому, получив из базы знаний информацию о том, что X является понятием высокого уровня общности, система может использовать вспомогательную схему дискурса.

Особая специфика характерна для синтеза текста в диалоге, так как диалог порождается несколькими авторами, но вместе с тем является единым связным текстом, причем реплики собеседников имеют очень высокую степень связности и дополняют друг друга прагматически и синтаксически. Это составляет основную трудность при синтезе текста в диалоговых системах, поэтому чаще синтезируют изолированные абзацы. Но эту особенность можно обратить на пользу при условии, что модель диалога будет максимально отражать естественные закономерности сочетания единиц диалога.

Рассмотрим наиболее современные (иерархические) модели диалога [Ferrari, 1986; Селivanова, 1989]. Суть подхода такова: выделяют ограниченное число минимальных единиц диалога — речевых актов (РА), представляющих простейшие действия в диалоге типа «спросить» и «ответить». Речевые акты, сочетаясь друг с другом, образуют более сложные действия типа «описать», «аргументировать», причем сочетаются не любые РА, а строго определенные цепочки (выявленные исследователями эмпирически). Каждая цепочка РА является единицей более высокого уровня и выполняет только ей присущую функцию. Кроме того, она имеет пограничные знаки, отделяющие ее от другой цепочки того же уровня. Из допустимых комбинаций единиц второго уровня состоят единицы

третьего и т. д. Верхний уровень рассматривает как единое целое весь диалог, тоже имеющий пограничные знаки, в частности установление контакта и его прекращение. Число уровней может быть разным, так как зависит не только от объективных особенностей текста диалога, но и от методов моделирования. В [Селиванова, 1989] предлагается следующий набор РА: указательный, побудительный, вопросительный, выделяющий, информирующий, отказ, согласие, выражение, подсказка, подтверждение контакта и (или) понимания.

Структура верхнего уровня, описывающего диалог, такова:

БЕСЕДА : = установление контакта

{настройка}

мини-диалог

(обсуждение результатов/

настройка на окончание беседы)

прощание

УСТАНОВЛЕНИЕ КОНТАКТА : = приветствие

{узнавание}

НАСТРОЙКА : = напоминание об известном/

описание факта, ситуации, цели

МИНИ-ДИАЛОГ : = поручение/

обсуждение планов/

комментарий/

коррекция совместных действий/

совместное решение

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ : = утверждение о выполнении цели

{замена цели/

отказ от цели)

{признание заслуг/

благодарность}

НАСТРОЙКА НА ОКОНЧАНИЕ БЕСЕДЫ : = смена темы/

дальнейшие планы

ПРОЩАНИЕ : = «До свидания», «Пока» и т. п. языковые выражения.

Основная единица диалога — МИНИ-ДИАЛОГ. Разные цели собеседников порождают разные МИНИ-ДИАЛОГИ: поручение, обсуждение планов и т. д., состоящие из обменов репликами, тоже имеющих структурное единство. Еще более дробной единицей является реплика, которая состоит из речевых актов.

На пути создания хороших систем генерации текстов имеется немало трудностей. Не изучены законы соединения отдельных фрагментов текстов в единое повествование (отметим ранние работы [Севбо, 1969, 1981]). Поэтому в современных ИС системы анализа и синтеза текста опираются на соображения «здорового смысла» [Файн, 1982], а не на лингвистическую теорию, которой пока нет. Не решены фундаментальные проблемы, связанные с такими явлениями, как фокус внимания или схемы дискурса для тех жанров текстов, которые характерны для ИС.

3.4. Машинное творчество

М. Г. Гаазе-Рапопорт, Р. Х. Зарипов

Суть проблемы

Машинное творчество является программной имитацией процесса создания произведений, которые можно считать результатами творческой деятельности. В этой проблеме выделяются два аспекта: использование ЭВМ для исследования человеческого творчества и его закономерностей, в частности модельная проверка гипотез о структуре и особенностях творческого процесса; использование

ЭВМ для интенсификации творческого процесса в диалоге с человеком с целью создания творческих произведений.

Сложность проблемы усугубляется отсутствием четкого определения понятия творчества. Под творчеством принято понимать процесс создания нового. Однако, после того, как найден алгоритм решения задачи, считавшейся творческой, ее решение перестает считаться творческим процессом. Таким образом, под творческими процессами понимается решение задач, для которых в настоящее время неизвестны алгоритмы отыскания решения.

Перечислим основные особенности произведений, которые являются результатом творческого процесса и считаются творческими.

1. В основе всех произведений, независимо от их характера, всегда лежат определенные исходные данные: тема (цель), вид произведения, набор (часто весьма большой, но всегда конечный) исходных «кирпичиков» (слов, нот и т. п.), из которых строится произведение.

2. Все произведения строятся с использованием более или менее формализованных закономерностей, правил и ограничений (правила грамматики, правила стихосложения, типовые структуры сказок и др.) [Пропп, 1928, 1969; Колмогоров, 1963, 1965а, б; 1966].

3. Отклонения от правил (так называемые «творческие находки») в массе своей имеют случайный характер и могут быть воспроизведены искусственно включением в машинную программу случайных составляющих.

Основной задачей машинного моделирования творческого процесса является использование известных и поиск новых закономерностей человеческого творчества. Процесс человеческого творчества, как и мышление, во многом протекает неосознанно, и человек не фиксирует на многих его процедурах своего внимания. Основу творческой деятельности человека составляют глобальные процедуры мышления, не ориентированные на конкретную деятельность, а приходящие любой из них, например: альтернативный выбор из множества имеющихся вариантов; установление связей между описаниями начальной и целевой ситуаций; вычленение классифицирующей структуры (фрейма), позволяющей отнести ситуацию к тому или иному классу ситуаций; обобщение описаний; прогнозирование; заимствование ситуации из некоторого их множества; варьирование (трансформация) ситуации и др.

При создании программ моделирования возникает задача оценки качества результатов их творчества. Это требует исследования вопросов восприятия и оценки людьми творческих произведений, а также отыскания физиологических коррелятов, объясняющих эти оценки. Первые попытки рассмотрения таких закономерностей применительно к абстрактной живописи рассмотрены в [Налимов, 1979].

Краткая история моделирования творческих процессов

Общий принцип генерации литературных текстов был сформулирован в трудах Раймонда Луллия (1235—1314). Этот принцип сочинения получил дополнительное подтверждение в принципе «усилителя мыслительных способностей» [Эшби, 1956]. Сущность его сводится к следующему. Имеется исходный набор элементов (символов, слов) из некоторого алфавита. Из этих символов случайным образом строятся последовательности достаточно большой длины, которые поступают в фильтр, выдающий на выход лишь в каком-то смысле приемлемые последовательности. Таким образом можно получить любое осмысленное высказывание, любой связный текст. Однако время генерирования и отсеивания бессмысленных комбинаций столь велико, что легко видеть практическую бесполезность подобного метода. Тем не менее этот метод, дополненный эффективными средствами сокращения перебора, учитывающими закономерности и свойства порождаемых объектов (текстов, музыкальных произведений), до сих пор лежит в основе подавляющего большинства способов машинного решения творческих задач.

В 50—60-е гг. появились первые программы, играющие в шашки [Strachey, 1952], шахматы [Shannon, 1950, 1956] и другие игры, первые системы, распо-

знания образы [Француз, 1966; Бонгард, 1967; Вапник, 1974] и др. Обзор состояния работ в области программирования игр дан в [Поспелов Д., 1966; Адельсон-Вельский и др., 1978, 1983].

Невозможность или неумение алгоритмически описать процедуру решения задачи привели к попыткам алгоритмического описания действий людей, решающих подобные задачи. На основе анализа этих описаний строятся алгоритмы, повторяющие действия, которые используются для решения задач. Выдвигались предположения, что подобные алгоритмы адекватно моделируют работу человеческого мозга [Ньюэлл и др., 1964].

Описанное направление получило наименование *эвристического программирования*. Практика показала, что эвристические программы часто приводят к цели, однако гарантировать достижение цели всегда, а также приобрести уверенность, что с их помощью получается оптимальное решение, невозможно. Если удастся найти точные оценки применимости и результативности эвристических программ, то они перестают быть эвристическими и превращаются в обычные программы для алгоритмизируемых задач.

Эвристические программы для доказательства математических теорем [Геллернтер, 1963, Ньюэлл и др., 1964] и программа «Общий решатель проблем» (GPS) позволяют решать только задачи, для которых заранее сформулированы возможные подцели и перечислены используемые операторы.

Ограниченность эвристического подхода демонстрирует система простых эвристических правил, рекомендуемых при решении школьных геометрических задач: 1) выпишите, что дано; 2) выпишите, что следует доказать; 3) сделайте чертеж; 4) выпишите перечень относящихся или могущих относиться к задаче теорем; 5) наметьте план решения, т. е. сформулируйте промежуточные подцели; 6) приступите к процедуре решения (доказательства). Выполнение этих правил помогает решить задачу, однако их недостаточно. Необходимы еще знания геометрии, удача, интуиция или знание при выборе подцелей, т. е. знание предметной области, из которой взята задача.

Таким образом, основной недостаток эвристических программ заключается в том, что, используя знания о поведении человека при решении творческих задач, они не используют в достаточной мере знания о свойствах и особенностях той предметной области, в рамках которой должна решаться задача.

Следующим шагом на пути совершенствования программ моделирования творческих процессов является переход к явному использованию знаний об этих процессах и особенностях произведений. Проиллюстрируем это положение на примере создания волшебных сказок, стихотворений и музыкальных произведений.

Генерация литературных произведений

Для решения задачи генерирования произвольных осмысленных текстов необходимо ввести ограничение структуры генерируемых предложений — использование лишь полных, грамматически правильных простых предложений, и ограничение предметной области, в которой должны получаться осмысленные тексты. Известны три способа введения ограничений на предметную область или, что то же, на представление знаний о рассматриваемой предметной области (подобное разделение условно, при задании предметных областей способы комбинируются).

1. Задание одного или нескольких текстов, характеризующих данную предметную область [Арсентьева, 1963а, б, 1965а, б].

2. Построение в памяти машины относительно полной модели предметной области вместе с возможными отношениями между ее элементами. Вводя в машину правила порождения текста, основанные на характерных для этой предметной области псевдофизических логиках и присущих им правилах вывода, можно ожидать, что система может генерировать осмысленные предложения и их цепочки, неявно содержащиеся в модели, но отличные от тех, которые были использованы при построении этой модели (см. § 2.2).

3. Введение правил, присущих отдельным жанрам.

В настоящее время наиболее хорошо исследована структура волшебных сказок. На основе их изучения было показано [Пропп, 1928], что в основе сказок лежит жесткая типовая структура и выделено конечное число вариантов структуры. Пропп и его последователями [Мелегинский, 1958, 1971, 1979; Новиков, 1974; Рошияну, 1974; Новик, 1975] установлено, что множество разнообразных персонажей сказки распадается на небольшое число групп (герой, антигерой, помощник, вредитель, даритель и др.), причем представитель каждой группы независимо от его конкретного воплощения выполняет в сказке определенную роль, а его функции и поведение в сказке четко заданы.

Важной особенностью сказочных текстов является значительное количество встречающихся в них типовых клишированных выражений типа «В некотором царстве, некотором государстве», «Жили-были», «И я там был, мед-пиво пил, по усам текло, в рот не попало» и др. и характеристик. Так, лес всегда дремучий, конь-помощник обычно имеет имя Сивка-Бурка и о нем, как правило, говорится «Бежит, земля дрожит» и т. п.

Идеи Проппа, развитые в [Bremond, 1966], послужили основой для создания программы сочинения волшебных сказок, разработанной Клейном (ФРГ). С помощью этой программы порождались тексты следующего типа: «Моревичи жили в отдаленном районе. Отец был Ерема. Мать была Василиса. Старший сын был Балдак. Средний сын был Марко. Младший сын был Борис. Николай тоже жил там. Николай был рожден чудесным образом. У Балдака был волшебный конь. В отдаленном районе появился медведь. Медведь завладел волшебным конем...» и т. д.

Для того чтобы тексты, генерируемые ЭВМ, не были столь «жесткими», возникла необходимость в развитии средств представления знаний о сказочных мирах и действиях персонажей в них. Для описания сказочных миров стали использовать семантические сети, фреймы, сценарии и продукции, а также их различные комбинации [Minsky, 1974; Schank, 1975]. С помощью этих средств на базе языков ЛИСП и ФРЛ, реализованных в системе ЛИСП—ФРЛ МЭИ, удалось построить программную систему TALE, способную генерировать разнообразные структуры русских волшебных сказок.

В системе TALE [Гаазе-Рапопорт и др., 1980; Гаазе-Рапопорт, 1983; Haase-Rapport et al., 1984] сказка представляется в виде трех последовательных частей: экспозиции, тела и постпозиции. Тело задает последовательности встреч, связанных соединительными элементами. Знания о сказочном мире и его свойствах задаются в виде системы фреймов, связанных в сеть. В форме фреймов записана как исходная информация, необходимая для генерирования сказки (фреймы типовых глаголов, типовых действующих лиц и т. п.), так и информация, получаемая в ходе работы программы. Использование принципа наследования, реализуемого с помощью родовидовых связей, позволяет учесть в ходе формирования конкретного варианта сказки, не только исходную информацию, но и данные, полученные в ходе генерации, для соответствующей корректировки исходных данных, необходимых для продолжения этой генерации.

На первом этапе работы программы (TALE-1) формируется цепочка глагольных фреймов, определяющая грубую структуру выбранной сказки. Вариант такой структуры в виде последовательности имен фреймов имеет вид

ЖИЛ ИМЕЛ НАЛЕТЕЛ УНЕС

ВСТРЕТИЛ ПОПРОСИЛ ПОМОГ РАССКАЗАЛ ДАЛ ПОШЕЛ-ДАЛЬШЕ

ВСТРЕТИЛ ПОПРОСИЛ ПОМОГ РАССКАЗАЛ ДАЛ ПОШЕЛ-ДАЛЬШЕ

ВСТРЕТИЛ ПОПРОСИЛ ПОМОГ РАССКАЗАЛ ДАЛ ПОШЕЛ-ДАЛЬШЕ

ВИДИТ ДОСТАЛ ПРЕОДОЛЕЛ ПОШЕЛ-ДАЛЬШЕ

ВИДИТ ДОСТАЛ ПРЕОДОЛЕЛ ПОШЕЛ-ДАЛЬШЕ

ВИДИТ ДОСТАЛ ПРЕОДОЛЕЛ ПОШЕЛ-ДАЛЬШЕ

ПРИШЕЛ

БИЛСЯ НЕ ПОБЕДИЛ БИЛСЯ НЕ ПОБЕДИЛ БИЛСЯ ПОБЕДИЛ

(ВЗЯЛ ВЕРНУЛСЯ)

В указанных в цепочке глагольных фреймах отсутствуют данные о действующих лицах и их типах, вместо которых стоят вопросы типа КТО?, КОГО?,

ЧТО?, С КЕМ? и т. п. Ответы на эти вопросы получаются при работе второй части программы (TALE-2), в ходе которой формируется развернутая и осмысленная структура сказки. Приведем пример одной из таких структур, полученных в результате работы системы TALE:

ОДНАЖДЫ В ОДНОМ ГОРОДЕ ЖИЛ ЦАРЬ
ЦАРЬ НЕ ИМЕЛ ДЕТЕЙ
ПОЕХАЛ ЦАРЬ НА ОХОТУ
ВСТРЕТИЛ ЦАРЬ БАБУ-ЯГУ
БАБА-ЯГА ДАЛА ЗЕЛЬЕ ПОЛЕЗНОЕ
ВЕРНУЛСЯ ЦАРЬ ДОМОЙ
РОДИЛСЯ ОТ ТОГО ЗЕЛЬЯ СЫН ИВАН-ЦАРЕВИЧ
ВЫРОС ИВАН-ЦАРЕВИЧ
УНЕСЛА БАБА-ЯГА ИВАНА-ЦАРЕВИЧА
ПРОЖИЛ ИВАН-ЦАРЕВИЧ У БАБЫ-ЯГИ НЕ ЗНАЮ-СКОЛЬКО
ИЗУЧИЛ ВСЕ КОЛДОВСТВО
НАДУМАЛ ИВАН-ЦАРЕВИЧ УИТИ ОТ БАБЫ-ЯГИ
БАБА-ЯГА ЗАДАЛА ЗАДАЧУ ДОСТАТЬ ПЕРО ЖАР-ПТИЦЫ
ИВАН-ЦАРЕВИЧ РЕШИЛ ЗАДАЧУ ДОСТАЛ ПЕРО ЖАР-ПТИЦЫ
БАБА-ЯГА ЗАДАЛА ЗАДАЧУ ДОСТАТЬ ДЕВИЦУ-ШАМАХАНСКУЮ-
ЦАРИЦУ
ИВАН-ЦАРЕВИЧ РЕШИЛ ЗАДАЧУ ПРИВЕЗ ЦАРИЦУ
БАБА-ЯГА ЗАДАЛА ЗАДАЧУ ДОСТАТЬ ПЕРСТЕНЬ ЦАРЬ-ДЕВИЦЫ СО
ДНА МОРСКОГО
ИВАН-ЦАРЕВИЧ РЕШИЛ ЗАДАЧУ И ПЕРСТЕНЬ ДОСТАЛ
ИВАН-ЦАРЕВИЧ ВЗЯЛ ЗАРАБОТАННОЕ
ВЕРНУЛСЯ ИВАН-ЦАРЕВИЧ ДОМОЙ

В приведенном примере мы имеем дело с достаточно примитивной, но вполне осмысленной структурой волшебной сказки. Последующие (не реализованные) части системы TALE (TALE-3 и TALE-4) позволяют расцветить получаемые структуры добавлением эпитетов, метафор и т. п.

Возможен иной подход к моделированию волшебных сказок [Haase-Report et al., 1984]. Идея этого подхода также основывается на исследованиях фольклора [Пропп, 1946; Ведерникова, 1975, 1980; Новик, 1975; Цивьян, 1975; Герасимова, 1978]. В соответствии с этими исследованиями действующие лица сказок живут в трех мирах, различающихся специфическими свойствами. Это — Реальный мир, в котором разворачивается экспозиция сказки, выдача Герою задания и встречи Героя с Помощниками или Дарителями; Иной мир, в котором действует Антигерой, хранится похищенная Награда и происходит встреча Героя с Антигероем и битва между ними, и третий, Промежуточный мир, в котором действует и живет страж Иного мира, каким, как правило, является Баба-яга. «География» всех этих трех миров достаточно бедная и характеризуется небольшим числом локусов — мест, где живут, встречаются и действуют персонажи сказки. Как правило, каждый локус определяется небольшим числом типовых описаний, из которых может быть выбрано описание для конкретной сказки.

Существенным отличием этой версии генерации от системы TALE, в которой персонажи наделялись только действиями, позволяющими разворачиваться сюжету, является наделение персонажей способностью совершать поступки, перемещаясь по системе локусов. Поступки персонажей, описываемые специальными фреймами [Поспелов Д. и др., 1980], могут содержать как постоянную часть, определяемую типом персонажа, так и переменную, ситуативную часть, зависящую от хода разворачивания сюжета.

Структура программы изображена на рис. 3.6. В блоке 1, являющемся начальной частью системы TALE, выбирается тип сюжета, система действующих лиц и формируется экспозиция. В блоке 2 хранятся ситуативные описания, связанные с персонажами, а в блоке 4 — постоянные характеристики дифференцированной системы персонажей. По информации блока 2 и текущей информации (блок 3) о каждом персонаже в данный момент времени формируются мотивы

и поступки персонажей. Информация в блоке 3 состоит из траекторий движений по локусам, отражающим предысторию поведения персонажей и данные о закономерностях, которым должна удовлетворять дальнейшая имитация поступков и передвижений в другие локусы. В блоке 5 хранятся фрагменты описаний локусов из всех трех миров сказки. Блок 6 осуществляет «сборку» порожденных в ходе имитации фрагментов сказки, выполняемую так же, как это было сделано в системе TALE.

Описанный подход близок к изложенному в [Yazdani, 1982]. Однако в этой работе сцепление фрагментов осуществляется по априорно заданным законам, мало отличающимся от продукций системы СТРИПС [Fikes et al., 1972]. Здесь же эта процедура управляется событиями, уже произошедшими при имитации перемещения по локусам персонажей и формировании поступков в этих локусах.

Стихотворные тексты принадлежат к числу наиболее изученных. Это позволяет предполагать, что их генерирование, даже с помощью слабых средств, может оказаться эффективным. Существенную роль здесь играет и неоднозначность восприятия таких текстов. Читатель как бы домысливает текст стихотворения, ориентируясь на его звуковую и эмоциональную направленность, и не всегда основывается лишь на конкретном содержательном смысле текста самого стихотворения. Типичными примерами таких стихотворений являются «автостихотворения», полученные стохастической программой [Lutz, 1960]. Остановимся на программе [Кац, 1978], которая показала, что из структурных составляющих стихотворения (грамматика, метрика, рифмы и семантика) для получения полной иллюзии осмысленного стихосложения достаточно первых трех составляющих. Семантика формируется из грамматики и воображения читателя. Выбор словаря не оказывает влияния на качество стихотворения и определяет лишь его настроение и тематику. Программа построена на упрощенной грамматике, когда предполагается, что каждая строка стихотворения состоит из одного подлежащего, одного сказуемого и нескольких определений и обстоятельств, причем каждый из перечисленных членов предложения может быть опущен.

Словарь состоит из нескольких сотен слов различного числа, рода и времени, разбитых на четыре части: 1) существительные и местоимения любого рода и числа, которые могут быть подлежащими; 2) прилагательные и притяжательные местоимения, представляющие собой набор возможных определений; 3) глаголы (непереходные), которые могут служить в предложении сказуемыми; 4) наречия и существительные с предлогами, которые могут быть обстоятельствами места, времени и образа действия.

Каждое слово словаря сопровождается информацией о метре, рифме и грамматике. В частности, указываются число слогов до и после ударения, значения рода, числа и времени, включая безразличные (нулевые). Рифмы используются только мужские и женские и определяются специальными таблицами.

В задании, которое получает машина, указывается число строк в каждой строфе, число слогов в строке и система рифмовки. Так, если нужно получить «онегинскую» рифму, то она задается в виде аБаБввГдЕЕдЖЖ, где прописными буквами обозначаются мужские, а строчными — женские рифмы и указы-

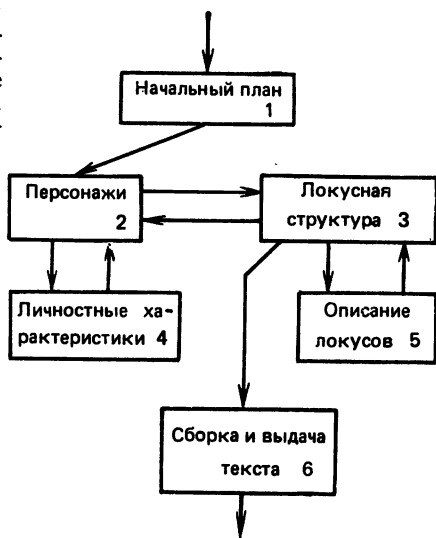


Рис. 3.6

вается, что строки с женскими рифмами содержат девять, а с мужскими — восемь слогов.

Работа программы начинается со случайного поиска рифмующихся слов с ударениями на последнем или предпоследнем слоге (в зависимости от вида рифмы). Найденная пара слов ставится на последние места в соответствующих строчках. После этого ищутся предыдущие слова.

При этом учитывается число слогов до и после ударения, чтобы выполнялся заданный размер. Затем проводится грамматический анализ, т. е. строке приписываются параметры — род, число и время строки, по которым выбираются последующие (справа налево) слова. При нарушении какого-либо из условий поиск повторяется. Таким образом заполняется каждая строка. При этом используются некоторые дополнительные правила. Так, если в односложном слове более трех букв, то оно, как правило, ставится на ударное место; если же написана почти вся строка и машина бракует все слова для последнего места, то на это место ставится какое-либо односложное междометие, союз или наречие. Например, на неудачном месте может оказаться «И», а на ударном «вновь».

В качестве примера стихов, сочиненных машиной, приведем полученное на базе словаря, почти полностью состоящего из слов сборника О. Э. Мандельштама «Камень» (изд. «Гиперборей», Пг., 1916):

Умирающий — в смятеньи.
Вновь, как тень, огни дрожат,
Вновь над бездною движения —
Где-то далеко — душа...
Крик смертельный рядом, зыбкий,
Тлели в хрустале глаза,
Шелест мечется с улыбкой,
Где-то в чаще небеса.

Сочинение одного редкого вида стихов имитирует программа «Скальд» [Кондратов и др., 1984; Кондратов, 1987]. Программа ориентирована на сочинение стихов типа скандинавских скальдических стихов [Стеблин-Каменский, 1978, 1979]. Несмотря на простоту содержания скальдических стихов, которые могут быть либо хвалебными (драпа), либо хулительными (нид), либо содержать изложение фактического события, эти стихи отличаются сложной и разнообразной системой рифмования, внутренней и обычной, аллитерациями и широким использованием своеобразной системы образов — кеннингов. Так, ворон мог представляться следующими кенningами: «гусь битвы», «оса трупов», «лебедь крови», «кукушка трупов» и т. п. Для корабля характерны кеннинги: «конь моря», «конь влаги», причем в таком двухсловном кеннинге вместо моря и влаги могли быть использованы и многие другие слова родственного содержания (фьорда, прибой, реки, волны, болота и т. п.), стоящие в родительном падеже.

Особенностью кеннингов является возможность многоступенчатого построения кеннингов от кеннингов. Так, кеннинг для меча мог иметь вид: огонь битвы или огонь треска скал, так как битва — это треск скал. Таким же образом формируются более длинные кеннинги, например, для воина это: «метатель огня व्यюги ведьмы луны коня корабельных сараев», где «конь корабельных сараев» — корабль, «луна корабля» — щит, «ведьма щита» — копьё, «вьюга копий» — битва, «огонь битвы» — меч и «метатель меча» — воин. Легко видеть, что процедура построения кеннингов допускает достаточно простую алгоритмизацию [Манин, 1979].

По сравнению с естественными скальдическими стихами в программе «Скальд» введены некоторые ограничения, например использование только односложных и двухсловных кеннингов. В основу программы положен словарь односложных слов русского языка, насчитывающий около 6000 слов. Каждое слово снабжено дополнительной информацией о его грамматической принадлежности, о том, может ли это слово выступать в роли кеннинга (с указанием при этом в качестве первого или второго слова кеннинга оно выступает), и о том, имеет ли это слово хулительный или хвалебный смысл. Слова словаря разбиваются на

10 тематических групп по основным темам стихов (муж, бард, меч, вран, щит, челн и др.). Развитая синонимия позволяет легко преобразовать стих о муже в стих о воине, мужчине и т. п., а стих о вороне в стих о вране и т. п. Для построения хулительного кеннинга часто бывает достаточным, чтобы лишь одно из двух слов кеннинга было хулительным (нид).

Перед началом работы программы заказчиком задается тема (одна из десяти), жанр (хулительный или хвалебный), объем стихотворения (количество двустиший), размер из 20 вариантов и один из пяти типов рифм. Для того чтобы стихотворение не превращалось в список кеннингов, предусмотрено введение по простым правилам других односложных частей речи (прилагательных, глаголов в повелительном наклонении и междометий), которым также приписываются индексы драпа и нид. Выбор слов стихотворения осуществляется по случайному принципу с последующей проверкой выполнения условий задания. Вид зачина и концовки выбирается из нескольких типовых случайным образом.

Дальнейшее развитие машинного моделирования процесса сочинения литературных произведений может идти путем: расширения областей творчества, подлежащих машинному моделированию (новые жанры текстов, например газетных заметок [Гвенциадзе, 1986], научных сообщений, анекдотов, пародий, афоризмов и т. п.); включения в разрабатываемые модели элементов «самообучения», использования предыдущего опыта функционирования этих моделей; включения в состав моделей диалога с человеком, обеспечивающего реализацию процесса совместной, кооперативной деятельности человека и машины при решении творческих задач (один из возможных путей решения такой задачи при сочинении коротких рассказов описан в [Mechan, 1980]); теоретической и экспериментальной проверки различных способов представления знаний о предметной области в машинных моделях творчества.

Краткая история моделирования музыкальных произведений

Машинной принято называть музыку, при порождении которой используется ЭВМ. Этот термин относится и к синтезированию звуков с помощью ЭВМ. Распространившийся в последние годы термин «компьютерная музыка» обозначает музыку, при сочинении и записи которой человеком используется музыкальный компьютер — новейший музыкальный инструмент с электронным синтезатором звука, управляемый персональной ЭВМ и снабженный фортепианной клавиатурой, видеоэкраном и устройством для печатания нот на бумаге.

Первые опыты сочинения машинной музыки появились в 1956 г. и были посвящены моделированию процесса сочинения музыки (в основном мелодий) традиционной структуры. Работы по использованию ЭВМ в музыке проводятся в нашей стране и за рубежом по следующим направлениям:

звукосинтез — синтез звуков (тембров), а также воспроизведение последовательности синтезированных звуков в соответствии с введенными в ЭВМ (закодированными) нотами, служащими своеобразным алгоритмом — системой указаний для музыканта;

анализ нотных текстов музыкальных произведений;

синтезирование нотного текста музыкальных композиций — собственно машинное сочинение музыки;

создание музыкально ориентированных языков программирования, предназначенных для ввода в ЭВМ музыкальной информации, ее машинной обработки и вывода результатов из ЭВМ, а также для составления машинных программ [Касслер, 1975];

ввод в ЭВМ музыкальных текстов для формирования банка музыкальных данных и вывод из памяти ЭВМ музыкальной информации в виде, удобном для непосредственного использования музыкантами-исполнителями или теоретиками [Зарипов, 1985].

Обзоры работ по этим направлениям приводятся в [Зарипов, 1965, 1971, 1983, 1984; Zaripov, 1979] (здесь даны и нотные примеры машинных композиций, полученные в экспериментах разных авторов), [Lincoln, 1970; Ludvova, 1975; Зарипов, 1988].

Известны три метода синтезирования музыкальных сочинений на ЭВМ.

1. *Метод марковских цепей.* В [Shannon, 1953] была предложена простая модель, дающая нотный текст на основе реализации процесса, описываемого цепью Маркова (аналогичная модель рассмотрена в [Добрушин, 1961] на примерах генерации фраз на русском языке). Однако любая модель явления не отражает в равной мере всех его свойств. Так и модель нотного текста, пригодная для исследования его статистических свойств, может быть неудовлетворительна для представления ее в качестве музыкального сочинения, которое должно кроме статистических обладать и другими характеристиками, свойственными первоначальному нотному тексту. Именно это и случилось с машинными моделями нотного текста, полученными методом марковских цепей ([Brooks et al., 1957; Olson et al., 1961; Havass, 1964; Kupper, 1967], см. в [Зарипов, 1971]). Метод марковских цепей не дает [Barbour, 1962] хороших результатов, так как традиционные мелодии не описываются цепью Маркова. В мелодии практически взаимосвязаны все ноты — они подчинены определенной логике развития. Этот метод предполагает локальную взаимосвязь нескольких (в большинстве экспериментов трех-четырех) соседних нот. При этом синтезируемая последовательность высот не зависит от ритма, ибо элементом марковской цепи является (во всех известных экспериментах) лишь высота ноты. Ни один вариант (последовательность высот) не может обеспечить аудиального сходства полученной мелодии (или ее фрагмента) с мелодиями-оригиналами, хотя фрагменты их высотных последовательностей совпадают.

2. *Структурный метод.* Основан на программировании правил композиции, которые удалось выявить при изучении человеческого творчества и сформулировать. Этим методом синтезируется одnogолосная и многоголосная, тональная и атональная музыка. Здесь выбор текущей ноты мелодии зависит от многих параметров — от предшествующей мелодической или ритмической фигуры, положения ноты на сильной или слабой доле такта, ладогармонической функции текущего такта (является или нет нота аккордовой). Практически все ноты мелодии взаимосвязаны некоторой корреляционной зависимостью, что, в частности, обеспечивается мелодическими структурами, организующими в мелодии повторение ритмических и мелодических фигур.

Первыми были опубликованы опыты по синтезированию мелодий под названием «Кнопка Берта» [Klein et al., 1956]. Четырехголосная «Сюита ИЛЛИАК» составлена из композиций, полученных на машине в каждом из четырех экспериментов. В первых двух частях учитывались простые правила учебной гармонизации, в последующих частях эти правила исключались, и по мере приближения к последней части музыка приобретала все более «нетрадиционный» характер [Hiller et al., 1957; Hiller, 1970].

В нашей стране первые машинные мелодии, названные «Уральскими напевами», получены в июне 1959 г. [Зарипов, 1960]. В них учитывалась структура, организующая повторность ритмических и звуковысотных последовательностей. В дальнейших экспериментах [Зарипов, 1963, 1965; Zaripov, 1969] вводились принципы синтезирования музыки, позволившие получить более качественные композиции, имитирующие мелодии классического типа.

В начале 60-х годов запрограммировано сочинение трехголосной атональной (додекафонной) музыки в стиле Шенберга [Gill, 1963]. Составлены программы для сочинения некоторых видов полифонической музыки (контрапункта) [Padberg, 1964; Степанов, 1967].

В [Зарипов, 1966, 1971; Zaripov, 1968] описан способ моделирования процесса решения учебных задач по гармонизации — этим имитируется сочинение четырехголосных хоралов. Составлена программа, выполняющая функции экзаменатора. Она указывает и исправляет ошибки в введенных в машину решениях задач по гармонизации, являясь прототипом обучающей системы в преподавании. В [Зарипов, 1967, 1971; Zaripov, 1979] описано сочинение песенных мелодий на заданный машине стихотворный ритм. В 1972 г. на основе принципа переноса инвариантных отношений (заимствования исходной ситуации)¹ была составлена программа для варьирования заданной мелодии — темы. Этим ими-

тировался неосознаваемый в композиторской практике процесс «творческой обработки» заимствованных мелодий [Зарипов, 1983].

В [Аксельруд, 1975] рассмотрено сочинение тональных мелодий в форме свободных джазовых импровизаций, появляющихся в результате определенного вида варьирования некоторой заданной мелодии — темы.

Один из модифицированных вариантов программы ABIOMELOS, созданной для сочинения музыки классического типа, при вводе в ЭВМ музыкальной темы выдает ее вариации с аккомпанементом, предназначенные для исполнения на фортепиано [Gagliardo, 1979].

В [Зарипов, 1984, 1987а, б] мелодия составляется не из отдельных звуков, а из более крупных элементов мелодии — интонаций (попевок, мелодических оборотов), которые выбираются из частотного словаря интонаций (аналога интонационной памяти композитора). Это имитирует процедуру заимствования [Зарипов, 1988].

3. *Метод заготовок.* Перебор возможных вариантов (и, следовательно, их сочинение) занимает большое место в различных видах творчества [Зарипов, 1983]. Мысль о применении ЭВМ для создания заготовок этих вариантов не раз высказывалась в нашей печати, например в поэтическом творчестве для составления заготовок рифм [Кондратов, 1963; Соболев, 1963]. В композиторской практике ЭВМ оказалась удобным помощником для заготовки фрагментов музыкальных сочинений (звуковысотных сочетаний, аккордов, тембров, последовательностей длительностей нот и т. п.). Сочинение таких заготовок «вручную» требует много времени кропотливого, не творческого труда. Уже в [Olson et al., 1962] отрывки машинной последовательности нот были использованы для составления мелодии. Из заготовок, полученных ЭВМ и удовлетворяющих определенным условиям, композитор отбирает наиболее отвечающие его творческому замыслу и включает их в свое сочинение уже без помощи машины.

Машинные заготовки пригодны и при сочинении музыкальных композиций классического стиля [Gagliardo, 1979]. В этой работе для исследования возможности сотрудничества человека и машины был выполнен математический анализ григорианских хоралов и с помощью программы ABIOMELOS получены после отбора и корректировок имитации этих хоралов.

Сложность выявления закономерностей классической музыки привела к переходу от идеи моделирования музыки к значительно легче осуществимой идее синтетизирования музыки, которая существенно отличается от традиционной. Она образуется по правилам, вводимым композитором. С помощью ЭВМ создается огромное количество музыкальных произведений конструктивистских направлений. Идеи и машинные эксперименты такого рода описаны в [Barbaud, 1966; Аствацатрян, 1977; Ксенакис, 1986] и в обзорах [Hiller, 1970; Моль, 1975].

Процесс машинного сочинения музыки

Мелодия — наименее осознаваемый элемент музыкального творчества. Ее фрагменты как бы сразу возникают в сознании композитора. Это делает мелодию удачным материалом для исследования и выявления неосознаваемых принципов интеллектуальной деятельности человека. Существенным этапом творчества является заимствование интонаций из ранее слышанных произведений — иногда довольно большой протяженности. Такое заимствование осуществляется не цитированием конкретных интонаций, а их варьированием или трансформацией разных сторон мелодии: ритма, лада, регистра, интервальных отношений и т. п. При этом сохраняется полиинвариантная структура, служащая носителем образа сходных между собой на слух, но разных интонаций [Зарипов, 1987а]. Таким инвариантом является интервально-метрическая структура нотной интонации $S^n = S^1/I^1 S^2/I^2 \dots S^{n-1}/I^{n-1}$, где I — интервал, определяемый числом ступеней между высотами соседних звуков ($I=0$ — прима, $I=1$ — секунда, $I=2$ — терция и т. д.), S — двунотная стопа, вычисляемая по соответствующей процедуре и принимающая два значения: $+$ или $-$. При $S=+$ образуется хорейческая стопа, от сильной ноты к слабой, при $S=-$ — ямбическая, от слабой ноты к сильной. Другие структуры интонации см. в [Зарипов, 1988].

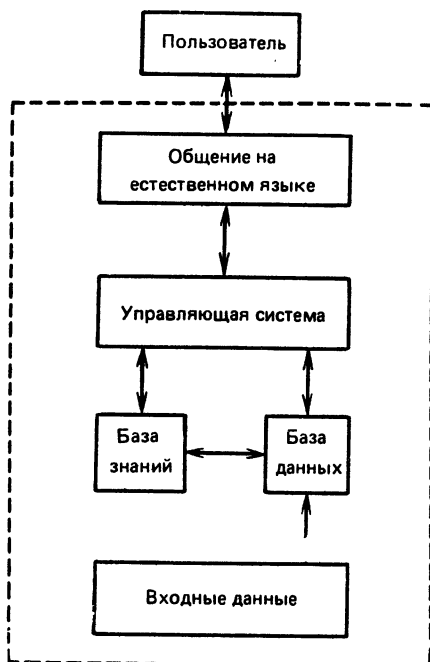


Рис. 3.7

«вхождения» алгоритма в множество конкретных мелодий. Отметим способ «сцепления» заимствованных интонаций. Они соединяются не «встык», а как бы вытекающая одна из другой — первые t элементов второй интонации совпадают с последними t элементами первой. Процесс сцепления по этой схеме вложенных интонаций одинаковой протяженности хорошо описывается марковской цепью SI -элементов. Таким способом строится не вся мелодия (поскольку традиционная мелодия не является марковской цепью ее элементов), а лишь ее однотактовые основные мелодические фигуры, которые включаются в мелодию в соответствии с так называемой масштабно-тематической структурой, предусматривающей повторность, в частности, двухтактовых ритмических и однотактовых мелодических фигур, ладогармонических и других функций и расчленение восьмитактовой мелодии на одно-, двух-, и четырехтактные построения [Зарипов, 1963, 1971, 1983, 1987; Zaripov, 1969, 1979].

Описанные принципы положены в основу программы порождения мелодий из интонаций (рис. 3.7). Для этой программы *входные данные* — массив мелодий, вводимых в ЭВМ для анализа; *база данных* — алфавит (нотные знаки) и данные анализа мелодий из блока выходных данных (частотный словарь интонаций); *база знаний* — множество всех запрограммированных правил. Их перечнем является упорядоченная последовательность параметров p_k : $M = (p_1, p_2, \dots, p_k, \dots)$, где p_k принимает несколько значений. Используется продукционная форма правил: «Если имеет место ситуация A и выполняется условие P , то с вероятностью q следует B , иначе C ».

Управляющая система формирует типовую структуру (фрейм) $M_c = (q_1, q_2, \dots, q_k, \dots)$, где q_i — одно из значений параметра p_i . Таким образом, из всего множества запрограммированных правил типовая структура M_c предусмат-

Займствование и «творческая обработка» (варьирование, сопровождающееся вводом новых, маскирующих элементов) — две глобальные процедуры, лежащие в основе любой творческой деятельности. Суть «творческой обработки» состоит в преобразовании описания заданной (займствованной) ситуации с сохранением ее инварианта (фрейма-классификатора). Творец — композитор, поэт или математик — не осознает глубинного механизма варьирования (как и займствования). Машинное варьирование позволяет шаг за шагом проследить весь путь преобразования, показывая его промежуточные этапы. Путем машинной имитации удалось вскрыть сущность займствования и «творческой обработки», формализовать эти процедуры и объяснить многие феномены, присущие творческой деятельности человека (например, показать механизм преобразования темы в вариацию на примере трансформации мелодии русской народной песни «По Дону гуляет казак молодой» в мелодию «Молодежной» И. О. Дунаевского [Зарипов, 1983]).

Проблема получения благозвучных сочетаний решается автоматически, поскольку мелодия синтезируется из интонаций, естественных для человеческого слуха. Для сочинения музыки определенного стиля разработан метод «погру-

ривает лишь некоторое его подмножество, которое и участвует в синтезировании текущей мелодии. Здесь же производится выбор элементов из базы данных, синтез фрагментов мелодии методом марковских цепей и их сцепление в соответствии с масштабно-тематической структурой. При взаимодействии правил, противоречивых на данном шаге, во избежание заикливания система правил автоматически меняется.

Обобщение на естественном языке — представление машинных результатов: мелодии с указанием различных параметров, ее типовой структуры M_c и комментариев (системы объяснений).

Имеется немало других систем, использующих ЭВМ для изучения процесса сочинения музыкальных произведений человеком. Примером может служить разработанная на материале армянской народной песни аналитико-информационно-поисковая система народной песни для хранения, коррекции, поиска и автоматической обработки данных с помощью ЭВМ [Гаспарян, 1981]. Не менее интересен частотный словарь интонаций [Зарипов, 1984]. С его помощью можно разрешить практически важные проблемы, одна из которых — автоматическое распознавание и поиск в некотором множестве мелодий (например, в сборниках песен советских композиторов) всех сходных между собой на слух мелодических фрагментов любой протяженности [Бахмутова и др., 1985; Зарипов, 1986]. Получить такой результат без ЭВМ — только по памяти и слуху — практически невозможно.

В настоящее время существуют способы кодирования нотных партитур достаточной сложности [Касслер, 1975]. Однако серьезным препятствием для ввода нотных массивов больших объемов является отсутствие реально действующей системы оптического считывания общепринятой нотации, а также автоматического преобразования печатного нотного текста в информацию на магнитных или иных носителях, пригодную для ввода в ЭВМ [Зарипов, 1985].

3.5. Когнитивная компьютерная графика

А. А. Зенкин

Суть проблемы

Средства интерактивной компьютерной графики (ИКГ) являются новым каналом общения между человеком и ЭВМ. ИКГ реализует две связанные между собой функции: иллюстративную и когнитивную (рис. 3.8). Одновременный вывод ИКГ-изображений в разные окна на экране дисплея создает у пользователя синтетический полиэкранный ИКГ-образ. Иллюстративная функция обеспечивает визуальную адекватность графического образа оригиналу, т. е. визуальную «узнаваемость» оригинала. Для этого необходимо, чтобы оригинал существовал объективно, «вне и независимо от нашего сознания». Когнитивная функция позволяет (при определенных условиях) изображать в наглядной графической форме внутреннее содержание оригинала [Пилиugin и др., 1985]. Удачный рисунок не только делает наглядной и понятной суть сложного вопроса, но и способен подсказать принципиально новое соображение, идею, гипотезу. Примером целенаправленного использования когнитивной функции ИКГ являются «логические круги» Леонарда Эйлера, с помощью которых в наглядной форме представляются все основные операции над множествами и классами и происходит визуализация силлогистики Аристотеля.

Отметим ряд особенностей когнитивной ИКГ: когнитивная ИКГ является средством прямого целенаправленного воздействия на интуитивные, образные механизмы мышления; динамизм ИКГ (например, в режиме мультимпликации) «включает в работу» специальные процедуры, эффективно отслеживающие динамику изменения ИКГ-образов во времени, т. е. процедуры обнаружения инвариантов и тенденций изменения ИКГ-изображений, а поиск таких инвариантов и тенденций является главной задачей любого и прежде всего абстрактно-математического научного исследования [Бурбаки, 1963]; когнитивная ИКГ позволяет

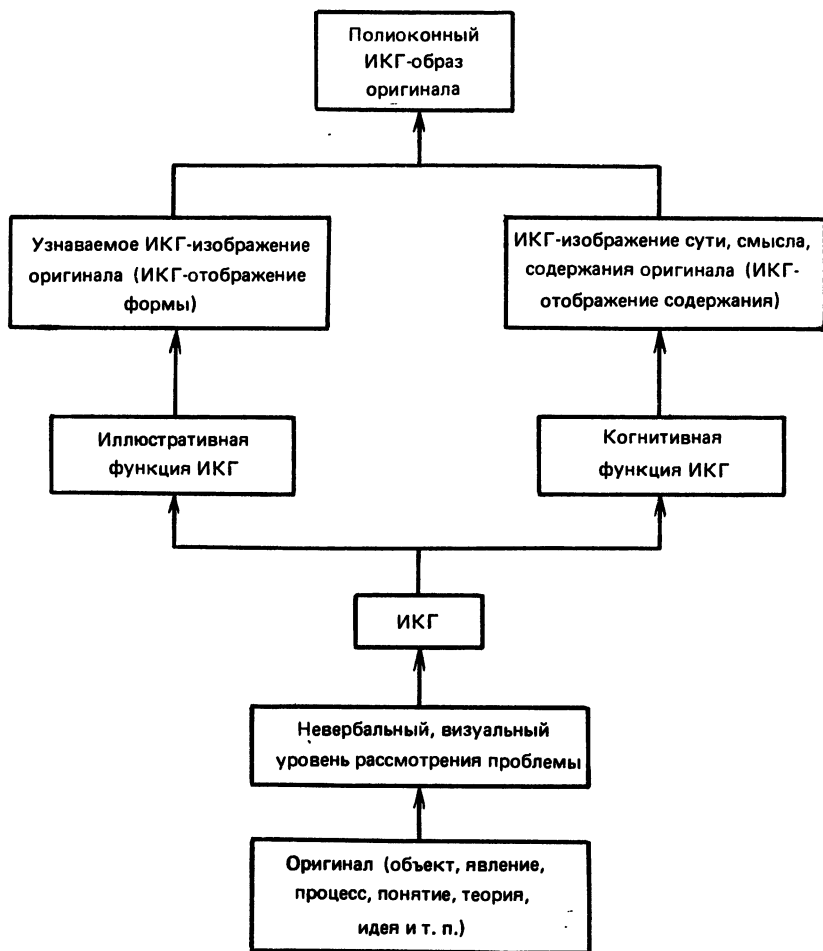


Рис. 3.8

визуализировать содержание проблемы, которая интересует исследователя, т. е. реализовать наиболее активную форму ИКГ-общения исследователя с проблемой; в этом случае ситуация настраивает исследователя на активный поиск нового знания с участием высших творческих механизмов его мышления.

В связи со сказанным становятся актуальными следующие задачи: разработка архитектуры когнитивных ИКГ-систем; разработка методологии применения когнитивных ИКГ-систем для интенсификации процесса научного творчества и создания качественно нового научного знания; использование когнитивных ИКГ-систем для экспериментального исследования высших механизмов интуитивного, образного мышления в условиях реального процесса творческого научного поиска.

Своеобразную «экологическую нишу» в проблеме когнитивной визуализации абстрактных понятий науки составляет направление исследований, находящееся на стыке науки и изобразительного искусства и связанное с представлением

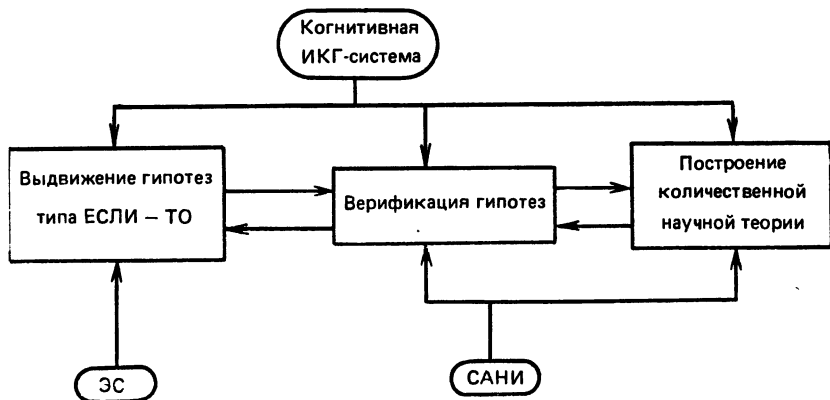


Рис. 3.9

внутреннего содержания абстрактных научных понятий средствами художественной графики. Примерами такого подхода могут служить работы голландского художника М. К. Эшера, который визуализировал идею геометрического порядка и принципа симметрии, и работы советского тополога А. Т. Фоменко [Фоменко, 1981], в которых визуализируется когнитивная суть абстрактных математических понятий. Нисколько не умаляя эстетического, художественного и научного значения этого направления, следует подчеркнуть, что здесь между смыслом, содержанием абстрактного научного понятия и его художественным образом стоит субъект — личность автора. В случае когнитивной ИКГ таким субъектом, наполняющим ИКГ-образ конкретным математическим содержанием, является исследуемая научная проблема. В настоящее время известны два вида систем представления, переработки и использования знаний: экспертные системы (ЭС) и системы автоматизации научных исследований (САНИ). Рассмотрим «движение» от незнания к знанию (рис. 3.9). Очевидно, что такой процесс является непрерывным, итерационным и диалектическим. Все его этапы тесно связаны между собой. Но отношение точных наук и науки об искусственном интеллекте к этим этапам до последнего времени различное. Точные науки основные усилия сосредоточивают на этапах проверки гипотез и построения количественных научных теорий, наука об искусственном интеллекте большое внимание уделяет этапу выдвижения гипотез. Все шире распространяются ЭС, в которых постепенно накапливаются знания о проблемной области и формируются гипотезы о существующих в ней закономерностях. Возникает возможность объединения двух подходов: основанного на качественных моделях и используемого на этапе формирования гипотез и теорий; основанного на количественных моделях, во многом характерных для ныне существующих САНИ. Объединение ЭС и САНИ создает инструмент мощной интеллектуальной поддержки для исследователя. Такая *гибридная технология* и служит основой для ИКГ-систем [Поспелов Г. и др., 1985; Поспелов Г., 1988].

Архитектура когнитивных ИКГ-систем

Общая схема интегрированной проблемно-ориентированной когнитивной ИКГ-системы приведена на рис. 3.10. Охарактеризуем ее основные элементы.

1. КП — конечный пользователь за сетевой ПЭВМ.
2. Диспетчер обеспечивает управление интерактивным доступом КП к системе
3. Основные режимы работы ИКГ-системы:

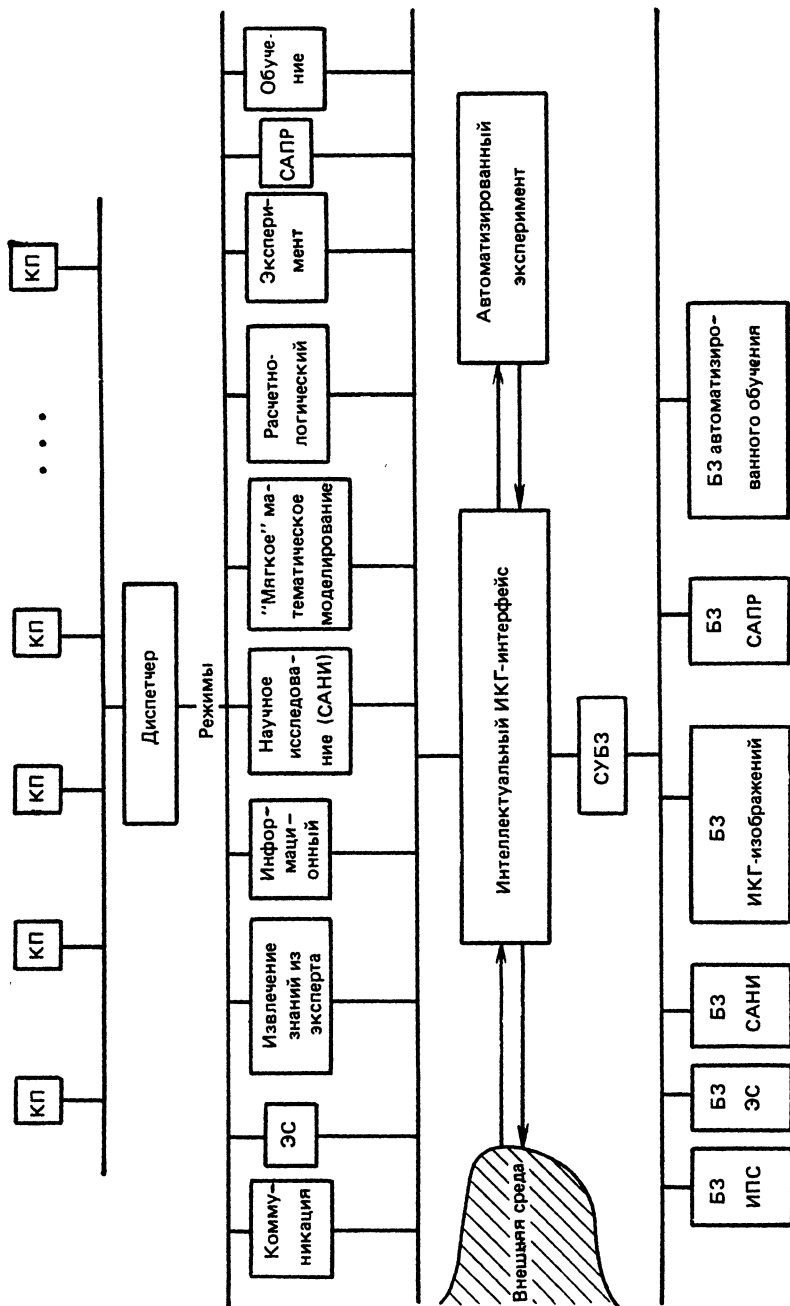


Рис. 3.10

коммуникация: обеспечение оперативной связи КП друг с другом и с внешними системами на уровне теледоступа;

ЭС: принятие решений в рамках работы с традиционной ЭС;

извлечение знаний из эксперта: формирование и наполнение базы знаний (БЗ) ЭС в режиме коллективного экспертного анализа ситуаций из проблемной области, представленных в форме ИКГ-изображений на дисплее ПЭВМ;

информационное обслуживание: работа с интеллектуальной информационно-поисковой системой по формированию и наполнению индивидуального банка фактографических и экспериментальных данных по вопросу, интересующему конкретного КП, а также поиск и анализ аналогий и прецедентов в других БЗ;

научное исследование: решение с помощью САНИ так называемых обратных задач научного исследования (*«жесткое» математическое моделирование*);

«мягкое» математическое моделирование: построение, конструирование и верификация новых научных теорий;

расчетно-логический: решение прямых (вычислительных) задач научного исследования на базе соответствующего интеллектуального пакета прикладных программ;

эксперимент: оптимальное управление экспериментом, сбор и первичная обработка экспериментальных данных;

САПР: поиск новых технических решений и интерактивное ИКГ-проектирование технических систем и технологических процессов;

обучение: интеллектуальное автоматизированное преподавание (обучение) типа проблемно-ориентированных систем автоматизированного обучения (CAO), основанных на единой ИКГ-методологии.

4. Внешняя среда: телекоммуникационные связи с другими когнитивными ИКГ-системами, системами коллективного пользования, суперЭВМ, интегрированными базами данных общего назначения, промышленными и технологическими.

5. Интеллектуальный ИКГ-интерфейс: управление ИКГ-общением КП с исследуемой проблемой во всех указанных режимах.

6. Автоматизированный эксперимент: управление экспериментальными установками и каналами обмена экспериментальной информацией с базами знаний в реальном масштабе времени.

7. СУБЗ: управление интегрированным комплексом взаимосвязанных БЗ.

8. Единая (интегрированная) база знаний: взаимосвязанные специализированные БЗ, которые, с одной стороны, благодаря единой СУБЗ максимально открыты и доступны для всех КП и друг для друга, а с другой — сохраняют автономию и специфику своих сегодняшних прототипов (БЗ ЭС, БЗ САНИ, БЗ САПР и т. д.). Эта единая БЗ с системной точки зрения повышает эффективность когнитивной ИКГ-системы. Структура баз знаний ЭС, САНИ и ИПС близка к традиционной; БЗ САПР — система математических моделей технических объектов и технологических процессов, экономических моделей, нормативов, ТЗ и т. д.; БЗ автоматизированного обучения — иерархическая система обучающих программных модулей; БЗ ИКГ-изображений — база знаний на основе ИКГ-изображений объектов, их свойств и отношений, а также постулатов и утверждений данной проблемной области.

Все подсистемы базы знаний и все режимы работы когнитивной ИКГ-системы ориентированы на широкое использование возможностей когнитивной ИКГ на всех этапах интерактивного процесса принятия решений в научных исследованиях.

Когнитивный ИКГ-язык

Естественный язык является универсальным в том смысле, что с его помощью можно записать, представить, зафиксировать в буквенно-цифровой форме любую информацию (даже графическое изображение или музыкальное произведение, хотя такое описание не всегда будет адекватным и эффективным). Иначе обстоит дело в случае графических форм представления информации. В [Александров и др., 1987] показано, что большинство баз видеоданных отно-

сится к специализированным или предметно-ориентированным системам (см. гл. 5), что объясняется сложностью задач использования и обработки видеoinформации. По-видимому, данная специфика вытекает и из природы графической формы представления информации. В связи с этим можно утверждать, что ИКГ-системы являются эффективными лишь в случае их априорной проблемной ориентации, хотя проблемная область не обязательно должна быть слишком узкой.

Сформулируем на содержательном, эвристическом уровне общие принципы подхода к построению проблемно-ориентированных когнитивных ИКГ-языков. Пусть A — некоторое множество аксиом-высказываний о свойствах объектов некоторой выделенной проблемной области D и U — множество всех высказываний, связанных с элементами-высказываниями из A , но не обязательно строго логически выводимых из A . Если найти способ визуализации A в виде конкретных ИКГ-изображений отношений, реализующихся в системе объектов из D , и зафиксировать способы трансформации (преобразования) этих ИКГ-изображений, то вместе с A могут быть визуализированы некоторые высказывания из U , среди которых могут оказаться и такие, которые, с точки зрения специалиста по проблемной области D , представляют собой принципиально новое знание. Способ визуализации A в виде специфических ИКГ-изображений и система правил преобразования этих ИКГ-изображений составляют *когнитивный ИКГ-язык* для проблемной области D с системой аксиом-высказываний A . Если взять другие D или A , то скорее всего получится другой когнитивный ИКГ-язык.

ИКГ-изображения и их упорядоченные последовательности (мультфильмы) формируют локальную ИКГ-БЗ для области D , аксиоматики A и выбранного ИКГ-языка. Затем с помощью (коллектива) экспертов по проблемной области D конструируется алгоритм интерпретации («расшифровки») ИКГ-изображений или, что то же, создается БЗ ЭС с продукциями типа:

ЕСЛИ <данное ИКГ-изображение содержит такой-то графический фрагмент, элемент, особенность, деталь, штрих, нюанс и т. п.>,

ТО <возможно, что справедлива такая-то содержательно значимая гипотеза-высказывание из U >.

Процесс ИКГ-генерации нового знания можно схематично описать следующей последовательностью основных этапов:

1) создается новое (или вызывается старое из соответствующей БЗ) ИКГ-изображение; 2) это ИКГ-изображение подвергается любым допустимым преобразованиям; 3) в каждом получаемом таким образом ИКГ-изображении визуально отыскиваются графические особенности; 4) эти особенности ИКГ-изображений анализируются с помощью продукций из БЗ ЭС и строится множество консеквентов этих продукций — содержательно-значимых гипотез-высказываний из U ; 5) полученные гипотезы-высказывания анализируются с точки зрения их значимости и новизны посредством, например, поиска по ключевым словам аналогов и прототипов в БЗ ИПС; 6) если обнаруживается новая содержательно-значимая гипотеза из U , то поиск (конструирование) доказательства этой гипотезы реализуется также с помощью когнитивного ИКГ-подхода.

В настоящее время высокоэффективные диалоговые и графические системы являются штатными средствами ПЭВМ. В ближайшем будущем интеллектуальность ИС будет определяться не столько обязательным наличием диалоговых или графических процессоров, сколько эффективностью их применения для решения сложных, нетривиальных, интеллектуально-трудных проблем.

Подавляющее большинство систем, реализуемых сегодня на ПЭВМ, широко используют ИКГ. В любом, даже чисто иллюстративном, применении ИКГ всегда можно обнаружить проявление когнитивной функции графики [Пилюгин и др., 1985; Федосов, 1986]. Однако ИС, в явном виде ориентированными на использование когнитивной функции ИКГ для генерации качественно нового знания, т. е. собственно когнитивными человеко-машинными ИКГ-системами, являются на сегодняшний день две системы: ДСТЧ для исследования проблем аддитивной теории чисел (проблемно-ориентирована на «половину» теории чисел) и

ДИАХИМ-86 для исследования химико-технологических проблем (проблемно-ориентирована на «всю» химию) [Зенкин, 1984, 1987] (см. § 9.3 кн. 1 данного справочника).

Принцип единства науки, производства и образования, который закладывается в качестве основы перестройки высшей школы, отражает диалектическое единство технологии «расширенного воспроизводства» человеческого знания: от его создания (наука), через массовое тиражирование в индивидуальных естественных интеллектах учеников (образование) к овеществленным формам знания (производство), к созданию «прибавочной стоимости» знания в виде опыта его применения как в образовании, что предполагает классификацию, систематизацию и структуризацию знания, так и в производстве, что обеспечивает широкомасштабную верификацию адекватности знания практике. Поэтому гипотетическая ИС должна позволять одновременно: создавать новое знание на уровне САНИ, применять знание на уровне САПР и АСУП, извлекать знание из получаемого при этом опыта на уровне ЭС и сразу внедрять это новейшее знание в учебный процесс на уровне систем автоматизированного обучения.

Огромный прогресс, достигнутый в области искусственного интеллекта за последнее десятилетие, связан прежде всего с реализацией международных и национальных программ создания ЭВМ V поколения. При этом наиболее значительными были успехи в двух направлениях: создания качественно нового поколения хардвее (суперЭВМ, сети ПЭВМ, теледоступ к международным БЗ, оптоэлектроника и т. п.) и интеллектуализации машин на базе в основном алгоритмических метапроцедур левополушарного мышления. Ближайшей задачей является освоение «мощности» второй половины человеческого мозга — метапроцедур его образного, интуитивного, творческого правополушарного мышления.

3.6. Компьютерные игры

Г. Г. Гнездилсва, О. А. Гончаров, Г. В. Сенин

Типы игр

Игровые программы, написанные для компьютеров, можно разделить на три класса: собственно игры, игрушки и живые картинки. В процессе игры человек должен достичь некоторой заранее определенной цели. Достижение этой цели обычно называют выигрышем. Традиционный пример игры — шахматы, цель которой — поставить мат королю соперника. В отличие от игр игрушки не ставят перед играющим определенной цели, но предоставляют ему возможности для манипуляций с ними. В качестве примера традиционных игрушек можно привести обычные детские автомобили. Примером этого класса компьютерных игр могут служить игры «Жизнь» и «Паучья графика».

Игра «Жизнь» моделирует жизнь поколений гипотетической колонии живых клеток, которые выживают, размножаются или погибают в зависимости от того, сколько соседей из восьми возможных оказывается у клетки. Так, если у клетки один сосед или ни одного, она погибает «от одиночества», если клетка имеет четырех или более соседей, она погибает «от перенаселения» и т. д. Размещив в начале игры клетки исходной популяции, игрок созерцает эволюцию колонии. Существуют конфигурации статические, повторяющиеся, перемещающиеся в определенном направлении. Поражает то, что простейшие правила «жизни» в зависимости от начальных условий порождают разнообразные комбинации, которые невозможно предсказать. Иногда это выглядит как довольно сложное «поведение» колонии и вполне заслуживает названия «Жизнь». Играющий в «Паучью графику» порождает узор на экране, управляя перемещением «паука». Построенная картина отчасти напоминает паутину.

К рассматриваемому классу игр, по-видимому, относятся диалоговые фильмы. Эти фильмы демонстрируются на видеоматрифононе, подключенном к персональным ЭВМ (ПЭВМ). В ходе демонстрации зритель имеет возможность

вмешиваться в ход развития сюжета с помощью соответствующих команд, подаваемых с ПЭВМ. Чтобы обеспечить эту возможность, сюжет фильма разветвляется на отдельные подсценарии, которые переключаются командами непосредственно в процессе сеанса.

Живые картинки — вид развлечения, бурно расцветший после появления у компьютеров графических терминалов. В живых картинках от человека не требуется почти никакой деятельности. Нужно только смотреть. Прототипами живых картинок можно считать различные виды калейдоскопов.

Перечислим несколько программ из [Гнездилова и др., 1988]. Программа «Пять кривых» строит на экране кривые, хорошо известные любителям математики: спираль Архимеда, улитку Паскаля, кардиоиду, трилистник и четырехлистник. Рисунок, вычерчиваемый на экране программой «Убегающий квадрат», образован вложенными квадратами. Вершины каждого следующего квадрата делят стороны предыдущего в заданном отношении, тем самым квадраты не только становятся все меньше и меньше, но и поворачиваются на некоторый угол. Узор, который рисует программа «Кружева», образован отрезками, соединяющими каждую вершину правильного восемнадцатиугольника со всеми другими его вершинами. НЛО, созданный программой «Прыгающий НЛО», появляется то в одной позиции экрана, то в другой, изредка издавая звуковой сигнал и меняя цвет при каждом «прыжке».

Собственно игры можно подвергнуть дальнейшей классификации и выделить: логические игры; игры на «везение»; игры на «ловкость»; обучающие и тренирующие игры.

Логические игры близки к традиционным математическим загадкам, головоломкам, упражнениям, но облеченным в игровую форму. Особенность этих игр заключается в том, что вам и вашему сопернику (соперникам) предоставляется возможность выбора из нескольких вариантов: нужно принимать решение. Примерами таких игр могут быть «Ним» и «Коровы и быки». Как и все логические игры, эти игры требуют не столько скорости мышления, сколько его глубины и точности, умения анализировать варианты. Логические игры наиболее приемлемы для людей, тяготеющих к сосредоточенной мыслительной работе, с медленным, но глубоким складом ума.

Компьютерные логические игры могут быть разделены на две группы в зависимости от того, какие функции возлагаются в игре на компьютер. В играх первой группы основное назначение компьютера — поддержка необходимой игровой обстановки. Компьютер порождает исходную ситуацию и организует диалог с игроком, обеспечивая ему возможность сделать очередной ход. Обязанностью компьютера является контроль правильности сделанного хода и соответствующее изменение игровой ситуации. От него требуется также слежение за тем, достигнута ли целевая ситуация. В играх второй группы компьютер помимо всех действий, выполняемых им в играх первой группы, должен еще делать игровые ходы. В этих случаях перед программистом встает проблема, как запрограммировать игру компьютера, как выбрать стратегию, которой тот должен придерживаться.

Назовем четыре наиболее употребительных способа программирования компьютера.

1. Оптимальная стратегия, т. е. стратегия, ведущая к победе или, если таковая невозможна, к минимальному проигрышу. Такую стратегию можно реализовать только для тех игр, для которых она известна и вычислима за приемлемое время. Примером игры с хорошо изученной выигрышной стратегией является «Ним».

2. Перебор вариантов. Компьютер перебирает возможные варианты продолжения игры и выбирает лучший. Сила переборных программ определяется тем, на сколько ходов вперед может «заглянуть» такая программа.

3. Перебор с применением эвристик. При переборе вариантов ходов используются некоторые соображения (эвристики), специфические для данной игры, позволяющие отбросить заведомо «плохие» ходы и выбрать «хорошие». Проблема в том, чтобы придумать эвристики. Иногда это удается.

4. Вероятностная стратегия. Компьютер делает очередной ход наугад. Эта стратегия наиболее проста в реализации и часто оправдывает себя в тех играх, где число допустимых ходов невелико и вероятность случайного выбора оптимального хода высока.

Только первая стратегия употребляется в чистом виде. Остальные, как правило, комбинируются друг с другом. Например, вероятностную стратегию можно сочетать с использованием эвристик: если эвристика применима в данной ситуации, то компьютер делает ход в соответствии с ней, иначе ход выбирается наугад.

Игры «на везение» — это азартные игры: рулетка, очко и т. п. Для подобных игр большого мастерства не требуется, достаточно благосклонности фортуны. Примером компьютерной игры «на везение» может служить игра «Фруктовая машина». Игроку в нее так: игрок делает ставку, компьютер наугад выбирает три фрукта. Ставка либо теряется, либо возвращается к игроку, увеличенная в несколько раз, в зависимости от того, какие именно фрукты были выбраны компьютером.

Игры «на ловкость» образуют практически бесконечную совокупность игр в придуманной, искусственной среде с заданными закономерностями. Игры на ловкость очень напоминают обычные подвижные игры, где надо «успеть», «додумать», «убежать», «поймать» и т. д. — скоростью в них основное. Примером служит игра «Не пересекай». По экрану движутся две ленты. Каждый из двух игроков управляет лентой своего цвета, которая в отсутствие управляющих сигналов движется в одном направлении, и поворачивает направо или налево по указанию игрока. Игра заканчивается, когда один из участников «врезается» своей лентой в границу игрового поля или в ленту противника.

Игры, тренирующие и обучающие, правильнее определить как обучение или упражнение в игровой обстановке. Тренирующие игры способствуют отработке навыков, которыми игрок уже обладает. Примером игр-тренажеров могут служить «Тренировка в счете» и «Мешанина». Первая использует вычислительные способности компьютера и имеет все основания заменить учителя-человека, для которого проверка арифметических знаний ученика является удручающе однообразной работой. Ученику предлагается избрать одно из арифметических действий, в котором он будет практиковаться: сложение, вычитание, умножение или деление. Далее программа выбирает случайным образом числа из заданных обучаемым диапазонов и использует их как операнды в арифметическом действии. Обучаемый должен правильно назвать результат. Проверив правильность ответа, программа выдает соответствующее уведомление и повторяет упражнение.

Обучающие игры помогают приобрести новые знания и навыки, полезные в практической деятельности. Например, игра «Анаконда» помогает изучать клавиатуру пишущей машинки. В этой игре змея, состоящая из алфавитно-цифровых символов, гонится за своим хвостом. На хвосте сидит человек, задача которого уничтожить змею, отщипывая от нее символ за символом. Если скорость, с которой человек нажимает на соответствующие клавиши, мала, то змея, постепенно удлиняясь, догоняет свой хвост и уничтожает человека. Игра «От одного до десяти» помогает малышу запомнить, в каком порядке идут числа от одного до десяти. Вопрос, который каждый ход предлагает программа, выглядит так: «Какое число идет за N?» (N — любое число от 1 до 10). Даже если ребенок пока еще не умеет читать, сыграв с вашей помощью в эту игру несколько раз, он легко научится понимать, что у него спрашивают. Правильный ответ награждается призом — красочной картинкой. Неправильный — влечет повторное задание того же вопроса.

Подклассом обучающих игр являются развивающие игры. Свое название они заслужили благодаря тому, что эффективно развивают наиболее важные и фундаментальные способности и навыки, такие как сообразительность, умение сосредоточиться, умение мыслить логически, пространственное воображение, умение искать закономерности, память.

Одной из первых игр, ориентированных на реализацию способностей к поис-

ку закономерностей, является «Приключение», сделанная в Массачусетском Технологическом Институте (США). В этой игре полный перечень и поведение игровых объектов не сообщаются игроку. Он сам в процессе игры должен выяснить, сколько их, как их найти и как с ними бороться. «Цель игры — исследование Пещеры и поиск ценностей, спрятанных там. Пещера полна опасностей. Гигантская змея, тролль и дракон препятствуют игроку на его пути. Банда кровожадных тварей практикуется в метании ножа в его незащищенное тело. Пират-клептоман постоянно ворует у него найденные им драгоценности. Но хуже всего то, что его лампа выгорает по прошествии определенного времени, а исследование пещеры в темноте — самоубийство». [Johnson, 1983].

Организация игр

Будем рассматривать *пошаговые игры* и *игры реального времени*. Пошаговые игры состоят из последовательности дискретных шагов (ходов). Время между ходами не влияет на течение и исход игры. Примером пошаговых игр являются компьютерные варианты традиционных настольных игр. В играх реального времени, аналогом которых служат подвижные игры, отдельные ходы, как правило, не рассматриваются. В играх реального времени играющий управляет поведением объекта в игрующей среде, постоянно оказывая на него управляющее воздействие (скачущая по ипподрому лошадь делает очередной прыжок только по приказу, убегающий от врага делает шаг только при нажатии определенной клавиши и т. д.).

Различие между пошаговыми играми и играми реального времени проявляется более всего при реализации. В пошаговых играх особое внимание уделяется стратегии, которой должен придерживаться компьютер-соперник (если он сам играет, а не только создает игровую ситуацию). В играх реального времени наибольшее внимание уделяется созданию разнообразной, увлекательной и красочной среды обитания игрующего персонажа. Кроме того, большое значение в таких играх имеет своевременная реакция компьютера на управляющие воздействия игрока, поскольку исход игры часто зависит от того, какое время проходит между двумя последовательными нажатиями на клавиши.

В ряде игр сложность игры зависит от уровня подготовки ее участников: чем успешнее реализуется игра, тем труднее становится играть. Среди таких игр можно дополнительно различать *игры с несколькими уровнями сложности, усложняющиеся и адаптирующиеся*. В играх с несколькими уровнями сложности игрок сам может выбрать уровень, на котором он будет играть. Сыграв несколько партий, можно изменить уровень сложности на более подходящий. В усложняющихся играх игроку, успешно закончившему партию, дается возможность сыграть еще одну партию на более высоком уровне сложности. Таким образом, игрок проводит столько партий, сколько уровней он сумеет преодолеть. Такой способ изменения уровня сложности не является наилучшим, поскольку основную массу времени игрок вынужденно проводит в несложных для себя партиях, что в конце концов может вызвать у него раздражение. По-иному обстоит дело в адаптирующихся играх. Здесь уровень сложности может не только увеличиваться, но и уменьшаться в зависимости от качества игры человека: игра всегда проходит на уровне сложности, примерно соответствующем уровню игрока, что создает у него положительный настрой, желание сыграть еще раз. Такая обстановка делает игру наиболее привлекательной.

Существуют *игры конечные*, в которых игрок может победить, и *игры бесконечные*, в которых победа никогда не наступает. Конечные игры имеют один недостаток. Если в такой игре человек все время побеждает, то она становится для него неинтересной, слишком простой. Так же быстро надоедает игра, если человек все время проигрывает. Поэтому конечные игры проигрывают в конкуренции бесконечным. Побеждая на очередном этапе бесконечной игры, человек чувствует возросшее мастерство. С другой стороны, сколько бы он ни играл — окончательная победа невозможна, и всегда остается уверенность, что он может улучшить свое последнее достижение. Поэтому бесконечные усложняющиеся игры в течение длительного времени притягивают к себе игроков.

Еще один возможный способ деления компьютерных игр на группы — рассмотрение *игр со случайными событиями и детерминированных игр*. Для того чтобы игра стала действительно интересной, она должна обладать непредсказуемостью, неповторемостью. В детерминированных играх каждая игровая ситуация является единственно возможным следствием предшествующей ей ситуации и управляющего воздействия игрока. Игрок быстро изучает поведение игры, развитие игры перестает для него быть неожиданным и игра становится скучной. Это нестрашно, когда собственное поведение игрока (а значит, и общее течение игры) подвержено неизбежной случайности. Нельзя, например, повториться при игре в «не пересекай». Никогда пальцы не смогут выдерживать жесткий ритм нажатия клавиш, точного соблюдения их последовательности и интервалов между ними. Конечно, общая стратегия может выдерживаться более-менее неизменной, и рисунок игры будет напоминать прежние, но картина каждый раз будет хоть чуть-чуть, а иной.

Во многих играх можно легко повторять свои ходы (в частности, в пошаговых играх). В этих случаях важно, чтобы не повторялась программа-партнер. Разнообразие поведения игровой программы достигается, как правило, за счет случайного выбора последующей игровой ситуации из нескольких возможных (разумеется, только в тех играх, в которых это возможно). Именно такие игры называются играми со случайными событиями. Примерами случайных шагов в действиях ЭВМ могут служить: выбор «поля действия» (точки на экране) посредством случайного задания его координат, случайная смена движения объекта на экране (вправо-влево-вверх-вниз), случайный выбор из нескольких заданных объектов (природа которых может быть самой разной: числа, рисунки, изображения предметов и т. п.), случайная смена цвета, неожиданный звуковой сигнал и т. п.

Разнообразие игровых ситуаций, к которому, как правило, стремится большинство авторов компьютерных игр, не позволяет игре приесться, однако оно не должно быть неуправляемым. Желательно, чтобы новые черты появлялись в игре при определенных условиях и в определенном порядке. Тогда у игрока будет возможность привыкнуть к новой ситуации и поупражняться в ней. Идеальным было бы возникновение по мере усложнения игры новых объектов с неожиданным поведением и своеобразным взаимодействием с другими объектами и с игроком. Однако это плохо увязывается с детерминированным характером программирования, которое может обеспечить лишь конечный набор объектов и ситуаций, не исключая и «случайное» (фактически псевдослучайное) поведение объектов. На практике выходом из положения служит конечный набор ситуаций, расположенных по мере возрастания сложности (причем, таким образом, чтобы выигрыш в последней ситуации был практически недостижим).

В настоящее время наиболее распространенными являются бесконечные, усложняющиеся, недетерминированные игры реального времени.

Устройство игровых программ

В компьютерных играх зародилась, прошла «обкатку», а затем проникла в серьезные программы идея *прямого управления обстановкой*, получившая широкое применение в персональных компьютерах. Традиционные способы игры с помощью компьютера основывались на языке команд и текстовых описаниях обстановки игры. Влиять на ситуацию игрок мог только через явно выраженного посредника, исполняющего его приказы. Такая игра имела пошаговый и опосредованный характер, больше ориентировалась на аналитические способности игрока.

В современных игровых программах обстановка игры выглядит на экране как графическая, цветная, живая картинка, объекты которой перемещаются «по правилам игры», а некоторые из объектов поддаются прямому управлению с помощью небольшого числа клавиш. Участник отождествляет себя с некоторым объектом на экране: он убегает, преследует, достигает, отступает и погибает. Это непрерывная игра в реальном времени. Игрок может уйти в нее с головой,

принимая вымышленный мир за реальный, дисплей — за окно в этот мир, ощущая клавиши компьютера как продолжение своих пальцев.

Тот же принцип лежит в основе серьезных программных продуктов. Например, чтобы удалить файл, можно просто указать на его условное изображение на экране, а затем «подтащить» его к изображению урны. Такие приемы облегчают общение с программой, а значит, увеличивают круг пользователей ЭВМ. В настоящее время принцип прямого наглядного управления широко используется в так называемых офисных системах, предназначенных для автоматизации делопроизводства.

Строгих принципов написания игровых программ не существует, но некоторые общие соображения могут облегчить составление программы, а главное, ее дальнейшие изменения и усовершенствования.

Никакая игра не существует без правил. Это утверждение относится и к компьютерным играм. Всякая игра предлагает игроющему определенную обстановку и правила поведения в рамках этой обстановки. Это и есть то, что называют *правилами игры* и что составляет ее неотъемлемую внутреннюю характеристику. В игровой программе правила носят даже более строгий и четкий характер. При этом сама программа следит за тем, как игрок (ученик, (пользователь) их выполняет. С точки зрения программы (точнее, программиста) правила — это набор процедур, определенных над структурами данных (наборами переменных). Процедуры и данные в совокупности воплощают «внутренний мир» игровой программы. В процессе игры внутренний мир постоянно меняется: во-первых, по собственным законам, во-вторых, под воздействием игрока (игроков). Таким образом, в каждый момент времени игра находится в некотором состоянии, что игрок ощущает как текущую обстановку. Ей соответствует состояние внутренних переменных — наполнение структур данных программы.

Выделим следующие компоненты игровой программы: преамбула (начальная стадия), основной цикл, завершающие действия. Структуры данных определяются на начальной стадии. Иногда они очень просты (несколько целых чисел), в наиболее интересных компьютерных играх они имеют сложную форму. На начальной стадии экран занимает начальная «позиция» игры (часто картинкой, в простых случаях — числами). Начальная позиция представляет собой визуальный эквивалент начальных значений структур данных.

В преамбулу выносятся приветствие программы игроку (при этом может сообщаться имя ее автора и дата создания), а также вопрос о том, нужны ли правила. При утвердительном ответе на него программа выдает на экран объяснительный текст. Если ответ отрицателен (обычно в случае, если игра ведется не первый раз и правила твердо усвоены), то игра переходит на следующую стадию.

Основной цикл — это та часть программы, где собственно и происходит игра. Здесь параллельно и согласованно изменяются внутренние структуры данных (по правилам игры) и перестраивается их отображение на экране. Главной задачей программы в основном цикле является поддержка диалога с игроком. В общем случае программа принимает от него управляющие воздействия, выражающие его поведение в игре. (Иногда эти воздействия можно трактовать как ходы). Управление чаще всего происходит в форме нажатий на определенные клавиши клавиатуры. Другие устройства ввода (джойстик, мышь) используются реже.

После того, как программа восприняла управляющие воздействия игрока, необходимо произвести их анализ, изменить содержание внутренних структур данных по правилам игры и отобразить эти изменения на экране дисплея, чтобы игрок увидел результат своих действий в виде новой ситуации. Основной цикл (и все упомянутые действия программы) повторяется много раз до тех пор, пока не будет достигнуто условие окончания игры, также проверяемое внутри цикла. Таким условием может быть некоторая конечная позиция, определенная клавиша («конец игры»), ограничение на число повторений (шагов или ходов

игры) и др. Рассмотренная структура является лишь приблизительной и достаточно сильно варьируется в деталях для конкретных программ.

Современные психологи считают, что игровая обстановка является наилучшей для обучения практически любому виду деятельности. Стремление добиться поставленной в игре цели стимулирует игрока напрячь силы, активизировать способности. Познавательный потенциал обучающих компьютерных игр неограничен. Игры, возникшие как модель ситуации, взятой из реальной жизни, помогают изучать эту ситуацию и получать необходимые навыки. Те из них, которые воспроизводят социальные, экономические, логические, биологические и другие закономерности, позволяют познавать их в игровой обстановке. Однако обучают не только обучающие игры. Игры реального времени требуют быстрой и правильной реакции на происходящие события, чувства времени, умения точно выдерживать заданные временные интервалы, способности следить за несколькими объектами одновременно. Пошаговые игры — знания алгоритма выигрыша или ведущих к нему эвристик, умения искать закономерности, способности к быстрому и максимально полному перебору основных вариантов, памяти на события, произошедшие в текущем и в прошлых сеансах игры. Как правило, все компьютерные игры, за исключением немногочисленного класса игр «на везение», требуют от игрока умения сосредоточиться, способности интенсивно работать на протяжении иногда длительного времени. Развитие у игрока всех упомянутых качеств является безусловным достоинством компьютерных игр.

Глава 4

Распознавание образов и анализ изображений

Ю. И. Журавлев, И. Б. Гуревич

4.1. Проблема распознавания

Основные понятия

Рассмотрим методы синтеза при помощи обучения алгоритмических процедур преобразования и анализа информации для задач, алгоритмы решения которых неизвестны. Эти задачи возникают в связи с обработкой и преобразованием на ЭВМ структур, образованных из символов, т. е. структур, представляющих в программах искусственного интеллекта знания о проблемой области в целом и знания, относящиеся к конкретной задаче. Несмотря на универсальность подобных задач эти методы стали предметом интенсивных исследований и получили оформление в виде математической теории лишь для одного обширного класса задач преобразования и анализа информации — задач распознавания образов.

Под образом понимается структурированное приближенное (частичное) описание (эскиз) изучаемого объекта или явления, причем частичная определенность описания является принципиальным свойством образа. Образ допускает рекурсивное определение: символ является образом, список символов является образом. Списочная запись позволяет использовать одно и то же представление для описания образа произвольного типа независимо от его «содержания», а также один и тот же алгоритм для работы с образами с различным содержанием. Естественно допускать, что образ состоит из двух групп символов, представляющих соответственно переменные и постоянные характеристики объекта описания.

Описания служат для установления соответствия образов, т. е. доказательства их идентичности, аналогичности, подобия, сходства и т. п., осуществляемого

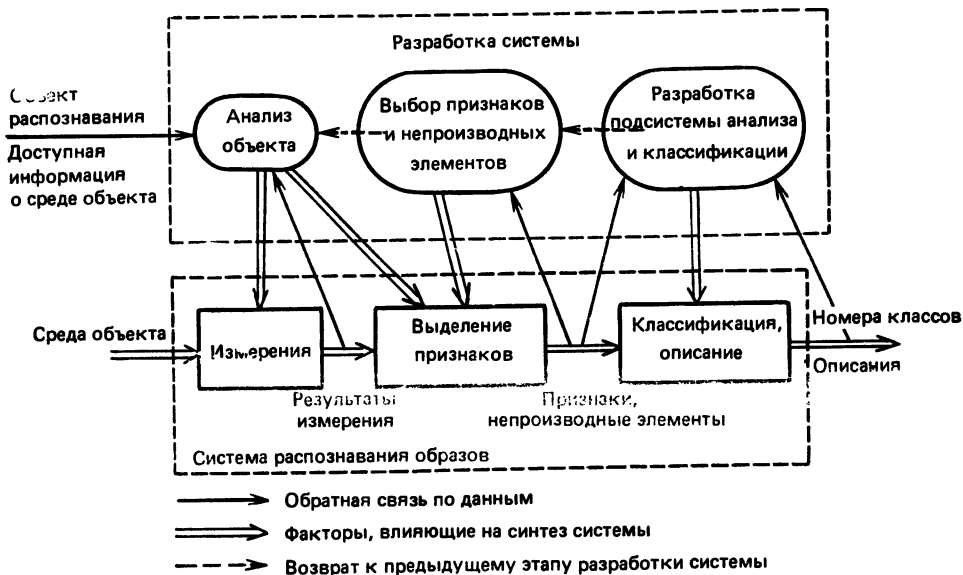


Рис. 4.1

сопоставлением. Сопоставление образов представляет собой основную задачу как распознавания, так и искусственного интеллекта в целом.

В различных задачах искусственного интеллекта понятию «образ» придается различный смысл. Так, в распознавании (в классических моделях) образ описывается вектором признаков, каждый элемент которого представляет числовое значение одного из признаков, характеризующих соответствующий объект. В структурной модели распознавания в качестве образа выступает высказывание, порожаемое грамматикой, характеризующей класс, которому данный образ принадлежит.

Термин «распознавание» относится в равной мере как к процессам восприятия и познания, свойственным живым организмам, так и к «механическим» аналогам (по функции и результату) этих процессов, исследование и синтез которых составляют предмет распознавания как раздела информатики. Целью создания автоматизированных вычислительных систем распознавания является автоматизация группы процессов восприятия и познания, связанных с поиском, выделением, идентификацией, классификацией и описанием образов на основе анализа реальных данных. Обычно поиск и выделение образов осуществляются на начальном этапе анализа в процессе обработки исходных данных и выполняются для того, чтобы получить промежуточные результаты (т. е. преобразовать исходные данные в другую форму), «лучше» представляющие образы с точки зрения решения соответствующей задачи. Следующий этап — разработка «классификатора» — включает анализ выборочных (преобразованных) данных, синтез модели, учитывающий изменчивость образов, принадлежащих некоторому классу, выбор из заданного набора характеристик некоторого подмножества, адекватно характеризующего отдельные классы объектов, определение методов выделения указанного подмножества и разработку алгоритма распознавания (классификации). На рис. 4.1 представлена общая структура системы распознавания и ее разработки.

К проблемам гносеологического обоснования распознавания и таких фундаментальных понятий распознавания, как образ, класс, распознающий алгоритм, обращались и философы [Тюхтин, 1976], и информатики, и математики. На становление и развитие теории распознавания существенное влияние оказали алгебраический подход к задачам распознавания и классификации (см. § 4.2) [Журавлев, 1978а], идея комбинаторной регулярности [Гренандер, 1979—1983] и концепция парадигматического символа [Watanabe, 1985].

В основе теории У. Гренандера лежит представление о структурированности мира, т. е. существовании регулярности, проявляющейся в виде постоянных связей, закономерностей. Отправной точкой служат объект (образ как таковой) и проблемы природы образа, прототипа, класса. Образы рассматриваются в рамках точного формализма, который используется в качестве основы для синтеза и анализа образов, что способствует пониманию того, каким способом образы строятся и обрабатываются. В результате процедуры, обеспечивающие описание, аппроксимацию, восстановление и распознавание образов, принимают вид естественных следствий процедур формирования и преобразования объектов. Образ следует соотносить с некоторым ансамблем возможных случаев, а порядок в таком ансамбле рассматривать как единообразное выполнение определенных свойств. Основным объектом служат комбинаторные регулярные структуры (конфигурации), логические конструкции, позволяющие определять типы регулярности. С формальной точки зрения речь идет о построении новых объектов посредством комбинирования заданных объектов в соответствии с определенными правилами построения объектов. Постулируется, что образы формируются из простых стандартных элементов (атомов), которые выбираются в соответствии с «физической» природой изучаемых объектов или явлений. В качестве таких элементов могут выступать абстрактные символы, множества, отношения или функции, но роль элементов в порождении регулярных структур идентична.

Концепция парадигматического символа опирается на формулу В. Витгенштейна «видеть нечто₁ как нечто₂». Здесь образ — противоположность хаосу, некая сущность, нечетко определенная, но допускающая приписывание ей имени, т. е. нечто. В этом смысле образ соответствует «нечто₂», причем последнее не существует как таковое в языке на том же уровне, как «нечто₁» присутствует на изображении. В некоторых случаях «нечто₂» имеет имя, в некоторых — не имеет его, в некоторых — структура безымянной конструкции должна описываться перечислением составляющих ее элементов и указанием хорошо определенного способа их соединения между собой. Как свидетельствуют многочисленные данные психологии восприятия и психиатрии, существенно, что «нечто₂» порождается посредством взаимодействия внешнего раздражителя и процессов мышления, обеспечивающих установление связи рассматриваемого объекта с другим, подобным ему или по меньшей мере родственным.

Замена в формуле Л. Витгенштейна глагола «видеть» на «распознавать» и «нечто₂» на «образ» приводит к идее класса, объединяющего ряд индивидуальных объектов, и к представлению о распознавании как об идентификации объекта в качестве элемента известного множества, т. е. как о процессе отображения, ставящего в соответствие различным элементам одного множества один и тот же элемент другого.

Распознавание обычно связывают с двумя функциями: отношением объекта к неизвестному классификатору классу объектов и идентификацией объекта в качестве элемента известного классификатору класса. Первая функция представляет собой процесс выделения новых классов — кластеризацию, а вторая — собственно распознавание. Это толкование распознавания опирается на представление об образе как о некотором объекте, выступающем в качестве выборочного образа объектов, составляющих некоторый класс. При этом предполагается, что некоторая группа объектов описывается (представляется, характеризуется) типичным примером, прототипом. С другой стороны, этимология понятия «образ» допускает толкование, не требующее существования «главного

объекта», являющегося родоначальником всех остальных разновидностей «образа». Эта двойственность отражается в эциентрическом и агрегативном подходах к определению образа и оказывает существенное влияние на методологию распознавания в целом.

Общая характеристика задач распознавания и их типы

Основной целью распознавания является построение на основе систематических теоретических и экспериментальных исследований эффективных вычислительных средств для отнесения формализованных описаний ситуаций и объектов к соответствующим классам. В основе такого отнесения (расознавания, классификации) лежит получение некоторой агрегированной оценки ситуации из ее описания. При условии установления соответствия между классами эквивалентности, заданными на множестве решений и множестве объектов распознавания (ситуаций), автоматизация процедур распознавания становится элементом автоматизации процессов принятия решений.

Задачи распознавания представляют собой дискретные аналоги задач поиска оптимальных решений, в которых по некоторой, обычно разнородной, быть может, неполной, нечеткой, искаженной и косвенной, информации требуется установить, обладают ли изучаемые (сложные, «комплексные») ситуации (объекты, явления) фиксированным конечным набором свойств, позволяющим отнести их к определенному классу, или по аналогичного рода информации о конечном множестве достаточно однотипных процессов следует выяснить, в какой области из конечного числа областей будут находиться эти процессы через определенный период времени.

Задачи распознавания имеют ряд специфических черт.

1. Это информационные задачи, которые решаются применением к доступным исходным данным преобразований, состоящих из двух основных этапов: а) приведение исходных данных к стандартному виду, удобному для распознавания — синтез формализованного описания ситуации (объекта) на основе имеющейся разнородной информации (эмпирических данных, результатов измерений, знаний о логических аспектах изучаемых явлений (процессов), сведений о конструкции, назначении и эксплуатационных характеристиках (возможно, предполагаемых) объекта, экспертных данных, имеющейся априорной семантической информации); б) собственно распознавание преобразованного формализованного описания в стандартизованную матрицу ответов, соответствующую выбору в качестве ответа (классификационного решения) одной из конечного фиксированного набора возможностей (указание принадлежности ситуации (объекта) определенному классу).

2. В этих задачах можно вводить понятие подобия между объектами (ситуациями), точнее, между их описаниями, формулировать обобщенное понятие близости в качестве основания для зачисления ситуаций (объектов) в один и тот же класс и разные классы.

3. В этих задачах можно оперировать набором прецедентов — примеров, классификация которых (в смысле решаемой задачи) известна и которые (в виде стандартных формализованных описаний) могут быть предъявлены алгоритму распознавания для настройки на задачу в процессе обучения.

4. Для этих задач трудно строить формальные теории и применять классические математические методы, поскольку в ситуациях, в которых они возникают, имеет место один из двух случаев: а) уровень формализации соответствующей предметной области и (или) доступная информация таковы, что не могут составить основу для синтеза математической модели, отвечающей классическим математическим или математико-физическим канонам и допускающей изучение классическими аналитическими или численными методами; б) математическая модель в принципе может быть построена, однако ее синтез или изучение связаны с такими затратами (сбор необходимой информации, вычислительные ресурсы, время), что они существенно превышают выигрыш, приноси-

мый решением, либо выходят за пределы существующих технических возможностей, либо делают решение задачи просто бессмысленным.

5. В этих задачах «по определению» существует «плохая» исходная информация, характеризующая сложную в семантическом и структурном отношении ситуацию (объект в некоторой среде), — это ограниченная, неполная (с пропусками), разнородная, косвенная (характеристики внешних проявлений процесса, причем не всегда относящиеся к принципиальным особенностям лежащего в его основе механизма), нечеткая, неоднозначная, вероятностная. В целом в этих задачах известно слишком мало для того, чтобы можно было пользоваться классическими методами решения (моделями), но все-таки известно достаточно, чтобы решение было возможно.

Известны следующие типы задач распознавания.

1. Отнесение предъявленного объекта (ситуации) по его формализованному описанию к одному из заданных классов — задача распознавания (обучение с учителем).

2. Разбиение множества ситуаций (объектов) по их формализованным описаниям на систему непересекающихся подмножеств (классов) — *задача автоматической классификации (таксономия, кластер-анализ, обучение без учителя)*.

3. Определение информативного набора признаков для построения формализованного описания объекта распознавания; оценка информативности отдельных признаков и их сочетаний — *задача выбора информативного набора признаков при распознавании*.

4. Построение формализованного описания объекта распознавания — *задача приведения исходных данных к виду, удобному для распознавания*.

5. Задача 1 с учетом динамичности объекта (ситуации).

6. Задача 2 с учетом динамичности объектов (ситуаций).

7. Задачи 5, 6, в которых решение должно относиться к некоторому моменту времени в будущем, — *задача прогнозирования*.

Все перечисленные задачи могут решаться при задании исходных данных в одной из следующих форм либо в их сочетаниях:

изображения, полученные в различных частях спектра излучений (оптические, инфракрасные, ультразвуковые и т. д.) различными способами (телевизионные, фотографические, лазерные, радиолокационные и т. д.) и преобразованные в цифровую форму;

сигналы (длинные числовые последовательности);

экспертные данные, числовые и другие виды информации;

серии изображений («фильмы») любого из перечисленных видов.

4.2. Математическая теория распознавания образов

Этапы развития и современное состояние

К середине 1970-х годов распознавание как самостоятельное научное направление достигло такой стадии развития, что возникла возможность создания математической теории распознавания. Одной из предпосылок явилось выделение и отработка ряда моделей алгоритмов распознавания.

Модели, основанные на использовании принципа разделения (R-модели), различаются главным образом заданием класса поверхностей, среди которых выбирается поверхность (или набор поверхностей), в некотором смысле наилучшим образом разделяющая элементы разных классов [Ту и др., 1978 (гл. 2); Еремин и др., 1979 (гл. 4)]).

Статистические модели основаны на использовании аппарата математической статистики. Применяются в основном в тех случаях, когда известны или могут быть определены вероятностные характеристики классов, например соответствующие функции распределения [Вапник и др., 1974; Горелик и др., 1985 (гл. 2, 3); Ту и др., 1978 (гл. 4, 6, 7)].

Модели, построенные на основе так называемого «метода потенциальных функций» (П-модели), базируются на заимствованной из физики идее потенциала, определенного для любой точки пространства и зависящего от расположения источника потенциала. В качестве функции принадлежности объекта классу используется потенциальная функция — всюду положительная и монотонно убывающая функция расстояния [Айзерман и др., 1970].

Модели вычисления оценок (голосования) (Г-модели) основаны на принципе частичной прецедентности. Анализируется «близость» между частями описаний ранее классифицированных объектов и объекта, который надо распознать. Наличие близости служит частичным прецедентом и оценивается по заданному правилу (посредством числовой оценки). По набору оценок близости вырабатывается общая оценка распознаваемого объекта для класса, которая и является значением функции принадлежности объекта классу [Журавлев, 1971; Журавлев и др., 1971; Гуревич др., 1974; Горелик и др., 1985 (гл. 3)].

Модели, основанные на исчислении высказываний, в частности на аппарате алгебры логики (Л-модели), в которых классы и признаки объектов рассматриваются как логические переменные, а описание классов на языке признаков представляется в форме булевых соотношений [Ледли, 1966; Горелик и др., 1985 (гл. 4)].

Становление распознавания служит отличной моделью развития математической теории обработки и преобразования информации, в процессе которого эвристические (по существу) методы получают строгое обоснование и начинают применяться в рамках формализованных регулярных процедур. Отметим, что распознавание сегодня является достаточно разработанным вариантом такой теории, поскольку позволяет разрешать ее основную задачу — синтезировать и выбирать алгоритмические средства для извлечения полезной информации.

Довольно долго подавляющее большинство приложений теории распознавания было связано с плохо формализованными областями — медициной, геологией, социологией, химией и т. д. Поэтому на первом этапе развития распознавания возникло множество методов и алгоритмов, применявшихся без серьезного обоснования для решения практических задач. При исследовании задачи на базе «правдоподобных» рассуждений предлагался нестрогий, но содержательно разумный метод решения и основанный на нем алгоритм; обоснование производилось непосредственно в эксперименте с задачами. Алгоритмы, приносившие успех при решении определенных практических задач, использовались несмотря на отсутствие математических обоснований.

Далее стало очевидным, что появление каждого эвристического алгоритма можно рассматривать как эксперимент, а со всем множеством экспериментов и их результатов работать как с новым для математики множеством объектов, т. е. изучать с помощью строгих математических методов множество некорректных процедур решения плохо формализованных задач. Поэтому второй этап развития теории распознавания отличался попытками, с одной стороны, ставить и решать задачу выбора в конкретной ситуации наилучшего в некотором смысле алгоритма, а с другой — переходить от описания отдельных некорректных алгоритмов к описанию принципов из формирования, т. е. строить единообразные описания для множеств эвристических, но успешно решающих реальные задачи процедур. Впервые в виде модели был представлен класс алгоритмов вычисления оценок. По мере распространения методов, основанных на идеологии распознавания, и роста числа практических задач обработки информации и анализа данных, решение которых удавалось сводить к схеме распознавания, выяснилось, что систематизация выбора наилучшего алгоритма распознавания непосредственно связана с формализацией теории распознавания.

Потребность в синтезе моделей алгоритмов распознавания определялась необходимостью фиксировать класс алгоритмов при выборе оптимальной или хотя бы приемлемой процедуры решения конкретной задачи. В свою очередь, построение таких моделей породило интерес к собственно «математическим» свойствам алгоритмов распознавания и в особенности к проблемам их строгого обоснования. Оказалось, что получение описания класса алгоритмов распозна-

вания представляет собой задачу, сходную с построением классического определения алгоритма. Следовательно, необходимым условием построения теории распознавания являлось проведение классических алгоритмических исследований для понятия «алгоритм распознавания», т. е. требовалось, как и в случае понятия «алгоритм», перевести интуитивные (эвристические, некорректные) представления на язык определений. Последнее эквивалентно необходимости построить математическую модель алгоритма распознавания, столь же убедительную (по критерию опроса), как и математическая модель понятия «алгоритм».

Анализ некорректных алгоритмов распознавания позволяет выделять и описывать не только отдельные алгоритмы, но и принципы их формирования. Эти принципы, действующие уже над подмножествами алгоритмов и формулируемые сначала также в плохо формализованном виде, затем могут реализовываться в виде точных математических описаний. На этом этапе эвристический характер имеет выбор принципа, а алгоритмы, порождаемые на основе соответствующего принципа, могут строиться стандартным образом. Именно в таком смысле формализация принципов построения распознающих алгоритмов приводит к появлению моделей распознающих алгоритмов.

Переход к моделям распознающих алгоритмов не привел ни к созданию универсальной модели, ни к формализации выбора модели для решения конкретной задачи распознавания, но позволил ставить и решать в рамках определенной модели задачу выбора алгоритма, экстремального по функционалу качества классификации или прогноза. В большинстве практических случаев класс таких задач невелик, так как в противном случае при синтезе модели алгоритмов распознавания, описании классов и выборе признаков объектов распознавания необходимо было бы использовать значительный объем априорной информации, которую можно получить, лишь располагая точной моделью изучаемых объектов и явлений.

Очередной этап развития распознавания был связан с изучением строения совокупности некорректных алгоритмов в целом. Поскольку оказалось, что обогащение модели часто не удается сопроводить эквивалентным улучшением результатов и существует естественная граница сложности любой модели, возникла идея выбирать алгоритмы из имеющихся семейств и, используя корректирующие операции над алгоритмами, строить из исходных алгоритмов оптимальный.

Одним из первых вариантов этой идеи явился так называемый «корректор по результатам», предусматривавший формирование решения задачи распознавания на основе результатов обработки исходной информации отдельными алгоритмами [Растргин и др., 1981]. Оказалось, однако, что не существует в некотором естественном смысле «хороших» простых операций, которые позволяли бы проводить необходимую коррекцию даже в простейшем случае.

В качестве выхода из этой ситуации был предложен алгебраический подход к задачам распознавания и классификации, обеспечивающий эффективное исследование и конструктивное описание алгоритмов распознавания, в том числе общее определение алгоритма распознавания, в рамках которого укладываются все существующие типы алгоритмов. Подход предусматривает обогащение исходных эвристических алгоритмов с помощью алгебраических операций и построение семейства алгоритмов, гарантирующего получение корректного алгоритма для решения изучаемого класса задач [Журавлев, 1976, 1977, 1978а, б; 1989].

В основе алгебраического подхода лежит идея индуктивного порождения математических объектов посредством обобщенного индуктивного определения. Выделяются базисные алгоритмы и модели распознавания и вводятся операции над ними, позволяющие последовательно порождать новые алгоритмы и модели. Выясняются условия, при которых данное семейство алгоритмов является базисным относительно введенных операций, а также свойства, которыми должна обладать модель для того, чтобы в ней нашелся алгоритм, правильно классифицирующий все объекты произвольной конечной выборки. Формируются методы построения семейства таких алгоритмов. Это семейство алгоритмов рассматривается как некоторая алгебра, операции которой позволяют на основе базиса

семейства алгоритмов строить такое расширение этого семейства, которое содержит корректный алгоритм, правильно классифицирующий конечную выборку по всем классам.

В алгебраическом подходе существенно используются особенности любой процедуры распознавания. Он предусматривает введение пространства оценок, промежуточного по отношению к исходным описаниям и допустимым ответам. Алгоритм распознавания при этом рассматривается как суперпозиция двух операторов: распознающего, который в качестве ответов формирует элементы, называемые оценками, и решающего правила, определяющего по оценкам окончательные ответы. Таким образом, необходимость иметь дело с «неудобными» пространствами исходных описаний и допустимых ответов уступает место возможности вести коррекцию в пространстве оценок (чаще всего оно представляет собой множество действительных чисел).

Важным в алгебраическом подходе является понятие полноты, связывающее отдельные задачи и модели алгоритмов: полнота некоторой задачи относительно модели означает, что при произвольном наборе априорных классификаций для рассматриваемых объектов в рамках модели может быть построен алгоритм, дающий всегда правильный ответ. Из полноты некоторой задачи относительно модели непосредственно следует существование в этой модели алгоритма, обеспечивающего абсолютную точность на материале обучения. Существенно что построение экстремального алгоритма оказывается в большинстве случаев задачей, сравнительно легко разрешаемой стандартными математическими методами.

Целью алгебраического подхода к распознаванию является получение алгоритма, обеспечивающего выделение из представленных данных всей полезной информации и получение решения, полностью и точно соответствующего «информативности» этой информации. Такое решение характеризуется минимальной (относительной) вычислительной сложностью, устойчивостью к шуму и искажениям в исходной информации и статистической надежностью. В процессе решения используются принцип прецедентности, формализация понятия обобщенной близости, автоматизация настройки алгоритма на задачу, в том числе автоматизация выбора класса алгоритмов, оптимального для рассматриваемого класса задач, и принцип коррекции окончательного решения посредством расширения базового множества моделей алгоритмов, используемых для его получения.

Решение имеет многоуровневый характер. На первом этапе строится эвристическая модель алгоритма, отражающая специфику задачи. На следующем этапе работа ведется с моделями семейств алгоритмов, порождаемых на основе принципа, выбираемого эвристически, стандартным образом. На этом этапе оптимизация алгоритма распознавания осуществляется для отдельных моделей. На третьем этапе искомый алгоритм синтезируется из алгоритмов, принадлежащих разным моделям.

Таким образом, алгебраический подход к обработке информации в задачах распознавания, прогнозирования и искусственного интеллекта обеспечивает реализацию идеологии, позволяющей синтезировать алгоритм, который при выполнении определенных нежестких и просто проверяемых условий точно решает конкретную задачу. Это своего рода методология автоматизации синтеза алгоритмов распознавания — САПР для алгоритмов распознавания и прогнозирования, позволяющая проанализировать предъявленную задачу, учесть ее особенности, а затем выбрать метод решения и на его основе предложить соответствующий алгоритм. Главное отличие алгебраического подхода от других методов распознавания состоит в том, что в них отсутствует этап три, а следовательно, нет реальной возможности получать точное решение. Смысл этого этапа состоит в снятии трудностей, возникающих на этапе 2 (неточность моделей эвристических алгоритмов распознавания, сложности, возникающие при реальном проведении оптимизации в многопараметрическом пространстве), и возможности обеспечивать получение абсолютно точного в указанном выше смысле решения в отличие от, как правило, локально-экстремальных решений этапа 2.

Параллельно процессу перехода в распознавании от эвристических алгоритмов к математической теории развивались исследования по использованию алгебраических методов для расширения типов исходной информации, допустимых в задачах распознавания. В этой связи отметим теорию образов У. Гренандера и развиваемую на основе алгебраического подхода дескриптивную теорию распознавания изображений (см. § 4.3) [Гренандер, 1979—1983; Гуревич и др., 1983; Гуревич 1984, 1985, 1986а, б, 1989]. Она предусматривает решение задач, связанных с получением формализованных представлений и описаний изображений как объектов распознавания и синтезом процедур их распознавания. При этом внутреннее строение, структура и содержание изображения рассматриваются как результат тех порождающих операций, с помощью которых изображение может быть построено из неприводимых элементов и объектов, выделяемых на изображении на различных этапах его анализа. Поскольку этот способ характеристики изображения является операциональным, весь процесс обработки и распознавания изображений, включая построение формализованного описания — модели изображения (траектория задачи), рассматривается как реализация над изображением системы преобразований, определенных на классах эквивалентности, которые представляют ансамбли допустимых изображений (ансамбли также задаются дескриптивно — системой прототипов и функционально полным относительно класса эквивалентности набором порождающих преобразований).

В процессе распознавания участвует иерархия формализованных описаний и представлений изображений в частности модели, относящиеся к различным морфологическим и масштабным уровням представления, многоуровневые модели, позволяющие в процессе распознавания выбирать и изменять степень подробности описания объекта распознавания.

Важнейшими положениями дескриптивной теории распознавания изображений являются: принцип порождения; формализация описания и представления изображения посредством учета многоуровневости конструкции описаний, использования в них синтаксической и реляционной информации, обобщения понятия «признак распознаваемого объекта» для задачи распознавания изображений, параллельного использования порождающих и параметрических моделей; двойственность процедур синтеза формализованных описаний и процедур распознавания, а также двойственность формализованных описаний и представлений; организация процесса распознавания с учетом методологии алгебраического подхода (схема «эвристика — модель — коррекция на множестве моделей»), обобщения понятия «алгоритм распознавания» для задачи распознавания изображений, иерархичности процедур преобразований и реализации их механизмом реверсивного алгебраического замыкания; логическая фильтрация, в основе которой лежит использование знаний об изображаемом объекте, предметной области, универсальных физических, логических и математических законах, целях, методах и средствах анализа; понятие траектории задачи, обеспечивающее сведение задачи выбора и синтеза преобразований в задаче распознавания изображений к решению задачи распознавания со стандартной информацией.

Отметим, что методология распознавания используется в информатике в двух качествах: по прямому назначению для решения задач распознавания в классическом смысле и как средство точного исследования плохо определенных задач. В последнем случае эта методология реализуется следующим образом. Пусть, например, данные, полученные в результате физического или имитационного эксперимента, в некотором ограниченном смысле характеризуют изучаемый объект или явление. Необходимо свести их воедино, чтобы установить закономерности имеющегося материала. Для этого выдвигается простая гипотеза, которой придается математический облик, и делается попытка «объяснить» этот материал с ее помощью. Последовательное использование ряда эвристик (реализаций гипотезы) может позволить угадать модель. В противном случае происходит переход к поиску в рамках модели, порождаемой эвристикой, а затем к поиску оптимального (адекватного) эвристического принципа — модели. Если соответствующего принципа не существует или им нельзя практически воспользо-

зоваться, то следует сформировать конгломерат принципов, обеспечивающий выделение «федеративного» принципа, — этот верхний уровень и соответствует возможностям и назначению алгебраического подхода.

Математическая постановка задачи распознавания

Пусть дано множество M объектов ω ; на этом множестве существует разбиение на конечное число подмножеств (классов) $\Omega_i, i = \overline{1, m}, M = \bigcup_{i=1}^m \Omega_i$.

Разбиение определено не полностью. Задана лишь некоторая информация I_ω о классах Ω_i . Объекты ω задаются значениями некоторых признаков $x_j, j = \overline{1, N}$ (этот набор всегда один и тот же для всех объектов, рассматриваемых при решении определенной задачи). Совокупность значений признаков x_j определяет описание $I(\omega)$ объекта ω . Каждый из признаков может принимать значения из различных множеств допустимых значений, например: $\{0, 1\}$ — признак не выполнен или выполнен; $\{0, 1, \Delta\}$, Δ — информация о признаке отсутствует; $\{0, 1, \dots, d-1\}$ — степень выраженности признака имеет различные градации, $d > 2$; $\{a_1, \dots, a_d\}$ — признак имеет конечное число значений; $d > 2$; $[a, b], (a, b), [a, b), (a, b]$ — произвольные числа или символы — $-\infty, +\infty$; значениями признака x_j являются функции некоторого класса; значениями признака x_j являются функции распределения некоторой случайной величины. Описание объекта $I(\omega) = (x_1(\omega), \dots, x_N(\omega))$ называют стандартным, если $x_j(\omega)$ принимает значение из множества допустимых значений.

Т а б л и ц а 4.1

Объект	Признаки и их значения			Класс
	x_1	x_j	x_N	
ω_1	$a_{1,1}$	$\dots a_{1,j}$	$\dots a_{1,N}$	Ω_1
...	
ω_{r_1}	$a_{r_1,1}$	$\dots a_{r_1,j}$	$\dots a_{r_1,N}$	
...
$\omega_{r_{i-1}+1}$	$a_{r_{i-1}+1,1}$	$\dots a_{r_{i-1}+1,j}$	$\dots a_{r_{i-1}+1,N}$	Ω_i
...	
ω_{r_i}	$a_{r_i,1}$	$\dots a_{r_i,j}$	$\dots a_{r_i,N}$	
...
$\omega_{r_{m-1}+1}$	$a_{r_{m-1}+1,1}$	$\dots a_{r_{m-1}+1,j}$	$\dots a_{r_{m-1}+1,N}$	Ω_m
...	
ω_{r_m}	$a_{r_m,1}$	$\dots a_{r_m,j}$	$\dots a_{r_m,N}$	
ω'	b_1	$\dots b_j$	$\dots b_N$	Ω_γ

Задача распознавания со стандартной информацией состоит в том, чтобы для данного объекта ω и набора классов $\Omega_1, \dots, \Omega_m$ по обучающей информации $I_0(\Omega_1, \dots, \Omega_m)$ о классах и описанию $I(\omega)$ вычислить значения предикатов $P_i(\omega \in \Omega_i)$, $i = \overline{1, m}$. Информация о вхождении объекта ω в класс Ω_i кодируется символами $1(\omega \in \Omega_i)$, $0(\omega \in \Omega_i)$, Δ (неизвестно, принадлежит ω классу Ω_i или нет) и записывается в виде так называемого информационного вектора

$$\tilde{\alpha}(\omega) = (\alpha_1(\omega), \dots, \alpha_m(\omega)), \quad \alpha_i \in \{0, 1, \Delta\}. \quad (1)$$

Стандартной информацией

$$I_0(\Omega_1, \dots, \Omega_m) \quad (2)$$

называют совокупность множеств $(I(\omega_1), \dots, I(\omega_m))$ и $(\tilde{\alpha}(\omega_1), \dots, \tilde{\alpha}(\omega_m))$ (предполагается, что среди информационных векторов нет вектора вида (Δ, \dots, Δ)).

Априорная информация в задаче распознавания с непересекающимися классами часто задается в виде так называемой *таблицы обучения* $T_{N,m}$ (табл. 4.1).

Синтез модели эвристического алгоритма распознавания

Проиллюстрируем построение такой модели на примере формализации принципа разделения, состоящего в том, что во многих задачах, где описания объектов задаются наборами значений числовых признаков (объекты суть точки n -мерного пространства), такие описания, принадлежащие разным классам, могут быть разделены поверхностями достаточно простого вида.

Рассмотрим одну из возможных формализаций. Воспользуемся простейшим классом разделяющих поверхностей — гиперплоскостями

$$\sum_{i=1}^n a_i x_i + a_{n+1} = 0. \quad (3)$$

Пусть множество допустимых объектов разделено на два класса: K_1, K_2 , $K_1 \cap K_2 = \emptyset$, и известно, что объекты S_1, \dots, S_m принадлежат K_1 , объекты S_{m+1}, \dots, S_q — K_2 . Эти объекты, вообще говоря, неравнозначны. Поэтому введем их числовые характеристики $\gamma(S_i) = \gamma_i$ — вес объекта S_i , $i = 1, 2, \dots, m, m+1, \dots, q$. Таким образом, множество алгоритмов характеризуется заданием параметров a_1, \dots, a_{n+1} — коэффициентов в уравнении гиперплоскости и $\gamma_1, \dots, \gamma_q$ — весов объектов, классификация которых проведена ранее. Процесс распознавания для $I(S) = (a_1, \dots, a_n)$ производится следующим образом

Пусть

$$f(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i + a_{n+1}. \quad (4)$$

Разделим объекты S_1, \dots, S_m на множества K_1^+ , K_1^- : $S_i \in K_1^+$, если $f(I(S_i)) \geq 0$; $S_i \in K_1^-$, если $f(I(S_i)) < 0$. Аналогично объекты S_{m+1}, \dots, S_q разделим на множества K_2^+ , K_2^- .

Рассмотрим величины

$$\gamma(K_1^+) = \sum_{S_i \in K_1^+} \gamma(S_i), \quad \gamma(K_1^-) = \sum_{S_j \in K_1^-} \gamma(S_j) \quad (5)$$

и аналогичные им величины $\gamma(K_2^+)$, $\gamma(K_2^-)$.

Вычислим $f(I(S))$. Сопоставим S с двумя числами: $\Gamma_1(S)$, $\Gamma_2(S)$ — значениями функции принадлежности S классам K_1 , K_2 соответственно. Если $f(I(S)) > 0$, то

$$\Gamma_1(S) = \frac{\gamma(K_1^+) + \gamma(K_2^-)}{\gamma(K_1^+) + \gamma(K_2^+)}, \quad \Gamma_2(S) = \frac{\gamma(K_2^+) + \gamma(K_1^-)}{\gamma(K_1^+) + \gamma(K_2^-)}. \quad (6)$$

При $f(I(S)) < 0$

$$\Gamma_1(S) = \frac{\gamma(K_1^-) + \gamma(K_2^+)}{\gamma(K_1^+) + \gamma(K_2^-)} \quad (7)$$

и аналогично $\Gamma_2(S)$.

По числам $\Gamma_1(S)$, $\Gamma_2(S)$ принимается решение о зачислении S в K_1 или K_2 . Эта процедура задается решающим правилом. Рассмотрим класс решающих правил, определяемых параметром $\delta \geq 0$:

если $\Gamma_1(S) - \Gamma_2(S) > \delta$, то $S \in K_1$;

если $\Gamma_2(S) - \Gamma_1(S) > \delta$, то $S \in K_2$;

если $|\Gamma_1(S) - \Gamma_2(S)| \leq \delta$, то решение не принимается, алгоритм отказывается от классификации S .

Построенная модель основывается на гипотезах: а) элементы классов K_1 и K_2 разделяются гиперплоскостью (по крайней мере значительная часть элементов, классификация которых представляет интерес); б) элементы классов неравнозначны по важности; меру этой важности можно выразить числом.

Реализация гипотез проведена при построении модели

$$M(a_1, \dots, a_{n+1}, \gamma_1, \dots, \gamma_q, \delta), \quad -\infty < \gamma_i, a_i < +\infty, \delta \geq 0. \quad (8)$$

Задание значений всех параметров модели определяет ее элемент — конкретный алгоритм распознавания.

Синтез экстремального в модели алгоритма распознавания

Из множества математических задач, возникающих на уровне моделей, следует выделить задачу синтеза алгоритмов, экстремальных по качеству распознавания в данной модели. Функционал качества алгоритма может определяться различными способами. Обычно задается способ построения объектов каждого из классов. Оценивается для фиксированного алгоритма из данной модели, какую долю объектов он классифицирует правильно, т. е. относит к данному классу. Полученная величина усредняется по классам и называется функционалом качества алгоритма. Задача состоит в том, чтобы для модели найти алгоритм с максимальным значением функционала качества.

Например, может быть задан следующий закон порождения классов K_1 , K_2 . Пусть описания $I(S)$ объектов S являются наборами $(a_1(S), \dots, a_n(S))$ числовых признаков $-\infty < a_i(S) < +\infty$, $i = \overline{1, n}$. В n -мерном пространстве заданы два нормальных распределения с математическими ожиданиями m_1 , m_2 и дисперсиями σ_1 , σ_2 . Производится случайный выбор точек (описаний объектов) и разыгрывается по заданным законам класс, в который они зачисляются. После этого объект S , занесенный, например, в K_1 , с вероятностью p причисляется к обучающей выборке, с вероятностью $1-p$ — к контрольной. То же самое делается с объектами из K_2 . Пусть таким образом сформированы обучающая и контрольная выборки. В первую из них зачислены из K_1 объекты S_{11}, \dots, S_{1m} , из K_2 — S_{21}, \dots, S_{2l} ; во вторую из K_1 — объекты S_{31}, \dots, S_{3v} , из K_2 — S_{41}, \dots, S_{4u} . В модели строится алгоритм A , который по описаниям $I(S_{11}), \dots, I(S_{1m})$, $I(S_{21}), \dots, I(S_{2l})$ дает максимальное значение функционала качества $\Phi(A) = q'/q''$, где q' — число объектов из контрольной выборки, правильно классифицированных алгоритмом A ; $q'' = v + u$ — число объектов в контрольной выборке.

Значение $\Phi(A)$ есть случайная величина, и ее характеристики (моменты) дают представление о точности модели на определенном типе задач распозна-

алгоритмы распознавания $\{A\}$ такого алгоритма A , на котором функционал F_A , характеризующий качество распознающего алгоритма A , обращается в максимум. В классе АВО функционалом качества, заданным на пространстве распознающих алгоритмов, является ошибка распознавания, возникающая при использовании конкретного алгоритма.

Уточним понятие ошибки распознавания. Довольно часто задачи распознавания и классификации обладают важной особенностью: число объектов, принадлежность которых к определенным классам известна, составляет незначительную часть от числа гипотетически возможных («допустимых») объектов. В таких ситуациях следует вычислять ошибку или некоторую «обобщенную» ошибку по всем объектам, принадлежность которых определенным классам уже известна, и минимизировать именно этот функционал. С другой стороны, при использовании ограниченного «экспериментального материала» переход к другому экспериментальному материалу может привести к кардинальному изменению функционала ошибки. Для практических приложений распознавания альтернатива такова: либо в принципе недопустимо решать задачу распознавания на ограниченном материале, либо необходимо пользоваться функционалом ошибки по этому материалу. Снятие этой проблемы связано с отысканием критериев устойчивости статистической выборки по функционалу ошибки [Вапник, 1979].

4. Поскольку оптимизационные процедуры реализуются на специально выбираемом пространстве, то для синтеза оптимального алгоритма необходимо определить класс алгоритмов, на котором производится оптимизация, таким образом, чтобы можно было ее реально осуществить. Это значит, что нужна модель, единообразно представляющая широкий класс алгоритмов распознавания, которая должна задаваться рядом параметрических объектов, характеризующих данный класс алгоритмов распознавания. Подобная модель позволяет установить взаимно однозначное соответствие между алгоритмами и набором числовых параметров (класс алгоритмов — область многомерного пространства). При этом, если задан конкретный алгоритм A , принадлежащий рассматриваемому классу алгоритмов распознавания $\{A\}$, то можно сопоставить с ним значение функционала обобщенной ошибки F_A и, следовательно, определить функционал, оценивающий качество алгоритма распознавания, на точках соответствующего параметрического пространства. Значения этого функционала должны вычисляться эффективно. Задание параметрической модели алгоритма обеспечивает возможность подбирать и изменять его параметры таким образом, чтобы значение функционала качества изменялось в нужном направлении. Если существует эффективный способ вычисления функционала F_A , то принципиально возможно построение алгоритма A^* , на котором достигается экстремум F_A . Хотя известно, что последняя задача не всегда может быть решена до конца, в тех случаях, когда абсолютно экстремальный алгоритм может быть найден, имеется гарантия, что при данном исходном материале в данном классе алгоритмов для заданного контрольного материала не существует лучшего алгоритма распознавания, чем A^* [Журавлев, 1971, 1978a].

Часто основная «различающая» информация заключена не в отдельных признаках, а в различных их сочетаниях. Наибольшую известность среди алгоритмов, позволяющих учитывать информацию, заключенную в комбинациях признаков, приобрели так называемые тестовые алгоритмы [Дмитриев и др., 1966], программа «Геометрия» (блок «Кора») [Бонгард, 1967] и алгоритм «Кора» [Вайнцвайг, 1973]. Вычислительная сложность (объем перебора) заставляла в двух последних алгоритмах ограничиться конъюнкциями сложности три.

Класс АВО доводит идею использования совокупностей признаков до логического конца: поскольку не всегда известно, какие сочетания признаков наиболее информативны, то в АВО степень схожести объектов вычисляется при сопоставлении всех возможных (или определенных, когда известны сочетания признаков, обладающие наибольшей разделяющей силой) сочетаний признаков, входящих в описание объектов. Для вычисления соответствующих оце-

нок близости объектов в АВО имеются несложные аналитические формулы, снимающие перебор при реализации процедуры распознавания (точнее, при настройке параметров алгоритма на задачу в процессе обучения). Кроме того, АВО позволяет учитывать различия в информативности (разделяющей силе) признаков и их сочетаний, а также в репрезентативности объектов, включенных в таблицу обучения.

5. Важным отличием АВО от других классов алгоритмов распознавания являются значительно более слабые требования к исходной информации, так как они не предусматривают наличия сведений о моментах и других статистических характеристиках. Исходный материал может представляться не только в числовой форме, но и описаниями на естественном языке.

6. Класс АВО позволяет решать статические и динамические задачи — к последним часто сводятся задачи прогнозирования.

Алгоритм распознавания, основанный на принципе прецедентности или частичной прецедентности, сравнивает описания распознаваемого объекта $I(\omega')$ с $T_{N,m}$ и принимает решение о том, к какому классу следует отнести этот объект. Решение выносится на основе вычисления степени сходства распознаваемого объекта (строки) со строками, принадлежность которых к заданным классам известна.

Пусть заданы стандартные описания объектов $\{\omega^{\bar{i}}\}$, $\omega^{\bar{i}} \in Q_i$ и $\{\bar{\omega}^i\}$, $\bar{\omega}^i \in \bar{Q}_i$. Необходимо определить принадлежность классу Q_i , $i = \overline{1, m}$, предъявленного для распознавания объекта ω' . Если введен способ определения величины близости для некоторых частей описания $I(\omega')$ и соответствующих частей описаний $\{I(\omega^{\bar{i}})\}$ и $\{I(\bar{\omega}^i)\}$, то можно сформировать „обобщенную близость“ между объектом ω' и

множествами объектов $\{\omega^{\bar{i}}\}$ и $\{\bar{\omega}^i\}$ соответственно. В простейшем случае обобщенная близость приравнивается сумме близостей между частями описаний. В результате характеристику вида $G_i(\omega') = G_i^{\bar{i}} - G_i^i$, где $G_i^{\bar{i}}$ и G_i^i — значение соответствующих обобщенных близостей, естественно считать значением функции принадлежности объекта ω' классу Q_i . Величина $G_i(\omega')$ называется оценкой объекта ω' по классу Q_i (иногда будем обозначать ее как $G(\omega', Q_i)$).

Описания объектов $\{\omega'\}$, предъявленные для распознавания, переводятся алгоритмом распознавания в числовую матрицу $\{G_i\}_{\{\omega\} \times m}$ — матрицу оценок. Эта процедура включает два этапа: сначала подсчитывается оценка ω' по каждой строке из $T_{N,m}$, а затем эти оценки используются для получения суммарных оценок по каждому из классов Q_i . Применение решающего правила к матрице оценок позволяет получить матрицу $\{\alpha_i\}_{\{\omega\} \times m}$ информационных векторов объектов $\{\omega'\}$.

Рассмотрим процедуру построения оценок $G_i(\omega')$, используемую в тестовых алгоритмах и АВО. В основе тестовых алгоритмов лежит понятие теста [Чегис и др., 1958]. Тестом таблицы $T_{N,m}$ называется совокупность столбцов x_{t_1}, \dots, x_{t_q} , таких, что после удаления из $T_{N,m}$ всех столбцов, за исключением имеющих номера t_1, \dots, t_q , в полученной таблице $T_{N-q,m}$ все пары строк, принадлежащих разным классам, различны. Тест $\{x_{t_1}, \dots, x_{t_q}\}$ называется тупиковым, если никакая его часть не является тестом.

Пусть $\{T\}$ — множество всех тупиковых тестов $T_{N,m}$ и $T = \{x_{t_1}, \dots, x_{t_q}\} \in \{T\}$. Выделим в описании распознаваемого объекта $I(\omega')$ часть $(b_{t_1}, \dots, b_{t_q})$, соответствующую признакам x_{t_1}, \dots, x_{t_q} , и сопоставим ее со всеми частичными описаниями $(a_{r_{t_1}}, \dots, a_{r_{t_q}})$ объектов $I(\omega_r)$ таблицы $T_{N,m}$, $r = t_{i-1} + 1, \dots, t_i$

$i = \overline{1, m}$. Подсчитаем число совпадений $\Gamma_T(\omega', \Omega_i)$ — частичных описаний $(b_{t_1}, \dots, b_{t_q})$ со всеми частичными описаниями $(a_{r_{t_1}}, \dots, a_{r_{t_q}})$ объектов i -го класса.

Величина $\Gamma_T(\omega', \Omega_i)$ представляет собой число строк этого класса, близких к распознаваемой строке ω' по тесту T , т. е. оценку строки ω' для класса Ω_i по тесту T . Аналогично вычисляется оценка для ω' по остальным тестам (для всех классов). Величина

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{T \in \{T\}} \Gamma_T(\omega', \Omega_i) \quad (10)$$

представляет собой оценку объекта ω' по классу Ω_i .

Известны разновидности тестовых алгоритмов, в которых при формировании оценок $\Gamma_T(\omega', \Omega_i)$ учитываются различия в представительности («важности») отдельных строк таблицы $T_{N,m}$ и признаков, включенных в стандартные описания объектов. Для этого используются числовые коэффициенты — веса признаков и веса объектов. Чаще всего веса задаются из эвристических соображений, исходя из специфики решаемой задачи, с помощью экспертной оценки и т. п. Для тестовых алгоритмов была предложена мера важности признака — информационный вес [Дмитриев и др., 1966]:

$$p(x_j) = r_{x_j}(N, m) / r(N, m), \quad (11)$$

где $r(N, m)$ — число тупиковых тестов таблицы $T_{N,m}$; $r_{x_j}(N, m)$ — число тупиковых тестов $T_{N,m}$, содержащих признак x_j . Чем в большее число тупиковых тестов входит признак x_j , тем больше его информационный вес $p(x_j)$, тем значительнее его роль в описании объектов $T_{N,m}$.

Если учитываются веса признаков $p(x_1), \dots, p(x_N)$ и объектов таблицы $T_{N,m}(\omega_1), \dots, \gamma(\omega_m)$, то каждое совпадение частичного описания распознаваемого объекта $(b_{t_1}, \dots, b_{t_q})$ с частичным описанием объектов из $T_{N,m}(a_{r_{t_1}}, \dots, a_{r_{t_q}})$, соответствующих некоторому тесту T , оценивается величиной

$$\Gamma_T(\omega', \omega_r) = \gamma(\omega_r) (p(x_{t_1}) + \dots + p(x_{t_q})). \quad (12)$$

В результате оценка объекта ω' по классу Ω_i (10) принимает следующий вид:

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{T \in \{T\}} \sum_{r=r_{i-1}+1}^{r_i} \Gamma(\omega', \omega_r). \quad (13)$$

Переход от тестовых алгоритмов к АВО связан с расширением видов подмножеств множества признаков, по которым проводится сопоставление распознаваемого объекта с объектами из $T_{N,m}$, и построением эффективных формул вычисления оценок $\Gamma(\omega', \Omega_i)$ для различных случаев задания подмножеств множества признаков (в АВО они называются опорными множествами алгоритма распознавания). В тестовых алгоритмах в качестве системы опорных множеств алгоритма используются множества тупиковых тестов (при отсутствии пересечений классов в $T_{N,m}$ и бинарных признаках). В АВО рассматриваются два случая: наличие [Журавлев, 1971; Журавлев и др., 1971] и отсутствие ограничений на систему опорных множеств алгоритма [Гуревич и др., 1974]. В первом случае наиболее распространенными являются системы опорных множеств, составленные из всех подмножеств множества признаков фиксированной длины q , $q=2, \dots, N-1$, либо из всех непустых подмножеств множества признаков.

Рассмотрим полный набор признаков $\langle x_1, \dots, x_N \rangle$ и выделим систему подмножеств множества признаков (систему опорных множеств алгоритма) S_1, \dots, S_l . Удалим произвольный поднабор признаков из строк $\omega_1, \omega_2, \dots, \omega_m$, и обозначим полученные строки через $\tilde{S}_{\omega_1}, \tilde{S}_{\omega_2}, \dots, \tilde{S}_{\omega_m}, \tilde{S}'$.

Правило близости, позволяющее оценить похожесть строк $\tilde{S\omega}'$ и $\tilde{S\omega}_r$, состоит в следующем. Пусть «усеченные» строки содержат q первых признаков, т. е. $\tilde{S\omega}_r = (a_1, \dots, a_q)$ и $S\omega' = (b_1, \dots, b_q)$, и заданы пороги $\varepsilon_1, \dots, \varepsilon_q$, δ . Строки $\tilde{S\omega}_r$ и $S\omega'$ считаются похожими, если выполняется не менее чем δ неравенств вида $|a_j - b_j| \leq \varepsilon_j$, $j = 1, \dots, q$. Величины $\varepsilon_1, \dots, \varepsilon_q$, δ входят в качестве параметров в модель класса алгоритмов типа АВО.

Рассмотрим процедуру вычисления оценок по подмножеству S_i . Для остальных подмножеств она полностью аналогична. В таблице $T_{N,m}$ выделяются столбцы, соответствующие признакам, входящим в S_i ; остальные столбцы вычеркиваются. Проверяется близость строки $S_i \tilde{\omega}'$ со строками $S_i \tilde{\omega}_1, \dots, S_i \tilde{\omega}_{r_i}$, принадлежащими классу Ω_i . Число строк этого класса, близких по выбранному критерию классифицируемой строке $S_i \tilde{\omega}'$, обозначается через $\Gamma_{S_i}(\omega', \Omega_i)$; эта величина представляет собой оценку строки ω' для класса Ω_i по опорному множеству S_i . Аналогично вычисляются оценки для остальных классов: $\Gamma_{S_1}(\omega', \Omega_1), \dots, \Gamma_{S_m}(\omega', \Omega_m)$.

Применение подобной процедуры ко всем остальным опорным множествам алгоритма позволяет получить систему оценок

$$\Gamma_{S_1}(\omega', \Omega_1), \Gamma_{S_1}(\omega', \Omega_m), \dots, \Gamma_{S_j}(\omega', \Omega_1), \dots, \Gamma_{S_j}(\omega', \Omega_m).$$

Величины

[illegible]

представляют собой оценки строки ω' для соответствующих классов по системе опорных множеств алгоритма S_A . На основании анализа этих величин принимается решение либо об отнесении объекта ω' к одному из классов Ω_i , $i = 1, \dots, m$, либо об отказе от его распознавания. Решающее правило может принимать различные формы, в частности, распознаваемая строка может быть отнесена к классу, которому соответствует максимальная оценка, либо эта оценка будет превышать оценки всех остальных классов не меньше чем на определенную пороговую величину η_1 , либо величина отношения соответствующей оценки к сумме оценок для всех остальных классов должна быть не менее величины некоторого порога η_2 и т. д. Параметры типа η_1 и η_2 также включаются в модель АВО.

Определение класса АВО сводится к формализации следующих этапов процедуры распознавания: 1) выделяется система опорных множеств алгоритма, по которым производится анализ распознаваемых объектов; 2) вводится понятие близости на множестве частичных описаний объектов; 3) задаются правила: а) вычисления оценки для пар объектов по значению степени подобия эталонного и распознаваемого объектов; б) формирования оценок для каждого из эталонных классов по фиксированному опорному множеству на основе оценок для пар объектов; в) формирования суммарной оценки для каждого из эталонных классов по всем опорным множествам; г) принятия решения, обес-

печивающего на основе оценок для классов отнесение распознаваемого объекта к одному из классов или отказывающего этому объекту в классификации.

Фиксация структурных параметров, т. е. способа выбора системы опорных множеств, типа функции близости, правил вычисления оценок и решающего правила, определяет выбор подкласса АВО, а задание соответствующих параметров — конкретный алгоритм типа АВО. Модель класса АВО — параметрическая, т. е. имеет место взаимно однозначное соответствие между конкретными алгоритмами и наборами числовых параметров. В таком случае задание конкретного алгоритма, принадлежащего рассматриваемому классу, позволяет сопоставить с ним значение некоторого функционала качества распознавания (например, число ошибок и отказов от распознавания на таблице обучения) и, следовательно, определить последний на точках параметрического пространства алгоритма.

Если строить вычислительную процедуру по приведенному описанию алгоритма, то при большой мощности системы опорных множеств требуется значительное число машинных операций. Так, при выборе в качестве системы опорных множеств алгоритма всех подмножеств множеств признаков мощности q число опорных множеств равно $C_{N,q}$, а число слагаемых в формуле, определяющей величину $\Gamma_{S_A}(\omega', \Omega_i)$ (6), равно $(r_i - r_{i-1}) C_N^q$.

Поскольку эффективность (в вычислительном смысле) вычисления функционала качества в АВО определяется эффективностью процедуры вычисления оценок, то можно построить оптимальный алгоритм. В случаях, когда может быть найден абсолютно экстремальный алгоритм, имеется гарантия, что при заданном исходном материале $T_{N,m}$ в данном классе не существует лучшего алгоритма распознавания.

Известны два метода, комбинация которых позволяет находить достаточно простые формулы для практически интересных моделей АВО при условии, что используются пороговые функции близости (принимают значения 0 или 1) и $p(S) = p_{x_1} + \dots + p_{x_q}$ (вес опорного множества равен сумме весов входящих в него признаков).

1. Первый метод [Журавлев, 1978а] использует свойство оценок для класса $\Gamma(\omega', \Omega_i)$, когда в качестве оценки для класса по опорному множеству S_u , $u = \overline{1, l}$, используется оценка вида (12)

$$\Gamma_{S_u}(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) (p_{t_1} + \dots + p_{t_q}) B_{S_u}(\omega', \omega, \omega_r), \quad (15)$$

где t_1, \dots, t_q — совокупность единичных координат характеристического вектора, определяющего опорное множество S_u ; B_{S_u} — функция близости частичных описаний объектов $s\omega'$ и $s\omega_r$, принимающая значение «1» или «0» в зависимости от числа выполненных неравенств вида $|a_j - b_j| \leq \varepsilon_j$, $j = t_1, \dots, t_q$.

Пусть $V_j(\omega', \omega_r)$ — число различных значений числа опорных множеств $S_u \in S_A$, содержащих некоторый фиксированный признак x_j и таких, что $B(S\omega', S\omega_2) = 1$. Можно показать, что оценка для класса Ω_i принимает вид

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) \sum_{j=1}^N p_j V_j(\omega', \omega_r). \quad (16)$$

Если число различных значений $V_j(\omega', \omega_r)$ невелико, то внутренняя сумма свертывается в небольшое число слагаемых (практически исключается суммирование по системе опорных множеств) и сложность вычисления $\Gamma(\omega', \Omega_i)$ становится пропорциональной длине обучающей выборки. Доказано, в частности, что при использовании пороговой функции близости, значение которой определяется лишь числом выполненных и невыполненных неравенств вида

$|a_j - b_j| < \epsilon_j$, величины $V_j(\omega', \omega_r)$, $j=1, \dots, N$, принимают не более двух значений.

Приведем аналитические формулы, обеспечивающие эффективное вычисление оценок $\Gamma(\omega', \Omega_i)$ для двух способов введения ограничений на систему опорных множеств алгоритма [Журавлев, 1971, 1978а].

А. S_A совпадает с системой всех подмножеств множества признаков $\{1, \dots, N\}$ мощности g .

А1. Функция близости принимает значение 1, если выполняется не менее δ (порог по числу выполненных) неравенств вида

$$|a_{t_u} - b_{t_u}| \leq \epsilon_{t_u};$$

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) \left(\sum_{j=0}^{\delta} C_{z(\omega', \omega_r)}^{q-j} C_{N-z(\omega', \omega_r)}^j \right) \quad (17)$$

где $z(\omega', \omega_r)$ — число выполненных неравенств вида $|a_j - b_j| < \epsilon_j$ для пары (ω', ω_r) , $j=1, \dots, N$.

А2. Функция близости принимает значение 1, если выполнены все неравенства вида $|a_{t_u} - b_{t_u}| \leq \epsilon_{t_u} (\delta = 0)$, т. е. в опорное множество входят только совпадающие по порогу ϵ_{t_u} признаки:

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) C_{z(\omega', \omega_r)}^q. \quad (18)$$

Б. S_A совпадает с системой всех непустых подмножеств множества признаков $\{1, \dots, N\}$.

Б1. Функция близости та же, что и в случае А1:

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) \left[(2^{z(\omega', \omega_r)} - 1) \sum_{j=0}^{\delta} C_{N-z(\omega', \omega_r)}^j \right]. \quad (19)$$

Б2. Функция близости та же, что и в случае А2:

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r \in \Omega_i} \gamma(\omega_r) (2^{z(\omega', \omega_r)} - 1). \quad (20)$$

2. Второй способ получения эффективных формул вычисления оценок [Гуревич и др., 1974] основан на следующих двух утверждениях:

если система опорных множеств S_A состоит из непересекающихся подмножеств, то оценка $\Gamma(\omega', \Omega_i)$ для такого алгоритма равна сумме оценок для алгоритмов, опорными множествами которых являются подмножества, образующие S_A (каждому подмножеству соответствует отдельный алгоритм):

$$S_A = \bigcup_u S_u; \quad \Gamma_A(\omega', \Omega_i) = \sum_u \Gamma^A_u(\omega', \Omega_i). \quad (21)$$

Если характеристическая функция алгоритма класса АВО является элементарной конъюнкцией, то задача построения эффективной формулы вычисления оценок обеспечивается методом 1.

В некоторых случаях априори известны подмножества признаков, которые следует учитывать при сопоставлении распознаваемого объекта с объектами таблицы обучения. Эти подмножества не всегда совпадают с частными случаями (17)–(20); они могут иметь различную длину, задаваться запрещенные или «избыточные» комбинации и т. п. В [Гуревич и др., 1974] аналитические

формулы вычисления оценок получены для произвольных опорных множеств. Вывод формул основан на введении характеристической булевой функции системы опорных множеств алгоритма f_{S_A} и установлении взаимно однозначного соответствия между подмножествами множества признаков и булевыми векторами длиной N (вершинами единичного N -мерного куба) [Журавлев, 1974].

В тех случаях, когда множество единиц f_{S_A} образует в единичном N -мерном кубе интервал или сумму непересекающихся интервалов, можно построить эффективные формулы вычисления оценок. Напомним, что подмножество вершин единичного N -мерного куба называется интервалом, если оно соответствует некоторой элементарной конъюнкции. Очевидно, что все грани, ребра и вершины единичного N -мерного куба являются интервалами.

Система опорных множеств организована следующим образом (соответствующий интервал представлен ребром, соединяющим вершины): в нее включены все признаки, входящие в дизъюнктивную нормальную форму (ДНФ) характеристической функции без отрицания, не включены признаки, входящие в ДНФ с отрицанием, а по остальным признакам происходит полная вариация, т. е. рассматриваются подмножества, как включающие, так и не включающие эти признаки.

Эффективная формула для вычисления оценок в тех случаях, когда характеристической функции системы опорных множеств соответствует интервал и функция близости предполагает, что $\delta=0$, имеет вид

$$\Gamma(\omega', \Omega_i) = \frac{1}{r_i - r_{i-1}} \sum_{\omega_r^* \in \Omega_i} \gamma(\omega_r^*) \cdot 2^{z^*(\omega', \omega_r^*)}, \quad (22)$$

где ω_r^* — «эффективный» для ω' объект таблицы обучения. В (22) учитывается вклад только тех объектов $T_{N,m}$ («эффективных»), постоянная часть которых близка (в смысле используемой функции близости) постоянной части ω' ; $z^*(\omega', \omega_r^*)$ — число выполненных неравенств вида $|a_j - b_j| < \epsilon_j$ на варьируемой части.

Если характеристической функции соответствует сумма непересекающихся интервалов (представляется ортогональной ДНФ), то при вычислении оценок по (14) она применяется к каждому интервалу отдельно, после чего результаты суммируются.

Сложность формулы вычисления оценок в АВО при произвольной S_A пропорциональна сложности ДНФ, представляющей характеристическую функцию системы опорных множеств алгоритма. Это означает, что построение простой формулы вычисления оценок $\Gamma(\omega', \Omega_i)$ связано с задачей минимизации булевых функций в классе ДНФ, а точнее — с задачей построения кратчайшей ортогональной ДНФ или ДНФ, в которой каждый интервал имеет небольшое число пересечений с соседними. В общем случае задача такого синтеза неразрешима и потому следует пользоваться приближенными алгоритмами, обеспечивающими получение «достаточно простых» ортогональных ДНФ или ДНФ с небольшим числом взаимных пересечений интервалов [Гуревич и др., 1974; Журавлев, 1974].

Таким образом, если для вычисления расстояний $\rho_j(a_j, b_j)$, $j = \overline{1, N}$, существует эффективный алгоритм и число операций при одном таком вычислении не превышает некоторой величины Q , то число операций при вычислении всех величин $\Gamma(\omega', \Omega_i)$, $i = \overline{1, m}$, не превышает $2QNm$. Число операций при распознавании одного объекта в фиксированном алгоритме класса АВО пропорционально «площади» таблицы $T_{N,m}$ с коэффициентом пропорциональности, не превышающим $2Q$. Обоснование сведения задачи построения экстремальных алгоритмов класса АВО к отысканию экстремумов функции многих переменных выполнено в [Журавлев, 1971]. Для оптимизации могут быть применены методы переборного типа (при небольшом числе параметров), градиентного типа или случайного списка.

В последнее время были изучены модели АВО с опорными множествами, задаваемыми на $T_{N,m}$ локальными окрестностями невысоких порядков — так называемыми двухиндексными опорными множествами (подмножество признаков задается пересечением нескольких строк и столбцов таблицы обучения).

Алгебраический подход к задаче распознавания

В алгебраическом подходе используются особенности строения любой процедуры распознавания. При решении конкретной задачи распознавания рассматриваются информация, на базе которой происходит распознавание, перечень классов и объекты ω', \dots, ω^q , для которых надо решить, к каким из заданных классов они принадлежат. Любой алгоритм распознавания должен по стандартной информации вида (2) и описанию $I(\omega)$ определить информационный вектор этого объекта (1), т. е. вычислить значения предикатов $P_i(\omega \in \Omega_i)$, $i = \overline{1, m}$.

Таким образом, любой алгоритм распознавания переводит задачу распознавания z с q распознаваемыми объектами и m классами в матрицу ответов — информационную матрицу, строками которой являются информационные векторы для каждого из распознаваемых объектов ω^j , $j = \overline{1, q}$:

$$\begin{pmatrix} \alpha_{11} & \dots & \alpha_{1i} & \dots & \alpha_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_{j1} & \dots & \alpha_{ji} & \dots & \alpha_{jm} \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_{q1} & \dots & \alpha_{qi} & \dots & \alpha_{qm} \end{pmatrix}, \quad (23)$$

где элемент матрицы α_{ji} принимает значения 1, 0 или Δ и указывает значение, которое вычислил распознающий алгоритм для свойства $P_i(\omega^j)$, $i = \overline{1, m}$, $j = \overline{1, q}$.

Так как применение разных распознающих алгоритмов к одной и той же задаче z может приводить к получению различных информационных матриц, возникает проблема построения корректирующих методов, которые позволяли бы, сопоставляя информационные матрицы различных алгоритмов, вырабатывать единую матрицу ответов, причем по возможности с минимальным числом ошибок. Это непростая проблема, поскольку не существует удобных операций над элементами «да», «нет», «не знаю» (1, 0, Δ), обладающих естественными свойствами типа ассоциативности, коммутативности и т. д. [Журавлев, 1978а]. Поэтому выполнение корректирующих операций над информационными матрицами невозможно.

Перезод исходной обучающей информации в информационную матрицу ответов специфичен. Например, в статистических алгоритмах распознавания сначала формируется матрица вероятностей $\|P_{ji}\|_{q \times m}$, где P_{ji} — вероятность принадлежности j -го объекта i -му классу, а затем по этой матрице вероятностей строится матрица окончательных ответов; в АВО сначала строится матрица оценок (голосов) $\|\Gamma_i(\omega^j)\|_{q \times m}$, где $\Gamma_i(\omega^j)$ — числовая оценка принадлежности j -го объекта i -му классу, а затем по этой матрице принимается окончательное решение о принадлежности распознаваемого объекта определенному классу.

Процесс переработки распознающим алгоритмом исходной информации в информационную матрицу ответов распадается на две последовательные стадии. На первой исходная информация переводится в числовую матрицу стандартного размера с числом строк, равным числу распознаваемых в задаче z объектов, и числом столбцов, равным числу классов, рассматриваемых при решении z . Вторая стадия состоит в переработке этой числовой матрицы в матрицу окончательных ответов с тем же числом строк и столбцов.

Разделение процесса переработки информации в распознающем алгоритме на две стадии обосновывается теоремой, утверждающей, что всякий алго-

ритм распознавания можно представить в виде двух последовательно выполняемых алгоритмов [Журавлев, 1978а]. Первый алгоритм В переводит обучающую информацию и описания распознаваемых объектов в числовую матрицу размерности $q \times m$, а второй алгоритм С переводит последнюю в матрицу ответов, составленную из символов 1, 0 и Δ и имеющую ту же размерность.

Алгоритм С можно сделать одинаковым для всех распознающих алгоритмов. Это — пороговое решающее правило с положительными порогами

$$C(\|c_{ji}\|_{q \times m}) = \|C(a_{ji})\|_{q \times m}, \quad (24)$$

т. е. решающее правило применяется к числовой матрице $\|a_{ji}\|_{q \times m}$ поэлементно;

$$C(a_{ji}) = \begin{cases} 1, & a_{ji} > d_2, \\ 0, & a_{ji} < d_1, \quad 0 < d_1 < d_2, \\ \Delta, & d_1 \leq a_{ji} \leq d_2, \end{cases} \quad (25)$$

где d_1 и d_2 — произвольные фиксированные положительные числа.

Основная часть процесса переработки информации приходится на алгоритм В — распознающий оператор, который переводит исходную информацию для данной задачи z в числовую матрицу стандартных размеров, т. е.

$$B(I, \Omega^m, \omega^q) = \|a_{ji}\|_{q \times m}. \quad (26)$$

Если имеется несколько или семейство распознающих алгоритмов, то, исключив из рассмотрения составляющую С, получим набор соответствующих распознающих операторов. На них как на отображениях на матрицы стандартных размеров можно определить операции сложения, умножения и умножения на число.

Пусть B_1 и B_2 — распознающие операторы:

$$B_1(I, \Omega^m, \omega^q) = \|a_{ji}^1\|_{q \times m}, \quad B_2(I, \Omega^m, \omega^q) = \|a_{ji}^2\|_{q \times m}. \quad (27)$$

Суммой распознающих операторов B_1 и B_2 называется распознающий оператор $B^+ = B_1 + B_2$, определяемый соотношением

$$B^+(I, \Omega^m, \omega^q) = \|a_{ji}^1 + a_{ji}^2\|_{q \times m}. \quad (28)$$

Применение распознающего оператора B^+ к задаче z состоит в применении к этой задаче распознающих операторов B_1 и B_2 и поэлементном сложении полученных матриц.

Аналогично определяется произведение $B(\cdot) = B_1 \cdot B_2$ распознающих операторов B_1 и B_2 :

$$B(\cdot)(I, \Omega^m, \omega^q) = \|a_{ji}^1 \cdot a_{ji}^2\|_{q \times n}. \quad (29)$$

В этом случае используется не стандартное, а поэлементное перемножение соответствующих матриц.

Умножение распознающего оператора B на число определяется как

$$(c \cdot B)(I, \Omega^m, \omega^q) = \|c \cdot a_{ji}\|_{q \times m}. \quad (30)$$

Введенные операции сложения, умножения и умножения на число распознающих операторов обладают всеми свойствами сложения и умножения чисел, а именно коммутативностью, ассоциативностью, дистрибутивностью и т. д. Если задано семейство алгоритмов распознавания $\{A\}$, то согласно теореме о представлении алгоритма распознавания можно задать семейства распознающих операторов $\{B\}$ и решающих правил $\{C\}$. Замыкание $L\{B\}$ операторов семейства $\{B\}$ операциями (28) и (30) называют линейным замыканием $\{B\}$, а множество

$$L\{A\} = L\{B\} \cdot \{C\} \quad (31)$$

— линейным замыканием семейства алгоритмов $\{A\}$. Операторы линейного замыкания представим через операторы B_g , принадлежащие $\{B\}$, т. е.

$$a_1 B_1 + \dots + a_g B_g + \dots + a_k B_k, \quad (32)$$

где a_g — действительные числа, $g = \overline{1, k}$.

Замыкание $\mathfrak{R}\{B\}$ операторов семейства $\{B\}$ относительно операций (28)—(30) называют алгебраическим замыканием $\{B\}$, а множество

$$\mathfrak{R}\{A\} = \mathfrak{R}\{B\} \cdot \{C\} \quad (33)$$

алгебраическим замыканием семейства алгоритмов $\{A\}$. В таком случае можно перейти к записи операторных многочленов, т. е. выражений вида

$$\Sigma C_{g_1} \dots C_{g_k} \cdot B_{g_1} \dots B_{g_k}, \quad (34)$$

где C_{g_1}, \dots, C_{g_k} — константы; B_{g_1}, \dots, B_{g_k} — распознающие операторы исходных распознающих алгоритмов.

Совокупность операторов из $\mathfrak{R}\{B\}$, представимых такими многочленами степени не выше k , называют алгебраическим замыканием $\mathfrak{R}_k\{B\}$ множества $\{B\}$, $k = 2, 3, \dots, v, \dots$, а множество

$$\mathfrak{R}_k\{A\} = \mathfrak{R}_k\{B\} \cdot \{C\} \quad (35)$$

— алгебраическим замыканием степени k семейства $\{A\}$ алгоритмов распознавания.

Введение операций над распознающими операторами позволяет расширять исходную совокупность распознающих операторов, а следовательно, и распознающих алгоритмов. Действительно, если $L(B_1, \dots, B_k)$ — операторный многочлен, то $L(B_1, \dots, B_k) \cdot C(d_1, d_2)$ — распознающий алгоритм. В данном случае символ произведения означает последовательное выполнение соответствующих частей алгоритма; $C(d_1, d_2)$ — пороговое решающее правило с порогами d_1 и d_2 .

Расширенные таким способом распознающие алгоритмы обладают очень сильными корректирующими свойствами. Доказано [Журавлев, 1978а; Журавлев и др., 1980], что даже в тех случаях, когда в исходном семействе распознающих алгоритмов отсутствует алгоритм, правильно решающий данную задачу распознавания z , при выполнении просто проверяемых допущений относительно исходной обучающей информации и описания распознаваемых объектов в расширении существует искомым алгоритм, причем он может быть выписан в явном виде.

Для иллюстрации рассмотрим в качестве исходного множества некорректных распознающих процедур (эвристика) совокупность АВО. Алгоритмы этого класса задаются параметрами $\epsilon_1, \dots, \epsilon_N$ (порог точности для признаков), p_1, \dots, p_N (веса признаков), $\gamma_1, \dots, \gamma_m$ (веса объектов в таблице обучающей информации), l (число признаков в опорных множествах алгоритма — этот параметр используется при задании лишь некоторых разновидностей АВО).

Рассмотрим задачу распознавания z с обучающими объектами $\omega_1, \dots, \omega_r, \omega_r = (a_{r1}, \dots, a_{rm})$, $r = \overline{1, r_m}$, классами Q_1, \dots, Q_m , которые могут быть пересекающимися; решение z предполагает распознавание объектов $\omega^1, \dots, \omega^q$, $\omega^j = (b^{jN}, \dots, b^{jN})$, $j = \overline{1, q}$. Пусть также в множестве возможных значений признака x_t , $t = \overline{1, N}$, введен способ измерения расстояния $\rho_t(y, z)$.

Установлено, что при выполнении естественных условий в алгебраическом расширении может быть явно построен распознающий алгоритм, правильно решающий заданную z . Распознающий оператор этого распознающего algo-

$$R_A = (d_1 + d_2) \sum_{j=1}^q \sum_{i=1}^m C_{ji} B_{ji}^{k_{ji} i}. \quad (36)$$

В этом выражении для распознающего алгоритма все обозначения сохраняют свой смысл, а k_{ji} — степень операторного многочлена (порядок алгебраического замыкания). В качестве распознающих операторов B_{ji} выступают фиксированные распознающие операторы из класса АВО и, следовательно, каждый из распознающих операторов полностью описан, т. е. заданы все параметры $\varepsilon(j, i)$, $\rho(j, i)$, $\gamma(j, i)$ и $k(j, i)$. Значения параметров вычисляются с помощью специальных процедур по обучающей информации I и описаниям распознаваемых объектов. Каждый распознающий оператор B_{ji} задается набором из $2N + r_m + 1$ чисел. Величина k_{ji} также вычисляется по обучающей информации и описаниям распознаваемых объектов, но во всех случаях [Журавлев и др., 1980]

$$k_{ji} \leq \{(\ln g + \ln m + |\ln(d_1 + d_2)| - |\ln i|) [|\ln(1 - (q + m - 2 - \eta)^{-1})|^{-1} + 1], \quad (37)$$

где η — сколь угодно малое число. Показатель степени k_{ji} можно понизить до $q-1$, но при этом исчезает возможность простой записи распознающих операторов.

Константы C_{ji} не всегда удастся определять только на основе обучающей информации. В таком случае вся исходная информация делится на две части: объекты, включенные в первую часть, рассматриваются как материал обучения, остальные объекты образуют распознаваемый массив $\omega^1, \dots, \omega^q$. На самом деле для последних объектов известна принадлежность к заданным классам. В этом случае считается, что $C_{ji} = 1$, если $\omega^j \in \Omega_i$, и $C_{ji} = 0$, если $\omega^j \notin \Omega_i$.

В алгебраическом подходе доказываются, что распознающий оператор типа R_A сохраняет классификацию, если объекты $\omega^1, \dots, \omega^q$ заменяются объектами из их окрестности. Это означает, что алгоритм с распознающим оператором типа R_A имеет ненулевой радиус устойчивости классификации. Отсюда несложно выводится теорема о том, что если границы классов достаточно гладки и число q достаточно велико, то алгоритм с распознающим оператором типа R_A дает правильный ответ почти всегда, т. е. с вероятностью, близкой к 1.

Этот факт верен, если выполнена так называемая *гипотеза компактности*. Если информации о строении классов нет и содержательные соображения не дают оснований полагать, что требования гипотезы компактности выполнены, то с помощью аппарата [Вапник и др., 1974] можно доказать, что распознающий алгоритм с распознающим оператором типа R_A дает правильный ответ на всей совокупности допустимых объектов распознавания с гарантированной вероятностью. Эта вероятность тем ближе к 1, чем больше величина q — число объектов, зачисляемых во вторую часть разбиения обучающего массива, которую называют *контрольной*.

При разбиении совокупности объектов с известной классификацией на две части контрольную группу целесообразно делать по возможности большей. При этом необходимо обеспечить выполнение двух условий: на обучающей информации классы должны различаться; для всякой пары объектов ω^u, ω^v из контрольного массива должен существовать хотя бы один объект ω_r из обучающего массива и хотя бы один признак x_t , $r=r(u, v)$, $t=t(u, v)$, такие что

$$\rho_t(a_{rt}, b_{ut}) \neq \rho_t(a_{rt}, b_{vt}). \quad (38)$$

Другими словами, это означает, что для любых объектов из контрольного массива в обучающем массиве найдутся объект и признак, такие, что некий способ измерения расстояния ρ_t в множестве значений x_t -го признака дает различные значения для расстояний от найденного обучающего объекта до первого и второго контрольных.

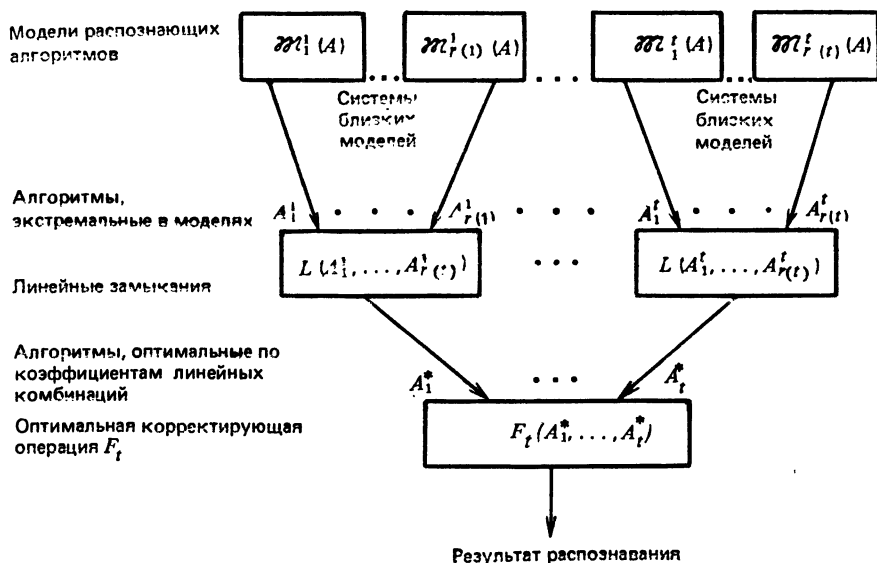


Рис. 4.2

На основе изучения различных моделей алгоритмов оказалось возможным сформулировать общее определение распознающего или классифицирующего алгоритма и изучить свойства множества таких алгоритмов. Множество распознающих алгоритмов является алгеброй, причем операции этой алгебры обладают набором свойств, позволяющих изучить множество распознающих алгоритмов. С помощью алгебраических методов решены задачи нахождения базисов в совокупности алгоритмов распознавания, выяснены условия, при которых модель содержит алгоритм, который абсолютно точно классифицирует любую заданную конечную выборку объектов, а также эффективно решена задача выбора экстремального алгоритма не для отдельной модели, а для множества распознающих алгоритмов. Если набор нестрогих алгоритмов имеет систему просто проверяемых свойств, то его можно исполнить с помощью формальных процедур; в пополненном наборе алгоритмов будет существовать алгоритм, дающий точное решение каждой задачи данного вида.

Введение дополнительных ограничений на классы решаемых задач и множества эвристических алгоритмов позволяет не только доказать теорему существования, но и указать методы поиска в пополнениях таких корректных алгоритмов. Следовательно, вместо построения формальных моделей в областях, плохо поддающихся формализации, достаточно построить семейство «интуитивно разумных» алгоритмов, затем ввести алгебру на множестве решаемых задач и построить алгебраическое замыкание «интуитивного» семейства алгоритмов. В этом замыкании принципиально разрешима любая задача из множества задач, связанных с исследованием плохо формализованных ситуаций. При этом совокупность задач должна быть формально описана: задано множество исходной информации и множество вопросов (предикатов), на которые следует ответить в процессе решения задач.

Основной результат алгебраической теории распознающих алгоритмов состоит в том, что для описания класса алгоритмов, правильно классифицирующих конечную выборку по всем классам, достаточно взять любую полную модель, рассмотреть линейное замыкание совокупности ее распознающих опера-

горов и присоединить к ней любое корректное решающее правило. Такой алгоритм окажется оптимальным по любому функционалу качества распознающего алгоритма.

Модель распознающих алгоритмов является *полной*, если в ее линейном замыкании существуют такие распознающие операторы, что матрицы, полученные из начальной информации в результате их применения, образуют базис в пространстве числовых матриц. Решающее правило является корректным, если для любой конечной совокупности допустимых объектов существует матрица числовых оценок (получаемая в результате применения распознающего оператора), такая, что применение к ней этого решающего правила переводит ее в истинную информационную матрицу окончательных ответов.

Большая часть построений в рамках алгебраического подхода выполнена в расширениях модели АВО, однако аналогично они могут быть проведены для любых моделей, расширения которых являются полными и, следовательно, корректными.

Схема применения алгебраического подхода для решения задач распознавания представлена на рис. 4.2 [Журавлев, 1976] и состоит из следующих этапов: формирование моделей распознающих алгоритмов; выбор экстремальных алгоритмов в модели; разбиение экстремальных алгоритмов на классы; построение в каждом классе линейного замыкания; выбор алгоритма, оптимального по коэффициентам линейной комбинации и параметрам решающего правила; выбор оптимальной корректирующей операции и применение ее для корректировки оптимальных алгоритмов.

4.3. Распознавание изображений

Суть проблемы

Изображение как форма представления информации играет важную роль в современной науке и технике. Оно выступают и как результат, и как объект исследований (или измерений) в физике, астрономии, космонавтике, геологии, метеорологии, картографии, биологии, медицине и др.

При представлении исходной информации в виде изображений принято выделять три основных направления: машинную графику, обработку изображений и распознавание зрительных образов (изображений), которые в значительной мере пересекаются. В этой связи начинают говорить о новой области информатики — распознавании и анализе изображений [Павлидис, 1986].

Задачи машинной графики заключаются в порождении изображений по исходной информации, представленной не в виде изображений. Приложения машинной графики разнообразны, причем сложность программ и вычислительных процедур зависит от конкретной задачи. Так, речь может идти (в порядке возрастания сложности) о воспроизведении графиков функций или экспериментальных зависимостей, композиции графического представления для машинных игр и синтезе сцен для пилотажных тренажеров. В первом случае изображения (графики) не изменяются во времени, во втором — изображения динамические, а в третьем — сцены, воспроизводимые в тренажере, не только изменяются во времени, но и должны создавать иллюзию глубины. Интенсивно развивается машинная живопись и синтез мультипликационных фильмов на ЭВМ.

Обработка изображений связана с решением задач, в которых как входной, так и выходной информацией являются изображения. Примерами таких задач служат сжатие изображений и устранение шума, улучшение качества размытых изображений и изображений, снятых с недодержкой или передержкой, сведение изображений, характеризующегося значительным диапазоном яркостей, к изображению с двумя уровнями яркости, получение нового изображения на основе нескольких или множества заданных (например, синтеза изображений поперечных сечений тела человека по латеральным рентгенограммам).

Задачи распознавания изображений заключаются в получении описания заданного изображения или отнесении предъявленного изображения к одному из заданных классов. Распознавание изображений — это в определенном смысле задача, обратная задаче машинной графики. При решении задачи распознавания изображение задается, в процессе распознавания оно преобразовывается в некоторое «абстрактное» описание (набор чисел, набор символов, граф, ДНФ, множество порождающих процедур и т. д.). Последующая обработка должна заканчиваться зачислением исходного изображения в один из известных классов. Примером такой задачи может служить обнаружение аномалий на рентгенограмме или создание читающего автомата для автоматической сортировки почтовой корреспонденции — устройство должно обеспечивать считывание с конвертов цифрового кода (обычно написанного стилизованными цифрами) и его «распознавание».

В проблеме распознавания изображений возникли внутренние научные задачи, представляющие значительный интерес. В первую очередь это задачи, связанные с алгебраизацией теории распознавания изображений, структуризацией алгоритмов распознавания изображений, оценкой алгоритмической сложности задачи распознавания изображений, автоматизацией синтеза соответствующих эффективных процедур, формализацией описания изображения как объекта распознавания, регуляризацией выбора системы представлений изображения в процессе распознавания и др. Эти задачи и составляют основу математической проблематики дескриптивной теории распознавания изображений, разрабатываемой для создания синтезированного комплекса методов математических средств обработки информации в задачах распознавания и анализа изображений.

Изображение как способ представления информации при распознавании

Исходная информация, используемая при формализации описаний объектов распознавания и классификации, обычно представляется в одной из двух форм: как результаты наблюдений, измерений, осмотра, экспертной оценки и т. п.; как изображения. В первом случае можно говорить об одномерных сигналах и использовании для их представления модели стандартной обучающей информации в смысле алгебраической теории распознавания, во втором — о многомерных сигналах, в частности о двух- и трехмерных изображениях, и использовании для их представления цифровых матриц. Эти две основные формы представления информации различаются в основном отсутствием (в одномерном случае стандартной информации) и наличием (в случае изображений) отношений (частичного) порядка, превращающих неупорядоченное множество характеристик (признаков) объекта в целостный объект.

Изображения обладают информационной емкостью, компактностью и наглядностью, а зрение является наиболее естественным для человека механизмом восприятия информации о внешнем мире и из внешнего мира. Это к тому же наиболее древний в эволюционном смысле способ представления и восприятия информации. В процессе восприятия изображения человек, судя по всему, не строит словесное описание изображения, но оперирует им как неким целостным образом или системой таких образов, прибегая к неязковому внутреннему представлению. При разработке методов и систем автоматизированного распознавания изображений приходится отыскивать способы эффективной формализации изображений для того, чтобы иметь возможность работать с представлениями (описаниями), отражающими семантику изображения, информацию, заключенную в его внутренней структуре и структуре внешних связей части реального мира (сцены), воспроизводимой с помощью изображения.

Изображения обычно несут большой объем информации, являющейся в ряде случаев избыточной. Во-первых, они обладают статистической избыточностью, заключающейся в том, что элементы изображения, расположенные близко друг к другу, имеют близкие значения яркости. Во-вторых, изображения обладают психовизуальной избыточностью, заключающейся в том, что часть

информации, переносимой изображением, может быть исключена (потеряна) без последствий для восприятия этого изображения человеком. Эта избыточность связана с особенностями зрительного восприятия человека. В-третьих, существует избыточность, определяющаяся целями обработки и анализа изображений. Скажем, при использовании изображений в диагностических задачах требуется не представить, обработать и проанализировать все изображение в целом, а либо отыскать на нем «объект» определенного вида, либо убедиться в отсутствии такового. Другой вид избыточности последнего типа определяется «семантической» природой изображения, позволяющей использовать контекстную информацию при обработке, анализе и распознавании (учет запрашиваемых отношений порядка, частичных отношений порядка и тому подобных условий и ограничений, отражающих особенности физической и логической организации реального мира). В-четвертых, существует избыточность, свойственная временным последовательностям изображений и выражающаяся в повторении на них не изменяющихся во времени деталей.

При работе с изображениями возникают четыре основные проблемы: получение формализованных описаний (моделирование) изображений; разработка и выбор математических средств обработки, анализа и распознавания изображений; программная, архитектурная и аппаратная реализация математических методов работы с изображениями; оценка качества изображений.

Специфика задач анализа и распознавания изображений определяется необходимостью достижения компромисса между противоречиями, отражающими требования к анализу, природе зрительного восприятия, способы получения, формирования и воспроизведения изображения и существующие математические и технические возможности работы с ними. Основным является противоречие между природой изображения и анализом, основанным на применении формального аппарата (в сущности, модели) объекта: для использования преимуществ представления информации в виде изображения необходимо придать этой информации «неизобразительный» вид, ибо соответствующие алгоритмы приспособлены для переработки символических описаний. Изображения по природе своей прекрасный объект для применения параллельных методов обработки, однако большинство известных методов распознавания имеют последовательный характер, в частности потому, что организация распознавания предусматривает учет результатов, получаемых на промежуточных этапах.

При формализации описания изображения рассматриваются два способа представления: растровые представления, воспроизводящие пространственную информацию, т. е. числовые матрицы, и представление через систему объектов, выделяемых на изображении. Поскольку реализация некоторых процедур обработки и анализа изображений привязана к виду представления изображения (так, поточечные операции могут выполняться при растровом представлении, операции, определенные на множествах точек (областях), — при растровом и контурном представлении, операции над линиями — при контурном представлении), то при работе с одним изображением иногда приходится попеременно иметь дело с несколькими его моделями. Хранение и передача изображений связаны с их сжатием и кодированием, что, в свою очередь, налагает ограничения на способы представления изображений.

До сих пор отсутствуют систематические математические методы формализации и анализа изображений (см. § 6.2). Подавляющее большинство методов работы с изображениями являются эвристическими, и достоинства их определяются успешностью преодоления или «изобразительности» изображения «неизобразительными» средствами.

Таким образом, в задачах распознавания под изображением понимается информация, организованная в виде некоторой квадратной числовой матрицы размером $2^{14} \dots 2^{18}$, записанной на машинный носитель и воспроизводящей свойства изображаемого объекта (сцены) и деформации, которые связаны со способом и процессом получения изображения (в частности, освещением и позицией «наблюдателя»). Число значений, которые может принимать каждый элемент этой матрицы, равно некоторой целой положительной степени числа 2

(большей 1 и, как правило, не большей 10). Под обработкой изображения понимается применение к нему системы преобразований, приводящих изображение к виду, удобному для распознавания или улучшающего качество изображения. Под распознаванием изображения понимается применение к нему системы преобразований, обеспечивающих извлечение из изображения полезной информации о свойствах изображаемого объекта или процесса.

Классы и преобразования изображений

По способу представления и обработки выделяются четыре класса изображений [Павлидис, 1986].

Класс 1: тоновые и многоцветные изображения (телевизионные изображения). Изображения этого класса являются наиболее точным представлением реального мира. Задаются они матрицами с целочисленными элементами, причем в большинстве задач размер этих матриц велик (наиболее распространены матрицы 512×512). Поэтому для хранения изображений используются более эффективные структуры данных.

Цветные изображения представляются либо тремя матрицами (по одной для красного, зеленого и синего цветов), либо одной матрицей, в которой различные биты каждого элемента соответствуют определенному цвету. Цветные изображения можно рассматривать как матрицы, образованные трехмерными векторами.

Класс 2: бинарные и выполненные в нескольких цветах изображения (печатный текст). Бинарные (черно-белые) изображения могут представляться матрицами с затратой одного бита на элемент, а также в виде карт, поскольку они содержат явно выделяющиеся области одного цвета. Именно поэтому эти два типа изображений объединены в один класс, хотя матричное представление удобно лишь для бинарных изображений.

Класс 3: непрерывные кривые и прямые (контурные областей, сигналы и графики).

Класс 4: точки и многоугольники — множества изолированных точек, стоящих друг от друга достаточно далеко. Отличие этого класса от предыдущего определяется формой представления. Изображения этого класса чаще всего используются в машинной графике, причем несмотря на то, что воспроизводится изображение классов 1 или 2, внутреннее представление относится к данному классу. Такая ситуация типична для анализа описаний трехмерных объектов, проектируемых на плоскость экрана для получения воспроизводимого изображения.

В прикладных задачах используются в основном следующие виды изображений четвертого класса:

- 1) многогранники, аппроксимирующие поверхности, причем обычно грани треугольные; проекции изображения представляют собой многоугольники;

- 2) кривые, аппроксимирующие поверхности: на поверхности твердого тела вычерчиваются кривые, описания которых используются затем в качестве проекций, представляющих собой изображения класса 3;

- 3) участки поверхностей высшего порядка; эти изображения близки к изображениям вида 1) с тем отличием, что в качестве элементов, образующих поверхность объекта, используются не плоские многоугольники, а участки поверхности высшего порядка.

Введение рассмотренных выше классов изображений позволяет описывать процедуры обработки и анализа изображений как преобразования, переводящие изображения из класса в класс или (несколько реже) не изменяющие их классовую принадлежность [Гуревич, 1985].

Между классами изображений выполняются следующие преобразования.

Переход из класса 1 в класс 2 представляет собой сегментацию и позволяет выделить области с приблизительно одинаковыми цветами и (или) яркостями.

Переход из класса 2 в класс 3 обеспечивается двумя основными видами преобразований — построением контура и утончением. Первое отображает об-

ласть в замкнутую кривую, а второе — в граф, который называют остовом области.

Переход из класса 3 в класс 4 обеспечивается преобразованиями, определяющими критические точки контура (сегментацией кривых). В случае многоугольников критическими точками являются вершины.

Переход из класса 4 в класс 3 обеспечивается преобразованиями типа интерполяции (проведение гладкой кривой через ряд точек) и приближения (проведение гладкой кривой вблизи заданных точек).

Переход из класса 3 в класс 2, если входное изображение представляет контур, обеспечивается преобразованием «заполнение», которое часто принимает форму преобразования «затенение», когда яркость или цвет области изменяются в соответствии с известным законом; если входное изображение представляет остов, то область может быть восстановлена с помощью преобразования «расширение».

Переход из класса 2 в класс 1. Представление изображения малым числом цветов часто неприемлемо из эстетических соображений, так как глаз человека легко обнаруживает контуры. Сгладить такое изображение можно низкочастотной фильтрацией или добавлением низкочастотного шума.

Обычно переходы $1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 4$ происходят при распознавании, а переходы $4 \rightarrow 3$, $3 \rightarrow 2$, $2 \rightarrow 1$ — при решении задач машинной графики. В общем случае распознавания (а также обработка) изображений включает преобразования в обоих направлениях, а также внутри классов. Так, улучшение качества изображения обеспечивается преобразованиями, не изменяющими классовую принадлежность, а сжатие обычно приводит к переходу из класса 1 в класс 2.

Важный класс задач предусматривает реализацию преобразований, связывающих двумерные изображения и трехмерные объекты. Проекцией называют преобразование, переводящее трехмерный объект в двумерное изображение или двумерное поперечное сечение объекта в одномерный набор. Обратным проектированием называют преобразование, обеспечивающее восстановление твердого тела или его поперечного сечения по его проекциям. Эти преобразования широко используются в трансаксимальной томографии и машинной графике. В последней часто необходимо воспроизвести проекцию трехмерного твердого тела по его описанию. Хотя с помощью соответствующих геометрических преобразований несложно спроектировать все точки объекта на плоскость воспроизведения, полученное изображение не будет искомой проекцией, поскольку на нем будут присутствовать точки, которые не должны быть видны из-за того, что на самом деле они заслонены другими. Для решения этой проблемы вводятся преобразования по удалению скрытых линий и скрытых поверхностей.

Преобразования, не выводящие изображение за пределы класса, сравнительно просты. Так, фильтрация, обеспечивающая повышение контраста, удаление высокочастотного шума и т. д. являются примерами преобразований, применяемых к изображениям классов 1 и 2, изменение системы координат (в том числе, перенос и поворот) — примером преобразований, применяемых к изображениям классов 3 и 4. К изображениям любого класса может применяться разложение в ряд (чаще всего — в ряд Фурье).

Типы задач распознавания изображений

При работе с изображениями возникают следующие задачи распознавания:

- 1) сопоставление двух изображений в целом для установления их принадлежности к одному классу (определяется, представляют ли изображения один и тот же объект или сцену);

- 2) сопоставление изображения в целом с набором или серией последовательных (по времени) изображений, представляющих некоторый класс изображений (т. е. объектов или сцен) (цель — та же, что и в задаче 1);

- 3) задачи 1 и 2 для нескольких классов;

- 4) поиск на предъявленном на распознавание изображении некоторой ре-

гулярности/нерегулярности (объекта, ситуации), на которую следует обратить внимание, хотя она и не задавалась в априорном перечне эталонов (ассоциативный поиск; ограниченно детерминированный набор классов — задача логической и семантической фильтрации в сочетании с самообучением);

5) поиск на предъявленном на распознавание изображении регулярности (нерегулярности) фрагмента заданного вида;

6) разбиение множества изображений на непересекающиеся подмножества (задача автоматической классификации);

7) автоматическая классификация на одном изображении (разбиение изображения на однородные области, группы объектов, сегментация области, выделение признаков объектов);

8) совместное решение задач 6 и 7;

9) автоматическое выделение непронизованных элементов, характерных объектов изображения, признаков-объектов, пространственных и логических отношений для синтеза формализованных представлений и описаний изображения;

10) приведение изображения к виду, удобному для распознавания; автоматический синтез формализованных представлений и описаний изображения;

11) решение задач 9 и 10 в диалоговом режиме;

12) решение задач 1—11 при наличии динамических объектов, сложной фоновой обстановки (в том числе динамических и статических помех) и с учетом способов получения, формирования и представления изображений;

13) восстановление по фрагментам кадров в последовательностях изображений, фрагментов изображений и изображений по фрагментам, элементов траектории задачи и траектории задачи по элементам;

14) выбор и формирование траектории решения задачи распознавания изображений.

Краткая история

Задача распознавания изображений возникла в связи с разработкой читающих автоматов [Ковалевский и др., 1980]. Затем интересы теории распознавания сосредоточились на результатах наблюдений, измерений, осмотра, экспертных оценках и др., т. е. на задачах, в которых отсутствуют отношения порядка, превращающие неупорядоченное множество признаков объекта в целостный объект с определенной структурой.

Основная часть задач анализа изображений решалась в рамках проблемы машинного зрения, которая формулировалась как понимание сцены, «описываемой» изображением — некоторой матрицей пикселей. В этой проблеме выделялись такие направления, как обработка изображений, распознавание образов, анализ сцен, интерпретация изображений, оптическая обработка информации, обработка видеоинформации и понимание изображений. Главными темами этих направлений были обработка сигналов, классификация (распознавание) и понимание изображений.

Системы обработки сигналов обеспечивают преобразование исходного изображения в другое изображение, обладающее заданными характеристиками (например, обеспечивается увеличение отношения сигнал-шум или подчеркиваются отдельные детали изображения с тем, чтобы облегчить человеку его анализ). Содержание изображения при этом часто значения не имело. Эта группа задач обычно отождествляется с обработкой изображений. Применение цифровых методов, методов обработки видеосигналов и оптических методов позволяет решать эти задачи с высоким быстродействием.

Задачи классификации сводились к отнесению изображений к одному из заданных классов. С изображения снимаются значения некоторого фиксированного набора признаков. Решение о близости изображения соответствующему классу принимается методами статистической теории решений, реализуемыми в многомерном пространстве признаков. Особое внимание уделяется построению оптимальных правил принятия решений.

Для решения проблемы понимания изображений разрабатывались программы, обеспечивающие построение по заданному изображению его описания,

а также описания воспроизводимой с его помощью сцены. На начальном этапе исследований понятие «анализ сцен» использовалось для того, чтобы подчеркнуть различия между обработкой двухмерных изображений и трехмерных сцен. Для решения этой задачи требуются знания о соответствующей проблемной области, а также сложные методы анализа изображений.

В это же время предпринимались попытки построить процесс распознавания на основе понятия «образ». Они сводились в основном к следующим двум направлениям:

1) изучение «образа» как такового: что представляют собой образы разных типов, каковы эти типы, как образы строятся, формируются, возникают, какова их структура, какие подобразы можно выделять в целостном образе, насколько они могут или должны быть элементарны, какие отношения существуют на подобразах и непроецируемых образах (процесс распознавания при этом строится с учетом механизма формирования образа);

2) построение системы распознавания на основе имитации способностей человека к распознаванию, т. е. путем выявления эвристических беспереборных приемов обработки и анализа информации, собственных зрительной системе человека, и использования этих эвристик, их вариантов и обобщений в качестве алгоритмов распознавания.

Второе направление существенно зависело от успехов в области психофизиологии и психофизики восприятия, а также нейрофизиологии. В наиболее целостном виде эта методология была реализована в перцептронном подходе к распознаванию. Перцептрон был предложен в качестве относительно простой модели, воспроизводящей некоторые принципы работы мозга человека, и стал своего рода мостом между эвристическим подходом и математическим исследованием процедур обучения в системах распознавания [Розенблатт, 1965; Ивахненко, 1975]. Однако тщательное изучение этой модели показало, что круг интересных практических задач, поддающихся решению в ее рамках, довольно узок.

В дальнейшем решение задач распознавания изображений основывалось на попытках перенести в эту область методы обработки одномерных сигналов и ввести в процесс распознавания структурную информацию. Господствующим стало направление обработки изображений со свойственным ему приоритетом разработок измерительной аппаратуры и применении традиционных методов обработки сигналов. При этом «обработка» сводится к измерению и воспроизведению сигнала, а основными математическими средствами анализа служат разновидности статистических фильтров и спектральный анализ. В результате стали доминировать технические задачи и их решение выдавалось за продвижение к конечной цели — разработке систем машинного зрения (автоматизированных систем обработки и распознавания изображений).

Существующее положение

Задача распознавания изображений отличается от классической постановки математической теории распознавания. Объектами распознавания могут служить изображения, полученные в различных частях спектра излучений (оптические, инфракрасные, ультразвуковые и т. д.) разными способами (телевизионные, фотографические, лазерные, радиолокационные, радиационные и т. д.), преобразованные в цифровую форму и представленные в виде некоторой целочисленной матрицы. Для кодирования изображения обычно используются методы эффективного кодирования. В качестве средств представления изображений используются растровое представление, позиционное представление, древовидные структуры данных, эскизы Марра и фреймовые конструкции. Способ получения и формирования изображения на процесс распознавания не влияет, но может учитываться в качестве дополнительной информации.

В обработке и анализе изображений выделяют несколько уровней обработки информации. На нижнем уровне (предварительной обработки) на изображении выделяются непроецируемые элементы и простейшие признаки (яркостные переходы, ориентация яркостных переходов). На промежуточном уровне

выделяются обобщенные признаки (линии и области), а также извлекается информация о форме объектов (ориентации поверхностей и их наложении друг на друга) (уровень сегментации). На высшем уровне работа ведется с объектами, целью является получение описания сцены и в основе обработки лежит использование знаний о проблемной области.

При переходе к задачам распознавания изображений возникают математические задачи, связанные с формальным описанием изображения как объекта анализа. Последний опирается на рассмотрение параметров четырех типов: контраста, яркости (цвета), формы и текстуры. При формальном описании изображений как объектов анализа и формировании и выборе процедур распознавания изображение (его строение, структура и содержание) рассматривается как результат дескриптивных (порождающих) операций. Эти операции позволяют построить изображение из подизображений и других объектов более простой природы, т. е. непрямых элементов и объектов, выделяемых на изображении на различных этапах работы с ним. Поскольку этот способ характеристики изображения является операциональным, процесс обработки и распознавания изображения, включая построение формального описания — модели изображения, можно рассматривать как реализацию на изображении системы преобразований, которые определены на классах эквивалентности, представляющих ансамбли допустимых изображений. Из этого следует, что работа ведется с иерархией формальных описаний изображений, т. е. в процессе распознавания используются модели изображения, относящиеся к разным морфологическим уровням (многоуровневые модели), которые позволяют выбирать и изменять в процессе решения степень подробности описания объекта распознавания.

Содержание изображения характеризуется информацией трех типов: идентифицируемые объекты с хорошо определенной структурой; идентифицируемые объекты с плохо определенной структурой; неидентифицируемые объекты. При распознавании изображения должна использоваться информация, отражающая механизм формирования изображения в целом и объектов, на нем представленных.

Учет структуры изображения сводится к определению того, какие подизображения, объекты и производные элементы можно выделить на изображении, насколько они могут или должны быть элементарными, в каких отношениях пребывают. В результате удается использовать иерархическую структурную информацию, содержащуюся в изображении: изображение описывается через некоторую систему объектов, каждый объект описывается с помощью более простых объектов, последние снова описываются с помощью более простых объектов и т. д.

Вводить структурную информацию в процесс распознавания можно двумя способами. Во-первых, можно сохранить перечень признаков как основной принцип формализации, причем в описание вводятся признаки, которые характеризуют взаимосвязи (отношения) отдельных признаков; признакам присваиваются веса, указывающие степень их существенности для описания объекта; отдельные признаки объединяются в комплексы и рассматриваются как один признак.

Второй способ использует то обстоятельство, что такая информация представляет свойственную реальному миру регулярность. Математическим аналогом этой идеи служит аппарат комбинаторных регулярных структур, учитывающий иерархичность строения изображений как объектов распознавания [Гренандер, 1979—1983]. Обращение к комбинаторным структурам позволяет, оперируя ограниченными количествами производных элементов и правил комбинирования, с помощью неограниченного их применения к исходным производным элементам и результатам применения отдельных комбинаторных преобразований получать практически неограниченное разнообразие описаний.

Математическая постановка задачи распознавания изображений

Формализация задачи распознавания с помощью понятия стандартной информации (1), (2) и матриц вида (23) и (26) применима при работе с такими видами информации, как результаты наблюдений, измерений, осмотра, экспертные оценки и т. п. Модели объектов распознавания в этих случаях сравнительно просты, а требования к организации и представлению исходной информации минимальны — допустимо ограничиваться таблицами типа «признак — значение признака». Основная особенность таких моделей состоит в том, что они представляются некоторым набором характеристик, взаимоотношения которых игнорируются. При этом предполагается, что каждый объект отождествляется с некоторой точкой многомерного признакового пространства, а класс объектов представляется «компактным» множеством таких точек.

При распознавании изображений построение модели изображения как объекта распознавания становится самостоятельной задачей, разрешаемой в процессе распознавания. Изменяются, кроме того, понятия начальной и финальной информации. Процесс распознавания изображений имеет многоуровневый характер, при котором исходная модель изображения в результате применения к ней процедуры распознавания преобразовывается в модель изображения, относящуюся к другому морфологическому (масштабному) уровню; к полученной модели снова применяется процедура распознавания и т. д.; правило остановки определяется видом результатов, требующихся при решении рассматриваемой задачи.

При анализе и распознавании изображений обрабатываемая информация представляется числовой матрицей, воспроизводящей свойства изображаемого объекта (сцены) и деформации, связанные со способом и процессом получения изображений. В таком случае для формализации процесса обработки (в широком смысле) изображений следует определить три множества (моделей) изображений, на которых постулируется существование классов эквивалентности, и множества допустимых преобразований, заданных на классах эквивалентности. Введение классов эквивалентности на множестве моделей изображений отражает гипотезу, согласно которой всякое изображение обладает определенной регулярностью или смесью регулярностей различных типов. Задача анализа и распознавания сводится при этом предположению к разделению изображений, сохраняющих собственную регулярность, и изображений с нарушениями собственной регулярности (могут ставиться задачи обнаружения на изображении регулярности или нарушения регулярности определенных типов).

Рассмотрим следующую схему решения задачи распознавания изображений (рис. 4.3). Пусть I — истинное изображение изучаемого объекта. Процесс получения, формирования, дискретизации и т. д. (все процедуры, необходимые для того, чтобы с изображением можно было работать) можно рассматривать как передачу истинного изображения по каналу с помехами. В результате предметом анализа служит не истинное, а реальное наблюдаемое изображение I^* . В процессе анализа последнее должно быть классифицировано, т. е. должен быть определен его прототип в истинном классе эквивалентности K_i либо на наблюдаемом изображении I^* следует обнаружить регулярность (регулярности) заданного вида I^R .

Таким образом, можно определить множества $\{I\}$, $\{I^*\}$ и $\{I^R\}$, $\{I\} = \bigcup_i K_i$,

преобразования формирования $\{T^F\}$ и распознавания $\{T^A\}$ изображений:

$$T^F: I \Rightarrow I^*, \quad (39)$$

$$T^A: I^* \Rightarrow I^R. \quad (40)$$

Итак, распознавание изображений сводится к определению на классах эквивалентности множества $\{I\}$ алгебраических систем преобразований $\{T^F\}$ и $\{T^A\}$, применению их к наблюдаемым изображениям I^* для: а) анализа «на-

зад» — разделения изображений в соответствии с характером регулярности (восстановление истинных изображений, т. е. указание классов эквивалентности, к которым они относятся); б) анализа «вперед» — поиска на изображении I^* регулярностей определенного вида I_R и их локализации.

Такая постановка задачи позволяет определить класс процедур распознавания изображений, характеризующихся фиксированной структурой процесса, интерпретация (конкретная реализация) которой зависит от целей и типа анализа. В процессе распознавания выделяются следующие основные этапы:

1. Синтез модели наблюдаемого изображения $\mathfrak{R}(I^*)$ — приведение изображения к виду, удобному для распознавания, т. е. получение формализованного описания изображения, пригодного для обработки его соответствующими преобразованиями — алгоритмическими процедурами распознавания.

2. Логическая фильтрация изображения — преобработка изображения, обеспечивающая его предварительную классификацию, которая необходима для выбора множества преобразований $\{T^F\}^{-1}$. Предполагается существование соответствия между типом и (или) характером модели $\mathfrak{R}(I^*)$ и классом эквивалентности, определенным на множестве $\{T^F\}$, а также наличие слабой эквивалентности на множестве $\{T^F\}$, что дает возможность сопоставлять в смысле такой слабой эквивалентности подмножества $\{T^F\}$ с классами эквивалентности моделей истинных изображений $K_i[\mathfrak{R}(I)]$.

3. Установление класса эквивалентности истинного изображения $K_i[\mathfrak{R}(I^*)]$, порождающего наблюдаемое изображение I^* . Для этого к модели $\mathfrak{R}(I^*)$ применяются обратные преобразования формирования $\{T^F\}^{-1}$. Кроме того, на основе анализа этапа 2 выдвигается гипотеза о классе эквивалентности, истинном для $\mathfrak{R}(I^*)$. Это дает возможность применять к модели истинного изображения-прототипа $\mathfrak{R}(I^*)$ преобразования T^F для проверки допустимости порождения рассматриваемого I^* в соответствующем классе эквивалентности и сопоставлять результаты применения $\{T^F\}^{-1}$ к $\mathfrak{R}(I^*)$ и T^F к $K_i[\mathfrak{R}(I^*)]$. И те и другие преобразования могут в соответствии с методологией алгебраического подхода применяться в форме линейных и алгебраических замыканий соответствующих преобразований:

$$L\{T^F\}^{-1} : \mathfrak{R}(I^*) \Rightarrow K_i[\mathfrak{R}(I^*)]', \quad (41)$$

$$L\{T^F\} : K_i[\mathfrak{R}(I^*)] \Rightarrow \mathfrak{R}(I^*)'. \quad (42)$$

При достижении эквивалентности $K_i[\mathfrak{R}(I^*)]'$ и $K_i[\mathfrak{R}(I^*)]$, $\mathfrak{R}(I^*)$ и $\mathfrak{R}(I^*)'$ или эквивалентности промежуточных результатов преобразований процессов анализа «вперед» и «назад» процесс прекращается. Этот механизм восстановления класса эквивалентности, истинного для $\mathfrak{R}(I^*)$, называется процедурой реверсивного алгебраического замыкания [Гуревич, 1984, 1985].

4. По $K_i[\mathfrak{R}(I^*)]$ выбираются преобразования T^A , поскольку предполагается также соответствие подмножеств преобразований T^A и $K_i[\mathfrak{R}(I)]$:

$$K_i : \{T^A\} \Rightarrow T^A(K_i). \quad (43)$$

5. «Распознавание» — выявление на I^* искомым регулярностей I^R одновременным применением к $\mathfrak{R}(I^*)$ анализа «вперед» и к $\mathfrak{R}(I^R)$ преобразований $\{T^A\}^{-1}$, т. е. анализа «назад». Для установления того, может ли искомая регулярность $\mathfrak{R}(I^R)$ порождаться моделью наблюдаемого изображения $\mathfrak{R}(I^*)$, применяется процедура реверсивного алгебраического замыкания:

$$L[T^A(K_i)] : \mathfrak{R}(I^*) \Rightarrow \mathfrak{R}(I^R), \quad (44)$$

$$L[T^A(K_i)]^{-1} : \mathfrak{R}(I^R) \Rightarrow \mathfrak{R}(I^*)'. \quad (45)$$

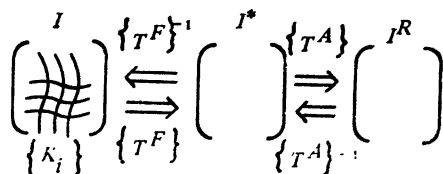


Рис. 4.3

По достижении эквивалентности $\mathfrak{R}(I^R)$ и $\mathfrak{R}(I^R)'$, $\mathfrak{R}(I^*)$ и $\mathfrak{R}(I^*)'$ или эквивалентности промежуточных результатов преобразований анализа «вперед» и «назад» процесс прекращается. При отсутствии эквивалентности выполняются новые итерации для этапов 1—5 с другими $\mathfrak{R}(I^*)$ и гипотезами о $K_i[\mathfrak{R}(I^*)]$.

Формализация понятия «алгоритм распознавания изображений»

При сведении анализа изображений к решению задачи распознавания необходимо правильно выбрать алгоритм распознавания из множества известных. При распознавании изображений, как и при решении задач распознавания со стандартной обучающей информацией, систематизация выбора наилучшего алгоритма связана с формализацией теории распознавания изображений, в частности понятия алгоритма распознавания изображений, что потребовало ввести понятие модели распознающего алгоритма.

Построение модели распознающего алгоритма и определение соответствующего класса алгоритмов распознавания не сводятся к механическому переносу в область изображений понятия модели распознающего алгоритма, выработанной в математической теории распознавания, и непосредственному использованию формальных представлений известных моделей распознавания, изученных в классической теории распознавания.

Тесная связь процессов описания и распознавания изображений, итерационность процесса распознавания, использование моделей изображения, относящихся к разным морфологическим (масштабным) уровням, означают, что алгоритм распознавания изображений включает, помимо распознающего оператора и решающего правила, оператор, осуществляющий приведение изображения к виду, удобному для распознавания:

$$R_j^n(I_n) = P_n(I, \Omega^m, \omega^j), \quad n = \overline{1, l}, \quad (46)$$

где I_n — изображение, соответствующее морфологическому уровню описания n ; $P_n(I, \Omega^m, \omega^j)$ — модель изображения на n -м морфологическом (масштабном) уровне описания, полученная в результате применения оператора R_j^n . Распознающий оператор B (26) применяется к модели P_n , а результат его применения может представлять собой модель изображения P_{n+m} , относящуюся к «нижнему» морфологическому (масштабному) уровню.

Класс процедур распознавания изображений характеризуется некоторой фиксированной структурой, которая задается последовательным применением троек операций R_j^n (46), B (26) и C (25); интерпретация процедуры, т. е. конкретные виды преобразований $\{T^F\}$ и $\{T^A\}$, определяются целями и типом проводимого анализа:

$$T^F: I_n \Rightarrow P_n(I^*), \quad (47)$$

$$T^A: P_n(I^*) \Rightarrow R, \quad (48)$$

где $P_n(I^*)$ — модель наблюдаемого изображения, соответствующая морфологическому (масштабному) уровню n ; I^R — регулярность, выявляемая на изображении в процессе решения задачи распознавания.

Дескриптивный подход к распознаванию и анализу изображений

Для задач анализа и распознавания изображений пока отсутствует единый теоретический каркас в виде алгебраического подхода. При построении математической теории распознавания изображений возникает ряд математических задач, разделяющихся на две группы: построение формального описания изображения; разработка процедур анализа и распознавания изображений. Их решение составляет основу для систематизации и автоматизации синтеза моделей представления информации (изображений и знаний) и эвристических алгоритмов преобразования информации в процессе распознавания изображений [Гуревич и др., 1983; Гуревич, 1984, 1985, 1988, Горелик и др., 1985 (гл. 6)].

Цель дескриптивной теории распознавания и анализа изображений состоит

в систематизации выбора и формировании процедур анализа изображений, обеспечивающих выделение из заданных изображений полезной информации и получение решения, точно соответствующего «информативности» этих изображений. Основными средствами достижения цели служат формализация и систематизация описаний, представлений и процедур анализа изображений. Предметом дескриптивной теории распознавания является изучение внутреннего строения изображений в зависимости от тех операций, с помощью которых изображение может быть построено из подизображений и объектов более простой природы, т. е. модели изображений и преобразования, определенные на классах эквивалентности моделей изображений, а также преобразования и выводы представлений изображений. Источниками дескриптивной теории распознавания изображений являются алгебраический подход к задачам распознавания и классификации Ю. И. Журавлева [Журавлев, 1976, 1978а, б], теория образов У. Гренандера [Гренандер, 1981] и информационная теория зрения Д. Марра [Марр, 1987].

Остановимся на принципиальных положениях дескриптивной теории.

1. Принцип порождения. В процесс распознавания включается информация, отражающая процесс формирования изображения. Выясняется структура изображения: какие элементы можно в нем выделять, насколько они могут или должны быть элементарны и в каких отношениях пребывают. Описание сложного объекта на изображении строится как иерархическая структура, образованная более простыми объектами, т. е. возникает возможность в явном виде использовать и представлять иерархическую (древовидную) структурную информацию, которую несет изображение. Применительно к распознаванию такой метод задания изображений может реализовываться как обнаружение и обработка регулярных структур. Представление изображения иерархической структурой приводит к комбинаторным регулярным структурам. Обращение к ним позволяет, оперируя ограниченными количествами атомарных («непроизводных») элементов и правил комбинирования, с помощью неограниченного (например, рекуррентного) применения последних к исходным элементам и результатам реализации соответствующих рекуррентных процедур получать неограниченное разнообразие описаний. Использование комбинаторной регулярности в качестве механизма описания структуры изображений обеспечивает экономное расходование средств описания.

Подобный способ задания («индуктивного порождения») некоторого класса объектов носит в математике название обобщенного индуктивного определения. Оно осуществляется по следующей схеме: 1) задаются исходные объекты определяемого класса; 2) задаются правила, позволяющие из уже определенных объектов получать другие объекты определяемого класса; 3) объектами определяемого класса являются только те, которые построены в соответствии с пп. 1) и 2) этого определения.

Смысл принципа порождения состоит в том, что формализованное описание изображения при распознавании задается как некоторая связанная между собой структурными отношениями система объектов, выделяемых на изображении с помощью системы преобразований и задаваемых преобразованиями, указывающими допустимый способ их построения. Эти преобразования («порождающие процедуры») обладают функциональной полнотой относительно соответствующего класса эквивалентности, индуцированного на множестве идеальных изображений.

2. Формализованное описание. В качестве исходной информации используется формализованное описание изображения — модель изображения. Все преобразования, применяемые к формализованному описанию изображения, вводятся ради достижения одной из следующих целей: получение нового формализованного описания изображения; приведение изображения к виду, удобному для распознавания; получение агрегированной оценки формализованного описания, т. е. переход от пространства исходных информационных пространств оценок, на котором реализуются процессы принятия классификационных решений при распознавании.

3. Формализованное представление изображения. Под формализованным представлением понимается формальная схема, предназначенная для получения в явном виде объектов изображения и порождающих процедур, т. е. стандартизованное формальное описание форм и поверхностей, образующих изображение. Под формализованным описанием понимается реализация формализованного представления. В качестве исходной информации при синтезе формализованных представлений изображения можно использовать значения яркостей, определяемые геометрическими свойствами (пространственной организацией и отражательной способностью видимых поверхностей, используемых при распознавании изображения, должна включать характерные объекты изображения, которым можно ставить в соответствие признаки, указывающие значения таких переменных, как яркость, ориентация, размеры, местоположение; характерные объекты должны соответствовать реальным физическим «особенностям» поверхностей. Это означает, что структура и свойства объекта изображения, воспроизводимые при помощи физических ограничений, играют центральную роль в процессах получения информации о поверхности.

Следующие четыре аспекта определяют специфику формализованного представления изображения.

Масштабная и морфологическая многоуровневость. При распознавании изображений используется система представлений, включающая несколько слоев формализованных описаний изображения, каждый из которых соответствует своему масштабному или морфологическому уровню. Разделение на морфологические уровни определяется типами и сложностью характерных объектов и производных элементов, задаваемых и порождаемых на каждом уровне. Разделение на масштабные уровни определяется масштабом характерных объектов и производных элементов, задаваемых и порождаемых на каждом уровне, а также соответствующими масштабными преобразованиями. Многоуровневость представления обеспечивает регуляризацию выбора системы порождающих преобразований и создает информационную избыточность, компенсирующую «частичность» и информационную неполноту формализованных описаний, соответствующих отдельным уровням. Одним из вариантов задания морфологических уровней служат классы Павлидиса [Павлидис, 1986].

Синтаксическая и реляционная информация. Информация, характеризующая синтаксическую структуру изображения, используется при синтезе представлений. Реляционная информация играет заметную роль при задании представления в виде формальной конструкции, поскольку обеспечивает возможность определения связей между объектами представления.

Признаки. Используются при распознавании изображений в двух качествах: как производные элементы, сборка которых посредством порождающих преобразований обеспечивает получение характерных объектов представления; как характеристики, которые ставятся в соответствие характерным объектам для того, чтобы зафиксировать изменения переменных, снимаемые с изображения и полезные, например, для определения изменений ориентации видимой поверхности относительно наблюдателя и расстояний до него. Признаки определяют соотношение используемой при формировании описания глобальной и локальной информации, содержащейся в изображении.

При работе с изображениями возникают новые типы признаков. Так, двухмерный характер объекта распознавания позволяет воспроизвести числовую величину, отражающую свойства локального участка изображения (распределение значений пикселей на этом участке; наличие или отсутствие некоторого характерного объекта на этом участке; тип формы объекта, выделяемого на участке; мощность локальной окрестности). Кроме того, к полезным локальным признакам, т. е. признакам, вычисляемым по локальной окрестности изображения, относятся характеристики, основанные: на шенноновской мере, вычисляемой для распределения типов окрестностей элементов изображения; на распределении типов булевых функций, задаваемых на окрестностях элементов изображения; на распределении типов частично определенных булевых функций, задаваемых на окрестностях элементов изображения; на числовых

оценках свойств графов связности и частично определенных графов связности однородных частей окрестности элементов изображения.

Признаки, используемые при синтезе модели изображения, классифицируются по функции (порождающие (дескриптивные) — производные элементы, признаки — характерные объекты, признакопорождающие процедуры и параметрические), по характеру отображаемой информации (глобальные и локальные), по способу получения (измеряемые или выделяемые на изображении и вычисляемые), по математическому аппарату (статистические, алгебраические, топологические, спектральные, геометрические, матричные), по типу изображений, к которым они относятся (яркостные, бинарные, текстурные), по типам объектов, представляемых с их помощью (основные, контурные, сегментационные).

Заметим, что в представлении «первоначальный эскиз» для задания информации, характеризующей изменения яркости, распределения и геометрические характеристики двумерных изображений, используются производные элементы следующих типов: пересечения нулевого уровня, пятна, обрывы, нарушения непрерывности яркостных переходов, границы, линии. В представлении «двухспловиномерный эскиз» для задания информации, характеризующей геометрические свойства видимых поверхностей (ориентация, глубина, контуры нарушений непрерывности), используются производные элементы следующих типов: локальная ориентация, расстояние от наблюдателя, нарушения непрерывности.

Модели. В процессе распознавания изображения представляются моделями двух типов — дескриптивными (порождающими) и признаковыми (параметрическими). Первые отражают структурную организацию изображения и являются той информацией, которая подвергается обработке в процессе преобразования представлений; вторые — реальные физические свойства изображаемых объектов, а также свойства числовой матрицы, которой задается наблюдаемое изображение. Признаковые модели удобнее использовать для получения агрегированных оценок формализованного описания. Дескриптивные модели скорее характеризуют процессы перехода между представлениями различных уровней, в то время как признаковые модели целесообразнее ставить в соответствие результатам преобразований представлений, относящимся к конкретным уровням.

Установление соответствий этих двух типов моделей применительно к некоторому уровню или шагу процесса распознавания может служить критерием управления ходом процесса и правилом останковки.

4. Двойственность. Процесс распознавания изображений характеризуется наличием двойственности двух типов: между формализованным описанием изображения и процедурой распознавания, а также между формализованным описанием и формализованным представлением изображения. Под двойственностью понимается свойство внутренней симметрии, присущее ряду аксиоматических теорий и выражающееся во взаимозаменяемости некоторых основных понятий. При распознавании изображений изменяются понятия начальной и финальной информации. Многоуровневость процесса распознавания приводит к тому, что он сводится к последовательности преобразований формализованных описаний изображений, относящихся к различным морфологическим или масштабным уровням. При этом формализованное описание изображений выступает в роли как начальной, так и финальной информации, поскольку при анализе изображений в качестве искомого результата может требоваться и классификационное решение, и формализованное описание, соответствующее условиям, определяемым задачами.

Двойственность второго типа связана с тем, что тип выбираемого представления в значительной мере определяет средства формирования соответствующего формализованного описания (характерные объекты, производные элементы), последнее, в свою очередь, влияет на типы формализованных представлений, включаемых в траекторию преобразований, реализующую процесс распознавания (это связано, в частности, с проблемой допустимости порождающих преобразований).

5. Характеризация процесса распознавания. Процесс распознавания изображений подчиняется нескольким основным принципам, определяющим методы формирования распознающих преобразований, структуру этих преобразований, организацию процесса в целом и модель алгоритма распознавания изображений.

А. При выборе и формировании распознающих преобразований основой является методология алгебраического подхода к задачам распознавания и классификации, включающая три основных этапа: выбор и применение базовых эвристических процедур, синтез и параметризацию модели эвристических процедур и выполнение оптимизации в модели, синтез процедуры решения задачи посредством коррекции на множестве базисных моделей эвристических процедур (с использованием механизмов линейного и алгебраического замыкания).

Б. Модель алгоритма распознавания изображений состоит в отличие от классической модели алгоритма распознавания не из двух, а из трех элементов: оператора приведения изображения к виду, удобному для распознавания (46); распознающего оператора (26), решающего правила (24).

Главной функцией оператора приведения является получение такого формализованного описания изображения, которое удобно для применения к нему распознающего оператора (26). В классе алгоритмов вычисления оценок по двумерной информации [Гуревич, 1986б] в качестве оператора приведения используется сетка дискретизации с ячейками произвольной формы (правильной), каждая из которых покрывает совокупность смежных пикселей исходного изображения. Конкретный способ подсчета оценки для локальной окрестности определяется набором структурных параметров, указывающих размер ячейки, ее форму, мощность, ориентацию, способ подсчета (правило усреднения, правило большинства, то же с учетом весов пикселей по вертикали, диагонали, горизонтали и т. д.), веса пикселей, значения порогов дискретизации. Это означает, что оператор приведения задается параметрической моделью, с которой можно работать точно так же, как и с другими моделями эвристических процедур распознавания.

Указанная тройка (46), (26), (24) является стандартным элементом процесса распознавания изображений.

В. Структура процесса распознавания имеет иерархический характер в силу двойственности, существующей между применением распознающих процедур и формализованными описаниями изображений. Преобразования представлений осуществляются не только «по вертикали» — между различными морфологическими и масштабными уровнями, но и «по горизонтали» — при оптимизации выбора формализованного представления и описания в пределах одного уровня. Таким образом, траектория процесса распознавания включает вертикальные участки (контроль за процессом осуществляется посредством преобразований по горизонтали) и горизонтальные участки (контроль за процессом осуществляется посредством преобразований по вертикали). Основным механизмом контроля служит установление соответствия между порождающими и параметрическими моделями на отдельных уровнях иерархии описания изображения.

Г. Достижение соответствия формализованных описаний, синтезируемых в процессе распознавания, а также проверка выполнения условий правила остановки процесса осуществляются с помощью механизма реверсивного алгебраического замыкания. Смысл преобразований (41) — (45) состоит в том, что процесс распознавания имеет итерационный и реверсивный характер; в результате достижение соответствия обеспечивается на промежуточном этапе (в смысле формализованных представлений и описаний) и преобразования применяются параллельно к исходным моделям и к финальным моделям, если их вид задан, либо к соответствующим моделям-гипотезам. Этот механизм используется и при вертикальных, и при горизонтальных преобразованиях иерархии формализмов изображений. При этом основным структурным элементом служит тройка операторов (46), (26), (24), а увеличение разрешающей мощности

(и сложности) преобразований осуществляется по схеме «базовая эвристика — модель эвристической процедуры — коррекция на множестве базисных моделей».

6. Логическая фильтрация. В процессе распознавания изображения интенсивно используются знания о предметной области, характере задачи, физических реальностях объекта изображения (сцены), универсальных физических, логических и математических законах, которым подчиняется и объект изображения (например, геометрические закономерности), способах получения и формирования изображения и обстоятельствах, этим процессам сопутствовавших. Эти знания используются при разбивании на классы эквивалентности, выдвижении гипотез о типе и характере финальной информации в процессе распознавания, выборе и сопоставлении базовых эвристик и моделей, формировании правил остановки и контроля процесса распознавания, выборе и назначении производных элементов и характерных объектов, типов и уровней формализованных представлений изображения. Основные формы, в которых представляются эти априорные знания, — семантическая информация и системы физических ограничений. Основным механизмом реализации этих знаний служит логический анализ, в частности логический вывод.

7. Комплекс преобразований, обеспечивающих решение конкретной задачи распознавания изображений, образует траекторию решения задачи. Формирование траектории аналогично этапу обучения или настройки алгоритма на задачу в классическом варианте распознавания. Очевидно, что для задач распознавания изображений одного класса (анализ идеальных изображений, относящихся к одному и тому же классу эквивалентности) траектории должны обладать близостью в пространстве преобразований; кроме того, должна иметь место и близость соответственных формализованных представлений и описаний. В таком случае задача синтеза траектории решения новой задачи сводится к задаче выбора оптимальной траектории из множества траекторий, поставленных в соответствие некоторому классу эквивалентности.

Другим вариантом служит задача восстановления траектории по ее фрагментам, поставленным в соответствие типам описаний, представлений и их преобразований. Эта задача может возникать в связи с проблемой эффективного использования знаний о задаче и, в частности, на этапе логической фильтрации. Пучки траекторий задачи могут служить основой для синтеза соответствующих параметрических моделей, на которых может ставиться и при определенных условиях разрешаться задача оптимизации выбора и т. д., что приводит решение задачи о синтезе траектории к схеме алгебраического подхода «эвристика — модель — коррекция».

Важнейшей проблемой дескриптивной теории распознавания изображений в настоящее время является систематизация средств описания изображений в задачах распознавания — признаков, производных элементов характерных объектов и методов синтеза формализованных представлений и описаний — порождающих процедур.

4.4. Алгоритмические базы знаний

Современная теория распознавания, как и лежащая в ее основе идеология алгебраического, а применительно к изображениям — дескриптивного подхода, является отражением стремления регуляризовать выбор и синтез с помощью обучения алгоритмических процедур преобразования и анализа информации, предназначенных для решения таких информационных задач, для которых соответствующие алгоритмы неизвестны. Эти задачи при всей функциональной и предметной разнородности имеют ряд принципиальных особенностей, позволяющих рассматривать их как новый, самостоятельный в теоретическом и прикладном отношениях, класс задач преобразования информации.

Исторически эти задачи связывают с попытками использовать опыт специалиста высокой квалификации для решения так называемых задач с плохой

структурой (диагностические, классификационные или прогностические задачи). В современной терминологии задачи такого рода относят к проблемам разработки экспертных систем, однако не следует забывать о значительном объеме работ, выполненных в связи с этими проблемами в рамках эвристического программирования в 1950—1960-е годы [Вычислительные, 1967; Фейгенбаум, 1973]. Работы этого периода, посвященные созданию экспертных систем нулевого поколения, привели к ряду интересных выводов. Оказалось, что, вообще говоря, практически можно решать задачи с «плохой» и даже противоречивой информацией, можно пользоваться эвристическими алгоритмами (эмпирической аксиоматикой), экспериментально убеждаясь в допустимости их использования. Оказалось также, что все попытки создавать универсальные методы решения указанных задач в рамках традиционной математической логики приводят к проблеме алгоритмической неразрешимости, т. е. к переборным методам решения. В сущности, достоинства эвристик и заключаются именно в том, что они позволяют в частных случаях разумно использовать доступную информацию для сокращения или упорядочивания перебора.

В дальнейшем в качестве выхода из этого положения было предложено обратиться к расширениям классической математической логики и математической статистики введением в них средств, позволяющих учитывать ненадежность и противоречивость информации и эвристичность правил вывода. Любой эмпирической аксиоме и эвристическому правилу вывода, включаемым в экспертную систему, ставится в соответствие мера (оценка), характеризующая их достоверность; точно так же оценки достоверности ставятся в соответствие цепочкам вывода.

Таким образом, первый способ расширения предусматривает структуризацию исходной информации, выбор систем эмпирических аксиом и эвристических правил вывода, сопоставление неформальных математических аксиом и правил вывода с оценками достоверности, оценивание эвристических выводов и их результатов в целом и принятие решений по оценкам.

В эти же годы вплоть до середины 1970-х годов интенсивно развивались и методы расширения, рассмотренные в данной главе. Методология второго типа расширения предусматривает выбор основных эвристических процедур, построение на основе эвристики соответствующей модели, параметризацию последней и переход к постановке и решению на эвристической модели оптимизационных задач.

В результате накопилось значительное количество разнообразных эвристических алгоритмов, каждый из которых, как правило, формулируется на некотором частном языке, не сопровождается указаниями или рекомендациями об условиях его применения или сведениями о соотношении с другими эвристическими алгоритмами. Это множество алгоритмов можно рассматривать как основу для формирования алгоритмических баз знаний современных экспертных систем. В то время как в экспертных системах нулевого поколения алгоритмические базы знаний фиксированы, это множество эвристических алгоритмов можно рассматривать как набор экспериментальных точек, каждая из которых представляет собой результат применения одного эвристического алгоритма. Такой подход позволяет ввести новый способ структуризации эвристических процедур, обеспечивающий переход к единому языку их описания. Этот переход реализуется с помощью параметризации описания алгоритмов, что, в свою очередь, позволяет «натягивать» на эвристики — экспериментальные точки алгоритмическую оболочку и решать на подобных параметрических моделях оптимизационные задачи.

После того, как в результате введения единого способа описания любой алгоритм можно представить как процедуру преобразования структурированной информации в некоторую матрицу оценок стандартного вида, возможна новая структуризация (расширение) — заданием операций над матрицами. В этом случае можно отказаться от оптимизации в параметрической модели и синтезировать новые алгоритмы в виде многочленов над базисными алгоритмами, связанными с операциями сложения, умножения и умножения на скаляр.

Таким образом, возникает проблематика систематического изучения алгоритмических баз знаний экспертных систем, связанная с автоматизацией синтеза новых алгоритмов по экспертным функционалам качества. Основными элементами этой проблематики являются сжатие пространства исходных алгоритмов, разработка средств единообразного описания алгоритмов, структуризация на основе выделения мажорантных алгоритмов и указания радиуса устойчивости и порядка в локальной окрестности [Журавлев, 1965, 1966], структуризация на основе алгебраического подхода, представляющая собой континуальное расширение конечного числа базисных алгоритмов. Другую ветвь этой проблематики составляют методы и средства формализации в задачах преобразования информации при расширении пространства допустимых исходных информаций.

Глава 5

Восприятие и первичная обработка зрительной информации

5.1. Зрительное восприятие человека и машинное зрение

В. В. Александров, Н. Д. Горский

Суть проблемы

Через зрительный канал в мозг человека поступает от 70 до 90% информации. Поэтому при создании ИС, воспринимающих информацию извне, необходимо учитывать особенности зрительного восприятия человека. Основными функциями зрения человека являются: общая ориентировка в окружающей среде (оценивание размера и формы объектов в пространстве дальности до объектов, пространственных взаимоотношений объектов, обнаружение движущихся объектов и т. д.); узнавание объектов (идентификация знакомых объектов сопоставлением их с названиями-метками); запоминание наблюдаемых сцен (кратковременная и долговременная зрительная или образная память); обучение и получение новой информации (выработка новых понятий на основе наблюдаемых образов, «называние» или оречевление новых образов).

Восприятие видеоданных в ИС логично рассматривать в такой последовательности: запоминание, выделение общей информации об окружающей среде, распознавание образов, содержательная интерпретация наблюдаемой сцены.

Прежде всего поток света попадает в сенсорное устройство, например, в виде двухмерной матрицы светочувствительных элементов, каждый из которых на выходе дает значение яркости соответствующей точки изображения. Эта яркость вместе с координатами (или номером) светочувствительного элемента называется *элементом изображения* (или *пикселом*; сокращенно *пел* от англ. picture element). Изображение, составленное из пикселов, является аналогом изображения на сетчатке глаза человека. И в том и в другом случае входной поток данных ограничен из-за конечного числа чувствительных элементов и их ограниченной разрешающей способности. Это изображение можно поместить в память ЭВМ, вывести на экран дисплея, записать на внешние носители и т. д., т. е. осуществить функцию запоминания принятого видеобраза. Однако двухмерная матрица не всегда является удобной формой для запоминания видеоданных. Например, для передачи видеоданных из одного устройства в другое используется упорядоченная последовательность элементов яркости, для опознавания объектов — их иерархическая структуризация и т. д. Формы представления видеоданных в ЭВМ могут быть разными — в зависи-

мости от задачи, которую надо решить. Преобразование одной формы в другую часто не взаимно однозначны. После нескольких преобразований изображений, как правило, невозможно восстановить исходную матрицу элементов яркости. Поэтому основной проблемой при создании ИС с элементами восприятия является поиск таких форм представления видеоданных в ЭВМ, которые были бы удобны и для запоминания (хранения) изображений, и для проведения с ними широкого круга преобразований.

Из-за невозможности построить прямые аналогии между представлением в ЭВМ и представлением в коре головного мозга приходится переходить от имитационного уровня моделирования зрительного аппарата к функциональному, т. е. к моделированию реакций системы, а не ее структуры. В то же время психологические и физиологические исследования свидетельствуют о неразрывной связи зрительных и мыслительных процессов. Отсюда возникает вторая проблема машинного зрения: построение системы представления знаний, обобщающей опыт восприятия человека, для обеспечения полноценного восприятия видеоданных.

Реализация в ИС трех основных функций зрения — ориентации, узнавания, понимания (в особенности последних двух) связана с установлением соответствия между элементами структуры, представляющей видеоданные в памяти, и элементами (фактами) базы знаний. Фактически эта большая проблема распадается на три подзадачи: выбор адекватного представления видеоданных и формы представления знаний, описывающих опыт восприятия в некоторой предметной области, а также разработка механизмов установления соответствия между образами (изображениями) и понятиями.

Краткая история

Идея о том, что глаз есть часть мозга, принадлежит Клавдию Галену, жившему в I в. до н. э. Первые модели глаза, в которых сетчатка рассматривалась как светочувствительный элемент, реагирующий на лучи, преломленные хрусталиком, были построены Ф. Платтером и И. Кеплером в начале XVII в. [Демидов, 1987]. Большую роль в исследованиях зрительного восприятия сыграли работы, в которых рассматривались процедуры, выполняемые нейронными структурами. В [Lettvin et al., 1959] исследовался глаз лягушки и было обнаружено четыре типа нейронных структур, реагирующих на определенные изображения на сетчатке: детекторы перепада яркости (края образа), детекторы движущегося края, детекторы затемнения и детекторы движущихся малых объектов [Эндрю, 1985]. Работа этих детекторов тесно связана с поведением лягушки.

Например, перемещение маленького темного предмета в поле зрения вызывает реакцию нападения, а затемнение поля зрения — реакцию бегства.

В работе [Хьюбел, 1974] отмечалось, что зрительное поле сетчатки у млекопитающих отображается на зрительные отделы коры, т. е. в коре как бы существует слепок изображения с сетчатки. Были найдены специфические нейронные комбинации: «простые» клетки, реагирующие на прямые линии определенного направления, «сложные», обнаруживающие углы, движущиеся линии, «сверхсложные», обнаруживающие отрезки прямых определенной длины. Дальнейшее развитие эти идеи нашли в [Зрительное, 1975; Глезер, 1985], где были описаны нейроны, реагирующие на решетки определенной частоты, и выдвинута гипотеза о голографическом принципе запоминания видеоданных в памяти.

Наиболее известной моделью восприятия, основанной на нейтроподобных структурах и использующей результаты работ по нейрофизиологии, стал *персептрон* [Розенблатт, 1965]. В этой модели светочувствительные элементы, моделирующие сетчатку, соединялись с А-элементами, моделирующими нейроны коры, а выходы А-элементов замыкались на вход единственного R-элемента, имитирующего решающее устройство (рис. 5.1). Веса связей настраивались в процессе обучения (предъявления образов разных классов) так, чтобы R-элемент выдавал различные сигналы при появлении

изображений, относимых к разным классам. В дальнейшем были разработаны модели более сложных перцептронов [Минский и др., 1971]. Среди других сходных устройств имитационного моделирования зрительного восприятия известны пандемониум [Selfridge et al., 1960] и матрицы [Штайнбуха, 1968]. Интересные идеи по организации обучающихся распознающих машин были выдвинуты в [Бонгард, 1967].

С развитием запоминающих устройств ЭВМ и созданием быстродействующих процессоров были разработаны алгоритмы кодирования изображений для их экономного хранения и быстрой передачи по каналам связи [Цифровое, 1980, 1981; Прэтт, 1982], улучшения качества изображений и устранения на них помех [Обработка, 1979; Ярославский, 1979], выделения на изображениях различных признаков, таких как яркие переходы, углы, периметр и площадь замкнутых областей, направление контура и т. д. [Розенфельд, 1972; Прэтт, 1982]. В то же время было показано, что работа перцептрона Розенблатта эквивалентна применению простого линейного решающего правила в пространстве признаков.

В последнее время все больше исследователей считают, что системы восприятия визуальной информации должны использовать в своей работе не только поступающие извне видеоданные, но и знания об окружающей среде — своеобразные модели мира [Психология, 1978; Прэтт, 1982; Тимохин, 1983]. Первые такие системы разделяли во времени процессы анализа изображений и их интерпретации на основе имеющихся знаний. В дальнейшем было предложено, чтобы эти подсистемы работали параллельно, непрерывно обмениваясь информацией [Rosenthal et al., 1984; Александров и др., 1986].

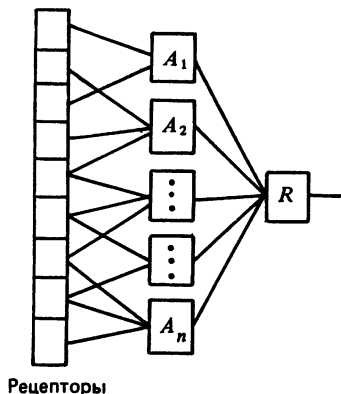


Рис. 5.1

Особенности зрительного восприятия

1. Проблемная ориентация зрения. Во многих случаях глаз ошибается в достаточно простых ситуациях. Возникают различные зрительные иллюзии [Толанский, 1975]. Часть из них предопределена генетически, другие могут вырабатываться [Рок, 1980]. Имеются основания считать, что иллюзии — следствие работы восприятия не как универсальной, а как проблемно-ориентированной в каждый момент системы. Это означает, что система «глаз — мозг» постоянно подстраивается под ситуацию и в текущий момент обладает существенно меньшим разнообразием реакций, чем потенциально она допускает [Александров и др., 1985]. Перестройка системы управляется контекстом наблюдения и состоянием самой системы. Организм постоянно ищет информацию, необходимую ему для действий, и выделяет наиболее важную в данный момент [Арбиб, 1976]. Отсюда следует, что система восприятия ИС должна обладать избирательностью и проблемной ориентацией.

Поскольку в каждый момент важны различные признаки и особенности образов, необходима автоматическая перестройка системы на текущий контекст. Информация о нем может поступать в систему как изнутри (определяться решаемой задачей), так и извне (определяться поступающими видеоданными). В последнем случае должны быть выделены признаки контекста, многие из которых можно получить на основе анализа зрительных иллюзий (например, сходящиеся в одну точку линии на изображении есть признак контекста «перспективное видение», приводящего к вполне определенному восприятию размера объектов на изображении). При столкновении с новой ситуацией перестройка восприятия должна происходить «сверху вниз» от глобального контек-

ста (модели мира) до набора локальных признаков объектов — по аналогии с живыми организмами [Бонгард, 1967].

2. Глобальность и локальность зрения. В восприятии участвует глобальная и локальная информация. В нейрофизиологических моделях [Хьюбел, 1974] опознавание объекта описывалось как последовательное объединение все более и более сложной информации о локальных пространственных признаках (путь от частного к общему). Дальнейшие исследования показали, что одним этим механизмом локальность и глобальность видения объяснить нельзя. Сейчас развивается точка зрения, что процесс идет от глобального восприятия к локальному [Величковский и др., 1979]. Это подкрепляется данными о том, что через зрительный нерв первыми передаются низкие пространственные частоты, соответствующие грубым, крупным деталям изображения, а затем высокочастотные составляющие, соответствующие мелким локальным деталям [Глезер, 1985]. Таким образом, в зрительном тракте имеет место как бы «проявление» (во времени) изображения, аналогом которого является анализ изображений в технических системах по принципу от грубого к точному [Wong et al., 1978; Sloan et al., 1979; Александров и др., 1985]. В то же время нельзя отказываться и от механизмов, основанных на анализе локальной информации, работающих аналогично иерархическому набору детекторов с восходящим потоком данных.

3. Множественность форм видения (константность восприятия). Человек узнает лицо, предмет или пейзаж и глядя на него непосредственно, и по черно-белой или цветной фотографии, и по контурному рисунку, и по картине художника. В широких пределах восприятие безразлично к размеру, яркости, контрастности, ориентации объекта. Существуют две основных гипотезы о механизме этого явления [Рок, 1980]: восстановление в мозгу путем некоторой компенсации неискаженного или эталонного изображения; восприятие по относительным характеристикам или признакам. В последнем случае практический вывод состоит в том, что идентификация наблюдаемого объекта, представленного в той или иной форме, происходит на основе как обобщенных классифицирующих признаков, характеризующих данный объект относительно классов других объектов, так и индивидуальных признаков, характеризующих данный объект относительно других объектов этого же класса. В существующих ИС используется преимущественно первый путь — компенсация различных искажений.

4. Межполушарная асимметрия восприятия. Физиологическими исследованиями [Spreng, 1973; Балонов и др., 1976; Глезер, 1985; Меерсон, 1986] установлено, что функции полушарий мозга в процессах восприятия значительно различаются. Правое полушарие связано с конкретным предметным мышлением, называемым также пространственно-образным, а левое — со зрительным обучением, т. е. с выработкой обобщенных понятий и признаков на основе зрительных образов. В системах восприятия зрительной информации в ИС необходимо реализовать оба механизма.

5. Получение и использование знаний. Восприятие предполагает непрерывный процесс уточнения и соотнесения конкретных наблюдаемых образов и абстрактных понятий, уже существующих в памяти и являющихся знаками для различных классов объектов реальности. Того, что человек никогда не видел, нельзя «узнать», т. е. описать в терминах ранее виденных объектов или соотнести с уже существующими знаниями. Эти знания могут существовать в виде обобщенных признаков, выделяемых левополушарным механизмом восприятия, а могут быть и врожденными [Лурья, 1973]. По-видимому, существуют также не выражаемые словами образные знания — результат работы правополушарных механизмов.

В системе восприятия ИС можно использовать три типа знаний: базу знаний, создаваемую разработчиком и описывающую модель мира, в котором работает система; накапливаемые знания в виде классифицирующих образы признаков и новых понятий, описывающих классы образов, — результат работы аналогов левополушарных механизмов; накапливаемые знания в виде конкретных образов и их элементов (частей) — результат работы аналогов правополушарных механизмов.

Представление видеоданных

В системах восприятия представление видеоданных — это структура данных (логическая или физическая), с которой взаимодействуют аппаратные и программные компоненты системы обработки изображений и от которой зависит сложность и эффективность алгоритмов дальнейших преобразований видеоданных и построения образной иерархии. Удачно выбранная структура данных позволяет сократить объемы передаваемой и хранимой в системе информации, избежать трудоемких операций преобразования видеоданных из одной формы в другую. Наиболее часто в системах восприятия ИС используются следующие представления видеоданных [Чукин, 1983; Павлидис, 1986]:

матричное или прямое, где каждый элемент описывает яркость или цвет элемента изображения;

векторное, где каждый элемент описывает отрезок прямой линии определенной длины и ориентации;

синтаксическое, элементами которого являются алфавит символов (простые, часто встречающиеся на изображении конфигурации типа отрезка, дуги, узла и т. п.) и объекты, конструируемые по определенным правилам из этих символов [Завалишин и др., 1974; Фу, 1977; Русын, 1986];

на основе ортогональных преобразований, где элементами являются коэффициенты разложения функции яркости по некоторой системе базисных функций [Обработка, 1979; Ярославский, 1979; Прэтт, 1982];

пирамидально-рекурсивное, где каждый элемент описывает яркость или цвет области изображения определенной формы и различных размеров (от элемента яркости до изображения в целом) [Tanimoto et al., 1975; Multiresolution, 1984; Александров и др., 1985].

На основе перечисленных форм и их комбинаций представляются и затем анализируются как двумерные, так и трехмерные изображения (см. § 5.2, 5.3).

Развиваются также морфологическое описание, основанное на выводах интегральной геометрии и математической морфологии [Кирчук и др., 1984], преобразование Хофа, дающее удобные средства работы с линиями [Дуда и др., 1976; Прэтт, 1982]. В рамках этих представлений элементарные объекты имеют помимо положения и другие характеристики (например, размер, форму).

Следующий этап — переход от внутреннего представления видеоданных к образной иерархии, которую можно рассматривать как взаимосвязанный набор признаков, деталей и частей образов, имеющих на изображении. Обычно начинают с так называемой предобработки изображения — устранения шумов и искажений в сенсорных устройствах или в потоке видеоданных, поступивших в систему; выравнивания яркости и усиления (если необходимо) контрастности; других операций, не затрагивающих семантику изображения, но субъективно (для человека) или объективно (для выполнения последующих преобразований) улучшающих его качество.

На этапе интерпретации выделенным на изображении символам и их пространственным и иерархическим отношениям должны быть присвоены именованные метки, описание которых имеется в базе знаний. При этом возможно либо построение образной иерархии и интерпретация ее элементов, либо построение модели мира и поиск на изображении признаков и символов, соответствующих исходным объектам и понятиям [Минский и др., 1976; Прэтт, 1982].

При реализации систем восприятия зрительной информации возникает множество проблем, главными из которых являются: обеспечение инвариантности восприятия окружающей среды в широком классе изменения условий наблюдения, искажений наблюдаемых объектов, видов сенсорной информации, описывающей среду, и обеспечение самообучаемости системы в процессе работы, автоматического извлечения новой информации о среде, обобщения, классификации и запоминания образов.

Несомненно, что в ближайшее время будут развиваться системы машинного зрения, в основе которых лежит функциональный подход к моделированию восприятия. Здесь должны найти решение проблемы представления и инвари-

антного описания видеоданных в ЭВМ; автоматического выбора значимых для решения поставленной задачи признаков и элементов символического описания образа; формирования и пополнения базы знаний, используемой для интерпретации изображения. Все большую роль будут играть исследования синтеза образов и изображений, моделирование процесса воображения на основе организации образной памяти и генерации изображений по их описанию.

5.2. Восприятие двумерных изображений

Н. Д. Горский

Основные определения

Восприятие двумерных изображений имеет специфику, которая отражается на применяемых методах и разрабатываемых моделях. Прежде всего считается, что наблюдаемые объекты являются плоскими, т. е. их третьим измерением (глубиной) можно пренебречь без ущерба для достижения поставленной цели. В качестве примеров таких задач можно привести распознавание букв, анализ биомедицинских видеоданных, интерпретацию спутниковых или аэрофотоснимков и т. п. Вторая особенность связана с тем, что объекты обычно предполагаются неперекрывающимися, что значительно упрощает задачу их идентификации.

Будем считать, что на входе системы восприятия изображение задано квадратной матрицей отсчетов $N \times N$ элементов яркости. Каждый элемент характеризуется координатами i, j , выраженными целыми числами, и яркостью s . Элемент обозначается $s(i, j)$. Если яркость всех элементов имеет только две градации (обычно 0 и 1), то изображение называют *бинарным* или *графическим*; если яркость может принимать много фиксированных значений, обозначающих уровни шкалы серого цвета (обычно 256), то *тоновым черно-белым*; если яркость задается вектором, каждая компонента которого соответствует интенсивности одного цвета, или спектрального диапазона, то *цветным* или *многоспектральным*.

Элемент яркости $s(i, j)$ часто рассматривается вместе с его окрестностью — соседними элементами яркости. Обычно соседними считаются такие элементы, координаты которых отличаются от (i, j) не более чем на единицу по каждому компоненту. В этом случае каждый элемент имеет восемь соседей. Иногда соседними считаются только ближайшие элементы по горизонтали и по вертикали. Если задано некоторое подмножество элементов изображения, то связными называются те элементы, между которыми можно найти цепочку соседних элементов из этого подмножества. Подмножество (группа) элементов называется *связной*, если попарно связаны все входящие в него элементы.

При анализе изображений обычно выделяют совокупности элементов с общими свойствами. Так, резкие перепады яркости, как правило, соответствуют *контурам объектов*; компактные связанные области, отличающиеся по яркости или иным характеристикам от окружающих частей изображения, часто являются различными объектами. На изображениях различают также *текстуры* — участки с однородными свойствами, в которых повторяется некоторый рисунок или структура, образуемые небольшим (относительно размеров участка) числом элементов изображения. Примерами текстур являются изображения травы, кирпичной кладки, леса, сфотографированного с большой высоты, песка, воды, ткани и т. п.

Формы представления изображений в ЭВМ

Наиболее распространенной формой представления видеоданных в ЭВМ является двумерная *матрица отсчетов*, каждый элемент которой описывает яркость или цвет элемента изображения с соответствующими координатами.

Основное достоинство матричного представления — сохранение на логическом (а часто и на физическом) уровне структуры обрабатываемого изображения — пространственной организации элементов яркости или цвета. Поэтому матричное представление видеоданных называют также *прямым*. Простота такого представления позволяет легко организовать последовательные или параллельные просмотр и обработку элементов изображения. При прямом представлении часто используются локальная обработка каждого элемента изображения на основе просмотра его соседей, например свертка с некоторым ядром в окрестности размером 3×3 , 5×5 пикселей и т. д.; локальные операции выделения контуров, определения градиента яркости и пр. На основе прямого представления созданы различные специализированные процессоры обработки изображений, в том числе векторные, матричные, конвейерные [Денисов и др., 1984; Granlund et al., 1985].

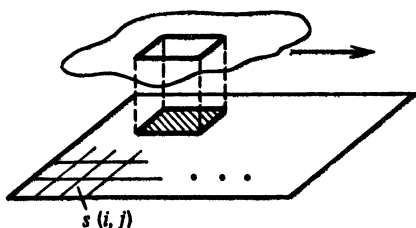


Рис. 5.2.

Многие авторы [Uhr et al., 1982; Granlund et al., 1985] отмечают трудности анализа изображений на основе прямого представления. Прежде всего они связаны с большими затратами на получение глобальных характеристик, обобщенных описаний обрабатываемых изображений. Локальные операции над окрестностями элементов эквивалентны просмотру изображения через небольшое отверстие в листе бумаги, который движется по всему полю (рис. 5.2). Такие операции удобны для выделения локальных признаков и предварительной обработки (в частности, улучшения качества, кодирования), но недостаточны для решения задач анализа, идентификации, распознавания. Для этих задач приходится переходить от прямого представления к какому-либо иному, например к *представлению набором признаков*, выделенных на основе выполнения локальных операций, *синтаксическому представлению* и т. д.

Для представления видеоданных в ЭВМ широко используется описание изображения *коэффициентами* некоторого *ортогонального* (обычно двухмерного) *преобразования*, например, Фурье, Адамара, Карунена — Лоева и т. д. В этом случае вместо яркости в памяти хранятся значения коэффициентов разложения двухмерной функции яркости по системе базисных функций. Каждый такой коэффициент является интегральной характеристикой изображения, он характеризует всю область определения функции яркости. Интегральность коэффициентов позволяет легко реализовывать глобальные операции преобразования изображения, такие как фильтрация помех, коррекция искажений и т. п. [Ярославский, 1979; Прэтт, 1982]. Коэффициенты двумерных преобразований используются и в качестве обобщенных признаков изображений, например при анализе текстур, распознавании отдельных классов изображений [Дуда и др., 1976; Ярославский, 1979; Прэтт, 1982]. Поскольку спектральный состав большинства реальных изображений резко неоднороден (преобладают низкочастотные составляющие), коэффициенты двумерных преобразований успешно применяются для кодирования и сжатия видеоданных [Цифровое, 1981; Прэтт, 1982].

В то же время при представлении видеоданных на основе двумерных преобразований затруднительно проведение локальных операций над изображением или отдельными его частями, что во многих случаях препятствует решению задач анализа изображений. Глобальные признаки, извлекаемые из спектральных коэффициентов, адекватно характеризуют лишь те изображения, которые можно рассматривать как стационарные сигналы (например, текстуры), но часто неприменимы при обработке и анализе других классов изображений, характеризующихся локальной неоднородностью, таких, например, как реальные сцены со множеством объектов. Возникают и чисто технические трудности при реализации двумерных преобразований для больших объемов входных дан-

ных, хотя за последнее время разработаны параллельные высокопроизводительные процессоры для выполнения так называемых быстрых алгоритмов дискретных преобразований Фурье, Адамара и др. [Ярославский, 1977; Параллельная, 1985; Трахтман, 1987].

Синтаксические методы представления, называемые также *структурно-лингвистическими*, основаны на описании изображений составных объектов в виде иерархической структуры более простых преобразов с использованием аппарата формальных грамматик. Так часто представляются графические изображения простых сцен, описываются объекты в системах машинной графики и автоматизации проектирования и т. д. Преимуществом синтаксического представления является описание изображения в терминах укрупненных, семантически значимых единиц, что хорошо соответствует особенностям человеческого восприятия. Это представление в наибольшей степени соответствует образной иерархии, получение которой является одной из основных целей анализа изображительной информации.

При синтаксическом представлении различают лингвистические и структурные подходы. В первом случае составной объект изображения описывается с помощью выделенного набора непроеизводных элементов (алфавита) и правил их соединения (законов построения «слов» и «фраз») [Шлезингер, 1976; Фу, 1979], например описание изображений букв на основе комбинаций отрезков линий различной формы. Во втором случае на множестве непроеизводных элементов формируются признаки, описывающие различные классы объектов [Завалишин и др., 1969; Русы, 1986]. Однако и при том и при другом подходе существуют проблемы выбора и выделения непроеизводных элементов. Как правило, адекватный их набор удается выделить лишь для узкого класса изображений (или объектов на них); трудно найти «семантически полный» базис непроеизводных элементов или признаков, позволяющий описывать или различать произвольные изображения из выбранного класса; возникает трудность с распараллеливанием процесса обнаружения непроеизводных элементов, поскольку число их в реальных задачах может быть достаточно велико. Таким образом, основные проблемы здесь связаны с самим процессом получения синтаксического представления видеоданных.

Пирамидально-рекурсивное представление объединяет ряд структур и способов организации данных (пирамиды, конусы, квадродеревья, рекуррентно-рекурсивные структуры), которые обладают общими чертами — иерархичностью и регулярностью [Васин, 1984; Multiresolution, 1984; Александров и др., 1985]. В этом случае обрабатываемое изображение описывается упорядоченной последовательностью нескольких изображений различного разрешения, сходящихся к исходному и располагаемых обычно одно под другим (отсюда термин «пирамида»). В наиболее распространенном варианте исходное изображение разбивается на равные квадратные блоки (возможно, пересекающиеся), затем процедура разбиения рекурсивно повторяется для каждого блока до тех пор, пока его размер не станет равным размеру элемента исходного изображения. Каждому блоку приписывается значение, называемое «яркостью» или «цветом» и являющееся его обобщенной характеристикой. По завершении процесса получается набор изображений, каждое из которых состоит из блоков определенной величины. Эти изображения последовательно уточняют друг друга и сходятся к исходному. Построение пирамиды может идти и обратным путем — от низших уровней к высшим путем объединения элементов (блоков).

Регулярность пирамидальных структур предопределяет их удобную реализацию и эффективное использование в ИС, например параллельную обработку [Александров и др., 1985; Fountain, 1986]. Иерархия описаний различной степени общности способствует контекстно-независимой структуризации входных видеоданных, которая может служить для моделирования индуктивных (при вводе данных) и дедуктивных (при анализе пирамиды «сверху вниз») процессов человеческого восприятия. На пирамидальной структуре удобно выделять как локальные (на нижних уровнях), так и глобальные (на верхних уровнях) характеристики изображений. Ее уровни представляют фактически исходное

изображение, пропущенное через различные фильтры, поэтому пирамида обладает теми же свойствами, что и представление на основе двухмерных преобразований [Burt, 1982; Александров и др., 1985]. Пирамидально-рекурсивная структура не является иерархией объектов, но удобна для представления видеоданных (извлечения признаков различной степени общности, идентификации образов и объектов различных уровней и размеров). В частности, пирамидально-рекурсивное представление применяется для извлечения производных элементов различных уровней иерархии при переходе к синтаксическому представлению.

Предварительная обработка изображений

Предварительная обработка изображений проводится для улучшения качества видеоданных и устранения искажений, возникающих в процессе получения и представления видеоданных в ЭВМ (например, из-за недостаточной прозрачности атмосферы при аэросъемке, движения снимаемого объекта, помех в каналах передачи данных, низкой чувствительности элементов сенсорной системы и т. п.). Задачу устранения искажений называют также *реставрацией изображений*. И улучшение качества, и реставрация изображений тесно связаны с задачей выделения информации из видеоданных, являясь первыми этапами анализа, на которых подготавливаются входные данные для алгоритмов выделения признаков.

Перечислим основные методы улучшения качества изображений и модели компенсации различных искажений (подробнее см. [Обработка, 1978; Ярославский, 1979; Прэтт, 1982]):

- изменение общей или локальной (на отдельных участках) яркости и (или) контрастности изображения на основе преобразований его гистограммы яркости;

- подчеркивание границ (перепадов яркости) на изображении для обнаружения контуров или улучшения субъективного (с точки зрения человека) качества изображения;

- подавление на изображении шумов и помех на основе различных видов фильтрации;

- коррекция геометрических искажений изображения на основе преобразования координат его элементов с целью изменения масштаба изображения, его поворота, сдвига, аффинных или проективных преобразований;

- реставрация нерезких изображений — устранение ошибок фокусировки оптической части сенсорной системы, рассеивающего влияния атмосферы;

- реставрация смазанных (вследствие движения камеры или объекта) изображений;

- компенсация искажений, вносимых зернистостью фотопленки, ограниченной разрешающей способностью устройств ввода изображений.

Преобразование значений яркостей элементов изображений часто применяется при малой яркости, недостаточной контрастности входного изображения, плохой различимости деталей «в тенях» или «в светах». При этом гистограмма яркости имеет ярко выраженный пикообразный характер или охватывает лишь небольшую часть возможного диапазона яркостей. Для улучшения качества таких изображений применяются два основных метода: задание функции изменения шкалы яркости и «выравнивание» гистограммы (всего изображения или, возможно, в отдельных зонах с малой контрастностью). Выравнивание гистограммы означает приведение ее к примерно равномерному виду (возможно, с потерей некоторого числа уровней квантования яркости), что обеспечивает улучшение субъективного восприятия изображения, повышает различимость деталей.

Характеристическое описание изображений

При характеристическом описании изображений происходит переход от набора характеристик изображения, его точек или фрагментов к описанию простейших семантически интерпретируемых частей изображения. Возможны два

пути построения такого описания: при априорно заданном наборе характеристик и при наборе, формируемом в процессе анализа изображений. Первый путь 'дает более простые процедуры описания, но требует смены набора при смене характера изображений или наличия очень большого набора «на все случаи жизни», что явно неэффективно. Второй путь связан с поиском набора характеристик исходя из содержания обрабатываемых видеоданных. Существенное его отличие от первого пути состоит в том, что нужно выделить совокупность элементов изображения, обладающих некоей общностью, а затем сопоставить с ней метку-имя. В этом случае внешней информацией, сокращающей перебор, является указание критерия однородности или общности элементов изображения, объединяемых в одну совокупность (примерами таких критериев являются: близкая яркость связной группы точек, однородность текстуры в некоторой области изображения, выравнивание в одну линию точек перепадов яркости и т. д.). Выбор критерия, как правило, возлагается на разработчика системы восприятия, который должен обладать определенным опытом.

Для формирования характеристик часто используют *сегментацию изображения* — разделение или разбиение его на связные области по сходству свойств входящих в него элементов. Сегментацию применяют и для выделения на изображении объектов (точнее, отделения их от фона), поскольку совокупности элементов изображения, соответствующие объектам, часто имеют характеристики, явно отличающиеся от характеристик элементов фона. Существует большое число методов сегментации, различающихся способом задания критерия однородности (или сходства) элементов изображения, относящихся к одному сегменту-символу или объекту.

Наиболее распространенным критерием является яркость элементов изображения. Для цветных изображений вместо яркости используют сравнение цветов элементов изображения. На тоновых черно-белых изображениях используются текстурная сегментация, когда для установления сходства элементов изображения сравниваются значения параметров текстур в их окрестностях, и контурная сегментация, когда в одну область объединяются элементы со сходными значениями и направлениями градиентов яркости. Для графических изображений популярным критерием однородности является цвет элементов, часто в сочетании с другими условиями, например с формой области (обычно ищут простые формы или заранее указывают нужную форму: квадрат, прямая и т. п.).

Среди методов сегментации выделим [Денисов, 1985]:

пороговые методы, когда в одну область объединяются все элементы, значение критерия для которых превышает установленный для данной области изображений порог;

методы наращивания областей, когда сначала выбираются некоторые точки-эталоны, а затем к ним последовательно подсоединяются наиболее похожие (в смысле заданного критерия) соседние элементы;

релаксационные методы, в которых на основе анализа окрестностей элементов изображения итеративно оцениваются вероятности их принадлежности различным сегментам;

методы аппроксимации, когда на основе априорной информации проверяется допустимость объединения элементов изображения в одну группу (например, точки контурного перепада аппроксимируются некоторой кривой).

Для описания прямых линий на изображении успешно применяется преобразование Хоха [Дуда и др., 1976; Павлидис, 1986]. Исходной для него служит контурная псевдограница изображения, а в качестве результата применения преобразования получаются коэффициенты уравнений прямых в полярных координатах, которыми можно аппроксимировать контурную псевдограницу.

Для описания выделенных на этапе формирования характеристик областей изображения применяются различные методы [Прэтт, 1982; Анисимов и др., 1983], во многом пересекающиеся с методами идентификации. При описании формы области применяется запоминание коэффициентов разложения в ряд Фурье функции кривизны границы области. При этом запоминаются особен-

ности формы (углы, дуги, прямые, разветвления, пересечения). Для этих же целей часто используются значения *центральных моментов функции яркости* (считая, что фон равномерен).

Топологические характеристики применяют для достижения инвариантности описания в различных условиях наблюдения, при возможных аффинных, проективных или перспективных искажениях формы. Например, используются значения чисел Эйлера для описываемой области, описание выпуклой оболочки и дефицита выпуклости, различные производные периметра и площади и т. д. [Русын, 1986].

Отметим, что «двухмерное зрение» часто рассматривается как предварительный этап (так называемое зрение низкого уровня) для описания трехмерных сцен, которое проводится с учетом таких важных характеристик, как ориентация поверхностей объектов в пространстве, тени от объектов, расстояния до них, их движение и т. п.

5.3. Восприятие трехмерных сцен

В. С. Шнейдеров

Отображение трехмерных сцен

Машинные методы отображения трехмерных сцен должны строиться таким образом, чтобы наиболее полно выполнялись привычные для глаз условия наблюдения, соответствующие непосредственному рассматриванию трехмерного мира, т. е. чтобы возникала иллюзия реальности.

Зрительное восприятие трехмерных сцен осуществляется как при монокулярном зрении (одним глазом), так и при бинокулярном зрении (двумя глазами). При монокулярном зрении главную роль играют линейная перспектива, блики, тени и полутона, маскировка, «оглядывание», зрительная память, ощущение различия в напряжении мышц глаза при изменении кривизны хрусталика для получения на сетчатке глаза резкого изображения предметов, находящихся на различных расстояниях от наблюдателя [Литвак и др., 1975].

Линейная перспектива вызывается тем, что если вдоль линии взгляда находятся параллельные линии, то они будут казаться сходящимися. Это приводит также к изменению относительных размеров предметов и градиента плотности. Известно, что уменьшение углового размера рассматриваемого предмета соответствует увеличению дистанции наблюдения. Если восприятие предмета не изолировано от восприятия окружающих предметов, то относительный видимый размер предмета служит эффективным средством оценки его удаленности. Этот эффект используется в живописи, где перспектива создается путем изменения размеров предметов.

Проекционное изображение предметов, находящихся на одинаковом расстоянии друг от друга, по мере удаления предметов становится более плотным. Например, кажущееся расстояние между шпалами на железной дороге по мере их удаления от наблюдателя уменьшается. Здесь градиент плотности описывает отношение между расстоянием предметов от глаза и их проекционным отношением.

Под *маскировкой* понимается заслонение дальних объектов ближними. Применение этого эффекта для создания иллюзии глубины было предложено Гельмгольцем.

Все объемные реальные тела при освещении отбрасывают тени. Именно по теням ученые оценивали высоту гор на фотографиях Луны.

Удаленные объекты обычно более светлы, расплывчаты, находятся как бы в дымке. Для использования этого эффекта можно применять модуляцию яркости, изображая передние грани предмета более яркими, чем задние [Ван Дам, 1986].

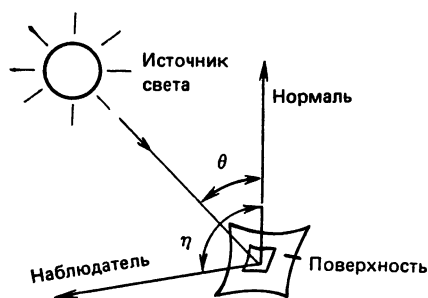


Рис. 5.3

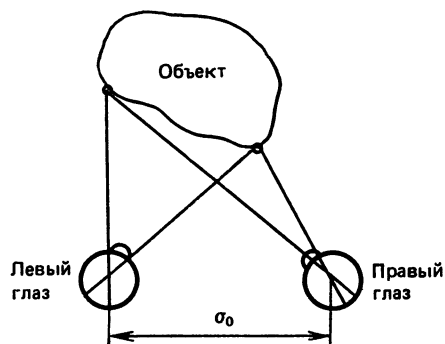


Рис. 5.4

Яркость конкретной поверхности зависит как от отражательной способности поверхности, так и от ее расположения относительно источника света. Модель отражения света от небольшого элемента поверхности (рис. 5.3) включает три основных объекта, расположенных в трехмерном пространстве: наблюдатель, поверхность и удаленный источник света. Источник света предполагается значительно удаленным от поверхности, так что лучи достигают поверхности практически параллельными. По закону Ламберта если поверхность является диффузной, то яркость света одинакова во всех направлениях, т. е. при любом угле наблюдения η . При этом интенсивность отраженного света пропорциональна косинусу угла между направлением на источник света и вектором, перпендикулярным поверхности, или нормалью к этой поверхности θ . Освещенность оказывается наиболее сильной, когда поверхность перпендикулярна направлению на световой источник. Эта модель может быть расширена на случаи любых других неламбертовских поверхностей. Для блестящих поверхностей, порождающих зеркальные блики, количество отраженного света определяется углами между нормалью к поверхности и двумя направлениями: на световой источник и на точку наблюдения. Сумма составляющих рассеянного, диффузного и зеркального отраженного света определяет интенсивность света поверхности.

Эффект «оглядывания», т. е. рассматривания трехмерной сцены с различных точек, может быть достигнут с помощью «кинетических» методов (см. с. 205).

При совместном действии всех перечисленных факторов у человека возникает эффект глубины.

В настоящее время появляются индикаторы стереоскопического видения (основанные на бинокулярном зрении). При бинокулярном зрении основную роль играет глазной базис σ_0 , т. е. расстояние между зрительными осями. Среднее или нормальное значение глазного базиса равно 65 мм. Из-за того, что глаза человека смотрят на объект с разных позиций, изображения одного и того же объекта для каждого глаза будут различаться (рис. 5.4). Различия левого и правого изображений учитываются в больших полушариях коры головного мозга человека, в результате чего создается ощущение трехмерности [Валюс, 1962].

Для получения изображений трехмерных сцен требуется: синтезировать два изображения (стереопару) одной и той же сцены, полученных с двух ракурсов, аналогичных позициям левого и правого глаза наблюдателя; обеспечить наблюдателю раздельное рассматривание левым и правым глазом соответствующих изображений стереопары, воспроизводимой на экране дисплея. Для выполнения этих условий стереосистема, как правило, должна содержать две

передающие телевизионные камеры, разнесенные в пространстве на некоторое расстояние, равное главному базису.

Для раздельного рассматривания изображений существует широкий набор технических средств [Леонов, 1970; Катус, 1975]. Наибольшее распространение для отображений стереопар в машинной графике получили цветные дисплеи. При этом одно изображение (для левого глаза) строится линиями синего цвета, а другое (для правого глаза) — красного цвета. Пространственное разделение изображений с помощью цветных очков, которые надевает наблюдатель, основано на цветоизбирательной проекции двух изображений стереопары в левый и правый глаз наблюдателя. Для этого осуществляется подбор цветных светофильтров с определенными спектральными характеристиками, устанавливаемых в очках наблюдателя.

При машинном синтезе стереоскопических изображений можно воспользоваться следующей особенностью восприятия объемных изображений: для вполне удовлетворительного восприятия в целом достаточно, чтобы только один глаз зрителя воспринимал изображение с полной четкостью. Для другого глаза можно синтезировать изображение с четкостью, уменьшенной в три-четыре раза, причем наблюдателю все равно, каким глазом он будет воспринимать изображение стереопары с полной четкостью, а каким — с уменьшенной.

Аппаратурные и программные средства вычислительной техники и, в частности, машинной графики позволяют решить проблему максимального приближения машинных изображений (по цветовым оттенкам, бликам, теням) к цветным фотографиям трехмерных сцен. Программистам удается воспроизвести практически любые световые эффекты. Новейшие высокопроизводительные растровые системы способны формировать изображение, состоящее из 1000 многоугольников, с частотой 12 кадров в секунду. Проблемой остается сокращение времени синтеза изображений. Громадные объемы вычислений, вызванные расчетом световых эффектов, требуют применения мощной вычислительной техники.

Построение трехмерных сцен средствами машинной графики происходит приблизительно так же, как у художника. После рождения замысла он набрасывает контуры предметов. В машинной графике это соответствует так называемой «проволочной» модели. По ней выверяется форма предметов. Затем художник стирает те линии, которые не видны из точки, где находится глаз наблюдателя. В машинной графике для этого используются алгоритмы удаления невидимых линий [Фолл и др., 1985]. Затем художник «покрывает» проволочную конструкцию «материалом», т. е. закрашивает поверхности, добиваясь эффекта реалистического светового решения. В машинной графике это делает алгоритм закраски.

Алгоритмы построения теней от источников света напоминают алгоритмы удаления скрытых поверхностей, поскольку в них определяется, какие поверхности могут быть видны из точек, где расположен источник света. Поверхности, одновременно видимые из точки наблюдения и из источников света, не находятся в тени, а те из них, которые видны из точки наблюдения, но невидимы из светового источника, лежат в тени. При распределенных источниках света нужны сложные вычисления как для полных теней, так и для полутеней.

Анализ трехмерных сцен

При анализе трехмерных сцен в ИС возникает ряд проблем технического характера, вызванных размытостью границ изображения, флюктуационными помехами, нарушением геометрического подобия между воспринимаемыми и воспроизводимыми изображениями. Более сложными являются проблемы анализа трехмерных сцен на уровне человеческого восприятия. Например, для современных ИС непосильна задача захвата и переноса деталей, сваленных в кучу, а не разложенных на ровной поверхности, или задача анализа описания сцены типа: «В комнате, напротив окна стоит стол, слева от стола торшер, справа висит картина, на которой...» и т. д.

При восприятии изображений человеком реализуется неразрывная связь чувственного познания и мышления. «Понимать» — значит видеть вещи определенным образом, но нельзя «видеть», не понимая. В любом чувственном восприятии присутствует мышление, которое определяет направленность восприятия [Грегори, 1972]. Восприятие трехмерных сцен сопровождается построением «мысленной картины» (концептуальной модели), воплощающей в себе знания о наблюдаемых объектах, которыми располагает исследователь. При восприятии трехмерных сцен необходимо выполнять в обратном порядке последовательность операций синтеза трехмерных сцен.

Вначале следует из полутонного изображения выделить контуры предметов или края. Затем, описав контур, определить, что это за предмет. Отметим, что воспринять предмет — это значит выделить его из окружающего пространства. Предмет имеет форму, фон относительно бесформен; предмет имеет характер вещи, фон выглядит как неоформленный материал. В выделении предмета из фона проявляется избирательность восприятия. В психологии эмпирически выявлены некоторые принципы организации, которыми можно влиять на отношения «предмет — фон». Так, в качестве предмета прежде всего выделяются замкнутые линии; симметричные конфигурации легче, чем несимметричные, воспринимаются как предметы; в том случае, когда поле заполнено однородными элементами, например точками, предметы образуют те из них, которые пространственно расположены близко друг к другу; если поле заполнено разнородными элементами, то предмет образуют те из них, которые имеют сходство по форме или цвету [Литвак и др., 1975].

Согласно основным принципам зрительного восприятия главную роль в опознании изображений играет контур. Он несет основную информацию об объекте. Фигуру нельзя воспринимать отдельно от контура. Если контур неясен и широк (когда одна часть поля постепенными оттенками переходит в другую), то форма остается неопределенной. Имеются основания предполагать, что первой операцией обработки зрительного объекта является выделение контура, т. е. пространственное дифференцирование точек световой интенсивности методами сканирования по градиенту. Полученная таким образом информация об объекте (контур) запоминается и служит входной для следующего этапа обработки — выделения локальных признаков. Контуры фигуры представляют собой области с высоким содержанием информации и характеризуются инвариантностью при трансформациях цвета и яркости [Грановская, 1974].

В ИС выделение края предмета оказывается сложной проблемой. Поиск края начинается с поиска «элементов края». На следующем этапе система должна соединить найденные элементы края для получения более длинных линий и сопоставить ее с имеющимися в системе трехмерными моделями элементарных «строительных блоков». Края сами по себе порождают небольшую разницу в интенсивностях и с ней очень трудно работать, поскольку информация от телевизионной камеры и других сенсоров искажена в гораздо большей степени. Кроме того, объекты вносят значительную долю искажений в виде грязи, текстуры, теней и многократных отражений. Если устройство поиска линий слишком чувствительно, то порождается масса ошибочно воспринимаемых краев; если оно недостаточно чувствительно, на рисунке отсутствуют некоторые настоящие линии. Ряд линий, порожденных шумами, выражен четче реальных, и их нельзя исключить, не привлекая истинные знания об окружающем мире.

Таким образом, трудности восприятия трехмерных сцен вызваны тем, что представление о трехмерном мире ИС должна получить из двухмерных проекций; яркость света в каждом элементе изображения определяется совместным действием многих факторов, которые трудно отделить друг от друга (например, яркость элемента зависит от освещения сцены, отражательной способности поверхностей сцены, включая цвет, текстуру и зеркальность, тумана или пыли в атмосфере, геометрических и хроматических искажений в линзах телекамеры, способа дискретизации изображения и т. п.); ИС должна обработать громадные объемы данных.

Анализ движения

Если два дискретных события сменяют друг друга быстрее чем через 1/16 с, то они воспринимаются как одновременные и сливаются в единый образ. Если события разворачиваются слишком медленно (с промежутками свыше 8—10 с), то такая смена событий не будет восприниматься как движение, представление о «явлении» исчезнет и вместо него будет восприниматься ряд отдельных состояний объекта. Например, в то время как секундная стрелка часов перемещается, «скачет» у нас на глазах, две другие стрелки кажутся неподвижными. Мы воспринимаем их показания как последовательность не связанных друг с другом состояний. Когда эти стрелки продвинулись с «восемью часов» на «десять минут девятого», мы не воспринимаем их движения непосредственно, а лишь фиксируем, что они переместились. Итак, любое естественное движение, которое должно быть воспроизведено, должно уложиться между этими двумя фиксируемыми временными границами.

Вычислительная техника позволяет пользователю создавать (программировать) движение (*динамическая машинная графика*). Переход от генерации статических изображений к генерации динамических изображений повышает реальность изображения. Благодаря этому можно наблюдать на дисплее поведение проектируемого автомобиля при аварии, изучать кинематическую схему «работающего» станка, движения еще не созданного робота-манипулятора. Задачи генерирования движения приходится решать при проектировании динамических объектов, машинного эксперимента, создании различных тренажеров, мультфильмов и т. д. Например, в тренажерах для обучения пилотов особое значение придается генерации чрезвычайно реалистичной картины динамики изменяющегося мира с частотой как минимум 30 кадров в секунду. Только при этом у обучаемого может возникнуть иллюзия полета.

Возможности динамической машинной графики позволяют повысить наглядность восприятия трехмерных сцен в монокулярных системах. Рассмотрим два примера. На экране дисплея генерируется сцена: белый цилиндр со случайно расположенными черными пятнами освещается рассеянным светом. Наблюдение осуществляется перпендикулярно оси цилиндра. Поскольку свет рассеян, то цилиндр не дает теней. Нерегулярность пятен на поверхности цилиндра не позволяет выявить вид поверхности. Однако если цилиндр вращать вокруг его оси, то по характеру движения пятен легко выявить форму отображаемой поверхности.

Аналогично можно решить проблему отображения трехмерной структуры молекулы. По двухмерному изображению восстановить ее структуру невозможно, но если вращать молекулу (точнее, не молекулу, а ее трехмерную модель, хранимую в памяти компьютера), то станет видно, какие части молекулы находятся впереди, а какие сзади. Такие методы отображения трехмерных объектов называются *кинетическими*. Они в какой-то мере используют известный в психологии зрения факт оглядывания предмета, когда наблюдатель для того чтобы лучше рассмотреть предмет, смотрит на него с разных точек зрения, меняя положение головы.

При создании систем анализа движения используют классификацию движения по направлению — перпендикулярное линии взгляда наблюдателя и вдоль линии взгляда, а также по величине перемещения (от кадра к кадру) — незначительное по сравнению с размерами объекта движения и сравнимое (или больше) с размерами объекта движения.

Наиболее простым методом машинного анализа движения является сопоставление двух последовательных во времени кадров. Каждый кадр представляет собой двухмерную матрицу яркостей. Предположим, что размер движущегося объекта значительно превышает длину пути, проходимого им за отрезок времени между двумя кадрами. При вычитании значений матрицы предыдущего кадра из последующего получаем так называемую *матрицу движения*. В этой матрице движущийся объект, который ярче, чем фон, дает положительную разность в направлении движения объекта и отрицательную в противоположном направлении. Сумма абсолютных значений разностей яркостей

двух кадров дает информацию о наличии движения, т. е. может служить индикатором обнаружения движения. По матрице движения можно вычислить параметры движения. Направление вектора скорости определяется отрезком, соединяющим центры областей «убывания» и «возрастания»; значение вектора скорости — скоростью изменения яркости.

Глава 6

Обучение

А. А. Мартиросян, Э. М. Погосян

6.1. Модели обучения

Неформальные модели

В психологии под обучением понимают способность к приобретению ранее неизвестных умений и навыков. В ИС неформальное понимание обучения трактуется аналогично. Говорят, что ИС обучилась чему-либо, если она стала способной к выполнению некоторых процедур или решению некоторых задач, которые до этого была выполнять неспособна. Конечно, такое определение широко и расплывчато. Под него, например, подходит случай, когда в память ИС закладывается готовая программа, которой ранее в ней не было. Поэтому часто специально подчеркивается, что обучение в ИС происходит таким образом, что она самостоятельно извлекает новую информацию из исходной или текущей информации. Другими словами, предполагается, что в процессе деятельности ИС анализирует имеющуюся информацию и на основе анализа извлекает из нее полезные закономерности.

При имитации в ИС процедур обучения активно используются модели обучения, известные в физиологии и психологии. Первые программы, демонстрирующие возможность обучения, — программы моделирования условных рефлексов и гомеостатических процессов [Гаазе — Рапопорт и др., 1987], опирались на чисто физиологические модели обучения, разрабатывавшиеся в школе Павлова. Позже такие жесткие модели уступили место моделям, опирающимся на *ассоциативную модель обучения*, согласно которой всякое обучение есть установление ассоциативных связей в нейроноподобных сетях.

На смену ассоциативной модели обучения пришла *лабиринтная модель*, опирающаяся на идеи когнитивной психологии. Модель предполагает, что процесс обучения состоит в эвристическом поиске в лабиринте возможных альтернатив и оценивании движения по лабиринту на основе локальных критериев. Наиболее исследованными на сегодняшний день являются модели, относящиеся к обучению по примерам.

Формальные модели

Обучение как математическая задача может быть отнесено к классу *оптимизационных проблем поиска описаний*.

Индивидуальная оптимизационная задача L есть пятерка

$$\langle X_L, Y_L, \rho_L, F_L, J_L \rangle,$$

где X_L и Y_L — множества входных и выходных записей; $\rho_L \subseteq X_L \times Y_L$ — отношение (или функция $\rho: X_L \rightarrow Y_L$); F_L — множество отношений ($f_L \subseteq X_L \times Y_L$

для всех $f_L \in F_L$), называемых описаниями; J_L — оператор качества для F_L , показывающий для каждого $f_L \in F_L$ степень его близости к ρ_L . Задача состоит в отыскании оптимального по J_L описания f^*_L из F_L .

Спецификация задачи часто оказывается неполной. Например, оператор качества J может быть плохо формализуемым, информация об отношении ρ может задаваться только примерами пар $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, для которых $x_i \rho y_i$, и т. д. Если спецификация полная, то обучение не нужно, так как получается традиционная оптимизационная задача. Для задач, относимых к обучению, характерна неполнота спецификации.

Множество индивидуальных оптимизационных задач $\{L\}$ с одними и теми же компонентами X, Y, F и J называется *оптимизационной проблемой*. Задача синтеза (поиска) описаний для проблемы состоит в построении алгоритма ω , который по спецификации произвольной $L \in \{L\}$ строит решение $\omega(L) = f^*$. При этом описание должно быть синтезировано с возможно меньшими вычислительными затратами, т. е. алгоритм ω должен быть в этом смысле оптимальным.

Для комбинаторных проблем задачи $L \in \{L\}$ часто ранжируются по числовому параметру $\text{size}(L)$, называемому *размером задачи* и являющемуся мерой сложности входной спецификации. Так, размером задач для problems нахождения кратчайшего пути в графах при условии полной спецификации может служить число вершин графа. При наличии размера у индивидуальных задач понятие оптимальности ω можно понимать стандартным образом: вводится *функция затрат* $C(\omega, L)$ и *функция сложности* $C(\omega, n) = \max\{C(\omega, L) \mid \text{size}(L) \leq n\}$. Алгоритм ω будет оптимальным, если не существует алгоритма ω' , такого, что $C(\omega', n) < C(\omega, n)$ для всех n (и для одного из n это неравенство строгое) [Гэри и др., 1982].

Для задач обучения естественное понятие размера часто отсутствует. Трудно представить, например, что могло бы служить в качестве функции size при синтезе описаний функций, спецификации которых задаются в виде (бесконечной) последовательности пар <аргумент→значение>. Отсутствие размера является одной из причин того, что сложность C в задачах обучения обычно плохо формализуема.

Известные методы решения задачи синтеза можно классифицировать по способу спецификации проблем, типу разрешенных алгоритмов ω , классу исследуемых проблем, критерию оценки синтеза и т. д. В настоящее время решение задач обучения характеризуется тремя подходами [Погосян, 1983].

1. В *теории статистических гипотез* [Кендэл и др., 1960] рассматривается множество M реализаций некоторого случайного объекта с распределением вероятностей $p(x)$ на M . Пусть W — произвольное подмножество M и $\{H\}$ — некоторое множество гипотез фиксированного типа, связанных с вероятностью $p(x \in W)$ и характеризующих ее. Требуется на основе выборки (обучающей последовательности) из M , полученной в соответствии с $p(x)$, выбрать наиболее подходящую гипотезу из $\{H\}$.

В качестве гипотез могут быть, например, следующие утверждения: « $p(x)$ есть пуассоновское распределение» или «распределение $p(x)$ — нормальное и имеет заданные средние и дисперсию» и т. д. Статистические методы применимы к случайным величинам и обнаруживают их специфические, статистические характеристики, которые часто являются симптоматическими по отношению к основным глубинным закономерностям исследуемых явлений [Дружинин, 1973]. В [Гаек и др., 1984] объединяются методы статистической теории на стадии выдвижения гипотез с логическими методами на стадии обоснования и построения следствий из этих гипотез. Другие методы теории статистических гипотез, применяемые в проблеме обучения, можно найти в [Вапник и др., 1974; Фукунага, 1979].

2. В *теории параметрической адаптации* [Шыпкин, 1968, 1970; Вапник и др., 1974] предполагается, что множество F описаний, среди которых ищется f^*_L , может быть охарактеризовано вектором параметров и выбор f^*_L сводится к по-

иску экстремума оператора качества, задаваемого функционалом вида

$$J(c) = \int_x Q(x, c) p(x) dx = Mx \{Q(x, c)\}.$$

Здесь $x = (x_1, \dots, x_i)$ — вектор дискретного или непрерывного случайного процесса с плотностью распределения $p(x)$; $c = (c_1, \dots, c_m)$ — вектор, компоненты которого характеризуют выбранное решение (описание); $Q(x, c)$ — функционал вектора c , зависящий от x ; Mx — математическое ожидание. Экстремум $J(c)$ находится из уравнения $\text{grad } J(c) = 0$. Поскольку в общем виде это уравнение не имеет аналитического решения, то можно переходить к разностному уравнению

$$c[t] = c[t-1] - \Gamma[t] \text{grad}_c J(c[t-1]),$$

где Γ — матрица $m \times m$, элементы которой, вообще говоря, зависят от текущего значения $c[t-1]$. Надлежащий выбор матрицы Γ должен обеспечить сходимость $c[t]$ к оптимальному значению c^* .

Если $p(x)$ неизвестна и ее нельзя предварительно восстановить, а также при отсутствии явно заданного функционала $J(c)$ переходят к другому разностному уравнению, которое по наблюдаемым значениям x, c и $\text{grad}_c Q(x, c)$ позволяет определить изменение вектора $c[t]$:

$$c[t] = c[t-1] - \Gamma[t] \text{grad}_c Q(x[t], c[t-1]).$$

В этом случае соответствующие итеративные алгоритмы называются *адаптивными* или *обучающимися*.

Алгоритмы параметрического обучения и их приложения в системах классификации, обучающихся моделях, антенных и кодирующих устройствах, фильтрах и т. д. описаны в [Айзерман и др., 1970; Цыпкин, 1970; Вапник и др., 1974].

3. *Теория индуктивного вывода* [Gold, 1967; Angluin et al., 1983] представляет собой дискретную математическую модель обучения по примерам. Множества X и Y счетные, искомое описание ρ в общем случае специфицируется посредством (потенциально бесконечной) последовательности троек вида $(x_1, y_1, a_1), (x_2, y_2, a_2), \dots$, таких, что $a_i \in \{0, 1\}$ и $x_i \rho y_i$; тогда и только тогда, когда $a_i = 1$ (т. е. тройки (x_i, y_i, a_i) представляют примеры и контрпримеры ρ). В качестве F выбирается множество процедур, например формальные грамматики, общерекурсивные функции и т. д.

6.2. Обучение по примерам

Типы задач

Система (человек или машина) может получать новые знания многими способами. Можно, например, вывести нужную информацию как логическое следствие имеющихся знаний, получить ее модификацией имеющихся знаний, рассчитывая на «аналогичность» ситуаций, попытаться вывести общий закон из имеющихся примеров. Приведем некоторые задачи, традиционно относимые к задачам обучения по примерам (ОП).

1. *Прогнозирование*. Дана последовательность чисел: 3, 5, 7 ... Чему равен следующий член этой последовательности?

2. *Идентификация (синтез) функций*. Имеется некоторый «черный ящик», относительно устройства которого можно судить по его поведению, подавая на вход сигналы и получая в ответ выходные. Требуется по этой информации сформулировать описание работы анализируемого устройства. Главное отличие этой задачи от предыдущей состоит в синтезировании общего закона, а не в прогнозировании его частного проявления. К задачам идентификации относятся, например, расшифровка структуры конечных автоматов [Трахтенброт и др., 1970], индуктивный синтез программ на языке ЛИСП [Biermann, 1978].

3. *Расшифровка языков.* Поиск правил синтеза текстов некоторого языка на основе анализа конкретных текстов на этом языке (расшифровки кодов, систем письменности и т. д.). Задачей такого же типа является обучение распознаванию образов.

Близкие по духу задачи рассматриваются в теории грамматического вывода [Biermann et al., 1972a].

4. *Индуктивный вывод.* В широком смысле в это направление вписываются все рассмотренные выше задачи. В узком смысле индуктивный вывод почти совпадает с проблемой расшифровки языков.

5. *Синтез с дополнительной информацией.* В качестве дополнительной информации может использоваться структура примеров, их родовидовая принадлежность и т. д. К дополнительной информации относят также контрпримеры. Контрпримеры часто помогают в решении задачи (например, в алгоритме Уинстона, решающем задачу об арках [Уинстон, 1978]). Дополнительная информация используется в других задачах синтеза. Например, возможность синтеза программ по парам вход — выход существенно расширяется, если с каждой парой вход — выход задается «траектория» ее вычисления (последовательность состояний программы без учета их тождественности) [Summers, 1977].

Все рассмотренные задачи в зависимости от предмета исследования (множество или отображение) относятся к одной из двух категорий: синтезу языков или синтезу функций. Отметим два свойства задач ОП (общие для функционального и языкового синтеза): 1) все они являются задачами нахождения описаний; 2) задаваемая в виде примеров входная информация (обучающая выборка) является недостаточной для однозначного формирования требуемого описания. В этом смысле задачи обучения некорректны. Например, при идентификации вычислимой функции по входным-выходным данным $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ существует бесконечное число вычислимых функций, графики которых содержат точки (x_i, y_i) , и нет логических оснований предпочесть одну из них другой.

Итеративные алгоритмы обучения

В большинстве задач ОП множество примеров потенциально бесконечно. Это означает, что хотя в каждый конкретный момент времени имеется конечное множество примеров, но число их может увеличиваться неограниченно. Например, при расшифровке конечного автомата можно подавать на его вход новые входные сигналы, получая в ответ новые выходные. Новые данные могут помочь отсеять некоторые описания и может наступить момент, когда останется единственный вариант, который не будет опровергаться последующими примерами. Такие ситуации возможны, хотя обычно не существует алгоритма, способного определить момент стабилизации решения.

Пусть задана (потенциально бесконечная) последовательность чисел y_0, y_1, y_2, \dots . Задача состоит в нахождении полинома P , такого, что y_0, y_1, \dots совпадают со значениями $P(x)$ при $x=0, 1, \dots$. Решение задачи будем искать с помощью итеративной процедуры, работа которой на k -м шаге состоит в обработке очередной порции примеров — последовательности y_0, y_1, \dots, y_k и выдаче на этой основе некоторого полинома P_k (гипотеза об искомом полиноме P). В качестве P_k выберем первый в фиксированном заранее перечислении всех возможных полиномов полином степени, меньшей или равной k , такой, что значения его при $x=0, 1, \dots, k$ совпадают с y_0, y_1, \dots, y_k (возможность перечисления полиномов и существование P_k очевидны). Если пример y_{k+1} , рассматриваемый на следующем шаге итерации, будет таким, что $P(k+1) \neq y_{k+1}$ (т. е. гипотеза не опровергается), то P_{k+1} будет совпадать с P_k . Если P_k и далее не опровергается, то оно будет искомым полиномом. С другой стороны, поскольку последовательность $y_0, y_1, \dots, y_k, \dots$ «полиномиальная», то после некоторого числа опровержений подходящий полином будет найден. Таким образом, после «достаточно большого» числа примеров последовательность гипотез стабилизируется на полиноме, удовлетворяющем условию задачи. Однако определить достаточное

число примеров алгоритмически невозможно, хотя очевидно, что если вместе с примерами в качестве дополнительной информации задана степень искомого полинома P , то последняя проблема разрешима.

Существует более изящный и менее сложный интерполяционный алгоритм для решения этой задачи, однако прием с перечислением «пространства гипотез» обладает большей универсальностью. Например, задача синтеза конечных автоматов по результатам экспериментов решается аналогично [Трахтенброт и др., 1970], причем число примеров, необходимых для стабилизации гипотез, можно подсчитать, если заранее задано количество состояний искомого автомата.

Таким образом, согласно наиболее распространенной в настоящее время парадигме [Gold, 1967] корректными являются методы генерации гипотез, которые в пределе (при исчерпании всех примеров) приводят к решению задачи.

Итеративные процедуры обладают существенно большими вычислительными возможностями, чем обычные алгоритмы. В теории алгоритмов эти процедуры получили название *алгоритмов предельных вычислений* (АПВ). Функция $f(x)$, вычислимая некоторым АПВ, представима в виде $\lim F(x, y)$ при $y \rightarrow \infty$, где F — некоторая общерекурсивная функция [Фрейвалд, 1974].

Предположение о предельной стабилизации гипотез является основой гипотико-дедуктивного подхода [Kugel, 1977], согласно которому решение задачи ОП включает четыре этапа: 1) наблюдение: сбор и накопление исходных данных (примеров); 2) обобщение: выдвижение «разумной» гипотезы H об искомом описании; 3) дедукция: выдвижение различных следствий из H или прогнозов на основе ее; 4) подтверждение: проверка прогнозов на совместимость с результатами новых наблюдений — оценка гипотезы H ; если H подтверждается, то остается в качестве текущей гипотезы и весь процесс повторяется сначала, в противном случае гипотеза H заменяется новой. Считается, что описанный процесс находит искомое описание, если оно было выдвинуто в качестве гипотезы при каком-либо прохождении второго этапа и при следующих прохождении этого этапа не заменялась новой.

В процессе выдвижения гипотез выясняются «разумные» способы выдвижения и критерии подтверждения гипотез.

Спецификация задач обучения по примерам

Для спецификации задачи ОП необходимо уточнить следующие ее характеристики [Angluin et al., 1983].

1. Класс искомым описаний. Поскольку речь идет о математической теории, то обычно это множества (языки) и функции.

2. «Пространство гипотез», т. е. множество формальных выражений, соответствующих возможным описаниям. Каждое из искомым описаний должно иметь в пространстве гипотез хотя бы одного своего представителя. Обратное неверно — пространство гипотез может представлять более широкий класс описаний, чем искомые.

3. Множество примеров для каждого описания, а также разрешенные последовательности этих примеров, называемые допустимыми представлениями этого описания.

4. Критерий успешности вывода, т. е. определение того, в каком случае гипотеза, на которой стабилизировался процесс решения задачи, считается приемлемой.

Например, задача синтеза (идентификации) вычислимых (т. е. частично рекурсивных) функций может быть специфицирована следующим образом. Класс искомым описаний есть множество всех вычислимых функций; пространство гипотез — множество всех синтаксически правильно построенных программ (скажем, Лисп-программ); множество примеров функции f — все возможные пары натуральных чисел $\langle x, f(x) \rangle$, такие, что f определена в точке x . Допустимым представлением является любая бесконечная исчерпывающая последовательность примеров (повторения допускаются). Наиболее часто упоминаемый критерий успешности состоит в том, что программа P , на которой стабилизировался про-

цесс нахождения описания множества $\{ \langle x, f(x) \rangle \mid x \in \text{dom } f \}$, должна заканчивать работу во всех точках $x \in \text{dom } f$ и для всех этих точек выдавать значения: $f(x)$ (P может быть определена и в таких точках, где f не определена). Говорят, что такая программа P идентифицирует функцию f . Другие встречающиеся критерии — точная идентификация (т. е. с учетом точек неопределенности f), идентификация с точностью до конечного числа точек и т. д.

Наиболее распространенными спецификациями задач *расшифровки языков* являются следующие. Класс описаний — множество формальных языков в фк-сированном алфавите Σ ; пространство гипотез — конкретный способ представления этих языков. Например, для задачи синтеза регулярных языков это могут быть регулярные выражения, конечные автоматы, левосторонние и правосторонние грамматики, а также формализмы, описывающие более широкие классы языков. Допустимым представлением языка L является любая бесконечная исчерпывающая последовательность слов из L . Такое представление языков называется *позитивным*. Если вместе с примерами слов из L в последовательности встречаются маркированные контрпримеры, то такое представление называется *позитивно-негативным*. Если в представлении присутствуют все возможные контрпримеры (т. е. последовательность включает все слова в алфавите Σ , маркированные знаками $+$ или $-$), то представление называется *полным*. Критерием успешности является точное соответствие найденного описания множеству позитивных примеров. В отличие от синтеза функций «сверхобобщение» обычно не допускается, поскольку это приводит к тривиальным решениям. Другими критериями могут быть совпадение с точностью до заданного числа слов, совпадение с точностью до любого конечного числа слов и т. п.

Задача *расшифровки языков* рассматривается и в вероятностной постановке. *Стохастический язык* — это множество слов в некотором языке с распределением вероятностей на своих элементах. Пространство гипотез составляют стохастические грамматики (т. е. грамматики с приписанными правилами вероятностями). Допустимым представлением является последовательность слов из языка. Подразумевается, что эта последовательность случайная, т. е. входящие в нее слова встречаются с соответствующей им в языке вероятностью. Наиболее распространенный критерий успешности — идентификация стохастической грамматики, порождающей заданный язык с вероятностью 1. Множество описаний, пространство гипотез и множество допустимых представлений задачи обычно стандартным и естественным образом вытекают из ее неформальной постановки; вариативность критерия успешности намного больше. Укажем несколько критериев, ослабляющих требование стабилизации гипотез. Выполнение условия на окончательную гипотезу требуется лишь «для всех гипотез, начиная с некоторого места». Например при синтезе функций с критерием успешности «по поведению» требуется, чтобы, начиная с некоторого места i , все гипотезы $P_i, P_{i+1}, P_{i+2}, \dots$ идентифицировали f . Таким образом, условие $P_i = P_{i+1} = \dots$ снимается: идентифицировать функцию становится легче. При частотной идентификации требуется, чтобы частота (точнее, ее нижний предел) «правильных» гипотез была не меньше заранее заданного числа ε . Такие же критерии рассматривались в задачах языкового синтеза.

Несколько другой характер имеет отказ от условия стабилизации гипотез в задаче *прогнозирования вычислимых функций*. Здесь требуется найти не общее описание, а различные следствия из него. Соответственно пространство гипотез составляют не программы для вычисления функций, а их результаты, т. е. множество всех натуральных чисел. Прогнозирование функции f считается успешным, если, начиная с некоторого места i , все выданные числа-гипотезы $h_i, h_{i+1}, h_{i+2}, \dots$ совпадают с числами $f(i), f(i+1), \dots$, заданными на вход итеративного алгоритма H : $H(f(0), f(1), \dots, f(i+j)) = h_{i+j} = f(i+j)$ для всех $j \geq 0$. Особенность задачи прогнозирования заключается в том, что ее формальные уточнения существенно зависят от принятой вычислительной модели итеративных алгоритмов.

Для того чтобы охарактеризовать эту зависимость и сравнить рассмотренные варианты задач языкового и функционального синтеза, достаточно пред-

ставить АПВ в виде трехленточной машины Тьюринга (M) со следующими особенностями. Первая (входная) лента предназначена для записи закодированных в алфавите машины бесконечных последовательностей (допустимых представлений множества примеров). Головка машины, работающая с этой лентой, является только считывающей и может перемещаться только вправо — по команде управляющего устройства. Вторая (рабочая) лента предназначена для вычислений на каждом шаге итерации. При этом входными параметрами являются очередной элемент допустимого представления, считанный со входной ленты, и записи, оставшиеся на рабочей ленте после предыдущего шага итерации — «накопленный опыт» машины. Головка рабочей ленты может и читать, и писать, может двигаться и вправо, и влево. Время от времени управляющее устройство дает команду головке третьей (выходной) ленты записать на ней некоторое число, которое интерпретируется как номер выдаваемой гипотезы в фиксированном заранее пересчете пространства гипотез. Последняя головка — только пишущая и движется только вправо. Печать номера на выходной ленте считается завершенной, если в конце его напечатан специальный знак — (разделитель). Программа управляющего устройства должна быть такой, чтобы печать каждого начатого номера завершалась.

Интерпретация работы такой машины для задач ОП очевидна. Гипотезу, выданную машиной M после считывания первых n элементов входной последовательности $a_1, a_2, \dots, a_n, \dots$, будем обозначать $M[a_1, a_2, \dots, a_n]$.

Исследуем возможности таких машин. Пусть S есть некоторая спецификация задач ОП, т. е. $S = \langle D, H, P, C \rangle$, где D — множество описаний; H — пространство гипотез; P — множество допустимых представлений; C — критерий успеха. Обозначим через $\text{Inf}_S(M)$ множество описаний из D , которое (после соответствующих кодировок допустимых представлений и пересчета пространства гипотез) машина M в состоянии синтезировать при заданных H, P и C . Другими словами, $\text{Inf}_S(M)$ — это множество индивидуальных задач из S , решаемых машиной M . Таким образом, при разных S $\text{Inf}_S(M)$ может быть множеством языков, синтезируемых M по их полному представлению, множеством функций, идентифицируемых M по входным-выходным данным и т. д. Класс $\text{Inf}_S(M)$ является мерой универсальности M при решении S . Множество описаний D' называется *идентифицируемым* (синтезируемым) при спецификации S , если существует машина M , такая, что $D' \subseteq \text{Inf}_S(M)$. Отметим, что идентифицируемость того или иного множества не зависит ни от алфавитов машины M , ни от способа кодирования информации, ни от способа пересчета (нумерации) пространства гипотез и самого его выбора. Независимость от выбора и способа нумерации пространства гипотез нуждается в оговорках, однако достаточно, чтобы H было множеством, обладающим так называемой геделевой нумерацией. Это следует понимать в том смысле, что класс $\text{Inf}_S = \{D' | D' \subseteq \text{Inf}_S(M) \text{ для некоторого } M\}$ инвариантен к изменению этих параметров.

Для большинства задач ОП не существует универсальных способов их решения (т. е. не существует АПВ, решающего все индивидуальные задачи: $D_S \notin \text{Inf}_S$).

Первый результат такого рода был получен в [Gold, 1967]: любое множество языков, содержащее все конечные и хотя бы один бесконечный язык, не может быть синтезировано по позитивному представлению. Отсюда следует не-синтезируемость по позитивным примерам множества регулярных языков. Подобные утверждения были затем доказаны для задач синтеза и прогнозирования вычислимых функций с различными критериями успешности, синтеза рекурсивных языков по полному представлению и др. [Бардзинь и др., 1972; Blum et al, 1975; Angluin, 1980a; Мартиросян 1986].

Сравним различные варианты спецификации задач ОП. Будем различать два типа машин. Машины первого типа, называемые всюду определенными, обладают тем свойством, что любой элемент входной последовательности когда-нибудь считывается и между любыми последовательными считываниями машина выдает на выходную ленту одну гипотезу. Другими словами, функция $M[a_1, \dots, a_n]$

определена для всех a_1, \dots, a_n и любого n . К машинам второго типа отнесем те, у которых $M[a_1, \dots, a_n]$ не определена хотя бы для одного набора (a_1, \dots, a_n) . Оказалось, что если S — спецификация задачи синтеза (языков или функций), то при проверке S -синтезируемости множества описаний можно ограничиться только всюду определенными машинами. Другими словами, по любой машине M можно построить всюду определенную машину M' , такую, что $\text{Inf}_S(M) \subseteq \text{Inf}_S(M')$.

Исключением из правила являются задачи прогнозирования. Рассмотрим три ее варианта, отличающихся друг от друга ограничениями на использование не всюду определенных машин. В первом варианте разрешается использовать только всюду определенные машины: $U \in \text{Inf}_{S_1}$, тогда и только тогда, когда существует всюду определенная машина M , такая, что для любого $f \in U$ имеем $M[f(0), \dots, f(n+1)] = f(n+1)$ для всех n , кроме конечного числа. Во втором варианте M не обязана быть всюду определенной, но если $f \in \text{Inf}_{S_2}(M)$, то, начиная с некоторого $n=n_0$, должно быть $M[f(0), \dots, f(n)] = f(n+1)$, причем для всех $k < n$ требуется, чтобы функция $M[f(0), \dots, f(k)]$ была определена. В третьем варианте разрешается, чтобы для некоторых k (но не более конечного числа) $M[f(0), \dots, f(k)]$ была не определена.

Для классов Inf_{S_1} , Inf_{S_2} и Inf_{S_3} введем специальные обозначения NV , NV' , NV'' (NV означает next value); имеют место соотношения $NV \subset NV' \subset NV''$, где \subset — строгое включение.

Рассмотрим несколько разновидностей задачи синтеза функций. Пусть S есть спецификация, при которой множеством описаний являются все вычислимые всюду определенные (общерекурсивные) функции, пространством гипотез — множество всех программ на каком-либо языке, единственным допустимым представлением функции f — последовательность $f(0), f(1), \dots$. Критерий успеха определим как точное угадывание: $M[f(0), \dots, f(n)] = \text{const}$ для всех n , начиная с некоторого n_0 , причем программа P , имеющая номер $M[f(0), \dots, f(n)]$, вычисляет именно функцию f . Задачу S обычно называют задачей синтеза общерекурсивных функций. Множество Inf_S обозначают через GN (от Gödel number).

Если ослабить критерий успешности, сняв условие стабилизации, то получим другую задачу — поведенческого синтеза общерекурсивных функций. Здесь требуется, чтобы для всех i программы, имеющие номера $M[f(0), \dots, f(n_0), \dots, f(n_0+i)]$, вычисляли одну и ту же функцию f (но не требуется, чтобы все программы были одинаковые). Множество Inf_S в этом случае обозначается обычно GN^∞ .

Соотношения между рассмотренными задачами такие [Подниекс, 1974]: $NV \subset NV' \subset GN \subset GN^\infty \subset NV''$. Результаты сравнения других вариантов спецификации можно найти в [Подниекс, 1975 ; $\text{Klette et al., 1980}$; $\text{Jantke et al., 1981}$; $\text{Osherson et al., 1982}$; $\text{Angluin et al., 1983}$; Case et al., 1983 ; Мартиросян, 1986] и др. Наиболее типичны случаи, когда имеет место строгое включение или когда эти задачи оказываются несравнимыми (т. е. соответствующие классы Inf_S несравнимы по теоретико-множественному включению). Например, задачи частичного прогнозирования функций всюду определенными и не всюду определенными машинами (соответствующие спецификации достаточно очевидны) несравнимы при разных заданных частотах: при $\delta < \varepsilon$ $NV(\delta)$ и $NV'(\varepsilon)$ несравнимы по включению [Подниекс, 1974]. Отсюда, в частности, следует, что нельзя гарантировать повышение частоты верных прогнозов за счет отказа от всюду определенных машин. Тем не менее теория индуктивного вывода не разбивается на отдельные независимые подтеории: не только методы, но и многие результаты для разных задач совпадают.

Возможности АПВ

При исследовании зависимости АПВ от типа перечисления примеров [Gold, 1967] оказалось, что позитивное представление языков недостаточно информа-

тивно для синтеза большинства типов языков. Однако в [Angluin, 1980a, b; Shinohara, 1982; Nix, 1983] приведены интересные классы языков, синтезируемых по позитивным примерам. В задаче синтеза стохастических грамматик ситуация иная: произвольный стохастический контекстно-свободный язык может быть с вероятностью 1 идентифицирован (т. е. для него будет найдена стохастическая контекстно-свободная грамматика) по случайной последовательности слов из языка, не содержащей явно негативных примеров [Horning, 1969]. Различные варианты языкового синтеза по позитивному представлению рассматривались также в [Angluin, 1980a, b, 1982b; Osherson et al., 1982; Angluin et al., 1983].

Если заранее известно, что перечисления порождаются только примитивно-рекурсивными функциями, то можно идентифицировать любые рекурсивные языки [Gold, 1967]. Этот же результат справедлив и для функционального синтеза [Blum et al., 1975; Мартиросян, 1986]: множество всех вычислимых функций идентифицируется по примитивно-рекурсивному перечислению своих графиков. Если допускать любые эффективные (алгоритмические) перечисления, то выигрыша не получается: множество вычислимых функций идентифицируется по эффективным перечислениям тогда и только тогда, когда оно идентифицируется по всем возможным представлениям. Другие возможные эффекты от ограничений подобного рода описаны в [Wiehagen, 1978]. В [Blum et al., 1975] показано, что описание, на котором стабилизируется АПВ, можно сделать независимым от того, с каким из допустимых перечислений оно работает. Этот результат останется справедливым в любом случае, если класс допустимых перечислений замкнут относительно эффективных преобразований [Мартиросян, 1986].

Ограничения на процесс решения задач ОП могут вытекать как из специфики данной задачи, так и из общих для обучения требований, например *согласованности*. Машина M называется согласующей, если всякая гипотеза, выданная M , согласована с тем начальным отрезком допустимого представления, на основе которого она генерирована.

Приведем содержательные примеры, показывающие, что требование согласованности ограничивает возможности АПВ. Пусть фиксирована некоторая спецификация $S = \langle D, H, P, C \rangle$ задачи ОП и M — некоторый АПВ. Пусть $d \in D$ — описание и $p_d \in P$ — допустимое представление d (для простоты предположим, что критерий успешности C включает условие стабилизации гипотез). Возможны три исхода при работе M с p_d : 1) M не стабилизируется ни на одной гипотезе, 2) M стабилизируется на гипотезе, удовлетворяющей C , и 3) M стабилизируется на гипотезе, не удовлетворяющей C . Машину M будем называть *надежной*, если ни для одного $d \in D$ и $p_d \in P$ третья из этих возможностей не реализуется. Иначе говоря, M либо идентифицирует данное d , либо не приходит ни к какому выводу. Заметим, что надежные машины наследуют свойства независимости от порядка. Всякая согласующая машина (АПВ) надежна. Без нарушения общности верно и обратное: для любой надежной машины M можно построить «равномощную» ей согласующую машину M' . Таким образом, класс описаний, синтезируемых надежными и согласующими машинами, один и тот же.

Надежные АПВ обладают рядом замечательных свойств. Например, если $U_1, U_2 \in \text{Infs}^R$, то $U_1 \cup U_2 \in \text{Infs}^R$, где Infs^R — класс множеств, S -идентифицируемых надежными АПВ. Это свойство аддитивности справедливо и для бесконечного числа слагаемых. Далее, если $U \in \text{Infs}^R$ и U' — «похожее» на U множество описаний (например, все описания из U' получаются применением к описаниям из U некоторой стандартной процедуры или описания из U' отличаются от описаний из U каждый раз только конечным числом примеров и т. д.), то часто удается по надежной M , идентифицирующей U , построить надежную M' , идентифицирующую U' .

Другим интересным свойством надежных алгоритмов в задачах синтеза функций (точнее, для задач, в которых критерий успешности допускает «сверх-обобщение данных») является то, что вместе с синтезом описания всегда можно обеспечить синтез любого его подописания, охватывающего более узкое множество примеров, чем исходное. Это свойство равномерности вне связи с надежностью рассматривается в [Фрейвалд и др., 1975]. Ограничение надежными АПВ

имеет и методологические преимущества. Оказалось [Мартиросян, 1986], что классы множеств, идентифицируемых надежными АПВ в разных вариантах задачи функционального синтеза, часто совпадают: $\text{Inf}_S^R = \text{Inf}_S^R$. Все приведенные результаты являются следствием требования надежности, т. е. в общем случае они неверны. Дополнительные сведения о надежности (согласованности) можно найти в [Angluin et al., 1983].

Иногда рассматривается понятие *относительной надежности* АПВ. Если D' — подмножество описаний (индивидуальных задач), то M называется D' -надежной, если условие надежности выполнено для всех описаний $d \in D'$. Например, если S есть задача синтеза вычислимых функций, T — класс всех всюду определенных вычислимых (общерекурсивных) функций, то T -надежность машины M означает, что по любому перечислению графика общерекурсивной функции машина M должна либо не стабилизироваться вообще, либо синтезировать в пределе алгоритм вычисления этой функции. T -надежные машины могут идентифицировать такие множества общерекурсивных функций, которые не могут быть надежно идентифицированы. Поэтому не следует ограничиваться согласующими алгоритмами.

Еще одним часто встречающимся понятием является *консервативность* АПВ. Консервативный АПВ изменяет выдаваемую гипотезу только в том случае, если вновь полученный элемент допустимого представления не согласован с ней. Консервативность не всегда можно обеспечить: например, существует множество вычислимых функций, такое, что U идентифицируется согласующей машиной, но не идентифицируется машиной, являющейся к тому же консервативной [Kugel, 1977; Мартиросян, 1986]. Аналогичное утверждение справедливо и для языкового синтеза [Angluin, 1980a]. Другие естественные ограничения обсуждаются в [Jantke et al., 1981].

В задаче индуктивного вывода входит также характеристика возможностей различных классов АПВ, т. е. определение множеств описаний, синтезируемых для задач ОП [Wiehagen, 1978; Angluin, 1980a; Zeugmann, 1983*]. Большинство таких характеристик сложно и содержательно не интерпретируется. Остановимся на тех, которые являются исключением из этого правила. Класс NV (класс множеств общерекурсивных функций, прогнозируемых всюду определенными АПВ) совпадает с классом всех эффективно перечислимых множеств общерекурсивных функций [Бардзинь и др., 1972]. Класс множеств общерекурсивных функций, идентифицируемых согласующими и одновременно консервативными АПВ, совпадает с NV [Мартиросян, 1986].

Дополнительная информация о решаемой задаче часто имеет существенное значение. Например, любую вычислимую функцию можно идентифицировать, если известно, что ее допустимое представление осуществляется примитивно-рекурсивной функцией [Gold 1967; Blum 1975]. В качестве дополнительной информации может служить и родовидовая принадлежность индивидуальной задачи. В [Jantke, 1978] показано, что можно таким образом подобрать систему множеств вычислимых функций $U_1, U_2, \dots, U_n, \dots$, что любая общерекурсивная функция f может быть идентифицирована по своему графику и подходящему номеру i , такому, что $f \in U_i$. Дальнейшее исследование синтеза с дополнительной информацией можно найти в [Freivald et al., 1979].

При создании эффективных процедур или методов необходимо исследовать вопросы сложности решения задач ОП. Если $S = \langle D, H, P, C \rangle$ — некоторая спецификация задач ОП, s — индивидуальная задача из S (задаваемая фиксацией какого-либо описания d), то обозначим через $CP(M, p_d)$ количество данных, требуемых машине M при работе с представлением p_d задачи s до момента стабилизации гипотез (CP — от Convergence Point). Будем говорить, что M_1 эффективнее по данным, чем M_2 , если $\text{Inf}_S(M_2) \subseteq \text{Inf}_S(M_1)$ и $CP(M_1, p_d) <$

* Zeugmann Th. Aposteriori characterizations in inductive inference of recursive functions// Elektron. Informations-verarb. Kybern. (EIK). — 1983. P. 559—594.

$\leq CP(M_2, p_d)$ для любого допустимого представления p_d индивидуальной задачи s из $\text{Inf}_s(M_2)$. В [Gold, 1967] показано, что метод перечисления пространства гипотез оптимален по этому критерию.

Желательно иметь понятие сложности, зависящее только от конкретной задачи, а не от ее представления, полагая, например, $CP(M, s) = \max\{CP(M, p_d) | p_d \in P\}$. Но при таком подходе в большинстве случаев $CP(M, s) = \infty$. Этот же недостаток присущ многим другим определениям сложности, имеющимся в литературе. К их числу относятся количество различных гипотез, выданных при работе с p_d до момента стабилизации, количество изменений гипотез до этого же момента и т. п. [Бардзинь и др., 1972, 1974; Фрейвалд, 1975; Княбер, 1977; Feldman et al., 1977]. В [Фрейвалд, 1975; Daley, 1977; Daley et al., 1983] предложен аксиоматический подход к определению сложности задач ОП. Большинство результатов по оценке сложности носят негативный характер: алгоритмов, которые были бы существенно лучше переборного, не существует [Gold, 1967].

Случаи, когда эффективные АПВ можно построить, чаще всего оказываются неинтересными с практической точки зрения. В теории индуктивного вывода особое внимание уделяется изучению предельного поведения синтезирующих алгоритмов, хотя в большинстве задач имеющееся реально множество примеров всегда ограничено. Такая ситуация способствует разрыву между теорией и практикой построения обучающихся систем.

6.3. Обучающиеся системы

Общие методы выдвижения гипотез

Способ выдвижения гипотез тесно связан с умением сравнивать гипотезы: лучшей считается гипотеза, которая «проще» и «более совместима с исходными данными», чем другие. Содержание, которое в каждом конкретном случае вкладывается в эти понятия, может меняться. Можно считать, что заданы два отношения α и β , называемые отношениями предпочтения и совместимости, такие, что

$$h_1 \alpha h_2 \Leftrightarrow \text{«} h_1 \text{ проще } h_2 \text{»};$$

$$h \beta V \Leftrightarrow \text{«} h \text{ совместима с обучающей выборкой } V \text{»}.$$

Тогда естественно полагать, что гипотеза h_1 лучше гипотезы h_2 при выборке V , если $h_1 \alpha h_2$ и $h_1 \beta V$, $h_2 \beta V$. Например, в задачах синтеза языков часто применяются следующие отношения:

$$h_1 \alpha h_2 \Leftrightarrow \text{«} L(h_1) \subseteq L(h_2) \text{»}, \quad h \beta V \Leftrightarrow \text{«} V^+ \subseteq L(h) \text{ и } V^- \cap L(h) = \emptyset \text{»},$$

где V^+ и V^- — множества позитивных и негативных примеров из V ; $L(h)$ — язык, определяемый гипотезой h .

Если для любых h_1 и h_2 из $h_1 \alpha h_2$ следует, что h_1 лучше h_2 при любой выборке V , с которой h_1 и h_2 совместимы, то отношение предпочтения α называется независимым от выборки. Для задач синтеза функций независимым от выборки отношением предпочтения является, например, упорядочение программ по их длине.

Если независимое от выборки отношение предпочтения вычислимо, можно предложить простую схему АПВ (Р), применяемую во многих приложениях.

Шаг 0. Вычислить наилучшую при данной выборке гипотезу. Потребовать новый пример. Перейти к шагу 1.

Шаг n ($n > 0$). Если текущая гипотеза совместима с примерами, то потребовать новый пример и перейти к шагу $n+1$. В противном случае перейти к шагу 0.

Эта схема будет сходиться к наилучшему описанию, если таковое действительно существует (в смысле отношения α). Отметим, что и промежуточные гипотезы будут наилучшими для каждой данной выборки описаниями.

Рассмотрим задачу синтеза конечных автоматов по позитивным и негативным примерам. Будем говорить, что автомат A_1 лучше автомата A_2 , если число состояний у A_1 меньше, чем у A_2 . Ясно, что нахождение наилучшего автомата возможно: упорядочив все автоматы по количеству состояний, выберем первый такой, который принимает данное множество позитивных и отвергает данное множество негативных примеров. Применение схемы P приводит к обычной схеме «идентификации по перечислению пространства гипотез», что позволяет синтезировать конечный автомат для любого регулярного языка [Gold, 1978]. Отметим, что задача нахождения автомата с минимальным числом состояний, совместимого с данным множеством примеров, NP-полна.

Аналогичное отношение предпочтения можно сформулировать для задачи синтеза языков по позитивному представлению: гипотеза $h_V \in H$ считается наилучшей для обучающей выборки V , если $V \subseteq L(h_V)$ и для всех $h \in H$, таких, что $V \subseteq L(h)$, выполнено $L(h_V) \subseteq L(h)$, где $L(X)$ обозначает язык, определяемый гипотезой X . Такое отношение предпочтения является независимым от выборки и даже вычислимым для многих случаев. Для некоторых из них найдены полиномиальные алгоритмы для вычислений h_V [Angluin, 1980b, 1982b; Shinohara, 1982].

Большое распространение получили отношения предпочтения, основанные на использовании теоремы Байеса. Пусть $\Pr(h)$ есть вероятность появления гипотезы h в (вероятностном) пространстве H ; $\Pr(V)$ — вероятность появления выборки V в пространстве примеров $\Pr(V/h)$ — вероятность того, что h совместимо с V . Тогда вероятность

$$\Pr(h/V) = [\Pr(h)\Pr(V/h)] / [\Pr(V)]$$

может быть использована для определения отношения α : $h_1 \alpha h_2$ тогда и только тогда, когда $\Pr(h_1/V) \geq \Pr(h_2/V)$. Задача состоит в максимизации $\Pr(h/V)$ или, что то же, максимизации $\Pr(h)\Pr(V/h)$. Ясно, что это отношение предпочтения не является независимым от выборки и схема P не работает.

Конкретные методы подсчета вероятностей зависят от предметной области [Horning, 1969; Вапник и др., 1974; Фукунага, 1979].

Связь отношения предпочтения с теоремой Байеса обсуждается в [Solomonoff, 1964]. Более подробный анализ отношений предпочтения можно найти в [Biermann et al., 1972a, b; Angluin et al., 1983].

При поиске метода решения задач ОП большое значение может иметь структурированность пространства гипотез. Отношение предпочтения можно рассматривать как один из способов структуризации. В противоположность простому перечислению структурированность позволяет исключить не одну, а группу неудачных гипотез.

Рассмотрим структуру пространства гипотез, часто применяющуюся в системах формирования понятий и индуктивного вывода. На множестве всех конъюнкций атомарных формул введем отношение предпочтения, определяемое следующим образом: $K_1 \alpha K_2$ тогда и только тогда, когда для некоторой подстановки σ множество атомов из $\sigma(K_1)$ есть подмножество множества атомов из $\sigma(K_2)$. Будем говорить в этом случае, что K_1 «общее» K_2 (например, $A(x) \& B(x) \& C(y)$ общее $A(a) \& B(a) \& C(f(b)) \& D(g(y))$). Задача обучения (формирования понятия) заключается в нахождении минимальной общей (по α) конъюнкции для заданного множества фактов, представленных атомарными формулами, не содержащими переменных. Если некоторая конъюнкция не согласована с каким-либо фактом, то и все менее общие конъюнкции можно исключить (при наличии отрицательных примеров можно исключать и более общие конъюнкции). Такого рода пространства гипотез и более сложные теоретически исследовались в [Plotkin, 1970, 1971; Reynolds, 1970]. Структуры, основанные на этой простой идее, используются во многих алгоритмах формирования понятий [Hayes-Roth et al., 1978; Vere, 1980; Dietterich et al.; 1981].

Структурированность пространства гипотез позволяет вводить в обучающуюся систему специальные диагностические компоненты, анализирующие причины, по которым гипотеза оказалась несовместимой с данными. Результатом работы

диагностических компонентов должно стать выявление структурных единиц гипотезы, обусловивших ее несостоятельность.

В [Shapirō, 1981] эта идея применяется к задаче индуктивного синтеза теорий по фактам. Пространство гипотез считается состоящим из всех возможных конечных множеств аксиом $A_1 \& \dots A_k \rightarrow A_0$ (A_i — атомарные формулы или их отрицания), т. е. фактически из всех возможных Пролог-программ; обучающая выборка состоит из атомарных формул (без переменных), помеченных символами И («истина») или Л («ложь»). Задача заключается в нахождении теории (Пролог-программы), в которой все позитивные примеры (формулы, помеченные символом И) были бы выводимы, а все негативные (помеченные Л) — невыводимы. Если текущая гипотеза оказалась несовместимой с фактами, то диагностическая процедура в интерактивном режиме выясняет, какая из аксиом, входящих в гипотезу, может быть причиной неудачи. Пролог-программы, содержащие эту аксиому, исключаются из дальнейшего рассмотрения.

Наибольшее число практических разработок основано на методах выдвижения гипотез, использующих идею локальной оптимальности гипотез. Представление о задачах обучения как об оптимизационных задачах характерно для многих подходов к проблеме обучения и в особенности для стохастических подходов с аналитическими функционалами качества [Цыпкин, 1970; Вапник и др., 1974]. Суть метода состоит в том, что выдвинутая гипотеза остается неизменной, если по мере поступления новых данных она продолжает оставаться оптимальной в своей окрестности. В противном случае она заменяется наилучшей в своей окрестности гипотезой.

В [Cook et al., 1976] можно найти пример такого «подъема на холм» для дискретной стохастической задачи — идентификации стохастических контекстно-свободных грамматик по случайным позитивным данным. Вводится комбинаторная мера $M[G, V]$, отображающая сложность грамматики G и одновременно степень ее совместимости с обучающей выборкой V . К текущей гипотезе можно применять определенные трансформации. Грамматики, получающиеся однократным применением какой-либо трансформации к текущей гипотезе, образуют ее окрестность. Если некоторая грамматика из этой окрестности оказалась лучше по мере M , она выбирается в качестве очередной гипотезы. В качестве первой гипотезы выбирается грамматика, порождающая в точности заданную обучающую выборку V (этой грамматике приписывается большая сложность и большая степень совместимости с V).

Рассмотренные до сих пор методы не всегда эффективны, поскольку ориентированы прежде всего на то, чтобы не пропускать в процессе перебора гипотез, совместимых с обучающей выборкой. Методы другого типа основаны на процедурах, которые исходя из обучающей выборки конструируют те свойства, которым должна удовлетворять разумная гипотеза. Такая идея использована, например, в эвристическом методе синтеза конечных автоматов по входным-выходным данным реализуемых ими функций [Biermann et al., 1972b]. Из всех возможных разбиений множества состояний канонического древесного автомата выбирается то, при котором в один блок попадают состояния, неразличимые для входных слов длины не более k (параметр k задается пользователем). Получаемый в результате склейки k -неразличимых состояний автомат может оказаться недетерминированным, однако при увеличении числа примеров и параметра k в пределе будет получен корректный детерминированный автомат. Этот метод был затем обобщен для синтеза грамматик [Brayer et al., 1977; Levine, 1982].

Обучение распознаванию образов

Вопросы обучения по примерам сравнительно хорошо исследованы для проблемы поиска алгоритмов классификации (ПАК). Пусть в проблеме ПАК L, T — конечное множество (универсум), $n = |X_L|$, $Y_L = \{0, 1\}$, $T = \{(X_1, X_0) | X_1, X_0 \subseteq X_L \& X_1 \cap X_0 = \emptyset\}$ — пары множеств (ПМ) из X_L (частично определенные предикаты на X_L); $T^n = \{(X_1, X_0) | (X_1, X_0) \in T \& X_1 \cup X_0 = X_L\}$ — абсолютные ПМ (предикаты на X_L); $T' \subseteq T^n$; ρ_L — конкретный предикат из T' , известный только оракулу (или учителю); Q_L — заданный частично определенный предикат из T , связанный

с ρ_L (дополнительная информация); F_L — множество алгоритмов $f: X_L \rightarrow Y_L$, предназначенных для моделирования ρ_L и именуемых алгоритмами классификации X_L ; J_L — оператор качества алгоритмов из F_L , например $J_L = \langle J_0, T_0, S_0 \rangle$, где $J_0(f) = \sum_{x \in X_L} (f(x) - \rho_L(x))^2$ — хеммингово расстояние классификации, полу-

ченной $f \in F_L$, от истинной ρ_L ; $T_0(x)$ — максимальное по всем $x \in X_L$ время вычисления значений $f(x)$ (время распознавания); $S_0(f)$ — память, используемая при хранении f в машине. Требуется построить оптимальный по J_L алгоритм классификации f_L^* из F_L .

Класс $A = \{\text{ПАК}\}$ определяется множеством всех конечных множеств X_L . Если f_L^* может быть построена без обращения к учителю с использованием лишь свойства Q_L , то проблема ПАК называется *инициально полной*; в противном случае — *инициально пустой*.

Синтез классификаций по примерам для инициально пустых ПАК рассматривается в основном в теории распознавания образов и называется обучением распознаванию. В теориях статистических гипотез и параметрической адаптации обучение распознаванию формулируется как разновидность проблемы минимизации среднего риска [Вапник, 1974], получаемый из проблемы ПАК при следующих дополнительных требованиях:

1) X_L есть множество n -мерных векторов с распределением вероятностей $P(x)$; каждый вектор представляет результат измерений множества объектов n датчиками с конечной и дискретной шкалой измерений;

2) компоненты произвольной абсолютной пары множеств $V = (X_1, X_0)$ на X_L являются реализациями случайных величин B_1^V и B_0^V соответственно;

3) на множестве X_L существуют совместные распределения вероятностей $p(\omega, x)$, $\omega = 0, 1$, которые задают вероятности событий типа: появился вектор $x \in X_L$ и $x \in X_1$ ($x \in X_0$); $p(\omega, x)$ характеризует множества элементов из X_L , о принадлежности которых будет задан вопрос учителю; при этом $p(\omega, x) = p(x)p(\omega|x)$;

4) каждая из рассматриваемых абсолютных пар множеств задается характеристической функцией $F(x, c)$ при некотором c , где $x \in X_L$; c — m -мерный вектор параметров;

5) оператор качества решений задается в виде

$$J(c) = \int (\omega - F(x, c))^2 dp(\omega, x). \quad (1)$$

Требуется найти минимум по c функционала $J(c)$, если неизвестна функция распределения $p(\omega, x)$, но задана случайная и независимая выборка $x_1\omega_1, x_2\omega_2, \dots, x_L\omega_L$ — обучающая последовательность.

Так, для персептрона оператор качества решений имеет вид

$$J(c) = \int \left(\omega - \theta \sum_{i=1}^m c_i \varphi_i(x) \right)^2 dp(\omega, x),$$

где $\theta(z)$ — пороговая функция, $\varphi_i(x)$ — некоторые преобразования на X_L (А-элементы).

Существуют три пути решения задачи (1). Первый относится к теории статистических гипотез и связан с идеей восстановления функций распределения $p(\omega, x)$ векторов первого и второго класса, а затем построением на их основе дискриминантной функции. Поскольку построение функций распределения является более сложной задачей, чем построение искомой дискриминантной функции, данный путь часто бывает нерационален.

Второй путь связан с организацией рекуррентной процедуры поиска параметров c , дающего минимум (1). Исходя из процедуры $c[t]$ для (1) имеем

$$c[t] = c[t-1] + \Gamma(t) \operatorname{grad}_c \xi(\omega(t) - F(x(t), c[t-1])). \quad (2)$$

Выбирая различные выпуклые функции ξ , получаем ряд известных алгоритмов [Цыпкин, 1968, 1970; Вапник и др., 1985*], в частности обучения трехслойных перцептронов и метода потенциальных функций [Айзерман и др., 1970]. В общем случае решение в соответствии с (2) гарантируется при неограниченном увеличении обучающей выборки. Достаточные условия сходимости (2) при конечных выборках и оценки длины этих выборок приведены в [Вапник и др., 1974].

Третий путь решения (1) относится (как и второй) к теории параметрической адаптации и связан с идеей замены (1) так называемой функцией эмпирического риска

$$J_{\text{эмп}}(c) = \frac{1}{l} \sum_{i=1}^l (\omega_i - F(x_i, c))^2, \quad (3)$$

построенной по случайной независимой выборке x_1, \dots, x_l и определяющей частоту неправильных классификаций на этой выборке. Метод состоит в поиске $c_{\text{эмп}}$, обеспечивающего минимум эмпирического риска. Теория призвана выявить условия, при которых переход от (1) к (3) возможен. Основой их является требование

$$p \left\{ \sup_c |J_{\text{эмп}}(c) - J(c)| > \varepsilon \right\} \xrightarrow{l \rightarrow \infty} 0 \quad (4)$$

равномерной сходимости частот $J_{\text{эмп}}$ к вероятностям $J(c)$ по классу S событий, каждое из которых определяется параметром c и указывает на множество ошибочно классифицированных векторов x при решающем правиле $F(x, c)$.

Необходимым и достаточным условием выполнения (4) является стремление к нулю последовательности

$$\frac{H^S(1)}{1}, \frac{H^S(2)}{2}, \dots, \frac{H^S(l)}{l}, \dots$$

при неограниченном увеличении длины выборки l . Здесь $H^S(l)$ — энтропия класса S решающих правил $F(x, c)$, определяет математическое ожидание логарифма от числа возможных разбиений $\Delta^S(x_1, \dots, x_l)$ выборки длины l в классе S . На практике вместо энтропии класса используется логарифм функции

$$m^S(l) = \max_{x_1, \dots, x_l} \Delta^S(x_1, \dots, x_l),$$

называемой функцией роста класса S , и соответствующее достаточное условие равномерной сходимости

$$\lim_{l \rightarrow \infty} \frac{\lg_2 m^S(l)}{l} = 0. \quad (5)$$

В [Вапник и др., 1974] доказано, что функция роста либо тождественно равна 2^l , либо мажорируется степенной функцией $1,5 \frac{l^{n-1}}{(n-1)!}$, где n — минимальное число, при котором нарушается равенство $m^S(l) = 2^l$. При этом во втором случае имеет место (5). Тем самым число $n-1$ — мера разнообразия или емкость класса решающих правил — определяет достаточные условия выполнения (4). В этом случае говорят об ограниченной, или конечной, емкости класса правил.

Таким образом, параметрическое обучение будет успешным, если емкость класса, в котором ищется решение, конечна, а само решение доставляет минимум эмпирического риска, т. е. по возможности с меньшим количеством ошибок разделяет элементы обучающей выборки разных классов.

Из разнообразных методов минимизации эмпирического риска чаще других используются два.

* В списке литературы неправильно указан 1984 г.

Метод обобщенного портрета [Вапиик и др., 1974]. Строится разделяющая гиперплоскость, минимизирующая число ошибок обучающей выборки. Если это число не равно нулю, строится вторая гиперплоскость, минимизирующая остаточные ошибки. Если ошибок еще много, то строится еще одна гиперплоскость и т. д. Алгоритмы этого типа реализуют разные идеи кусочно-линейной аппроксимации.

Алгоритмы частичной прецедентности [Журавлев, 1978]. Для случая двух классов и n -мерного булева пространства E_n , все векторы которого считаются допустимыми, этот алгоритм задается с помощью следующих процедур:

F_1 — разбиения произвольной пары множеств (X_1, X_0) на обучающую (X'_1, X'_0) и контрольную (X''_1, X''_0) , такие, что $X_1 = X'_1 \cup X''_1$ и $X_0 = X'_0 \cup X''_0$;

F_2 — выбора опорных множеств, указывающих совокупности признаков для анализа векторов;

F_3 — вычисления функций близости на множестве векторов;

F_4, F_5, F_6 — вычисления оценок (при сравнении вектора с вектором и вектора с классом векторов при заданных опорном множестве или классе опорных множеств);

F_7 — определения решающего правила;

F_8 — оценки оптимальности построенных распознающих алгоритмов.

Процедуры F_1 — F_8 перерабатывают произвольную заданную пару множеств в распознающий алгоритм в такой последовательности:

1. Процедура F_1 разбивает заданную пару множеств (X_1, X_0) на (X'_1, X'_0) и (X''_1, X''_0) .

2. Процедуры F_2 — F_7 по обучающей паре множеств (X'_1, X'_0) задают параметрическое множество распознающих алгоритмов $\{R\}$, каждый из которых определяется однозначно фиксацией конкретных значений конечного числа параметров. Будем считать, что области изменения параметров также конечны. Произвольный алгоритм распознавания R , полученный указанным образом, каждый булевый вектор S из E_n сопоставляет с одним из чисел: 0, 1, 2. При этом, если

$$R(S) = \begin{cases} 1, & \text{то } S \text{ принадлежит I классу,} \\ 2, & \text{то } S \text{ принадлежит II классу,} \\ 0, & \text{то } S \text{ не распознан.} \end{cases}$$

Таким образом, каждому алгоритму R соответствует пара множеств (X_1^R, X_0^R) , такая, что $S \in X_1^R \leftrightarrow R(S) = 1$, $S \in X_0^R \leftrightarrow R(S) = 2$, $S \in E_n / X_1^R \cup X_0^R \leftrightarrow R(S) = 0$.

3. Процедура F_8 выбирает такой алгоритм из $\{R\}$, при котором достигается минимальное число ошибок на контрольной паре множеств (X''_1, X''_0) . Алгоритм R^* объявляется экстремальным.

Представление проблемы обучения распознаванию как проблемы минимизации среднего риска предполагает пригодность статистической параметрической модели для исследования проблемы распознавания, в частности возможность описания компонентов искомой пары множеств в языке статистической теории (т. е. через понятия о среднем, дисперсии, корреляции, типе распределения и т. п.), что не всегда выполняется.

Например, допущение о том, что X_1 может быть описано через указание векторов средних значений и их дисперсий, практически означает следующее. Выбрана система n датчиков (измерителей, предикатов), которые преобразуют реализации некоего случайного объекта B_1 в множество векторов X_1 таким образом, что средние значения по каждой компоненте этих векторов описывают особенности B_1 , достаточно существенные, чтобы на их основе, также по их отклонениям от средних (дисперсии) отделять B_1 от B_0 . Другими словами, датчики должны быть выбраны для измерения значений тех признаков случайного объекта B_1 , которые существенны для отделения B_1 от B_0 , т. е. предполагается знание существенных для описания B_1 признаков. Однако известно, что выбор

существенных признаков, по которым B_1 и B_0 могут быть разделены, является трудной проблемой, сравнимой с проблемой распознавания.

Если система существенных признаков задана, то статистические методы способны лишь устранить возможные случайные отклонения от средних, измеряемых по этим признакам, значений, т. е. устранить помехи при условии, что измерители способны оперировать сигналами.

Поскольку статистический подход в проблемах распознавания часто связывается с восстановлением функции распределения вероятностей и, как правило, нормального распределения, которое определяется заданием среднего значения и дисперсии, то вышесказанное характеризует требования, необходимые для успешного применения этого подхода: 1) распознаванию подлежат множества реализаций случайных объектов B_1 и B_0 ; 2) задана система n измерителей, которые преобразуют множество реализаций случайных объектов B_1 и B_0 в множества n -мерных векторов X_1 и X_0 соответственно; 3) измерители таковы, что анализ средних значений и дисперсий полученных измерений в совокупности достаточен для отделения B_1 от B_0 , т. е. измеряются «существенные» характеристики B_1 и B_0 . Эти требования, как и требования параметрического представления класса распознающих алгоритмов, во многих проблемах распознавания невыполнимы, и проблема задается в своем исходном комбинаторном представлении в форме ПАК.

Комбинаторность проблем выражается, в частности, в следующих особенностях.

1. Высокая степень индивидуальности, различие как проблем из A , так и элементов из X_L при $L \in A$. Вследствие этого построение решения f^*_L и вычисление $f^*_L(x)$ при $x \in X_L$ связано с длительным процессом изучения и приспособления к специфике L и x , требующим дополнительной информации о них извне.

2. Оператор I может быть плохо формализуем или трудновычислим, вследствие чего непосредственное использование известных методов оптимизации, основанных на аналитических свойствах, становится невозможным.

Для комбинаторных задач распознавания известны логические, лингвистические, структурные методы, обучение понятиям, методы формирования понятий Ханга [Хант, 1970], методы растущих пирамидальных сетей [Гладун, 1970] и др. Отметим алгоритм «Кора» [Бонгард, 1967] формирования понятий в виде дизъюнкции элементарных разделяющих функций. Для построения последних из множества заданных признаков выделяются достаточные признаки для элементов обучающей выборки каждого класса, которые выполняются только на элементах этого класса и в совокупности покрывают весь этот класс. При опознании нового элемента подсчитывается число выполненных достаточных признаков каждого класса и решение принимается по большинству из них. В [Вазник и др., 1974] показано, что если значение эмпирического риска близко к нулю, то вероятность неправильной классификации с помощью алгоритма «Кора» уклонится от эмпирической оценки не более чем на

$$\varepsilon = (6t \ln n - \ln \eta) / l,$$

где n — размерность пространства векторов выборки; t — число достаточных признаков; η — верхняя оценка (4), $0 < \eta < 1$.

К особенностям логических методов обучения или индуктивной логики относятся использование малого числа примеров для формирования понятий, обобщение не только по признакам, но и по структуре и именам [Поспелов Д., 1986]. Один из них, основанный на принципах индуктивных рассуждений Бэкона — Милля, развивается в [Финн и др., 1981]. Логические методы формирования понятий являются основными для систем, основанных на знаниях. Они широко применяются в ситуационном управлении [Поспелов Д., 1981, 1986] и при моделировании шахматной игры [Адельсон-Вельский и др., 1978; Погосян, 1983].

Синтез классификаций для инициально полных ПАК исследуется в теории комбинаторной оптимизации (распознавание слов с заданными свойствами) [Гэри и др., 1982]. Если итеративность в распознавании образов обусловлена,

как правило, ограниченностью доступной для анализа информации, то при комбинаторной оптимизации — ограниченностью вычислительных средств и требованием быстрой окупаемости затрат при построении гипотез. При конкретных всюдю определенных предикатах Q_L получаем инициально полные проблемы ПАК, эквивалентные ограниченным проблемам распознавания слов со свойством Q_L [Гэри и др., 1982]. В частности, при X_L — множестве n вершинных графов, $Q_L(G)$ — предикате «граф G имеет гамильтонов цикл» получаем ограниченную проблему L распознавания гамильтонова цикла в графах с числом вершин не более n . Инициально полные проблемы формирования образов ситуаций в конечных играх типа шахмат сформулированы в [Погосян, 1983]. Ряд задач по построению классификаций в ситуационном управлении при заранее заданных критериях качества управления также относятся к инициально полным [Поспелов Д., 1986].

Формирование понятий

Описания в процессе обучения формулируются обычно в виде языковых текстов, интерпретируемых в данной проблемной области, а не в терминах, скажем, разделяющих гиперплоскостей, содержательная интерпретация которых усложнена. Задачи такого рода будем называть *задачами синтеза (формирования) понятий*. Неформально: их определить можно следующим образом [Michalski, 1983].

Дано множество наблюдений (фактов) F , представляющих специфические знания о некоторых объектах, ситуациях, процессах и т. д.; множество исходных предположений, выражающих ограничения, которым должно подчиняться искомое описание; контекстная информация, включающая предположения, налагаемые на F и F' , конструкции, рекомендованные для генерации кандидатов на роль результирующего описания, критерий успешности, характеризующий обязательные для результирующего описания свойства, а также любые другие знания, релевантные предметной области. Предполагается, что языки, в которых записывается данная информация, фиксированы. Требуется сформировать удовлетворяющее критерию успешности описание H , такое, что все факты из F выводятся в качестве следствий из H : $H \models F$. Заметим, что отношение \models не обязательно совпадает с отношением следования в исчислении предикатов и может задаваться как контекстная информация. Критерий успешности для задач формирования понятий включает ряд трудноформализуемых семантических и прагматических условий, налагаемых на H , таких как, интересность, новизна, наличие практических следствий и т. д. При этом классификационный аспект H (т. е. разбиение объектов на два класса — принадлежащих и не принадлежащих H) может оказаться не самым важным. Например, при формировании математического понятия АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ в описываемой далее программе АМ [Lemat, 1976] важнее, к каким следствиям оно может привести, а не какие объекты следует считать операциями, а какие — нет. Таким образом, постановка задачи формирования понятий шире, чем задачи обучения распознаванию образов.

Различают два типа задач формирования понятий: *обучение по примерам* (называемое также *обучением с учителем*) и *обучение на основе наблюдения* [Michalski, 1983]. При обучении по примерам множество F представляет собой набор синтаксически однородных описаний объектов (примеров и контрпримеров формируемого понятия): $F = \{e_i^+\} \cup \{e_j^-\}$. Требование «выводимости» $H \models F$ приводит к тому, что должны выполняться условия: $\forall i (H(e_i^+))$ (полнота); $\forall j \neg H(e_j^-)$ (совместимость).

При обучении на основе наблюдения факты из множества F могут иметь разную структуру, условия полноты и совместимости не выполняются и целью индуктивного предложения H является только описание («объяснение») фактов из F . Такая задача сложнее предыдущей. Примерами обучения на основе наблюдения является большинство задач, решаемых экспертными системами, поскольку экспертные решения часто основываются на обобщении или «объяснении» имеющихся данных (формирование теорий для множества химических со-

единений [Buchanan et al., 1978] или чисел [Lenat, 1976, 1984; Lenat et al., 1978], обнаружения закономерностей в данных [Soloway et al., 1977; Hayes — Roth et al., 1978; Pokorný, 1980; Гаек и др., 1984], выявление естественной классификации объектов [Michalski et al., 1981]).

При условии неполноты исходной информации центральную роль в решении задач формирования понятий должны играть процедуры обобщения (индукции). Примерами такого рода являются процедуры получения решающих правил для шахматных эндшпилей и миттельшпильных комбинаций [Michalski et al., 1977; Wilkins, 1982].

Рассмотрим обучающуюся систему [Уинстон, 1978], показывающую, как с помощью различных эвристик можно решать и задачу обучения на основе наблюдения, и задачу обучения по примерам. Эта система представляет собой четырехступенчатую совокупность процедур, позволяющих роботу ориентироваться в мире трехмерных геометрических тел (кубиков).

Первой ступенью иерархии программ служит подсистема анализа двухмерных изображений сцен, составленных из кубиков. Целью подсистемы является распознавание геометрических тел, составляющих сцену, и ее описание в терминах базовых свойств и отношений, таких, как **КЛИН**(x) (т. е. « x представляет собой объект, имеющий форму клина»), **СЛЕВА**(x, y) (« x слева от y »), **СПРАВА**(x, y), **НАД**(x, y), **ПОДДЕРЖИВАЕТСЯ**(x, y), **ПЕРЕД**(x, y), **СОВМЕЩАЕТСЯ**(x, y) и т. д. Полученное описание представляется в виде сети, вершины которой соответствуют объектам и их свойствам, а дуги (называемые указателями) — отношениям. Задача этого этапа укладывается в рамки модели обучения на основе наблюдения.

Программы второй ступени производят разбиение множества объектов на группы, с которыми можно обращаться как с самостоятельными объектами. Эта «естественная классификация» производится без учителя на основе поиска объектов, связанных цепочками указателей или находящихся в одинаковых отношениях к некоторому другому объекту, связывающему члены потенциальной группы.

Программы третьей ступени предназначены для формирования с помощью учителя новых понятий, имеющих более сложную природу, чем «естественные классификации». Центральной идеей здесь является использование контрпримеров.

Программы четвертой ступени позволяют роботу на основе накопленных знаний планировать целесообразное поведение в мире кубиков.

Рассмотрим механизм «естественной классификации» объектов. Первыми кандидатами в группу являются объекты, связанные друг с другом цепочкой указателей типа **ПОДДЕРЖИВАЕТСЯ** и **ПЕРЕД**. Не все такие цепочки порождают группу. Например, если некоторая часть сцены представляет собой композицию, состоящую из высокой башенки (четыре больших кубика) с широкой площадкой наверху и второй маленькой башенкой (три маленьких кубика) на этой площадке, то несмотря на то, что все элементы сцены связаны отношением **ПОДДЕРЖИВАЕТСЯ**, система выделит две группы (большую и маленькую башни), а не одну. Это объясняется различием в размерах объектов (система располагает указателями **БОЛЬШОЙ**, **МАЛЫЙ**). Таким образом, система сначала выдвигает гипотезу о группировании объектов, а потом проверяет ее.

Другой способ выдвижения гипотезы основан на наблюдении, что некоторое число объектов одинаково относятся к другому объекту. Ножки стола — типичная группа, образованная отчасти из-за того, что они одинаково относятся к крышке стола, и отчасти из-за того, что все они являются разновидностями брусьев и все стоят.

Проверка гипотезы производится следующим образом. Формируется список «общих отношений» в группе, т. е. отношений, в которых участвует более половины объектов группы. Для каждого объекта подсчитывается доля общих отношений, в которых участвует данный объект. Если a — максимальная доля из всех имеющихся, то объекты, доля которых составляет менее 80% от a ,

удаляются из группы и процесс повторяется снова до тех пор, пока оставшаяся группа не будет состоять из очень похожих друг на друга объектов. Информация о группировках включается в сеть, описывающую сцену, для дальнейшего использования на третьей и четвертой ступенях. Не может существовать универсальная процедура группирования, так как потребности различных программ, вызывающих процедуры группирования, могут быть различными. Например, группой можно считать тела, составляющие при подходящем совмещении некоторую заранее заданную фигуру.

Процедура обобщения по примерам работает следующим образом. Пусть системе предъявлена конструкция из трех прямоугольных брусков, соединенных в виде буквы П (назовем эту конструкцию аркой). В качестве первой гипотезы о понятии «арка» система выбирает описание сцены, полученное на предыдущих этапах. Однако по одному примеру трудно судить, какие из соотношений, присутствующих в описании, обязательны для арки, а какие факультативны. Для этого система использует новые примеры. Пусть вторая предъявленная сцена отличается от первой тем, что верхний брусок снят и положен перед двумя вертикальными. Такой пример маркирован учителем как отрицательный. Система сравнивает описание второго примера с описанием первого и находит, что единственным их различием является удаление из сети указателя ПОДДЕРЖИВАЕТСЯ. Отсюда делается вывод, что указатель имеет статус обязательного. Это выражается тем, что в сети, соответствующей текущему описанию (гипотезе), он заменяется на указатель ДОЛЖЕН ПОДДЕРЖИВАТЬСЯ. Программы нахождения аналогий и различий в сетях описаний играют на этом этапе важнейшую роль. Если описание негативного примера имеет несколько отличий от описания текущей гипотезы, то эти программы находят различие, фиксируемое на более высоком уровне сети, и объявляют его причиной расхождения, давая тем самым одному из отношений в текущей гипотезе статус обязательного. Результат сравнения сетей запоминается, поскольку может возникнуть необходимость возврата для пересмотра гипотезы о причине расхождения (из-за ее несовместимости с последующими примерами). Если текущий пример был позитивным, то различие в сетях обобщается с тем, чтобы обобщенное описание покрывало и старую гипотезу, и новый пример. Так, если в первом примере верхний брусок (горизонтальную часть буквы П) заменить треугольной призмой, то эта конструкция также будет представлять собой арку. Обучающаяся программа выводит из этого, что форма горизонтальной перекладины не существенна, т. е. свойство ПРИЗМА будет выброшено из гипотезы.

Из сказанного ясно, что качество обучающих примеров и порядок их подачи имеют большое значение. Наиболее информативными оказываются примеры, близкие к типичным для данного понятия, но не являющиеся таковыми.

Пример системы Уинстона показывает, что вместе с общелогическими правилами необходимо использовать специфические для данной предметной области знания. Осознание этого факта привело к появлению систем, в которых знания отделены от управляющей программы (хотя могут воздействовать на нее). Управляющая программа играет роль интерпретатора знаний и может быть более или менее универсальной для разных предметных областей.

Примером обучающейся системы, основанной на использовании эксплицитно представленной базы знаний, является программа АМ (Автоматический Математик) [Lepat, 1976, 1984]. АМ является системой формирования теорий в почти математическом смысле слова (предметной областью является именно математика). База знаний, содержащая вначале только пару десятков элементарных математических понятий, в процессе работы системы обогащается новыми, давая возможность выводить также «теоремы», связывающие эти понятия. АМ не имеет подсистемы дедуктивного вывода и не может оперировать понятием доказательства, поэтому ее теоремы носят эмпирический характер.

Каждое понятие в АМ представляется фреймом с набором характеристик. Каждая характеристика соответствует какому-либо вопросу, который может

быть задан относительно этого понятия, например: NAME («как называется понятие?»), DEFINITION («как определяется понятие?»), INTEREST («чем оно интересно?»), GENERALIZATION («частным случаем каких понятий является данное?»), CONJECTURES («какие выводы можно сделать?») и т. д. Всего в АМ фиксировано 25 таких характеристик. Не все характеристики могут быть заполнены, и задача системы состоит не только в формировании новых понятий, но и в уточнении уже сформированных.

Кроме фреймов, система располагает примерно 100 эвристиками, имеющими вид продукций <ЕСЛИ (условие), ТО (действие)>, в которых часть ЕСЛИ представляет собой некоторый набор предикатов, а часть ТО — набор действий, каждая из которых может быть либо предложением рассматривать новую задачу, либо инструкцией, каким образом нужно определять новое понятие или заполнить аспекты уже сформированного. Пример такой продукции:

ЕСЛИ	текущей задачей является <исследовать примеры понятия F >
И	понятие F есть оператор с областью определения A и изменения B
И	известно более чем 100 примеров из F
И	хотя бы один пример $b=f(a)$ является особым в B (например, является экстремальным в B),
ТО	для каждого такого b сформировать следующее понятие: NAME: <прообраз b по F > DEFINITION: $\lambda a F(a)$ есть b GENERALIZATIONS: A INTEREST: любой вывод с использованием F или F^{-1}
И	причиной для формирования этого понятия считать: «Стоит исследовать те $a \in A$, которые имеют необычные F -значения»
И	добавить к списку задач задачу: заполнить аспект CONJECTURES.

Продукции определяют поведение системы на нижнем уровне. Вопросы глобального управления, в том числе выбор продукции из множества применимых, выбор задач, на решение которых следует ориентироваться в данный момент, решают независимые процедуры, имеющие более общий характер, прототипом которых являются структуры [Ленат, 1975].

Особенностью АМ является автономность: система не нуждается в информации извне, поскольку сама порождает материал для своего исследования. К недостаткам следует отнести ее неспособность менять свои эвристики (обучаться новым продукциям).

Необходимость в эффективных обучающихся процедурах ощущается практически в любой области искусственного интеллекта. Диапазон реализованных систем чрезвычайно широк: от процедур пополнения базы знаний в универсальном решателе задач [Minton, 1985] до прогнозирования результатов на лошадиных бегах [Salzberg, 1985]. Много внимания уделяется обучению при решении трудных комбинаторных проблем, в частности в шахматном программировании [Skience, 1986].

Синтез программ по примерам

Большинство работ в этой области посвящено синтезу программ в ограниченном (но функционально полном) подмножестве языка Лисп. Прimitивами Лиспа, используемыми для этих целей, являются операции разбиения списков `car`, `cdr` и их суперпозиции, называемые базовыми функциями; операция сцепления списков `cons`; предикат `atom`; атомы `T`, `nil`, а также условные и функциональные вызовы. В общем случае синтезируемые программы имеют сле-

дующий вид (для двухместной синтезируемой функции):

$$(Fzx) = (\text{cond } ((p_1x) (f_1xz)))$$

$$((p_k x) (f_k x z))$$

$$(T \quad (Hx(F(bx)(Gxz))))),$$

где H и G — программы такого же типа, что и F (в частных случаях H или G могут опускаться); b — некоторая базовая функция; f_i — либо константа nil , либо комбинация car , cdr и cons , либо вызов.

Основной прием состоит в том, что для каждой пары вход-выходных списков (x, y) единым способом выписывается бесцикловая программа (называемая *траекторией* или *полутраекторией вычисления y по x*). Затем траектории, полученные по разным (x, y) , сравниваются на предмет выявления какой-либо закономерности в их структуре, позволяющей сформулировать условные и функциональные вызовы и записать общую для всех (x, y) программу. Таким образом, при известных траекториях вычисления задача сводится к восстановлению циклов в программе.

Возможность получения траекторий в Лиспе основана на следующих ограничениях: ни один атом во входном S -выражении x не встречается дважды; все атомы выходного S -выражения y , за исключением nil, встречаются в x ; единственной операцией, применяемой в процессе конструирования y по x , является присоединение некоторого атома или подписка x к имеющемуся в данный момент выходному S -выражению.

При этих условиях траектория может быть однозначно вычислена функцией ST:

$$ST(x, y) = \begin{cases} p(x), & \text{если } y = p(x), \text{ где } p \text{ — базовая функция;} \\ \text{nil}, & \text{если } y = \text{nil}; \\ \text{cons } ST(x, \text{car}(y)) \text{ } ST(x, \text{cdr}(y)) & \text{в противном случае.} \end{cases}$$

Например, при $x = (AB)$ и $y = (BA)$ траекторией будет

$$\text{ST}(x, y) = (\text{cons}(\text{cadr}(\text{cadr } x)) (\text{cons}(\text{car } x) \text{nil}))$$

(согласно принятому в Лиспе способу написания (`cadr x`) обозначает базовую функцию (`car(cdr x)`)).

Одна из первых методологий индуктивного синтеза программ изложена в [Summers, 1977]. В основе ее лежат не эвристики, а алгоритмы со строго характеризваемой областью применимости.

Ясно, что если по последовательности примеров $(x_1, y_1), \dots (x_n, y_n)$ можно построить базовые функции b_i , такие, что предикаты $(\text{ATOM } b_i x_j)$ ложны для всех $j \neq i$ и истинны для $j = i$, то программа

$$F(x) = (\text{cond}((\text{ATOM}(b_1x))(f_1x)) \dots ((\text{ATOM}(b_nx))(f_nx))),$$

где $(f_i x)$ есть $ST(x|y_i)$, будет удовлетворять всем $(x_i y_i)$.

Функции b_i выполняют роль указателей различий, выделения из входного списка ту часть, которая является атомом в случае x_i и не является атомом для остальных x . Например, для списков $x_1 = (A)$ и $x_2 = (A B)$ в качестве b_i можно брать функцию $(\text{cdr } x)$. Нахождение b_i в общем случае не представляет труда, однако для дальнейшего очень важно, чтобы они вычислялись каким-либо одним способом. В [Summers, 1977] приведен такой алгоритм вычисления b_i и доказывается, что, если, начиная с некоторого $i=k$, функции b_i и функции f_i могут быть выражены рекуррентными соотношениями $b_{k+n}(x) = b_n(b(x))$

и $f_{k+n} = C(f_n(b(x)), x)$, то функция F представляется в виде программы:

$$\begin{aligned} F(x) = & (\text{cond} ((\text{ATOM}(b_1x)) (f_1x)) \\ & ((\text{ATOM}(b_2x)) (f_2x)) \\ & \dots \dots \dots \\ & ((\text{ATOM}(b_kx)) (f_kx)) \\ & (T \quad (C(F(bx))x))). \end{aligned}$$

Эта теорема и ее обобщение устанавливают связь между рекуррентными соотношениями (повторяющимися шаблонами в траекториях) и рекурсивными схемами программы. Задача синтеза программ сводится к нахождению регулярных сегментов в траекториях примеров. Для этого можно применять, в частности, алгоритм унификации [Kodratoff, 1979]. Выбор унифицируемых траекторий в данном случае осуществляется с помощью эвристик.

В [Summers, 1977] предлагается и другой способ. В функциях f_i выбирается какой-либо константный список. Он заменяется на новую переменную z , начальное значение которой полагается равным замещенному списку. Затем ищется рекуррентное соотношение. Например, траекториями, полученными по примерам $\text{nil} \rightarrow \text{nil}$, $(A) \rightarrow (A)$, $(AB) \rightarrow (BA)$ и $(ABC) \rightarrow (CBA)$, будут $f_1 = \text{nil}$, $f_2 = (\text{cons}(\text{car } x) \text{ nil})$, $f_3 = (\text{cons}(\text{cadr } x) (\text{cons}(\text{car } x) \text{ nil}))$, $f_4 = (\text{cons}(\text{caddr } x) (\text{cons}(\text{cadr } x) (\text{cons}(\text{car } x) \text{ nil})))$. После введения новой переменной вместо nil траекторий f_3 и f_4 будут иметь вид

$$\begin{aligned} (f_3xz) &= (\text{cons}(\text{cadr } x) (\text{cons}(\text{car } x) z)), \\ (f_4xz) &= (\text{cons}(\text{caddr } x) (\text{cons}(\text{cadr } x) (\text{cons}(\text{car } x) z))). \end{aligned}$$

Рекуррентным соотношением между f_3 и f_4 будет

$$(f_4xz) = (f_3(\text{cdr } x)) (\text{cons}(\text{car } x) z).$$

Было доказано, что если найденное соотношение сохраняется и для следующих вход-выходных пар, то можно синтезировать корректную программу (без функции H):

$$\begin{aligned} (Px) &= (Fx \text{ nil}), \\ (Fzx) &= (\text{cond} ((\text{ATOM } x) z) \\ & (T \quad (F(\text{cdr } x) (\text{cons}(\text{car } x) z)))). \end{aligned}$$

Изложенный метод получил развитие в последующих работах. В [Biermann, 1978] синтезируются программы, представляющие собой набор функций F_i вида

$$\begin{aligned} (F_ix) &= (\text{cond} ((p_1x) (f_1x)) \\ & ((p_2x) (f_2x)) \\ & \dots \dots \dots \\ & (T \quad (f_{k+1}x))), \end{aligned}$$

где каждая f_j есть одна из функций nil , x , $(F_j(\text{car } x))$, $(F_j(\text{cdr } x))$, $(\text{cons}(F_mx), (F_nx))$, P_j имеет вид $(\text{ATOM}(b_jx))$ и для всех j $b_{j+1} = b_jw$, где w — базовая функция, не равная тождественной. Кроме того, ни одна из функций F_i не должна вызываться более чем одной функцией f_j . Такого рода программы называются *регулярными* (поскольку могут быть представлены конечными автоматами).

Предложенная в [Kodratoff, 1979] техника синтеза является усилением метода Саммерса. В данном случае регулярности ищутся не между траекториями, вычисляющимися y_i и y_{i+k} , а между y_i и частью y_{i+k} .

Среди списков y_{i+k} ищется такой, который можно разбить на подписки y_{i+k}^1 , y_{i+k}^2 и y_{i+k}^3 , такие, что y_{i+k}^2 связано некоторым рекуррентным соотно-

шением с y_i . После этого ставятся две новые задачи синтеза: по паре (x_i, y_{i+k}^1) и по паре (x_i, y_{i+k}^3) . Показано, что если H есть программа, синтезированная по первой подзадаче, а G — по второй, то результирующая программа будет иметь вид (6) с одним дополнительным оператором $(Px) = (Fxz_0)$, где z_0 — выбираемая в процессе работы алгоритма константа (результатом работы синтезированной программы считается P). Например, по парам $x_1 = (ABC)$, $y_1 = (ABCCBA)$ и $x_2 = (ABCD)$, $y_2 = (ABCDCCBA)$ можно построить разбиение y_2 на части $y_2^1 = (A)$, $y_2^2 = (BCDDCB)$, $y_2^3 = (A)$, причем траектория вычисления y_2^2 совпадает с траекторией y_1 при подстановке $\text{cdr } x$. Поскольку y_2^1 и y_2^3 выражаются через x как $(\text{car } x)$, то результирующая программа будет иметь вид

$$(Fxz) = (\text{cond}((\text{ATOM } x)z)$$

$$(\text{T} \quad (Hx(F \text{ cdr } x)(Gxz))))),$$

$$(Hxz) = (\text{cons}(\text{car } x)z),$$

$$(Gxz) = (Hxz),$$

$$(Px) = (Fx \text{ nil}).$$

В [Jouanpaud et al., 1977, 1979] также применяется техника разбиения на три части, однако авторы используют несколько иной базис примитивов Лиспа: $\text{icar}(ABC) = (A)$, $\text{igrac}(ABC) = (C)$, $\text{cdr}(ABC) = (BC)$, $\text{gcd}(ABC) = (AB)$, а также их композиции. По данному примеру $x \rightarrow y$ алгоритм разбивает y на три части: $y = \text{append}(py \text{ } sy)$, где s есть наибольший подпоследовательный с каким-либо подпоследовательным x . Поскольку вычисление s по x посредством базовых функций несложно, то задача сводится к синтезу функций p и s . Алгоритм находит наиболее короткие подвыражения x_1 и x_2 списка x , такие, что в x_1 и x_2 содержатся только те атомы, которые содержатся в py и sy соответственно. Применяя рекурсивно этот же алгоритм к примерам $x_1 \rightarrow py$ и $x_2 \rightarrow sy$ можно найти траекторию вычисления y по x , которая именно в силу своего построения обладает регулярной структурой и легко преобразовывается в программу. Особенностью метода является способность синтезировать программы по очень малому числу примеров.

Из приведенных примеров видно, что методы синтеза программ мало связаны со спецификой Лиспа. Они используют просто рекурсивную структуру программ. Такая постановка задачи синтеза содержится в [Бардзинь, 1982], где синтаксис и семантика программы не фиксируются, за исключением того, что язык программирования содержит итеративные DO-операторы вида $[I = a \text{ TO } b < \text{тело цикла} >]$. Задача состоит в восстановлении программ по протоколам (траекториям) их работы для частных примеров, т. е. в восстановлении DO-операторов по их разверткам. Формулируется понятие регулярности в текстах примеров и вводятся правила манипулирования этим понятием. При очень слабых ограничениях доказывается, что при большом числе примеров будет в конце концов синтезирована корректная программа (в дальнейших работах была показана также возможность синтеза WHILE-циклов). Примечательно, что часто можно восстановление программы сопровождать доказательством ее корректности. В этом смысле можно сказать, что здесь речь идет не только об индуктивном синтезе программ, но и о восстановлении индуктивных рассуждений вообще. Эта взаимосвязанность используется также в [Shapiro, 1982], где алгоритм индуктивного синтеза программ является составной частью интерактивной системы отладки Пролог-программ.

Большинство разработок по синтезу программ отличается высокая сложность вычислений. Одним из способов сокращения перебора является использование знаний. В [Biermann et al., 1979] знания, записанные в продукционной базе знаний, представляют собой некоторые схемы программ, которые после конкретизации с помощью вход-выходной информации приводят к спецификации искомой программы. Окончательная формулировка программы понимается как определение значений переменных в этой спецификации, которое также

производится с помощью базы знаний. В [Jantke, 1986; Lange, 1986] в качестве знаний используются спецификации класса программ, в котором ищется синтезируемая программа. Спецификации задаются посредством сигнатуры аксиом в исчислении первого порядка, определяющих отношения между основными примитивами в конструкции программ данного класса.

Более подробно с методами синтеза Лисп-программ можно ознакомиться в [Smith, 1982].

Системы индуктивного синтеза программ являются пока экспериментальными. Рассмотрим одно исключение из этого правила [Nix, 1983]. Представим ситуацию, когда непрограммирующий пользователь хочет изменить формат записи списка телефонных номеров таким образом, чтобы вместо записей

А. А. Иванов — 11—22—33 (сл.)	Иванов А. (сл.): 112233
Б. Б. Петров — 12—34—56 (д.)	Петров Б. (д.): 123456
Б. Б. Петров — 23—45—67 (сл.)	Петров Б. (сл.): 234567

Если формат записей в первом случае изобразить условно шаблоном «X1.X2.X3—X4—X5—X6□(X7).» (например, для первой записи X1=A, X2=A, X3=Иванов и т. д.), то шаблоном преобразованных данных будет «X3□X1. □(X7): X4X5X6.». Задача системы по заданным примерам синтезировать функцию, осуществляющую преобразование

$$X1.X2.X3 \square - \square X4 - X5 - X6 \square (X7) \rightarrow X3 \square X1. \square (X7): X4X5X6$$

Эта задача решается посредством последовательного синтеза исходного и результирующего шаблонов. Теоретическое исследование возможностей синтеза шаблонов по позитивным примерам выражений из языка, описываемого этим шаблоном, было предпринято в [Angluin, 1980 b]. Под шаблоном понимается любая конечная последовательность константных символов и переменных. Язык $L(p)$, определяемый шаблоном p , — это множество слов, получаемых подстановкой произвольных цепочек константных символов вместо переменных в p . Каждая строка из приведенной выше «телефонной книжки» дает пример слова из языка, определяемого своим шаблоном.

Было показано, что существует универсальный алгоритм, синтезирующий в пределе любой язык $L(p)$, причем по каждому конечному подмножеству $L \subseteq L(p)$ в качестве промежуточной гипотезы синтезируется шаблон p_L , являющийся наилучшим по критерию предпочтения « p' лучше p'' для L , если из $L \subseteq L(p')$ и $L \subseteq L(p'')$ следует $L(p') \subseteq L(p'')$ ». Кроме того, p_L является наименее общим шаблоном для L (в смысле подстановки шаблонов вместо переменных). К сожалению, задача определения p_L по меньшей мере NP-сложна. В то же время Д. Англин приводит примеры различных типов шаблонов, для которых эта задача полиномиальна. Для шаблонов, рассматриваемых в [Nix, 1983], задача нахождения p_L является NP-полной. Более того, если искомой программой является преобразование $p \rightarrow p'$ (p и p' — шаблоны), то нахождение p' даже при известном p также является NP-полной задачей. Тем не менее, удачный подбор эвристик позволил автору реализовать систему синтеза, как правило, справляющуюся с заданием за приемлемое время и с использованием малого числа примеров.

Отметим еще одну работу по синтезу шаблонов [Shinohara, 1982, 1986]. Если вводимая в машину текстовая информация подчиняется некоторому шаблону, система, контролирующая ввод, может через некоторое число примеров обучиться этому шаблону и известить оператора о том, что он может вводить только переменную информацию. Например, если вводимой информацией является библиография статей, задаваемых шаблоном.

«Автор: X1 Название: X2 Журнал: X3 Год (том): X4(X5) Номер X6», то система может автоматически набирать константные части шаблона, оставляя пользователю места для заполнения X1—X6. Используемый алгоритм имеет полиномиальную сложность и рассчитан на синтез так называемых регулярных шаблонов, т. е. шаблонов, в которых каждая из перечисленных частей встре-

чается только один раз (языки, определяемые такими шаблонами, являются регулярными).

В заключение главы отметим, что основными направлениями исследований в области обучения в настоящее время являются: выбор языков для формулирования эмпирических и теоретических утверждений; нахождение правил рационального индуктивного вывода для связи эмпирических утверждений; нахождение методов обоснования и оценки интересности множества выведенных теоретических утверждений.

Глава 7

Планирование решения задач

7.1. Планирование действий

А. Н. Аверкин, Е. И. Ефимов

Основные определения

Функционирование многих ИС носит целенаправленный характер (примером могут служить автономные интеллектуальные роботы). Типичным актом такого функционирования является решение задачи планирования пути достижения нужной цели из некоторой фиксированной начальной ситуации. Результатом решения задачи должен быть *план действий* — частично-упорядоченная совокупность действий. Такой план напоминает сценарий, в котором в качестве отношения между вершинами выступают отношения типа: «цель — подцель» «цель — действие», «действие — результат» и т. п. (см. § 1.5 и 2.2). Любой путь в этом сценарии, ведущий от вершины, соответствующей текущей ситуации, в любую из целевых вершин, определяет план действий.

Поиск плана действий возникает в ИС лишь тогда, когда она сталкивается с нестандартной ситуацией, для которой нет заранее известного набора действий, приводящих к нужной цели. Все задачи построения плана действий можно разбить на два типа, которым соответствуют различные модели: *планирование в пространстве состояний* (SS-проблема) и *планирование в пространстве задач* (PR-проблема).

В первом случае считается заданным некоторое пространство ситуаций. Описание ситуаций включает состояние внешнего мира и состояние ИС, характеризующиеся рядом параметров. Ситуации образуют некоторые обобщенные состояния, а действия ИС или изменения во внешней среде приводят к изменению актуализированных в данный момент состояний. Среди обобщенных состояний выделены начальные состояния (обычно одно) и конечные (целевые) состояния. SS-проблема состоит в поиске пути, ведущего из начального состояния в одно из конечных. Если, например, ИС предназначена для игры в шахматы, то обобщенными состояниями будут позиции, складывающиеся на шахматной доске. В качестве начального состояния может рассматриваться позиция, которая зафиксирована в данный момент игры, а в качестве целевых позиций — множество ничейных позиций. Отметим, что в случае шахмат прямое перечисление целевых позиций невозможно. Матовые и ничейные позиции описаны на языке, отличном от языка описания состояний, характеризующихся расположением фигур на полях доски. Именно это затрудняет поиск плана действий в шахматной игре.

При планировании в пространстве задач ситуация несколько иная. Пространство образуется в результате введения на множестве задач отношения типа: «часть — целое», «задача — подзадача», «общий случай — частный случай» и т. п. Другими словами, пространство задач отражает декомпозицию задач на

подзадачи (цели на подцели). PR-проблема состоит в поиске декомпозиции исходной задачи на подзадачи, приводящей к задачам, решение которых системе известно [Pohl, 1970; Levi et al., 1976]. Например, ИС известно, как вычисляются значения $\sin x$ и $\cos x$ для любого значения аргумента и как производится операция деления. Если ИС необходимо вычислить $\operatorname{tg} x$, то решением PR-проблемы будет представление этой задачи в виде декомпозиции $\operatorname{tg} x = \sin x / \cos x$ (кроме $x = \pi/2 + k\pi$).

Дадим классификацию методов, используемых при решении SS- и PR-проблем [Vanderburg, 1975; Levi et al., 1976].

1. Планирование по состояниям. Представление задач в пространстве состояний предполагает задание ряда описаний: состояний, множества операторов и их воздействий на переходы между состояниями, целевых состояний. Описания состояний могут представлять собой строки символов, векторы, двумерные массивы, деревья, списки и т. п. Операторы переводят одно состояние в другое. Иногда они представляются в виде продукций $A \Rightarrow B$, означающих, что состояние A преобразуется в состояние B .

Пространство состояний можно представить как граф, вершины которого помечены состояниями, а дуги — операторами. Если некоторая дуга направлена от вершины n_i к вершине n_j , то n_j называется дочерней, а n_i — родительской вершинами.

Последовательность вершин $n_{i_1}, n_{i_2}, \dots, n_{i_k}$, в которой каждая n_{i_j} — дочерняя вершина для вершины $n_{i_{j-1}}$, $j = \overline{2, k}$, называется путем длиной k от вершины n_{i_1} к вершине n_{i_k} .

Таким образом, проблема поиска решения задачи $\langle A, B \rangle$ при планировании по состояниям представляется как проблема поиска на графе пути из A в B . Обычно графы не задаются, а генерируются по мере надобности.

Различаются *слепые* и *направленные методы поиска пути*. Слепой метод имеет два вида: *поиск вглубь* и *поиск вширь*. При поиске вглубь каждая альтернатива исследуется до «конца», без учета остальных альтернатив. Метод плох для «высоких» деревьев, так как легко можно проскользнуть мимо нужной ветви и затратить много усилий на исследование «пустых» альтернатив. При поиске вширь на фиксированном уровне исследуются все альтернативы и только после этого осуществляется переход на следующий уровень. Метод может оказаться хуже метода поиска вглубь, если в графе все пути, ведущие к целевой вершине, расположены примерно на одной и той же глубине. Оба слепых метода требуют большой затраты времени и поэтому необходимы направленные методы поиска.

Метод ветвей и границ. Из формирующихся в процессе поиска неоконченных путей выбирается самый короткий и продлевается на один шаг. Полученные новые неоконченные пути (их столько, сколько ветвей в данной вершине) рассматриваются наряду со старыми, и вновь продлевается на один шаг кратчайший из них. Процесс повторяется до первого достижения целевой вершины, решение запоминается. Затем из оставшихся неоконченных путей исключаются более длинные, чем законченный путь, или равные ему, а оставшиеся продлеваются по такому же алгоритму до тех пор, пока их длина меньше законченного пути. В итоге либо все неоконченные пути исключаются, либо среди них формируется законченный путь, более короткий, чем ранее полученный. Последний путь начинает играть роль эталона и т. д.

Алгоритм кратчайших путей Мура. Исходная вершина x_0 помечается числом 0. Пусть в ходе работы алгоритма на текущем шаге получено множество дочерних вершин $\Gamma(x_i)$ вершины x_i . Тогда из него вычеркиваются все ранее полученные вершины, оставшиеся помечаются меткой, увеличенной на единицу по сравнению с меткой вершины x_i , и от них проводятся указатели к x_i . Далее на множестве помеченных вершин, еще не фигурирующих в качестве адресов указателей, выбирается вершина с наименьшей меткой и для нее строят-

ся дочерние вершины. Разметка вершин повторяется до тех пор, пока не будет получена целевая вершина.

Алгоритм Дейкстры определения путей с минимальной стоимостью является обобщением алгоритма Мура за счет введения дуг переменной длины.

Алгоритм Дорана и Мичи поиска с низкой стоимостью. Используется, когда стоимость поиска велика по сравнению со стоимостью оптимального решения. В этом случае вместо выбора вершин, наименее удаленных от начала, как в алгоритмах Мура и Дейкстры, выбирается вершина, для которой эвристическая оценка расстояния до цели наименьшая. При хорошей оценке можно быстро получить решение, но нет гарантии, что путь будем минимальным.

Алгоритм Харта, Нильсона и Рафаэля. В алгоритме объединены оба критерия: стоимость пути до вершины $g(x)$ и стоимость пути от вершины $h(x)$ — в аддитивной оценочной функции $f(x) = g(x) + h(x)$. При условии $h(x) \leq h_p(x)$, где $h_p(x)$ — действительное расстояние до цели, алгоритм гарантирует нахождение оптимального пути.

Метод планирования в пространстве состояний был использован в первом отечественном решателе ПРИЗ (пакет прикладных инженерных задач) [Тыугу, 1970]. В последней версии ПРИЗ представляет неориентированную сеть с вершинами двух типов: дескрипторы и спецификаторы. Спецификаторы — это функциональные выражения вида $f(x_1, x_2, \dots, x_n) = 0$, которые допускают явное разрешение хотя бы относительно одного из аргументов x_i . С таким спецификатором связано n ребер, каждое из которых соединено с одним из дескрипторов множества $\{x_1, x_2, \dots, x_n\}$. Два спецификатора соединены друг с другом некоторым дескриптором, если последний есть аргумент, общий для обоих спецификаторов. ПРИЗ решает задачи следующего типа. Дано множество исходных и целевых дескрипторов, необходимо найти такие допустимые разрешения для спецификаторов, которые образовали бы ориентированный путь, ведущий от исходных дескрипторов к целевым.

Алгоритмы поиска пути на графе различаются также направлением поиска. Существуют прямые, обратные и двунаправленные методы поиска. Прямой поиск идет от исходного состояния и, как правило, используется тогда, когда целевое состояние задано неявно. Обратный поиск идет от целевого состояния и используется тогда, когда исходное состояние задано неявно, а целевое явно. Двунаправленный поиск требует удовлетворительного решения двух проблем: смены направления поиска и оптимизации «точки встречи». Одним из критериев для решения первой проблемы является сравнение «ширины» поиска в обоих направлениях — выбирается то направление, которое сужает поиск. Вторая проблема вызвана тем, что прямой и обратный пути могут разойтись и чем уже поиск, тем это более вероятно.

2. Планирование по задачам. Этот метод приводит к хорошим результатам потому, что часто решение задач имеет иерархическую структуру. Однако не обязательно требовать, чтобы основная задача и все ее подзадачи решались одинаковыми методами. Редукция полезна для представления глобальных аспектов задачи, а при решении более специфичных задач предпочтителен метод планирования по состояниям. Метод планирования по состояниям можно рассматривать как частный случай метода планирования с помощью редукций, ибо каждое применение оператора в пространстве состояний означает сведение исходной задачи к двум более простым, из которых одна является элементарной. В общем случае редукция исходной задачи не сводится к формированию таких двух подзадач, из которых хотя бы одна была элементарной.

Поиск планирования в пространстве задач заключается в последовательном сведении исходной задачи к все более простым до тех пор, пока не будут получены только элементарные задачи. Частично упорядоченная совокупность таких задач составит решение исходной задачи. Расчленение задачи на альтернативные множества подзадач удобно представлять в виде И/ИЛИ-графа. В таком графе всякая вершина, кроме концевой, имеет либо конъюнктивно связанные дочерние вершины (И-вершина), либо дизъюнктивно связанные (ИЛИ-вершина). В частном случае, при отсутствии И-вершин, имеет место граф про-

странства состояний. Концевые вершины являются либо заключительными (им соответствуют элементарные задачи), либо тупиковыми. Начальная вершина (корень И/ИЛИ-графа) представляет исходную задачу. Цель поиска на И/ИЛИ-графе — показать, что начальная вершина разрешима. Разрешимыми являются заключительные вершины (И-вершины), у которых разрешимы все дочерние вершины, и ИЛИ-вершины, у которых разрешима хотя бы одна дочерняя вершина. Разрешающий граф состоит из разрешимых вершин и указывает способ разрешимости начальной вершины. Наличие тупиковых вершин приводит к неразрешимым вершинам. Неразрешимыми являются тупиковые вершины, И-вершины, у которых неразрешима хотя бы одна дочерняя вершина, и ИЛИ-вершины, у которых неразрешима каждая дочерняя вершина.

Алгоритм Ченга и Слейгла. Основан на преобразовании произвольного И/ИЛИ-графа в специальный ИЛИ-граф, каждая ИЛИ-ветвь которого имеет И-вершины только в конце. Преобразование использует представление произвольного И/ИЛИ-графа как произвольной формулы логики высказываний с дальнейшим преобразованием этой произвольной формулы в дизъюнктивную нормальную форму. Подобное преобразование позволяет далее использовать алгоритм Харта, Нильсона и Рафаэля.

Метод ключевых операторов. Пусть задана задача $\langle A, B \rangle$ и известно, что оператор f обязательно должен входить в решение этой задачи. Такой оператор называется ключевым. Пусть для применения f необходимо состояние C , а результат его применения есть $f(C)$. Тогда И-вершина $\langle A, B \rangle$ порождает три дочерние вершины: $\langle A, C \rangle$, $\langle C, f(C) \rangle$ и $\langle f(C), B \rangle$, из которых средняя является элементарной задачей. К задачам $\langle A, C \rangle$ и $\langle f(C), B \rangle$ также подбираются ключевые операторы, и указанная процедура редуцирования повторяется до тех пор, пока это возможно. В итоге исходная задача $\langle A, B \rangle$ разбивается на упорядоченную совокупность подзадач, каждая из которых решается методом планирования в пространстве состояний.

Возможны альтернативы по выбору ключевых операторов, так что в общем случае будет иметь место И/ИЛИ-граф. В большинстве задач удается выделить ключевой оператор, а только указать множество, его содержащее. В этом случае для задачи $\langle A, B \rangle$ вычисляется различие между A и B , которому ставится в соответствие оператор, устраняющий это различие. Последний и является ключевым.

Метод планирования общего решателя задач (ОРЗ). ОРЗ [Ньюэлл и др., 1964] явился первой наиболее известной моделью планировщика. Он использовался для решения задач интегрального исчисления, логического вывода, грамматического разбора и др. ОРЗ объединяет два основных принципа поиска: анализ целей и средств и рекурсивное решение задач. В каждом цикле поиска ОРЗ решает в жесткой последовательности три типа стандартных задач: преобразовать объект A в объект B , уменьшить различие D между A и B , применить оператор f к объекту A . Решение первой задачи определяет различие D второй — подходящий оператор f , третьей — требуемое условие применения C . Если C не отличается от A , то оператор f применяется, иначе C представляется как очередная цель и цикл повторяется, начиная с задачи «преобразовать A в C ». В целом стратегия ОРЗ осуществляет обратный поиск — от заданной цели B к требуемому средству ее достижения C , используя редукцию исходной задачи $\langle A, B \rangle$ к задачам $\langle A, C \rangle$ и $\langle C, B \rangle$.

Заметим, что в ОРЗ молчаливо предполагается независимость различий друг от друга, откуда следует гарантия, что уменьшение одних различий не приведет к увеличению других.

3. Планирование с помощью логического вывода. Такое планирование предполагает: описание состояний в виде правильно построенных формул (ППФ) некоторого логического исчисления, описание операторов в виде либо ППФ, либо правил перевода одних ППФ в другие. Представление операторов в виде ППФ позволяет создавать дедуктивные методы планирования, представление операторов в виде правил перевода — методы планирования с элементами дедуктивного вывода.

Дедуктивный метод планирования системы QAZ [Грин, 1973], ОРЗ не оправдал возлагавшихся на него надежд в основном из-за неудовлетворительного представления задач. Попытка исправить положение привела к созданию вопросно-ответной системы QAZ. Система рассчитана на произвольную предметную область и способна путем логического вывода ответить на вопрос: возможно ли достижение состояния B из A ? В качестве метода автоматического вывода используется принцип резолюций (см. § 2.4). Для направления логического вывода QAZ применяет различные стратегии, в основном синтаксического характера, учитывающие особенности формализма принципа резолюций. Эксплуатация QAZ показала, что вывод в такой системе получается медленным, детальным, что несвойственно рассуждениям человека.

Метод продукций системы STRIPS [Файкс и др., 1973]. В этом методе оператор представляет продукцию $P, A \Rightarrow B$, где P, A и B — множества ППФ исчисления предикатов первого порядка, P выражает условия применения ядра продукции $A \Rightarrow B$, где B содержит список добавляемых ППФ и список исключаемых ППФ, т. е. постусловия (см. § 1.4). Метод повторяет метод ОРЗ с тем отличием, что стандартные задачи определения различий и применения подходящих операторов решаются на основе принципа резолюций. Подходящий оператор выбирается так же, как в ОРЗ, на основе принципа «анализ средств и целей». Наличие комбинированного метода планирования позволило ограничить процесс логического вывода описанием состояния мира, а процесс порождения новых таких описаний оставить за эвристикой «от цели к средству ее достижения».

Метод продукций, использующий макрооператоры [Файкс и др., 1973]. Макрооператоры — это обобщенные решения задач, получаемые методом STRIPS. Применение макрооператоров позволяет сократить поиск решения, однако при этом возникает проблема упрощения применяемого макрооператора, суть которой заключается в выделении по заданному различию его требуемой части и исключении из последней ненужных операторов.

Метод иерархической системы продукций решателя ABSTRIPS [Sacredoti, 1974]. В этом методе разбиение пространства поиска на уровни иерархии осуществляется с помощью детализации продукций, используемых в методе STRIPS. Для этого каждой литере ППФ, входящей в множество P условий применения продукции, присваивается вес $j, j=0, k$, и на j -м уровне планирования, осуществляемом методом системы STRIPS, учитываются лишь литеры веса j . Таким образом, на k -м уровне продукций описываются наименее детально, на нулевом — наиболее детально как в методе системы STRIPS. Подобное разбиение позволяет для планирования на j -м уровне использовать решение $(j+1)$ -го уровня как скелет решения j -го уровня, что повышает эффективность поиска в целом.

Усовершенствованный метод планирования Ньюэлла и Саймона. В основу метода положена следующая идея дальнейшего совершенствования метода ОРЗ: задача решается сначала в упрощенной (за счет ранжирования различий) области планирования, а затем делается попытка уточнить решение применительно к более детальной, исходной проблемной области.

Комплексная схема нечеткого планирования

Недостатком большинства известных в настоящее время систем планирования [Chang et al., 1971; Sargiovanni et al., 1974; Guida et al., 1976] является их жесткая привязка к схеме планирования. Любая из них всегда ищет решение либо SS-проблемы, либо PR-проблемы. Связано это с фиксацией формы представления информации для планирования. Для классических моделей SS- и PR-проблем эти формы различны. Ясно, однако, что человек в своей деятельности успешно комбинирует шаги планирования из решения SS- и PR-проблем. Вторым недостатком является детерминированность систем планирования. В реальных ИС детерминированность планирования, как правило, не имеет места.

Обобщение нечетких SS- и PR-проблем заключается в допущении нечетких состояний и нечетких операторов перехода из состояния в состояние. Разбиение задачи на подзадачи имеет весовые коэффициенты на дугах со значениями из $[0, 1]$, которые интерпретируются как достоверности решения соответствующих подпроблем. Достоверность решения PR-проблемы определяется как минимум достоверностей решения ее подпроблем.

При переходе к обобщенной стратегии из решений нечетких подпроблем PR-проблемы, рассматриваемых как нечеткие SS-проблемы, можно получить решение нечеткой PR-проблемы, рассматриваемой также как нечеткая SS-проблема [Аверкин и др., 1978].

Общие принципы, которые могут быть положены в основу планировщика, использующего поиск в пространстве состояний и разложение задач на подзадачи, описаны в [Guida et al 1975, 1976]. Их обобщение на нечеткий случай содержится в [Аверкин и др., 1978; Аверкин, 1979, 1980].

Схемой SS-проблемы называется пара $M = (S, G)$, где S — множество состояний, G — множество отображений $g: S \rightarrow S$, называемых операторами. Путем из состояния $s_0 \in S$ в состояние $s_k \in S$ называется конечная последовательность $p = ((s_0, g_0), (s_1, g_1), \dots, (s_{k-1}, g_{k-1}), s_k)$, такая, что $g_i \circ s_i = s_{i+1}$ для $i = 0, k-1$. SS-проблема — это четверка $P = (S, G, i, f)$, где (S, G) — схема SS-проблемы, $i, f \in S$ — соответственно начальное и заключительное состояние. Путь x , ведущий из i в f , есть решение P , а множество всех подобных путей составляет множество решений.

Схемой PR-проблемы называется пара $N = (\Sigma, \Gamma)$, где Σ — множество проблем, Γ — множество отображений $\gamma: \Sigma \rightarrow \Sigma^+$, называемых операторами. Если $P \in \Sigma$, $\pi \in \Sigma^+$, то $\gamma_P(\pi)$ — отображение, представляющее проблему P в виде цепочки подпроблем $\pi = P_1 \dots P_n$. Для схемы $N = (\Sigma, \Gamma)$ накрывающий путь q из проблемы σ_0 в конечное множество проблем $\Sigma_k = \{\sigma_1, \dots, \sigma_n\} \in \Sigma'$ является конечной последовательностью, где $q = ((x_0, y_0), (x_1, y_1), \dots, (x_{k-1}, y_{k-1}), x_k)$, $x_i \in \Sigma^+$ для $i = 0, k$, $y_i \in \Gamma^+$ для $i = 0, k-1$; так что $x_0 = \sigma_0$, $x_k \in \Sigma_k^+$. PR-проблема представляет собой четверку $Z = (\Sigma, \Gamma, P_0, \Phi)$, где (Σ, Γ) — схема PR-проблемы, $P_0 \in \Sigma$ — начальная проблема, $\Phi \subset \Sigma$ — множество конечных проблем. Решение Z представляет собой накрывающий путь (Σ, Γ) из P_0 в $\Phi' \subset \Phi$, множество решений x_Z есть множество всех решений Z .

Приведенные определения рассматривают только синтаксис описания проблемы независимо от смысла используемой формальной схемы. В схеме SS-проблемы синтаксис и семантика могут совпадать в более сложных случаях, например в схеме PR-проблемы они должны различаться. Под семантикой здесь понимается способ, с помощью которого решение искомой проблемы получается из решений подзадач, к которым она свелась. Приведем формальное определение семантики сведения задачи к подзадачам.

Импликата проблемы P есть пара (π, ψ) , где $\pi = P_1 P_2 \dots P_k$ — цепочка проблем, ψ — отображение из $X_{P_1} \times X_{P_2} \times \dots \times X_{P_k}$ в X_P (X_{P_i} обозначает множество решений P_i). *Имплицативная схема* есть тройка $\Lambda = (P, \pi, \psi)$, такая, что P — проблема, (π, ψ) — импликата P . Множество T имплицативных схем называется *имплицативной сетью*. Множество проблем имплицативной сети

$$\Omega_T = \{x | (\exists \Lambda) ((\Lambda = (P, \pi, \psi)) \wedge ((x = P) \vee (x = \text{символ } \pi)))\}.$$

Объединим синтаксис и семантику подхода, основанного на разбиении проблемы на подпроблемы. PR-проблема $Z = (\Sigma, \Gamma, P_0, \Phi)$ представляет имплицативную сеть T тогда и только тогда, когда $\Sigma = \Omega_T$ и для каждого $\Lambda = (P, \pi, \psi) \in T$ существует в точности одно $\gamma \in \Gamma$, такое, что $P\gamma = \pi$, и для каждого $\gamma \in \Gamma$, и для каждого P из области определения γ существует в точности одно $\Lambda = (P, \pi, \psi) \in T$, такое, что $\pi = P\gamma$. Проблема P решена тогда и только тогда, когда X_P — пустое множество.

Если PR-проблема представляет имплицативную сеть, то проблема P_0 разрешима. Для существования решения достаточно, чтобы имплицативные схемы в имплицативной сети T существовали только для всех пар (x_i, y_i) ,

$i=0, k-1$, входящих в накрывающий путь схемы PR-проблемы. В этом случае синтаксис и семантика PR-проблемы не совпадают. В данном случае PR-проблема частично представляет импликативную сеть T .

Подчеркнем, что как синтаксис, так и семантика подхода разбиения проблемы на подпроблемы не предполагают предварительного определения проблемы. Поэтому можно в качестве проблемы выбрать SS-проблему.

Рассмотрим понятие, объединяющее подходы разбиения проблемы на подпроблемы и поиска в пространстве состояний. I-проблемой (объединенной проблемой) называется четверка $R=(B, \Gamma, P_0, T)$, где B — множество SS-проблем; (B, Γ) — схема PR-проблемы: $P_0 \in B$ — основная проблема; T — импликативная сеть, такая, что $\Omega_T \cap B \neq \emptyset$. Кроме того, R есть решение SS-проблемы P_0 . Учитывая связь между существованием импликативной сети и разрешимостью проблемы, легко показать, что если для заданной I-проблемы найден накрывающий путь x из P_0 на множество $\Phi' \subseteq B$, проблемы Φ' разрешимы как SS-проблемы, если x частично представляет импликативную сеть T , то и проблема R разрешима как SS-проблема.

Рассмотрим головоломку «Ханойская башня» [Нильсон, 1973]. Имеются три стержня 1, 2 и 3 и три диска различных размеров A, B, C с отверстием в центре, которые могут одеваться на стержни. В исходной позиции диски находятся на стержне 1; самый большой диск C — внизу, самый маленький диск A — наверху. Требуется перенести все диски на стержень 3, перемещая за один раз только один диск. Брать можно только самый верхний диск на стержне, причем его нельзя класть на диск, меньший по размерам. Используем для записи состояний и операторов классическую формализацию. Выражение ijk обозначает конфигурацию, при которой диск C находится на стержне i , диск B — на стержне j и диск A — на стержне k . Выражение xij обозначает действие, при котором диск x перемещается со стержня i на стержень j . С помощью этого формализма можно просто записать все состояния и переходы головоломки в виде треугольного графа, где вершины соответствуют расположению дисков на стержнях, а дуги — возможным переключиваниям дисков (рис. 7.1). На этой головоломке легко проиллюстрировать все основные понятия обобщенной стратегии проблем.

Представим головоломку в виде модели I-проблемы. Рассмотрим I-проблему $R=(B, \Gamma, P_0T)$, где $B=\{P_0, P_1, \dots, P_9\}$; $\Gamma=\{\psi\}$; $T=\{\Lambda_1, \Lambda_2, \Lambda_3\}$. SS-проблемы P_0, P_1, \dots, P_9 определяются следующим образом. На рис. 7.1 показана схема SS-проблемы $M=(S, G)$, где

$$\begin{aligned} P_0 &= (S, G, 111, 333), P_1 = (S, G, 111, 122), P_2 = (S, G, 122, 322), \\ P_3 &= (S, G, 322, 333), P_4 = (S, G, 111, 113), P_5 = (S, G, 113, 123), \\ P_6 &= (S, G, 123, 122), P_7 = (S, G, 322, 321), P_8 = (S, G, 321, 331), \\ P_9 &= (S, G, 331, 333). \end{aligned}$$

Схема PR-проблемы $N=(B, \Gamma)$ приведена на рис. 7.2; импликативная сеть T — на рис. 7.3, причем $\Lambda_1=(P_0, P_1P_2P_3, \psi)$, $\Lambda_2=(P_1, P_4P_5P_6, \psi)$, $\Lambda_3=(P_3, P_7P_8P_9, \psi)$, где $\psi(x_1, x_2, x_3)=x_1x_2x_3$.

Проблемы P_2 и P_4 — P_9 решаются перекладыванием одного диска и являются элементарными. Проблемы P_1 и P_3 решаются с помощью манипуляций только с дисками B и A и являются более простыми, чем P_0 . Проблемы P_1 и P_3 решаются, а проблема P_0 сводится к P_1, P_2 и P_3 аналогичной манипуляцией с дисками, синтаксис которой выражен оператором ψ , а семантика — отображением ψ .

Представление этой головоломки в виде PR-проблемы (рис. 7.2) является более компактным и наглядным, чем представление в виде SS-проблемы (рис. 7.1), а представление в виде I-проблемы (рис. 7.3) сочетает достоинство обоих и показывает взаимосвязь подпроблем и тех действий, которые нужно выполнить, чтобы решить головоломку.

Приведенные определения обобщаются на нечеткий случай, когда состояния системы, для которой строится модель решения проблемы, не являются точно заданным, а результаты действий системы неоднозначны. Например,



Нечеткой PR-проблемой называется PR-проблема, у которой элементам $\gamma \in \Gamma$ приписывается степень принадлежности $\mu_\gamma(\pi) \in [0, 1]$. α -решением нечет-

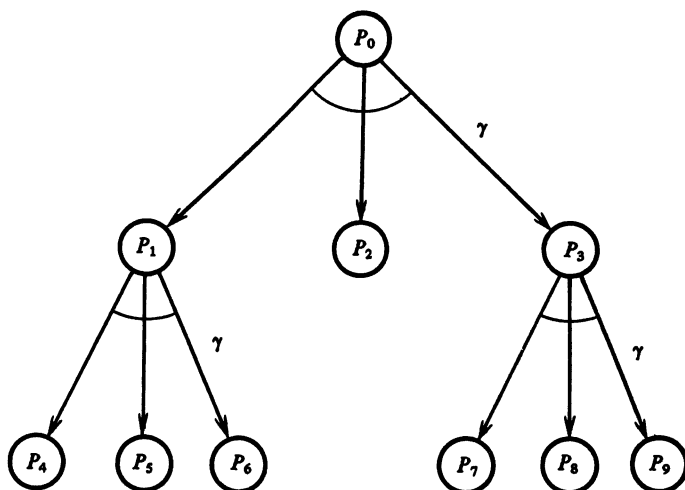


Рис. 7.2

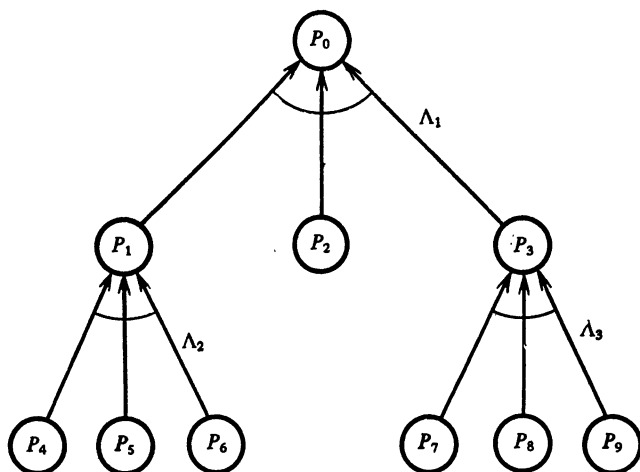


Рис. 7.3

кой PR-проблемы называется решение PR-проблемы, для которой $\min_i \gamma_{ij} \geq \alpha$, $\gamma_{ij} \in \gamma_i$.

Накрывающий путь в этом случае называется накрывающим α -путем. Степень принадлежности $\mu_\gamma(\pi)$ интерпретируется как степень возможности разбиения проблемы на подпроблемы.

Нечеткой импликативной схемой называется импликативная схема, все отображения ψ которой имеют степень принадлежности $\mu_\psi(\pi) \in [0, 1]$. Степень принадлежности интерпретируется как возможность получения решения проблемы из решений соответствующих подпроблем. Нечеткой импликативной сетью называется импликативная сеть с нечеткими импликативными схемами.

Нечеткая PR-проблема представляет нечеткую импликативную сеть, если кроме обычных условий для импликативной сети выполняется неравенство $\mu_{\psi}(\pi) \geq \mu_{\gamma}(\pi) \geq \alpha$, т. е. возможность разбиения на подпроблемы не меньше возможности получения решения для каждой пары соответствующих ψ и γ .

Аналогично определяется нечеткая I-проблема. Но построить α -решение нечеткой PR-проблемы по α -решениям ее нечетких подпроблем можно лишь в частных случаях и при наложении дополнительных условий. Пусть задана нечеткая PR-проблема, где Σ — нечеткие SS-проблемы, и существует простейшее α -решение Z , т. е. такое $\pi \in \Sigma^+$, что $\mu_{\gamma_P} \geq \alpha$, и если $\pi = P_1 \dots P_n$, то все проблемы P_i α -разрешимы, где $P_i = (S_i, G_i, J_i, F_i)$. α -решение проблемы P_0 существует при $\mu_{\psi}(\pi) \geq \mu_{\gamma}(\pi) \geq \alpha$ для импликативной сети. Запишем более конструктивное условие для этого. Пусть для всех i оператор g_i есть α -путь решения проблемы P_i и $F_i = J_{i+1}$. Тогда, если выполняется

$$\max_{S_n} \mu, [F_n \cap (F_{n-1} \cap (\dots (F_1 \cap (J \circ g_1) \circ g_2) \dots) \circ g_n)] \geq \alpha$$

и $\bar{g} = g_1 \dots g_n$, то \bar{g} — α -решение SS-проблемы P_0 .

Проблема вывода и планирования в неопределенных условиях ставилась в [Поспелов Д., 1981, 1986; Чесноков, 1984, 1985; Ефимов, 1985, 1986, 1987].

Особенности планирования целенаправленных действий

Дальнейшее развитие теории планирования было связано с построением «человеческих» моделей целенаправленной деятельности. Если представить человеческие рассуждения, связанные с планированием, как некоторую целенаправленную деятельность по решению интеллектуальных задач, то в модели планирования прежде всего необходимо учесть основные особенности человеческих рассуждений.

Пусть задан некоторый предметный мир, в котором действие ИС состоит в достижении целевых ситуаций s_k из некоторых исходных ситуаций s_n с помощью планов действий $V = b_{i_1}, \dots, b_{i_n}$, где b_{ij} — исполнительный модуль из данного набора B_0 . Задать ситуацию s в таком мире — это значит указать свойства $c_i \in C$ предметов $a_k \in A_0$ и отношения между ними $r_p \in R_0$, которые имели, имеют или будут иметь место в момент t . Модель предметного мира для такого действия ИС можно представить в виде $M_0 = \langle A_0, B_0, C_0, R_0 \rangle$, а задачу планирования действий в мире M_0 следующим образом: заданы исходная s_n и цель s_k ситуации, необходимо построить из исполнительных модулей $b_i \in B_0$ план действий V , который, будучи примененным к s_n , позволяет достичь s_k .

Перед человеком при решении этой задачи обычно возникают две проблемы. Во-первых, он, как правило, плохо представляет конкретную s_k , удаленную в будущее. Поэтому его устраивает достижение не конкретной, а любой ситуации из класса \bar{s}_k , удовлетворяющей определенным требованиям. Во-вторых, если даже он представляет s_k , то и тогда поиск плана действий затруднен из-за большой размерности пространства поиска. Итак, ИС необходимы более общие по отношению к M_0 модели миров.

Среди задач выделим класс элементарных. Остальные будем считать сложными, а их решения представлять в виде частично упорядоченной совокупности элементарных задач. Если решения различных сложных задач в каком-то смысле схожи, то они обобщаются. Так возникают обобщенные описания задач определенного типа.

Из решений элементарных задач ИС выстраивает решения сложных исходных задач. Однако найти таким образом решение сразу ей обычно не удается. Поэтому для перехода от исходных задач к элементарным используются типовые задачи. Вначале для заданной исходной задачи определяется смысловая структура ее исходных данных, т. е. ставится стратегическая задача и формулируется ее гипотеза решения. Далее каждая типовая задача гипотезы декомпо-

зируется, что приводит к постановке и решению тактических задач и, следовательно, к решению исходной задачи.

Наличие в базе знаний типовых и элементарных задач свидетельствует об иерархической структуре не только базы знаний, но и процедур поиска, при этом типовые задачи могут быть условно отнесены к стратегическому, а элементарные — к тактическому уровню поиска. Смысловые структуры требуемых результатов и исходных данных типовых задач обычно выявляются в результате осмысливания требуемых результатов или исходных данных тактических задач и недоступны непосредственному чувственному восприятию. Таким образом, на стратегическом уровне каждая тактическая ситуация (исходные данные или требуемый результат) оценивается по наличию в ней знакомых смысловых структур. Например, в шахматной игре смысловые структуры выражаются понятием «развитая позиция», «открытая позиция» и т. д. На тактическом уровне решаются спроецированные со стратегического уровня путем декомпозиции типовых задач тактические задачи, например образование проходной пешки в шахматной игре.

Итак, основу рассуждений ИС по планированию действий составляют структурированные знания и направленный эвристический поиск.

Пусть C_1 — множество, получаемое огрублением свойств $c \in C_0$, B_1 — множество элементарных задач, получаемое переименованием исполнительных модулей $b_i \in B_0$, $A_1 \equiv A$ и $R_1 \equiv R_0$. Тогда модель мира тактических задач можно представить в виде $M_0 = \langle A_1, B_1, C_1, R_1 \rangle$, постановку тактической задачи \hat{p} — в виде пары $\langle \hat{s}_n, \hat{s}_k \rangle$, а ее решение — в виде совокупности $\hat{V} = \hat{b}_1, \dots, \hat{b}_{i_n}$.

Очевидно, благодаря указанному огрублению в мире M_1 становятся неразличимыми отдельные состояния предметов и исполнительных модулей. Однако подобное упрощение, вызываемое «грубостью» органов чувств ИС, еще не позволяет значительно снизить размерность пространства поиска решений \hat{V} , поэтому необходимо дальнейшее обобщение M_1 уже на понятийном уровне.

В мире M_1 тактические ситуации \hat{s} описываются, как и ситуации s в мире M_0 , через свойства объектов и отношения между ними. Однако такие описания не носят целостного характера, так как не содержат в явном виде смысловых структур. Выявление таких обобщенных структур, а также формирование соответствующих им тактических ситуаций \hat{s} осуществляются на основе понятий, которыми располагает ИС, и означают осмысливание ею сложившихся или целевых ситуаций \hat{s} с точки зрения этих понятий, используемых в этом случае как программы-тесты. Таким образом, в мире M_2 указанные ситуации \hat{s} представляются в виде укрупненных предметов $\hat{a}_k \in A_2$, свойства которых $c_i \in C_2$ и отношения $\hat{r}_2 \in R_2$ между которыми, как и сами укрупненные предметы, определяются понятиями-тестами. Аналогично обстоят дела и с решениями \hat{V} в мире M_1 . Целостное описание \hat{V} означает описание этих решений именно как типовых задач. Таким образом, в мире M_2 имеют место типовые задачи $\hat{b}_i \in B_2$.

С учетом сказанного модель мира стратегических задач можно представить в виде $M_2 = \langle A_2, B_2, C_2, R_2 \rangle$, постановку стратегической задачи \hat{p} — в виде пары $\langle \hat{s}_n, \hat{s}_k \rangle$, а гипотезу решения — в виде совокупности $\hat{V} = \hat{b}_{i_1}, \dots, \hat{b}_{i_n}$.

Итак, модель действия ИС по планированию целенаправленной деятельности может быть представлена на теоретико-множественном языке в виде $\mathcal{M} = \langle M, Q \rangle$, где модель мира M — триада $\langle M_0, M_1, M_2 \rangle$, Q — комплексная стратегия поиска планов решения задач.

Комплексная стратегия поиска планов

Рассуждения человека наиболее полно исследовались при решениях научно-технических задач творческого характера [Пушкин, 1965; Тихомиров, 1969; Пойа, 1970, 1975; Шапиро, 1973; Дружинин и др., 1977]. Большое внимание при этом уделялось структуризации и обобщению знаний на основе модельной концепции мышления [Поспелов Д. и др., 1967]. Согласно этой концепции рас-

суждение профессионала, решающего творческую задачу, осуществляется по следующей схеме: исходная ситуация — структурированная и обобщенная модель исходной ситуации — порождение усеченного лабиринта и общих решений — выбор общего решения — формирование конкретного решения.

По модельной концепции мышления вначале путем структуризации и обобщения исходных данных исключаются несущественные детали, а затем строится лабиринт небольшой размерности; в то время как по поведенческой концепции мышления, получившей широкое распространение в теории переборных алгоритмов (в частности, при создании шахматных программ), вначале порождается полный лабиринт, учитывающий и несущественные детали, а затем он усекается с помощью эвристики.

Модельная концепция рассуждения, опирающаяся на представления предметного мира в виде модели $\mathcal{M} = \langle M_0, M_1, M_2 \rangle$, была впервые использована в комплексной стратегии Q системы СФИНКС [Ефимов, 1982]. Суть этой стратегии заключается в следующем. Пусть поставлена исходная задача $\bar{p} = \langle \bar{s}_n, \bar{s}_k \rangle$ и согласно \bar{p} — стратегическая задача $\hat{p} = \langle \hat{s}_n, \hat{s}_k \rangle$, где \hat{s}_n — результат осмысливания исходных данных \bar{s}_n , а \hat{s}_k — условия исходной задачи, выражающие смысл требуемого, пока неизвестного результата \bar{s}_k . Если \hat{s}_k было бы известно, то пара $\langle \bar{s}_n, \bar{s}_k \rangle$ задавала бы конкретизацию \bar{p} задачи \hat{p} , решение которой должно быть найдено в виде упорядоченной совокупности \bar{V} элементарных задач $\bar{b}_i \in B_1$. Но для этого, предварительно необходимо получить общее решение \bar{V} . Это и есть гипотеза решения. Она ищется в пространстве типовых задач $\bar{b}_i \in B_2$ и выражает в общих чертах, что и в какой последовательности надо сделать, чтобы достичь желаемого результата.

При построении гипотезы решения (обычно это делается в обратном направлении — от цели к средству ее достижения) используются планирующие эвристики «от цели к средству» и «различие между исходными данными и средствами достижения цели». Если требуемое средство не содержится в исходных данных, то они, в свою очередь, начинают играть роль очередных целей и все повторяется до тех пор, пока либо очередные средства не окажутся в наличии, либо будет установлено, что это вообще невозможно. В последнем случае делается попытка изменить или построить новую гипотезу решения.

Гипотеза решения ограничивает область дальнейшего поиска рамками сформированных типовых задач \bar{b}_i , декомпозиция позволяет поставить тактические задачи \bar{p}_i , т. е. сформировать компоненты \hat{s}_n^i и \hat{s}_k^i . На стратегическом уровне эти компоненты представляют, как уже было сказано, смысловые структуры, задающие требования, которым должны удовлетворять тактические ситуации. Например, смысловая структура «белая проходная» утверждает необходимость наличия в шахматной позиции хотя бы одной проходной пешки белых, при этом число пешек обеих сторон и их расположение в рамках смысловой структуры может быть произвольным. Таким образом, смысловая структура допускает определенную свободу как по количеству элементов, так и по их свойствам и связям. Это приводит к тому, что переход в результате декомпозиции типовых задач \bar{b}_i от смысловой структуры к конкретной является неоднозначным, т. е. является неоднозначной постановка по \bar{b}_i тактических задач \bar{p}_i .

Если тактическая задача поставлена, то механизмы построения ее решения аналогичны механизмам построения гипотез и решения. Тактическая задача может и не иметь решения или ее решение может не укладываться в рамки запланированного — результат выходит за соответствующую смысловую структуру. В этом случае срабатывает механизм обратной связи и либо осуществляется частичная корректировка построенной гипотезы, либо формируется новая гипотеза.

7.2. Планирование при синтезе программ

М.-И. Канович, Г. Е. Минц

Основные определения

Задача планирования может быть поставлена математически строго, если учитывать лишь те ее аспекты, которые могут быть описаны на уровне строгих формализмов. В такой трактовке задачу планирования часто называют *задачей автоматического синтеза программ*. В основе методов ее решения лежат приемы, традиционные для математической логики [Лавров, 1983; Минц и др., 1983а, б; Диковский и др., 1985]. Поиск плана действий в этом случае трактуется как поиск программы, решающей ту задачу, формулировка которой поступает на вход системы планирования, что приводит к схеме формулировка задачи → теорема → доказательство → программа. Таким образом, в системе планирования надо уметь реализовать три этапа: 1) трансляция описания исходной задачи в формулу логического исчисления, которая для этого исчисления выступает как формулировка теоремы, подлежащей доказательству; 2) поиск вывода в используемом исчислении; 3) извлечение из найденного вывода траектории, которая преобразуется в нужную программу.

Если второй этап представляется более или менее традиционным (см. гл.2), то первый и третий этапы требуют разработки специальных средств и методов. Первый этап связан со спецификой проблемной области, в которой решается задача, третий — с характером используемого логического исчисления и применяемых в нем методов доказательства теорем. Последнее утверждение требует некоторых пояснений.

Проблема *извлечения программы из доказательства* и вопрос о том, какие доказательства позволяют это, а какие нет, привлекали внимание математиков задолго до появления ЭВМ. Дедуктивные средства, обеспечивающие такое извлечение, стали называться *конструктивными*, или *эффективными*, а логики, использующие только эти средства, — *конструктивными логиками*. Общепринятое «конструктивное ядро» составляют так называемые *интуиционистские формальные системы* [Клини, 1957]. Основная черта таких формальных систем — согласованность с конструктивным пониманием суждений. Это означает, что по формальному доказательству предложения $\exists x A(x)$ можно построить натуральное число q , для которого верно $A(q)$, а по доказательству предложения $\forall x \exists y A(x, y)$, описывающего задачу построения результата y по данным x , вычисляемую функцию f , для которой верно $(\forall x) A(x, f(x))$, т. е. верна программа, которая по данному x строит искомое число $y = f(x)$. Построение программы по данному доказательству в интуиционистской системе может быть поручено ЭВМ. Размер программы, извлекаемой из доказательства, не намного больше размера самого доказательства, и при специальном представлении доказательства они могут непосредственно выполняться на ЭВМ.

Введем ряд понятий, связанных со схемой автоматического синтеза программ. Опишем модель предметной области (см. гл. 1), в которой формулируются и интерпретируются задачи. В задачах планирования, сводящихся к автоматическому синтезу программ, такие модели называют *вычислительными*. Вычислительная модель задается шестеркой:

$$W = \langle A, D, B, D_b, F, H \rangle.$$

Здесь A — множество атрибутов; D — множество соответствующих им доменов (см. § 1.1). Атрибуту $A_i \in A$ соответствует домен $D_i \in D$; B — множество функциональных зависимостей, определенных над атрибутами. Элементы B имеют вид $X \rightarrow Y$, где X и Y — некоторые подмножества A . Множеству B соответствует домен D_b , элементами которого выступают всюду определенные программы с входными параметрами X и выходными параметрами Y . Эти программы при подстановке в X значений из D_i вычисляют значения D_j тех атрибутов, кото-

рые соответствуют Y . Совокупность F есть множество описаний типов всех используемых в B функциональных зависимостей. Наконец, H задает совокупность отношений (предикатов) над множеством атрибутов A .

Напомним (см. § 1.1), что произвольное подмножество T декартова произведения $D_{i_1} \times D_{i_2} \times \dots \times D_{i_m}$ связано с понятием базы данных. В мате-

матической логике такое T , удовлетворяющее всем соотношениям вычислительной модели, называют моделью этой области. В теории баз данных [Ульман, 1983] система H выступает как *ограничения целостности* для реляционной базы данных.

В теории вычислительных моделей обычно предполагается, что некоторые величины могут принимать «пустые» значения ω и «пустой» величине соответствует атрибут Ω , для которого D_Ω есть $\{\omega\}$. Для остальных атрибутов A_i обычно предполагается, что D_i являются бесконечными. Введенное понятие вычислительной модели расширяется в работах по автоматическому синтезу программ [Лавров, 1983; Тыгу и др., 1985; Канович, 1986].

Решить задачу нахождения величин Y по заданным величинам X с помощью вычислительной модели означает, что на основе соотношений, зафиксированных в вычислительной модели, надо найти план действий по получению величин из списка Y по величинам из списка X . Эту процедуру можно разбить по крайней мере на два уровня: *концептуальный* и *процедурный* [Лавров, 1986]. На концептуальном <схемном, структурном> уровне используется только та часть вычислительной модели, которая обозначена в ее определении как H . На процедурном уровне в плане, полученном на предшествующем уровне (*абстрактной программы* в терминологии [Лавров, 1986]), происходит замена схем из H на их конкретные программные реализации, содержащиеся в D_B или конструируемые из них.

В более строгом смысле результат, получаемый на концептуальном уровне, может трактоваться как стандартная схема программы [Ершов, 1973; Лакхэм и др., 1975]. При этом используются следующие существенные обобщения: вводятся операторы присваивания, перехода и остановки; допускаются интерпретированные операторы типа $A = \text{root } G(X)$ решения «уравнения» $G(X) = 0$, операторы решения системы «уравнений», пересылки констант, выбора компонентов векторов и т. п.; в качестве параметров могут выступать параметры-процедуры.

Пусть, например, в предметной области фигурируют такие величины, как ускорение a и путь s , величины булева типа ДР (движение равномерное со скоростью v_1) и ДН (движение неравномерное), функциональная величина — скорость v , и выполняются известные соотношения из механики движения:

$$s = \int_{t_0}^{t_1} v(t) dt,$$

if ДР=true then val $v = v_1$,
if ДН=true then val $v = v_0 + at = 0$,
if $a = 0$ then ДР=true else ДН=true.

Здесь t_0 , t_1 , v_0 , v_1 заранее заданы. Для вычисления значения s по значению a можно предложить следующую «абстрактную» программу:

```
proc TASK (in a, out s);
  proc v (in t, out val v):
    if Q(a) then goto l1 else goto l2;
    l1: val v = v1; goto l;
    l2: val v = root G (val v, v0, a, t);
  l: end v;
  s = H(v);
end TASK;
```

Здесь Q — предикатный символ; G и H — функциональные символы. При их замене соответствующими программными модулями осуществляется переход от концептуального уровня к процедурному. Программа вычисления s по a на этом уровне будет иметь вид

```
proc TASK (in a, out s);  
  proc v (in t, out val v);  
    if a=0 then goto l1 else goto l2;  
    l1: val v=v1; goto l;  
    l2: val v=v0+at;  
  l: end v;  
  s:=Integral (v);  
end TASK;
```

Здесь Integral — стандартная процедура интегрирования, программа которой хранится в памяти системы.

Еще раз обратим внимание на тот факт, что извлечение программ из доказательств связано с особенностями конструктивных систем [Клини, 1957; Марков, 1962], так как многие специалисты, занимающиеся построением ИС, не придают ему должного значения. Если использовать в ИС исчисления классического типа, то можно столкнуться с тем, что программа из доказательства извлечена быть не может. Например, в классическом исчислении высказываний из двух формул $(B \rightarrow A)$ и $((A \rightarrow B) \rightarrow B)$ выводится формула $(C \rightarrow A)$. Может иметься процедура, реализующая «вычисление» $((A \rightarrow B) \rightarrow B)$ по входному параметру $(A \rightarrow B)$, являющемуся результатом работы процедуры такого же типа с входным параметром A . Выходными параметрами этих процедур являются соответственно B и $A \rightarrow B$. Однако из двух таких процедур нельзя сконструировать процедуру, реализующую зависимость $(C \rightarrow A)$.

При создании систем автоматического синтеза программ необходимо:

1. Уточнить и формализовать понятие решения вычислительной задачи: в каких случаях она должна считаться разрешимой и в каком смысле «абстрактная» программа решает задачу.

2. Исследовать проблему корректности исчислений, чтобы иметь возможность ответить на вопрос: если выводима некоторая формула, представляющая собой решение исходной задачи, то разрешима ли эта вычислительная задача с точки зрения семантики, заданной в модели проблемной области?

3. Исследовать полноту исчисления хотя бы относительно специального класса формул, чтобы получать ответы на вопросы типа: если исходная вычислительная задача имеет решение, то выводима ли соответствующая формула в исчислении?

4. Исследовать проблему конструктивности исчисления, т. е. уметь получать ответ на вопрос: содержится ли в выводе формулы вся необходимая информация для того, чтобы извлечь из этого вывода программу решения вычислительной задачи?

При практической реализации *планировщиков*, позволяющих автоматически синтезировать нужные программы, возникают следующие вопросы, связанные с оценкой сложности и эффективности их работы: насколько прост и нагляден перевод исходных формулировок задач проблемной области на язык формул выбранного исчисления? какова оценка эффективности поиска вывода в используемом исчислении? насколько проста и семантически интерпретируема для пользователя процедура извлечения программы из вывода? каково качество синтезируемых системой программ?

Наибольшие трудности вызывает проблема поиска вывода [Чень и др., 1983]. Здесь в полной мере проявляются все неприятные эффекты проблемы перебора в NP-полных и PSPACE-полных задачах [Гэри и др., 1982].

Пропозициональные модели предметных областей

Эффективность планировщиков прямо зависит от языка описания вычислительной модели и «мощности» того исчисления, которое используется для синтеза программ. Однако для многих практически интересных случаев оказывается

достаточным пропозициональный уровень (см. гл. 1). Известны следующие виды зависимостей такого уровня.

Функциональные зависимости [Ульман, 1983] или *безусловные предложения вычислимости* [Минц и др., 1983а, б] вида $(X \rightarrow Y)$, реализуемые процедурами с входными параметрами X и выходными параметрами Y .

Неявные зависимости [Канович, 1985] вида $(X \rightarrow \text{Circ}(Y))$, которые являются «схемным изображением» уравнения $g(X, Y) = 0$, однозначно разрешимого относительно каждой переменной из списка Y . В [Тыгу, 1984] такие переменные называются *слабосвязанными*.

Операторные зависимости [Диковский и др., 1985] или *условные предложения вычислимости* [Минц и др., 1983а, б] вида

$$((X_1 \rightarrow Y_1), (X_2 \rightarrow Y_2), \dots, (X_m \rightarrow Y_m)) \rightarrow (X \rightarrow Y),$$

реализуемые процедурами с входными параметрами типов $(X_1 \rightarrow Y_1), (X_2 \rightarrow Y_2), \dots, (X_m \rightarrow Y_m)$ и выходным параметром типа $(X \rightarrow Y)$.

Вариантные зависимости [Канович, 1987] вида $(X \rightarrow Y_1 \vee Y_2)$, которые описывают ветвления:

$$\text{if } q(X) \text{ then } Y_1 = p_1(X) \text{ else } Y_2 = p_2(X).$$

Если вернуться к примеру о вычислении пути по ускорению, то соотношение интегрирования можно отобразить функциональной зависимостью $(v, t_0, t_1 \rightarrow s)$ или операторной зависимостью $(t \rightarrow \text{val } v, t_0, t_1 \rightarrow s)$. Для дальнейших соотношений можно использовать неявные зависимости (например, $\text{ДН} \rightarrow \text{Circ}(\text{val } v, v_0, a, t)$), а необходимое разветвление задать вариантной зависимостью $(a \rightarrow \text{ДР} \vee \text{ДН})$.

В системе ПРИЗ и ее модификациях описание проблемной области и формулировки задач в ней переводятся на внутренний язык, эквивалентный по выразительной силе подязыку интуиционистского исчисления высказываний. В нем допускаются пропозициональные формулы лишь следующих двух видов: $X_1 \& X_2 \& \dots \& X_m \rightarrow Y$ (сокращенно $X \rightarrow Y$) и $(U^1 \rightarrow V^1) \& (U^2 \rightarrow V^2) \& (U^m \rightarrow V^m) (X \rightarrow Y)$ (сокращенно $(U \rightarrow V) \rightarrow (X \rightarrow Y)$). С логической точки зрения эти формулы — импликации, но с вычислительной точки зрения — функциональные зависимости. Формулы второго вида выражают функциональные зависимости высших порядков, аргументами которых являются функции.

При описании задач используются составные описатели, т. е. последовательности описаний и отношений (в частности, с помощью уравнений). Например, описание объекта «точка» имеет вид

```
point (X: numeric;
      Y: numeric)
```

При трансляции описаний класс point породит три отношения:

point \rightarrow point.X; point \rightarrow point.Y; point.X, point.Y \rightarrow point.

Рассмотрим в качестве примера описание задачи о минимаксе: найти строку матрицы, максимальный элемент которой является наименьшим среди максимальных элементов всех строк. Определим понятие максимума:

```
max: (vir
      arg: numeric;
      fun: numeric;
      maxval: numeric;
      (arg  $\rightarrow$  fun)  $\rightarrow$  maxval (D));
```

Здесь использована программа D для представления отношения, описываемого аксиомой $((\text{arg} \rightarrow \text{fun}) \rightarrow \text{maxval})$. Это отношение связывает максимальное значение функции с самой функцией, которая представляется подформулой $(\text{arg} \rightarrow \text{fun})$. Ключевое слово vir — указание транслятору описаний. Аналогично, используя ту же программу, можно описать понятие минимума:

```

min: (vir
      arg: numeric;
      fun: numeric;
      negfun: numeric;
      minval: numeric;
      maxval: numeric;
      negfun = -fun;
      minval = -maxval;
      (arg → negfun) → maxval (D));

```

Теперь легко перейти к описанию понятия минимакса.

```

minimax: (
  value: numeric;
  vir
  M: matrix;
  R1: max arg=M.j, fun=M.elem;
  R2: min arg=M.i, fun=R1.maxval, maxval=value);

```

Эта задача порождает, в частности, три аксиомы, в которых пропозициональные переменные M, I, J, E , MAXINROW обозначают соответственно вычислимость матрицы, номера ее строки, столбца, элемента и максимального элемента в строке:

$$(M \& I \& J \rightarrow E), \quad ((J \rightarrow E) \rightarrow \underset{\text{max}}{\text{MAXINROW}}), \quad (I \rightarrow \underset{\text{min}}{\text{MAXINROW}}) \rightarrow \underset{\text{min}}{\text{MINIMAX}}.$$

Для установления порядка применения программ к аргументам существенна лишь пропозициональная структура формул, используемых в системе ПРИЗ и ее модификациях. Но для того чтобы явно выписать результирующую программу, надо явно указать, как эти формулы реализуются. Во введенной выше сокращенной записи двух видов формул вычислительной модели, используемой в ПРИЗе, надо сделать небольшие дополнения, после которых они примут вид

$$\frac{X \rightarrow Y}{\underline{f}}, \quad \frac{(U \rightarrow V) \rightarrow (X \rightarrow V)}{\underline{g}}.$$

Функции \underline{f} и \underline{F} — реализации формул первого и второго типов, \underline{g} — последовательность \underline{g}_i , относящихся к отдельным членам в развернутой записи формулы второго вида. Таким образом, первое расширение исходной записи показывает, что значения переменных Y можно вычислить из значений переменных X с помощью функции \underline{f} , а вторая запись — что процедура \underline{F} порождает по функциям \underline{g} , вычисляющим V по U , новую функцию $\underline{F}(\underline{g})$, вычисляющую Y по X .

Синтез программ

На каждом шаге синтеза осуществляется переход вида $\Gamma \vdash P \rightarrow Q$, где Γ — множество ранее выведенных утверждений, P является выведенным. На первом шаге в качестве Γ выступает множество аксиом исчисления. При выводе используются правила структурного синтеза (SSR), задаваемые следующими тремя схемами:

$$1. \frac{\vdash X \rightarrow V; \Gamma \vdash X}{\Gamma \vdash V}, \quad (\rightarrow -);$$

Здесь $\Gamma \vdash X$ — список секвенций для всех X из \underline{X} .

$$2. \frac{\vdash (U \rightarrow V) \rightarrow (X \rightarrow V); \Gamma \vdash X; \underline{Z}, U \vdash V}{\Gamma, \underline{Z} \vdash V}, \quad (\rightarrow - -).$$

Здесь $\underline{Z. U \vdash V}$ — список секвенций для всех $(\underline{U} \rightarrow \underline{V})$ из $(\underline{U} \rightarrow \underline{V})$.

$$3. \frac{\Gamma, \underline{X} \vdash V}{\Gamma \vdash \underline{X} \rightarrow V}, (\rightarrow +).$$

Если все аксиомы это формулы первого типа, то для вывода используется стратегия движения от начальных состояний к целевым (*прямая волна*). При таком методе постепенно наращивается количество выведенных формул, пока не будет получен вывод нужного выражения. Если $\Gamma \vdash P \rightarrow Q$ — доказываемое выражение, то в конце процесса множество вычисленных формул состоит из P и всех переменных, входящих в Γ в качестве отдельных членов. На каждом шаге вывода в множество ранее полученных формул добавляются заключения Y формул, используемых в ПРИЗе (формул первого типа), если их посылки уже включены в это множество. Использованная таким образом формула далее в выводе не участвует. Цель $\Gamma \vdash P \rightarrow Q$ достигается, если на некотором шаге в множество выведенных формул включается Q . Если среди аксиом имеются формулы второго типа, то процедура вывода усложняется. Она становится эквивалентной выводу в интуиционистском исчислении высказываний [Волож и др., 1982].

Приведем в качестве иллюстрации вывод для задачи о минимаксе. С точностью до технических деталей он имеет вид

$$\frac{M \& I \& I \rightarrow F, (J \rightarrow F) \rightarrow \text{MAXINROW}}{M \& I \rightarrow \text{MAXINROW}, (I \rightarrow \text{MAXINROW}) \rightarrow \text{MINIMAX}} \\ M \rightarrow \text{MINIMAX}$$

При извлечении программы из доказательства используются стандартные интерпретации интуиционистских систем. Расширения формул применяются для оснащения аксиом из Γ замкнутыми лямбда-термами (реализациями). Эти аксиомы — исходные формулы (листья) дерева вывода. При движении по дереву вывода вниз в соответствии с применяемыми правилами происходит оснащение реализациями встречающихся по пути формул [Клини, 1957]. Лямбда-терм, сопоставленный с последней формулой $P \rightarrow Q$, будет схемой искомой программы. Для задачи о минимаксе такая программа будет иметь вид $m \min (\lambda_i \max (\lambda_j \text{get} (m, i, j)))$.

Описанные выше правила позволяют строить нерекурсивные программы. Методы, при которых получаются рекурсивные программы, используются значительно реже [Миц и др., 1983а, б]. Шаги вывода в этих методах (за исключением дедуктивного синтеза) такие же, как при синтезе нерекурсивных программ. К правилам вывода добавляется еще одно:

$$\frac{(A \rightarrow X), A \vdash X}{\vdash (A \rightarrow X)}.$$

Оценки сложности задачи планирования

Приведем ряд результатов, касающихся сложности решения задач планирования.

1. Анализ вычислительных задач оказывается PSPACE-полной проблемой даже при условии, что «пустых» величин Ω нет [Канович, 1987].

2. Для вычислительных моделей без функциональных величин, т. е. с пустыми D_v и F , проблема анализа вычислительных задач оказывается: а) PSPACE-полной для H , содержащих функциональные и операторные зависимости без Ω [Миц и др., 1983а, б]; б) NP-полной для H , содержащих только функциональные и варианты зависимости без «пустой» величины Ω [Канович, 1987]; в) полиномиальной (по времени работы планировщика) для H , содержащих только функциональные и неявные зависимости; г) можно построить планировщики с ли-

нейным временем работы для H только с функциональными зависимостями без Ω [Диковский, 1985] и для H с функциональными и неявными зависимостями [Канович, 1985].

В системе ПГИЗ D_v и F пусты, а в H имеются только функциональные и операторные зависимости без «пустой» величины Ω . В системе автоматического синтеза программ СПОРА [Лавров, 1983] используется исчисление предикатов, для которого разработана специальная стратегия вывода [Нейман, 1985]. В [Канович, 1985, 1986, 1987] предлагается специальное «исчисление вычислительных задач», для которого в H могут использоваться функциональные, неявные, операторные и варианты зависимости и в рамках которого попарно эквивалентными оказываются следующие три задачи: вычислительная задача разрешима; существует такое дерево анализа задачи, что все его висячие вершины (листья) являются аксиомами «исчисления вычислительных задач»; в любом дереве анализа для вычислительной задачи все висячие вершины оказываются аксиомами.

Дерево анализа для вычислительной задачи — это дерево из формул используемого исчисления, в корне которого находится формула, представляющая исходную задачу, каждая нетерминальная вершина дерева может быть получена по одному из правил вывода «исчисления вычислительных задач» из формул, расположенных в дереве непосредственно ниже, а терминальные (висячие) вершины не являясь заключением ни одного из правил вывода «исчисления вычислительных задач».

Планировщик в таком исчислении работает следующим образом: для заданной задачи конструируется какое-нибудь дерево анализа. Если все его висячие вершины являются аксиомами, то по этому дереву «собирается» программа. В противном случае формируется обоснование того, что задача неразрешима. Отсутствие детерминизма при логическом выводе в исчислениях стандартного типа приводит к экспоненциальному перебору возможных ветвей. Обнаружение при таком переборе тупика оказывается бесполезным (или почти бесполезным) для поиска на других ветвях дерева вывода. В «исчислении вычислительных задач» для полного решения задачи планировщику достаточно располагать каким-нибудь деревом анализа. При разумной стратегии построения деревьев анализа это позволяет получать сравнительно быстрые процедуры планирования [Канович, 1986, 1987].

В общем случае планировщику приходится решать PSPACE-полную задачу. Это означает, что все известные сейчас планировщики на почти всех вычислительных задачах работают экспоненциальное время. В реальности ситуация не столь плоха. Практически интересные задачи, как правило, допускают эффективное планирование. И хотя доля подобных задач с ростом параметров, определяющих их, стремится к нулю, именно они представляют интерес с точки зрения эффективности автоматического синтеза программ. Показателем практически интересных задач может служить степень взаимодействия подзадач в процессе решения исходной задачи [Канович, 1985]. Понятия «подзадача» и «условия подзадачи» возникают уже на стадии наивной интерпретации следующих зависимостей.

Функциональной зависимости $(F \rightarrow Y_1)$, где F — функциональная величина типа $(X \rightarrow Y)$, предполагающей предварительный синтез процедуры вычисления F , входными параметрами которой являются величины из списка X .

Операторных зависимостей, предполагающих синтез m процедур с входами «условиями подзадачи» X_1, X_2, \dots, X_m .

Вариантных зависимостей, предполагающих, что создаются две ветви вычислений: для первой известны значения всех величин из списка Y_1 , для второй — из списка Y_2 .

Рассмотрим простой (но типичный) пример. Пусть имеют место функциональная зависимость $D_1: (A, B, C \rightarrow E)$ и операторные зависимости $D_2: ((C \rightarrow E) \rightarrow E_1)$, $D_3: ((B \rightarrow E_1) \rightarrow E_2)$, $D_4: ((A \rightarrow E_2) \rightarrow E_3)$. Необходимо найти E_3 . Для этого следует решить подзадачу $(A \rightarrow E_2)$. Если воспользоваться зависимостью D_3 , то придем к поиску решения новой подзадачи $(B \rightarrow E_1)$. Если использовать зависимость D_2 , то придем к необходимости решения подзадачи $(C \rightarrow E)$, которая согласно D_1 разрешима только тогда, когда A и B известны. Степень взаимодействия подза-

дач в этом примере равна трем. Процесс является типичной схемой планирования в пространстве задач (см. § 7.1).

Для планировщиков, работающих в «исчислении вычислительных задач», синтез решения вычислительной задачи происходит за время $hq^r/l/r$, где h — константа, l — длина записи задачи, определяемая как суммарное число вхождений имен используемых величин в F и H , q — общее число «различных» аргументов величин из «условия подзадачи» в операторных зависимостях из H и «условий вариантов» в вариантных зависимостях из H , r — минимальная степень взаимодействия подзадач в исходной задаче [Канович, 1985, 1986, 1987]. Эта оценка носит квазиоптимальный характер. Планировщик работает долго только на тех задачах, которые не являются естественными, так как требуют предельного переплетения между собой всех подзадач, на которые разбивается исходная задача. Эксперименты показывают, что для задач, встречающихся на практике, время работы планировщика является полиномиальным. Память, необходимая для работы планировщика в «исчислении вычислительных задач», оценивается величиной bl^2 , где b — константа. Если в H нет вариантных зависимостей, то требуется линейный объем памяти dl , где d — константа [Канович, 1986, 1987].

В известных сейчас планировщиках получаемые программы далеки от оптимальных. Наблюдается тенденция ухудшения качества программ при уменьшении времени работы планировщика. При этом синтезировать оптимальные последовательные программы труднее, чем оптимальные параллельные программы. Это обстоятельство в связи с развитием ЭВМ новой архитектуры, ориентированной на параллельные процессы, может оказаться выгодным. Например, задача поиска минимальных по времени последовательного исполнения программ для решения вычислительных задач, у которых D_b пусто, а H состоит только из функциональных зависимостей Ω , оказывается NP-полной в сильном смысле [Канович, 1987]. С другой стороны, в «исчислении вычислительных задач» при условии, что D_b пусто, а H состоит из функциональных и неявных зависимостей, можно за линейное время kl (k — константа) синтезировать программу, параллельное выполнение которой требует минимального для данной задачи времени [Канович, 1987].

В системе ПРИЗ и ее модификациях вычислительная задача считается разрешимой, если соответствующая ей формула выводима в позитивном фрагменте интуиционистского исчисления высказываний [Минц и др., 1983а, б], т. е. в том и только том случае, когда достигается успех планировщиком системы ПРИЗ. Для «исчисления вычислительных задач» возможны две семантики для разрешимости [Канович, 1987].

Предельно неконструктивная; вычислительная задача считается разрешимой, если в любой базе данных (любой интерпретации), удовлетворяющей всем ограничениям вычислительной модели, имеет место некоторая функция типа $(X \rightarrow Y)$;

предельно конструктивная: обобщенная стандартная схема программы считается решением вычислительной задачи, если в любой интерпретации, удовлетворяющей всем ограничениям вычислительной модели, имеет место функция типа $(X \rightarrow Y)$, вычисляемая программой Π , получающейся из обобщенной стандартной схемы программы соответствующей конкретизацией предикатных и функциональных символов.

«Исчисление вычислительных задач» корректно и полно относительно обеих семантик. При первой семантике планировщик всегда выдает решение исходной задачи в виде правильной программы, при второй — класс всех пропозициональных формул, описывающих разрешимые вычислительные задачи, совпадает с классом всех формул, выводимых в интуиционистском исчислении высказываний. Это показывает, что «исчисление вычислительных задач» можно рассматривать как один из вариантов формализации предложенной в [Колмогоров, 1932] интерпретации интуиционистской логики как логики «задач».

Во многих ИС используются планировщики, возможности которых существенно шире, чем у планировщика системы ПРИЗ, или того, который применяется в «исчислении вычислительных задач». Исчисления, с которыми имеют дело многие планировщики систем автоматизированного проектирования, планирования и

управления, шире интуиционистских исчислений высказываний. При этом используются эвристические соображения, не имеющие аналогов в тех соотношениях, в которых описывались вычислительные модели. Примером может служить планировщик системы МАВР, предназначенной для ИС автоматизированного проектирования [Эрлих, 1984]. При его работе возникают ситуации, не разрешимые в теории, которая описывалась выше. Такие случаи заставляют искать иные пути построения системы для поиска планов действий. Одним из них может быть специальная логика решений, первоначальный набросок которой имеется в [Ефимов, 1982]. Ряд соображений, относящихся к проблеме планирования и способам построения планировщиков, можно найти также в [Гладун, 1977, 1987].

7.3. Поступки и поведение *

Д. А. Поспелов

Классификация типов поведения

В психологии под поведением понимают взаимодействие живых существ

с окружающей их средой и другими особями, опосредствованное двигательной и психической активностью. Специалисты, использующие в своей профессиональной деятельности понятие «поведение», определяют его под углом зрения целей своей науки, что приводит к обеднению этого понятия [Гааза-Рапопорт и др., 1987]. Будем называть некоторую деятельность *поведением*, если выполнены три условия: наличие цели; наличие наблюдателя; наличие оценки.

Не предполагается, что наличие цепи эквивалентно ее осознанию. Субъект может не осознавать цель в явной форме, а совершать некоторые действия, «потому что ему так хочется». Но сама деятельность должна осознаваться субъектом как таковая. Другими словами, из поведенческих актов исключаются безусловные рефлексы или деятельность внутренних органов, неконтролируемая специально созданием. Таким образом, под поведением понимается целенаправленная деятельность, а не деятельность вообще.

Второе условие вычленяет в этой деятельности еще более узкую область. В поведенческом акте должен принять участие осознаваемый наблюдатель. Его роль может выполнять сам субъект (самонаблюдение) или «мыслимый наблюдатель», который мог бы быть. Другими словами, второе условие требует возможности взгляда на свою деятельность со стороны, извне.

Третье условие требует, чтобы наблюдатель формировал оценку деятельности субъекта и высказывал к ней некоторое отношение. При этом важно, что сам субъект обладает свойством рефлексивности, т. е. может прогнозировать эту оценку, ставя себя на место наблюдателя. Третье условие исключает из поведения такие виды деятельности, как принятие пищи (если оно не носит ритуального или эпатирующего характера), сон, трудовую деятельность, не подверженную оценке, и т. п.

Подобно понятию поведения понятие цели в психологической литературе не имеет точного определения. Поэтому будем говорить лишь о неосознанных и осознанных целях. Будем предполагать наличие у субъекта некоей врожденной совокупности целей, носящей как физиологический, так и психологический характер. Примерами целей первого типа могут служить «утоление голода» или «продолжение рода»; примерами целей второго типа — «максимизация актов внимания» или «стремление избежать неприятные психологические ситуации».

Особенностью глобальных психологических целей является их индивидуализированность. Глобальные физиологические цели для всех людей одинаковы, но они также могут порождать поведенческие акты. Для достижения глобальных целей необходимо ставить и достигать промежуточные цели, которые могут быть прямыми подцелями глобальных целей или косвенными целями, внешне не связанными с глобальной целью. Косвенное целеполагание является характерной чертой поведения людей. Прямые цели и деревья целей более характерны для

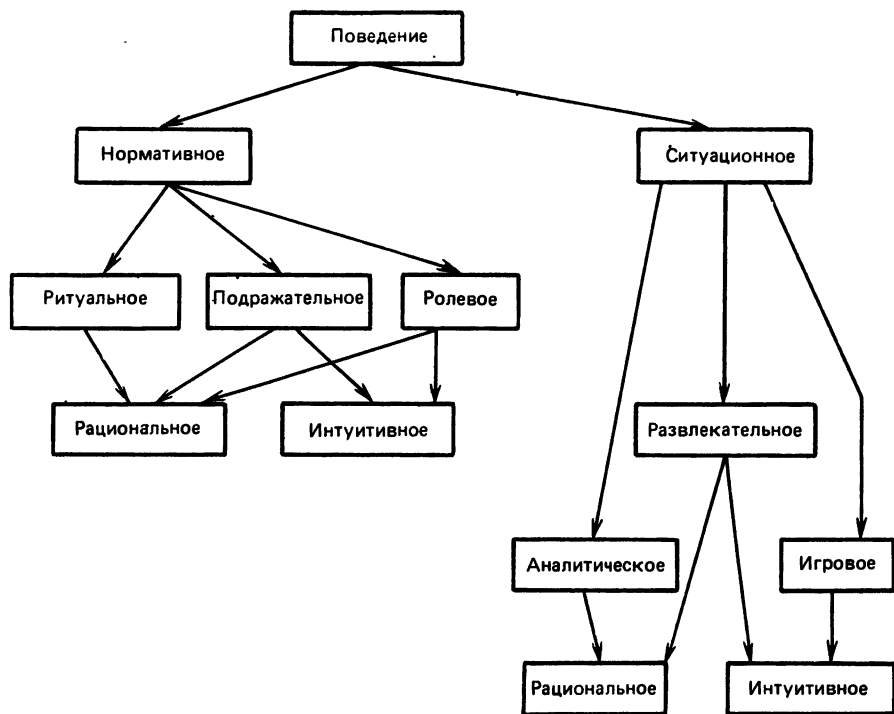


Рис. 7.4

решения задач или осуществления деятельности, которая не носит оценочного характера, т. е. не является поведением в нашем смысле.

Поведенческие акты органически связаны с априорными знаниями об оценках, которые субъект может ожидать после совершения действий. Субъект имитирует наблюдателя и его оценки. Умение рассуждать за наблюдателя предполагает знание о том, как он проводит свои рассуждения.

Наблюдателем для субъекта может быть некоторый персонифицированный человек или группа персонифицированных лиц (товарищ, жена, друзья, семья, дети и т. п.). Но наблюдателем может быть и некоторый наблюдатель «вообще», в котором персонифицируются неписанные морально-этические нормы, принятые в данном социуме, или свод юридических законов и норм, применяемых в данном государстве. Эти знания образуют фон, на котором планируется и строится поведение, снабженное самооценкой субъекта. Эта прогнозная самооценка может не всегда совпадать с апостериорной оценкой наблюдателя. Согласованность оценок характеризует полноту знаний, имеющихся у субъекта, реализующего поведенческий акт. Знания такого типа отличаются от знаний об окружающем мире. Они представляют собой субъективный слой знаний о формах поведения в этом мире.

Классификация типов поведения приведена на рис. 7.4.

Нормативное поведение определяется набором предписаний, характерных для того общества, к которому принадлежит индивид, совершающий акт поведения; *ситуационное* поведение — той реальностью, в которой оно возникает. Разница этих двух типов поведения иллюстрируется следующим примером. В зале кон-

серватории тишина; звучит завораживающая музыка. Нормативное поведение предполагает, что все зрители ведут себя тихо. Предположим, что какому-то зрителю стало плохо. Надо немедленно принять меры по спасению его жизни. Тогда вместо нормативного поведения возникает ситуационное. В сложившейся ситуации можно пренебречь требованиями нормативного поведения, нарушить тишину и помочь заболевшему покинуть зал.

Нормативные предписания не всегда зафиксированы в документах. Так, к нормативному поведению относится и такое, которое основано на некотором статистическом среднем, на положении «надо делать так, потому что все так делают» или на положении «мама меня учила, что так делать нельзя», или на каком-либо ином положении, обычно именуемом житейской мудростью. Нормативное поведение может также отражать стремление к определенному функциональному оптимуму, который достигается коллективом людей в результате согласованного поведения его членов. Этим трем случаям на рис. 7.4 соответствуют три вида поведения: *ритуальное*, *подражательное* и *ролевое*. Последние два вида нормативного поведения могут быть как *рациональными* (с осознанными целями), так и *интуитивными* (с неосознанными целями).

Наличие ситуационных правил поведения показывает, что нельзя обойтись только нормативными правилами. Отсутствие полного набора нормативных правил поведения и универсальности любого из сформулированных нормативных правил приводит к тому, что люди часто нарушают последние и вместо нормативного поведения реализуют ситуационное поведение.

Ситуационное поведение подобно нормативному распадается на три типа: *аналитическое*, *развлекательное* и *игровое*. Аналитическое поведение характеризуется осознанностью целей и плана их достижения. Оно полностью находится под контролем сознания. Развлекательное поведение возникает тогда, когда цели поведения, которые в данном случае могут быть осознанными и неосознанными, и действия, направленные на достижения этих целей, не причиняют никакого вреда другим индивидам, принося субъекту выигрыш. Игровое поведение связано с тем, что в процессе достижения своих целей субъект причиняет определенный вред окружающим.

В настоящее время в ИС воспроизводится лишь нормативное поведение. Остальные виды поведения пока не имеют таких описаний, которые были бы пригодны для формализации.

Модель поступка

Введем понятие *поступка*, которое будем считать базовым при описании поведения. Формально поступок может быть задан в виде двух графов G_1 и G_2 , которые соответственно называются мотивом (замыслом) и реализацией. Мотив представляет цель, которую ставит перед собой субъект. Пара $\langle G_1, G_2 \rangle$ определяет поступок. Знание этой пары позволяет наблюдателю дать словесную оценку поступка. Графы G_1 и G_2 имеют одинаковый набор помеченных вершин: I — субъект (я); H — он (некоторое персонифицированное лицо или коллектив, или объект неживой природы, или животное), на которого направлено действие или речевой акт; T — они (персонифицированное окружение субъекта, от которого возможна оценка данного действия); N — природа (окружающая среда); U — человечество «вообще» (свод неписаных морально-этических норм, принятых в данном социуме); S — свод зафиксированных в данном обществе (государстве) юридических законов.

В графах G_1 и G_2 допускаются связи двух типов: положительные и отрицательные, которые могут характеризоваться различными весами q_i .

На рис. 7.5 показан граф G_1 , в котором субъект хочет за счет положительного действия с весом q_1 получить ответное действие с положительным весом q_2 , превышающим q_1 , т. е. стремится получить некоторую личную выгоду от своего действия. При этом он предполагает на основе имеющихся у него знаний, что персонифицированное окружение положительно воспримет его деятельность (на рис. 7.5 этому соответствует положительная дуга от T к I с весом q_3). И хотя с точки

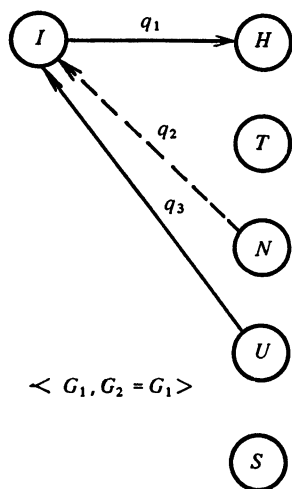


Рис. 7.5

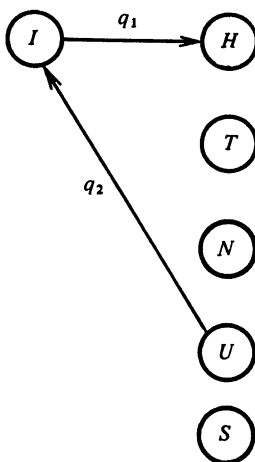


Рис. 7.6

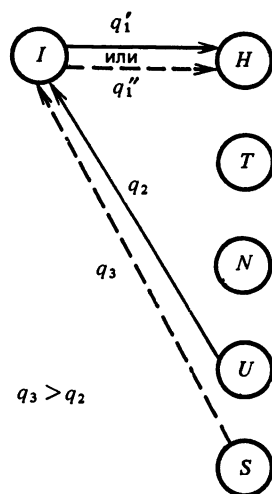


Рис. 7.7

зрения морально-этических норм субъект предполагает некоторую отрицательную оценку q_4 (испытывает «внутренние угрызения совести»), но общий выигрыш его остается положительным, так как $q_2 + q_3$ оказывается большим, чем $q_1 + q_4$.

Если граф G_2 совпадает с графом G_1 в том смысле, что $q'_2 + q'_3$ оказывается все же большим, чем $q_1 + q'_4$, то в качестве словесной оценки поступка, совершенного субъектом, может быть использован термин «корыстный поступок». Сам поступок при этом оказывается правильным с точки зрения субъекта, сформировавшего цель. Если же вместо графа G_1 реализация приводит к графу, отличному от него (например, к графу G_2 , показанному на рис. 7.6), то поступок объекта может быть охарактеризован словесно как «недалновидный» или «глупый». И этот поступок с точки зрения субъекта, поставившего цель, является неправильным.

На рис. 7.7 приведена структура поступка, для которого G_1 и G_2 совпадают. Словесная оценка этого поступка зависит от соотношения суммы положительных весов с отрицательным весом, полученным от N : поступок может быть назван смелым, мужественным или подвигом.

Наборы графов $\langle G_1, G_2 \rangle$ могут быть классифицированы. Можно выделить некоторые фреймные конструкции, которые являются базисными для словесных оценок. Например, фрейм, показанный на рис. 7.5, является базисным для понятия «корыстный поступок». Удаление в нем любых связей или невыполнение условия для q_1 приводит к изменению словесной оценки. На рис. 7.8 показан фрейм $\langle G_1, G_2 \rangle$, в котором G_1 и G_2 совпадают и который характеризует не корыстный поступок, а поступок иного типа. Его можно было бы назвать «благородным поступком». Фреймы на рис. 7.9, 7.10 характеризуются самыми общими словесными оценками (например, «добрый поступок» или «альтруистический поступок»).

В любом естественном языке имеется конечное число словесных оценок, используемых для характеристики поведения. Им соответствуют определенные фреймы $\langle G_1, G_2 \rangle$. Установление между ними соответствия позволяет в ИС «понимать» словесные оценки поведения. Фреймы-эталоны поступков хранятся в памяти робота и дают ему возможность при наличии достаточной информации классифицировать поступки [Шустер, 1983].

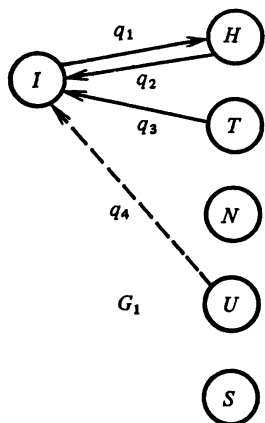


Рис. 7.8

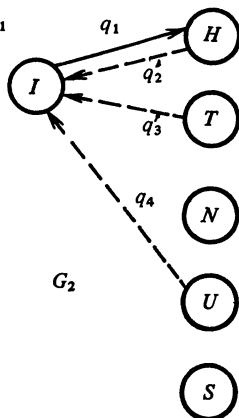


Рис. 7.9

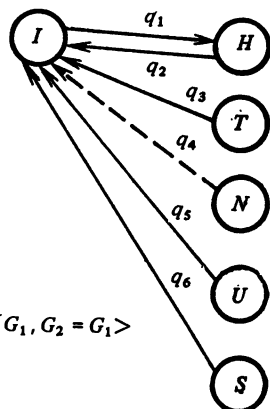


Рис. 7.10

Данная модель элементарного поведенческого акта довольно бедна. Многие аспекты поступков в ней не отражены. Поэтому возможны расширения этой модели. Например, в [Емельянов, 1981], где использована модель такого типа для оценки деятельности диспетчеров-управленцев в соответствии с их должностной инструкцией, введены в качестве действий не только действия, совершаемые де-факто, но и действия, заключающиеся в отказе от тех или иных действий. При этом прогноз реализации (граф G_2) формируется в виде двух альтернативных результатов: при совершении действия и при отказе от него. В более общей модели поступка возможно рассмотрение не одного воздействия I на H , а набора действий или отказов от них со взвешиванием ожидаемых результатов. Это приводит к постановкам задач теории принятия решений в их классическом смысле.

Траектории поведения

Под *траекторией поведения* субъекта будем понимать линейную упорядоченную последовательность вида $l_1, l_2, \dots, l_k, \dots, l_n$, в которой каждый элемент l_k есть либо поступок p_k , либо действие, не являющееся поступком. При этом в последовательности $\{l_k\}$ есть по крайней мере один поступок. Кроме того, последовательность $\{l_k\}$ будет считаться поведением лишь тогда, когда для всех p_k , входящих в нее, имеется одинаковая глобальная (физиологическая или психологическая) цель. В результате реализации некоторого поведения эта глобальная цель может достигаться или не достигаться. Кроме глобальной цели, в поведении может присутствовать набор рациональных целей или целей деятельности. Эти цели могут для субъекта иметь внешнее происхождение или возникать как под-цели при декомпозиции глобальной цели.

Для описания поведения недостаточно последовательности $\{l_k\}$. Характеристика того или иного поведения включает еще один параметр: расход ресурсов на реализацию $\{l_k\}$. Конкретный вид ресурсов, учитываемых при этом (время, энергетические затраты и т. п.), роли не играет. Поэтому будем понимать их как обобщенные ресурсы, измеряемые в некоторых условных единицах.

Введем следующие обозначения: W_k — оценка степени приближения к рациональной цели, R_k — суммарный ресурс, израсходованный на предшествующих шагах l_i , Q_k — суммарная оценка по весам реализации во всех поступках, совершенных до этого. Индекс k указывает на номер шага поведения. Тогда, например, поведение, описываемое условиями $\max W_k$, $R_k \geq 0$ и $Q_k > Q^* > 0$, характери-

зуют субъекта, который стремится достичь рациональной цели наилучшим образом, используя весь имеющийся у него ресурс, но при условии, что суммарная оценка его поведенческих актов (поступков) не будет ниже его «уровня порядочности» Q^* . Другим примером может служить поведение, характеризуемое как $max Q_k, R_k \geq R^* > 0$ и $W_k < W^*$, при котором субъект стремится максимизировать хорошее мнение о себе в условиях сохранения определенного уровня ресурсов и допустимого приближения к достижению рациональной цели.

Если для приведенных примеров использовать словесные оценки поведения, то субъект первого типа может быть охарактеризован как целеустремленный, упорный и порядочный (с уровнем порядочности Q^*). Второй субъект занимается деятельностью, не слишком утруждая себя, но чрезвычайно заботясь о мнении окружающих его людей. Легко представить самые разнообразные виды поведения в зависимости от соотношения введенных нами параметров. Подобная модель позволяет давать точную трактовку (с точностью до пороговых значений, которые, конечно, остаются размытыми и могут быть определены лишь экспертным путем) таких словесных понятий, как целесообразное поведение, сибаритское поведение и т. п. Подобные интегральные характеристики поведения могут формироваться у ИО в процессе ее взаимодействия с субъектами, существующими в окружающей среде.

Формирование поведения

На формирование последовательности $\{I_k\}$ оказывают активное влияние знания, которые субъект хранит в своей памяти. На основании априорной информации и накапливаемых сведений, формирующихся в результате реализации поступков, в системе знаний хранится информация о глобальном поведении H и T . Кроме того, в базе знаний хранится информация об U и S , а также о связи между действиями субъекта и откликами неживой природы N . Это и позволяет субъекту сформировать граф G_1 , характеризующий его очередной шаг. Если реализация поступка π_i оказалась успешной (совпала с прогнозом), то знания, использованные при формировании структуры поступка, получают увеличенную оценку доверия. При рассогласовании прогноза и реализации происходит перестройка знаний. При этом для H и T может меняться порог их оценки по шкалам интегральных оценок поведения (типа добрый — злой или ленивый — трудолюбивый), что, возможно, на следующем шаге заставит субъекта сменить прогноз их оценок. Для U и S вводятся необходимые уточнения, учитывающие всю ситуацию, в которой был совершен поступок π_i . При рассогласовании оценок по N в базу знаний вносится дополнительная информация о функционировании окружающей среды. Такое обучение системы реализуется с помощью хорошо известных алгоритмов обобщения и пополнения ситуаций (см. § 2.3 и гл. 6).

В настоящее время разработана специальная логика поступков [Котик и др., 1983]. В ней используется система фреймов-классификаторов, с помощью которых анализируются фреймы поступков. Кроме того, в этой логике имеются специальные средства для выбора тех или иных поступков в ответ на поступки партнеров.

СПИСОК ЛИТЕРАТУРЫ

- Абдрахманов и др., Абдрахманов А.А., Лозовский В.С., Лозовский С.В. ИТК - БИС - инструментально-технологический комплекс для систем семиотического моделирования // Искусственный интеллект. Кн.3. Программные и аппаратные средства. - М.: Радио и связь, 1990.
- Аверкин и др., Аверкин А.Н., Гвида Дж., Поспелов Д.А., Со-
1978 малывико М. Обобщенные стратегии в решателях проблем // Изв.АН СССР. Техн. кибернетика. - 1978. - N 5. - С.71-83.
- Аверкин, Аверкин А.Н. Нечеткая модель обобщенного решате-
1979 ля проблем // Семиотика и информатика. - 1979. - Вып.12. - С. 103-108.
- Аверкин, Аверкин А.Н. Нечеткие множества в моделях искус-
1980 ственного интеллекта // Вопросы кибернетики. Проблемы искусственного интеллекта. - М.: ВИНТИ, 1980. - С. 79-86.
- Аверкин, Аверкин А.Н. Нечеткое отношение моделирования в
1986 системах искусственного интеллекта // Автореф. дис. ... канд. физ.-мат.наук. - М., 1986. - 16с.
- Аверкин и др., Аверкин А.Н., Батыршин И.З., Блишун А.Ф., Силов
1986 В.Б., Тарасов В.Б. Нечеткие множества в моделях управления и искусственного интеллекта. - М.: Наука, 1986. - 312 с.
- Адельсон-Вельский Адельсон-Вельский Г.М. Программирование игр. -
и др., 1978 М.: Наука, 1978. - 255 с.
- Адельсон-Вельский Адельсон-Вельский Г.М., Арлазаров В.Л., Битман
и др., 1983 А.Р., Донской М.В. Машина играет в шахматы. - М.: Наука, 1983. - 207 с.
- Айзерман и др., Айзерман М.А., Браверман Э.М., Розоноер Л.И. Ме-
1970 тод потенциальных функций в теории обучения машин. - М.: Наука, 1970. - 240 с.
- Аксельруд, Аксельруд Ф.П. Моделирование джазовой импровиза-
1975 ции с помощью ЭВМ: Дипломная работа студента мате-матического факультета ДВГУ. - Владивосток, 1975. - 110 с.
- Александров и др., Александров В.В., Горский Н.Д. Алгоритмы и прог-
1983 раммы структурного метода обработки данных. - Л.: Наука, 1983. - 207 с.
- Александров и др., Александров В.В., Горский Н.Д. Представление и

- 1985 обработка изображений. Рекурсивный подход. - Л.: Наука, 1985. - 190 с.
- Александров и др., Александров В.В., Горский Н.Д., Мысько С.Н. Экспертные системы анализа изображений. Л.: ЛИИАН, 1986. - 37 с.
- Александров и др., Александров В.В., Горский Н.Д. Базы видеоданных: состояние и тенденция развития // Прикладная информатика. - М.: Финансы и статистика, 1987. С. 24-52.
- Алексеева и др., Алексеева Е.Ф., Стефаник В.Л. Экспертные системы - состояние и перспективы // Изв. АН СССР. Техн. кибернетика. - 1984. - N 5. - С. 153-167.
- Амосов и др., Амосов Н.М., Касаткин А.И., Касаткина Л.М. Активные семантические сети в роботах с автономным управлением // Тр. IV МОКИИ. - М., 1975, - Т.9. - С. 11-20.
- Амосов и др., Амосов Н.М. Алгоритмы разума. - Киев: Наукова думка, 1979. - 221 с.
- Анисимов и др., Анисимов Б.В., Курчанов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений. - М.: Высшая школа, 1983. - 295 с.
- Апресян 1974 Апресян Ю.Д. Лексическая семантика. Семиотические средства языка. - М.: Наука, 1974. - 366 с.
- Апресян 1978 Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др. Лингвистическое обеспечение в системе автоматического перевода третьего поколения. - М.: АН СССР, 1978. - 47 с.
- Арбиб, 1976 Арбиб М. Метафорический мозг: Пер. с англ. - М.: Мир, 1976. - 295 с.
- Аристотель, 1975 Аристотель: Соч. в 4 томах. - М.: Мысль, 1975-1983. - Т.1. Метафизика. О душе. - 549 с.
- Арсентьева, 1963а Арсентьева Н.Г. О синтезе предложений русского языка при помощи машины // Проблемы кибернетики. - 1963.- Вып.10. - С. 227-240.
- Арсентьева, 1963б Арсентьева Н.Г. О синтезе предложений русского языка при помощи машины // НТИ.- 1963.- N. 7,8.
- Арсентьева, 1965а Арсентьева Н.Г. О двух способах порождения предложений русского языка. // Проблемы кибернетики. - 1965.- Вып.14. - С. 189-218.
- Арсентьева, 1965б Арсентьева Н.Г. Исследование текстов, синтезированных машиной. // Проблемы кибернетики. - 1965. Вып.15. - С. 201-207.
- Аствацатрян, 1977 Аствацатрян Л. А. Роль ЭВМ в создании серийного материала симфонии // Матер.1-го Всесоюз. семина. по машинным аспектам алгоритмического формализованного анализа музыкальных текстов. - Ереван,

1977. - С.244-245.

- Балонов и др., 1976 Балонов Л.Я., Деглин В.Л. Слух и речь доминантного и недоминантного полушарий. - Л.: Наука, 1976. - 215 с.
- Барабаш и др., 1967 Барабаш И.Л., Варский В.В., Зиновьев В.Т., Кириченко В.С., Сапегин В.Ф. Вопросы статистической теории распознавания. - М.: Сов.радио, 1967. - 400 с.
- Бардзинь и др., 1972 Бардзинь Я.М., Фрейвалд Р.В. О прогнозировании обдерекурсивных функций // ДАН СССР. 1972. - Т.206, N 3. - С. 521-524.
- Бардзинь и др., 1974 Бардзинь Я.М., Кинвер Е.Б., Подниекс К.М. Об ускорении синтеза и прогнозирования функций // Ученые записки Латв.ИУ. - Рига, 1974. - Т.210. - С. 117-128.
- Бардзинь, 1982 Бардзинь Я.М. Некоторые правила индуктивного вывода и их применение // Семиотика и информатика. - 1982. - Вып. 19. - С.59-89.
- Батыршин, 1986 Батыршин И.З. Нечеткие множества с унифицированными шкалами // Методы и системы принятия решений. - Рига: РПИ, 1986. - С. 53-63.
- Бахмутова и др., 1985 Бахмутова И.В., Гусев В.Д., Зарипов Р.Х., Титкова Т.Н. Выявление и анализ сходных фрагментов в музыкальных произведениях // Вычислительные системы. - Новосибирск: ВЦ СО АН СССР, 1985. - Вып. 113. - С. 3-45.
- Блишун, 1987 Блишун А.Ф. Формализация обобщенной оценки и классификация нечетко описанных объектов // Нечеткие системы: моделирование структуры и оптимизация. - Калинин : КГУ, 1987. - С. 40-56.
- Блишун, 1988 Блишун А.Ф. Сравнительный анализ методов измерения нечеткости // Изв. АН СССР, Техн. кибернетика. - 1988. N 5. - С. 51-76.
- Бонгард, 1967 Бонгард М.М. Проблема узнавания. - М.: Наука, 1967. - 320 с.
- Борисов и др., 1970 Борисов А.Н., Осис Я.Я. Методики оценки функций принадлежности элементов размытого множества // Кибернетика и диагностика. - Рига : РПИ, 1970. - С. 123-125.
- Борисов и др., 1982 Борисов А.Н., Алексеев А.В., Крумберг О.А. Модели принятия решений на основе лингвистической переменной. Рига : Зинатне. - 1982. - 256 с.
- Братчиков и др., 1968 Братчиков И.Л., Фиталов С.Я., Цейтин Г.С. О структуре словаря и кодировке информации для машинного перевода // Материалы по машинному переводу. - Л.: ЛГУ, 1968. с.
- Брунер, Брунер Дж. Онтогенез речевых актов // Психолинг-

- 1984 вистика: Пер с англ. - М.: Прогресс, 1984. - С. 21-49.
- Брябрин, 1978 Брябрин В.М. Диалоговая информационно-логическая система // Семиотика и информатика. - 1978. Вып.9. - С. 17-24.
- Бурбаки, 1963 Бурбаки Н. Архитектура математики // Очерки по истории математики: Пер. с фр. - М.: ИЛ, 1963. - С. 245-259.
- Буторов и др., 1985 Буторов В.Д., Шереметьева С.О. Алгоритмизированный синтез формулы изображений // НТИ. Сер. 2. - 1985, - N 7. - С. 20-28.
- Вагин и др., 1982 Вагин В.Н., Викторова Н.П. Вопросы структурного обобщения и классификации в системах принятия решений // Изв. АН СССР. Техн. кибернетика. - 1982. - N 5. - С. 64-73.
- Вагин и др., 1984 Вагин В.Н., Кикнадзе В.Г. Дедуктивный вывод на семантических сетях в системе принятия решений. // Изв. АН СССР. Техн. кибернетика. - 1984. - N5. - С. 106-120.
- Вагин и др., 1985 Вагин В.Н., Викторова Н.П. Задачи обобщения в системах принятия решений: формирование классов объектов и отношений выбора на семантических сетях // Изв. АН СССР. Техн. кибернетика. - 1985. - N 5. - С. 3-17.
- Вагин, 1986 Вагин В.Н. Параллельная дедукция на семантических сетях // Изв. АН СССР. Техн. кибернетика. - 1986. - N 5. - С.51-61.
- Вайнцвайг, 1973 Вайнцвайг М.Н. Алгоритм обучения распознаванию образов "Кора" // Алгоритмы обучения распознаванию образов. - Сов. радио. 1973. - С. 110-116.
- Вальд, 1962 Вальд А. Последовательный анализ: Пер. с англ. - М.: ИЛ., 1962. - 328 с.
- Валюс, 1962 Валюс Н.А. Стереоскопия. - М.: АН СССР, 1962. - 104 с.
- Ван Дам, 1986 Ван Дам А. Машинная графика // Современный компьютер: Пер. с англ. - М.: Мир, 1986. - 109 с.
- Ван Хао, 1970 Ван Хао. Формализация и автоматическое доказательство теорем // Кибернетический сборник: Пер. с англ. - М.: Мир, 1970. - Вып. 7. - С. 180-193.
- Вапник и др., 1974 Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. - М.: Наука, 1974. - 415 с.
- Вапник 1979 Вапник В.Н. Восстановление зависимостей по эмпирическим данным. - М.: Наука, 1979. - 448 с.
- Вапник и др., 1984 Вапник В.Н., Глазова Т.Г., Кошечев В.А., Михальский А.И., Червоненкис А.Я. Алгоритмы и программы восстановления зависимостей. - М.: Наука,

1984. - 816 с.

- Варосян и др., 1982 Варосян С.О., Поспелов Д.А. Наметрическая прост-
ранственная логика // Изв. АН СССР. Техн. кибер-
нетика. - 1982. - N 5. - С. 86-99.
- Варшавский и др., 1984 Варшавский В.И., Поспелов Д.А. Оркестр играет
без дирижера. - М.: Наука, 1984. - 208 с.
- Васин, 1984 Васин Ю.Г. "Хорошо приспособленные" локальные
однородные методы обработки графической инфор-
мации // Автоматизация обработки сложной гра-
фической информации. - Горький: ГТУ, 1984. -
С. 131-159.
- Ведерникова, 1975 Ведерникова Н.М. Русская народная сказка. - М.:
Наука, 1975.
- Ведерникова, 1980 Ведерникова Н.М. Эпитет в волшебной сказке //
Эпитет в русском народном творчестве. - М.: МГУ,
1980. - с. 120-133.
- Вейценбаум, 1982 Вейценбаум Дж. Возможности вычислительных машин и
человеческий разум. От суждений к вычислениям:
Пер. с англ. - М.: Радио и связь, 1982. - 367 с.
- Величковский и др., 1979 Величковский Б.М., Зинченко В.П. Методологические
проблемы современной когнитивной психологии
// Вопросы философии. - 1979. - N 7. - С. 67-79.
- Величковский, 1982 Величковский Б.М. Современная когнитивная психо-
логия. - М.: МГУ, 1982. - 336 с.
- Волож и др., 1982 Волож Б., Мацкин М., Минц Г., Тютю Э. Система
ПРИЗ и исчисление высказываний // Кибернетика. -
1982. - N 6. - С. 63-70.
- Вычислительные, 1967 Вычислительные машины и мышление: Пер. с англ. -
М.: Мир, 1967. - 552с.
- Гаазе-Рапопорт, и др., 1980 Гаазе-Рапопорт М.Г., Поспелов Д.А., Семенова
Е.Т. Порождение структуры волшебных сказок / На-
учн. совет по комплексной проблеме "Кибернетика".
- М., 1980. - 20 с.
- Гаазе-Рапопорт, 1983 Гаазе-Рапопорт М.Г. Поиск вариантов в сочинении
сказок. // Зарипов Р.Х. Машинный поиск вариантов
при моделировании творческого процесса. - М.:
Наука, Физматгиз, 1983. - С. 219-223.
- Гаазе-Рапопорт и др., 1984 Гаазе-Рапопорт М.Г., Поспелов Д.А., Семенова
Е.Т. Имитация сказочного мира // Ученые записки
Тартуского ГУ. -Тарту, 1984. - Вып.688.- с.46-58.
- Гаазе-Рапопорт 1987 Гаазе-Рапопорт М.Г., Поспелов Д.А. От амебы до
робота: модели поведения. - М.: Наука, 1987. -
285 с.
- Гаврилова, 1984 Гаврилова Т.А. Представление знаний в экспертной
диагностической системе АВТАНТЕСТ // Изв. АН
СССР. Техн. кибернетика. - 1984. - N 5. -

- Гаек и др., 1984 Гаек П., Гавранек Т. Автоматическое образование гипотез: Пер. с англ. - М.: Наука, 1984. - 277 с.
- Гарднер, 1984 Гарднер М. А ну-ка, догадайся! Пер. с англ. - М.: Мир, 1984. - 212 с.
- Гаспарян, 1981 Гаспарян Э.Г. Аналитико-информационно-поисковая система обработки народной песни: Автореферат дис. канд.техн.наук. - Ереван, 1981. - 20 с.
- Гвенцадзе, 1986 Гвенцадзе М.А. Коммуникативная лингвистика и типология текста. - Тбилиси.: Ганатлеба, 1986. - 316 с.
- Гелернтер, 1967 Гелернтер Г. Реализация машины, доказывающей геометрические теоремы. // Вычислительные машины и мышление. - М.: Мир, 1967. - С. 145-164.
- Гитлина и др., 1981а Гитлина Л.С., Голендер В.Е., Розенблит А.Б. Обобщение информации об объектах, представленных графами // Семиотика и информатика. - 1981.- Вып. 17. - С.14-28.
- Гитлина и др., 1981б Гитлина Л.С., Голендер В.Е., Розенблит А.Б. Поиск закономерностей в классах объектов, представленных графами // Машинные методы обнаружения закономерностей. - Рига: Знание, 1981. - С. 77-82.
- Гладун, 1970 Гладун В.П. Формирование понятий путем обучения растущих сетей // Кибернетика.- 1970.- N 2. - С.
- Гладун и др., 1975 Гладун В.П., Вашенко Н.Д. Методы формирования понятий на ЦВМ // Кибернетика. - 1975. - N 2. - С.107-112.
- Гладун и др., 1977 Гладун В.П. Эвристический поиск в сложных средах. - Киев: Наукова думка, 1977. - 166 с.
- Гладун, 1987 Гладун В.П. Планирование решений. - Киев: Наукова думка, 1987. - 168 с.
- Глезер, 1985 Глезер В.Д. Зрение и мышление. - Л.: Наука, 1985. - 246 с.
- Глотов и др., 1976 Глотов В.А., Павелъев В.В. Экспертные методы определения весовых коэффициентов // Автоматика и телемеханика. - 1976. - N 12. - С. 95-107.
- Гнездилова и др., 1988 Гнездилова Г.Г., Гончаров О.А., Сенин Г.В. Персональный компьютер в играх и задачах. - М.: Наука, 1988. - 220 с.
- Голубев и др., 1987 Голубев С.А., Гукова С.М., Куршов Е.П., Осипов Г.С. Пустая система для диагностики технических объектов // Технология разработки экспертных систем: Тез. докл. - Кишинев, 1987.
- Горелик, Горелик А.Л. Игровой подход к построению прост-

- 1973 ранства признаков системы распознавания образов и явлений. // Кибернетика. - 1973. - N 5. - С. 15-21.
- Горелик и др., 1977 Горелик А.Л., Скрипкин В.А. Методы распознавания. - М.: Высшая школа, 1977. - 117 с.
- Горелик и др., 1985 Горелик А.Л., Гуревич И.Б., Скрипкин В.А. Современное состояние проблемы распознавания. Некоторые аспекты. - М.: Радио и связь, 1985. - 162 с.
- Горелов, 1980 Горелов И.Н. Невербальные компоненты коммуникации. - М.: Наука, 1980. - 104 с.
- Горелов, 1987 Горелов И.Н. Разговор с компьютером: психолингвистический аспект проблемы. - М.: Наука, 1987. - 256 с.
- Городецкий, 1989 Городецкий Б.Ю. Компьютерная лингвистика - моделирование языкового общения (Вступительная статья) // Новое в зарубежной лингвистике. Вып. XXIV. - М.: Прогресс, 1989. - 432 с.
- Городецкий и др., 1985 Городецкий Б.Ю., Кобозева И.М., Сабурова И.Г. К типологии коммуникативных неудач // Диалоговое взаимодействие и представление знаний. Новосибирск: ВЦ СО АН СССР, 1985. - С.64-78.
- Горячев и др., 1984 Горячев и др. Психологические основы совершенствования летной оценки авиационного тренажера - Ч. 1. - Ереван, 1984, - С. 146-148.
- Горячук и др., 1982 Горячук В.Ф., Лозовский В.С., Сокольников А.И. Базовая система ситуационного моделирования Н-Р/Н-РТ. Одесса, 1982. - 109 с. - Деп. в ВИНТИ 09.03.83. N 1218-83.
- Грановская, 1974 Грановская Р.М. Восприятие и модели памяти. - Л.: ЛГУ, 1974.
- Грегори, 1972 Грегори Р.Л. Разумный глаз: Пер. с англ. - М.: Мир, 1972. - 305 с.
- Гренандер, 1979-1983 Гренандер У. Лекции по теории распознавания образов: Пер с англ. - М.: Мир, 1979-1983. В 3-х томах.
- Гуревич и др., 1974 Гуревич И.Б., Журавлев Ю.И. Минимизация булевых функций и эффективные алгоритмы распознавания // Кибернетика. - 1974. - N 3. - С. 16-20.
- Гуревич и др., 1983 Гуревич И.Б., Журавлев Ю.И. О формализации принципов выбора алгоритмов анализа изображений // Методы и средства обработки сложноструктурированной семантически насыщенной графической информации: Тез. докл. 1 Всесоюз. конф. - Горький, 1983. - С. 20-21.
- Гуревич, 1984 Гуревич И.Б. Анализ изображений методом реверсивного алгебраического замыкания // Проблемы

искусственного интеллекта и распознавания образов: тез. докл. и сообщений науч. конф. с участием ученых из социалистических стран / Ин-т кибернетики АН УССР. - Киев, 1984. - С. 41-43.

Гуревич,
1985

Гуревич И.Б. Алгебраический подход к анализу и распознаванию изображений // Математические методы распознавания образов: Тез. докл. Всесоюз. конф. АН АрмССР. - Ереван, 1985 - С. 55-57.

Гуревич,
1986а

Гуревич И.Б. Методы распознавания и анализа изображений // Автоматизированные системы обработки изображений (АСОИЗ-86): тез. докл. II Всесоюз. конф. - М.: Наука, 1986. - С. 124-126.

Гуревич,
1986б

Гуревич И.Б. Определение класса алгоритмов вычисления оценок по двумерной информации для задач распознавания изображений // Методы и средства обработки графической информации: межвуз. сб. научн. трудов. - Горький: Горьков. ГУ, 1986. - С. 47-66.

Гуревич,
1989

Гуревич И.Б. Проблема распознавания изображений в задачах распознавания // Распознавание, классификация, прогноз. Математические методы и их применение. - М.: Наука, 1989. - Вып. 1. - С. 280-329.

Гусакова и др.,
1986

Гусакова С.М., Финн В.К. О формализации локального и глобального сходств // НТИ. Сер. 27 - 1986. - N 6. - С. - 16-19.

Гусакова и др.,
1987а

Гусакова С.М., Финн В.К. О новых средствах формализации понятия сходства // НТИ. Сер. 2. - 1987. - N 10. - С. 14-22.

Гусакова и др.,
1987б

Гусакова С.М., Финн В.К. Сходства и правдоподобный вывод // Изв. АН СССР. Техн. кибернетика. - 1987. - N 5. - С. 42-63.

Гери и др.,
1982

Гери М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. - М.: Мир, 1982. - 416 с.

Давыдов и др.,
1969

Давыдов Г.В., Маслов С.Ю., Минц Г.Е., Ореков В.П., Слисенко А.О. Машинный алгоритм установления выводимости на основе обратного метода // Исследования по конструктивной математике и математической логике. Записки науч. семинаров ЛОМИ АН СССР. - Л.: Наука, 1969. - Т. 6, - С. 17-32.

Девис,
1970

Девис М. Устранение лишнего из механических доказательств // Кибернетический сборник: Пер. с англ. - М.: Мир, 1970. - Вып. 7. - С. 160-179.

Дейт,
1980

Дейт К. Введение в системы баз данных: Пер. с англ. - М.: Наука, 1980. - 463 с.

Демидов,

Демидов В.П. Как мы видим то, что видим. - М.:

1987

Знание. 1987. - 54 с.

Денисов и др.,
1984

Денисов В.М., Матвеев Ю.Н., Очин Е.Ф. Принципы организации систем обработки изображений на базе клеточной логики // Зарубежная радиоэлектроника. - 1984. - N 1. - С. 3-25.

Диковский,
1985

Диковский А.Я. Решение в линейное время алгоритмических проблем, связанных с синтезом ациклических программ // Программирование. - 1985. - N 3. - С. 7-12.

Диковский и др.,
1985

Диковский А.Я., Канович М.И. Вычислительные модели с разделяемыми подзадачами // Изв. АН СССР. Техн. кибернетика. - 1985. - N 5. - С. 36-59.

Дмитриев и др.,
1966

Дмитриев А.Н., Журавлев Ю.И., Кренделев Ф.П. О математических принципах классификации предметов и явлений // Дискретный анализ. - Новосибирск: Ин-т матем. СО АН СССР, 1966. - Вып. 7. - С.3-15.

Добрушин
1961

Добрушин Р.Л. Математические методы в лингвистике // Математическое просвещение. -М.: Физматгиз, 1961. -Вып. 6. -С. 37-60.

Дрибас,
1982

Дрибас В.П. Реляционные модели баз данных. - Минск: БГУ, 1982. - 192 с.

Дружинин,
1973

Дружинин Н.К. Логика оценки статистических гипотез. - М.: Статистика, 1973. - 212 с.

Дуда и др.,
1976

Дуда Р., Харт П. Распознавание образов и анализ сцен: Пер. с англ. - М.: Мир, 1976. - 511 с.

Дулин,
1982

Дулин С.К. Исследование сетей с диссонансами // Изв. АН СССР. Техн. кибернетика. - 1982. - N 5. - С. 74-85.

Ежкова и др.,
1977

Ежкова И.В., Пospelов Д.А. Принятие решений при нечетких основаниях. 1. Универсальная шкала // Изв. АН СССР. Техн. кибернетика. - 1977. - N6. - С. 3-11.

Емельянов,
1981

Емельянов А.М. Метод анализа управляющей деятельности человека посредством фреймов и специальной модальной логики. // Изв. АН СССР. Техн. кибернетика. - 1981. - N 4. - С. 94-102.

Еремин и др.,
1979

Еремин И.И., Мазуров В.Д. Нестационарные процессы математического программирования. - М.: Наука, 1979. - 288 с.

Ершов,
1973

Ершов А.П. Современное состояние теории схем программ // Проблемы кибернетики. - 1973. - Вып. 27. - С. 87-110.

Ефимов,
1979

Ефимов Е.И. Автоматическое формирование понятий, описываемых замкнутыми формулами узкого исчисления предикатов // Семiotика и информатика. - 1979. - Вып. 13. - С. 86-108.

- Ефимов,
1982 Ефимов Е.И. Решатели интеллектуальных задач. - М.: Наука, 1982. - 320 с.
- Ефимов,
1985 Ефимов Е.И. Исчисление правдоподобностей в нестрогих человеческих рассуждениях // Изв. АН СССР. Техн. кибернетика. - 1985. - N 5. - С. 60-75.
- Ефимов,
1986 Ефимов Е.И. Детерминационное исчисление // Изв. АН СССР. Техн. кибернетика. - 1986. - N 5. - С. 62-81.
- Ефимов,
1987 Ефимов Е.И. Дедуктивные рассуждения с нечеткими кванторами // Изв. АН СССР. Техн. кибернетика. - 1987. - N 5. - С. 120-133.
- Ефимова,
1985 Ефимова С.М. П-графы для представления знаний. - Препринт. - М., 1985. - 40 с. - (ВЦ АН СССР).
- Ефимова и др.,
1986 Ефимова С.М., Суворов Е.В. Модель П-графов для представления знаний и способ ее аппаратной реализации на основе метода МЗ // Изв. АН СССР. Техн. кибернетика. - 1986. - N 2. - С. 32-47.
- Ефимова и др.,
1988 Ефимова С.М., Суворов Е.В. Поиск в базах знаний, опирающихся на модель П-графов, его аппаратная реализация на основе метода МЗ - Препринт. - М., 1988. - 64 с. - ВЦ АН СССР.
- Жолковский,
1974 Жолковский А.К. Модель "Смысл-Текст" // Энциклопедия кибернетики. - Киев: Украинская Советская энциклопедия. 1974. - Том 2.
- Жуковин и др.,
1983 Жуковин В.Е. Многокритериальные модели принятия решений с неопределенностью. - Тбилиси: Менциереба, 1983. - 102 с.
- Журавлев,
1965 Журавлев Д.И. Локальные алгоритмы вычисления информации. I. // Кибернетика. - 1965. - N 1. - С. 12-19.
- Журавлев,
1966 Журавлев Д.И. Локальные алгоритмы вычисления информации. II // Кибернетика. - 1966. - N 2. - С. 1-11.
- Журавлев,
1971 Журавлев Д.И. Экстремальные задачи, возникающие при обосновании эвристических процедур // Проблемы прикладной математики и механики. - М.: Наука, 1971. - С. 67-75.
- Журавлев и др.,
1971 Журавлев Д.И., Никифоров В.В. Алгоритмы распознавания, основанные на вычислении оценок // Кибернетика. - 1971. - N 3. - С. 1-11.
- Журавлев,
1974 Журавлев Д.И. Алгоритмы построения минимальных дизъюнктивных нормальных форм для функций алгебры логики // Дискретная математика и математические вопросы кибернетики. Т.1 - М.: Наука, 1974. - С. 67-98.
- Журавлев и др.,
1974 Журавлев Д.И., Камиллов М.М., Туляганов Ш.Е. Алгоритмы вычисления оценок и их применение. -

Ташкент: Фан, 1974. - 38 с.

- Журавлев, 1976 Журавлев Д.И. Непараметрические задачи распознавания образов // Кибернетика. - 1976. - N 6. - С. 93-103.
- Журавлев, 1978а Журавлев Д.И. Об алгебраическом подходе к решению задач распознавания и классификации // Проблемы кибернетики. - М.: Наука, 1978. - Вып. 33. - С. 5-68.
- Журавлев, 1978б Журавлев Д.И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. III // Кибернетика. - 1978. - N 2. - С. 35-43.
- Журавлев и др., 1980 Журавлев Ю.И., Зенкин А.И., Исаев И.В., Кольцов П.П., Кочетков Д.В., Рязанов В.В. Задачи распознавания и классификации со стандартной обучающей информацией // Журнал вычисл. матем. и матем. физики. - 1980. - Т. 20, N 5. - С. 1294-1309.
- Журавлев, 1989 Журавлев Д.И. Об алгебраических методах в задачах распознавания и классификации // Распознавание, классификация, прогноз. Математические методы и их применение. - М.: Наука, 1989. - Вып.1. - С. 9-16.
- Журавлев и др., 1989 Журавлев Ю.И., Гуревич И.Б., Распознавание образов и распознавание изображений // Распознавание, классификация, прогноз. Математические методы и их применение. - М.: Наука, 1989.
- Забежайло и др., 1982 Забежайло М.И., Финн В.К., Козлова С.П., Катамадзе Т.Г., Авидон В.В., Рабинков А.А. Об одном методе автоматического формирования гипотез и его программной реализации // НТИ. Сер. 2. - 1982. - N 4. - С. 20-26.
- Завалишин и др., 1974 Завалишин Н.В., Мучник И.Б. Модели зрительного восприятия и алгоритмы анализа изображений. - М.: Наука, 1974. - 344 с.
- Загоруйко, 1981 Загоруйко Н.Г. Методы обнаружения закономерностей. - М.: Наука, 1981. - 115 с.
- Загоруйко, 1972 Загоруйко Н.Г. Методы распознавания и их применения. - М.: Сов. радио, 1972. - 208 с.
- Заде, 1974 Заде Л.А. Основы нового подхода к анализу сложных систем и процессоров принятия решений // Математика сегодня: Пер. с англ. М.: Знание, 1974. - С. 5-48.
- Зарипов, 1960 Зарипов Р.Х. Об алгоритмическом описании процесса сочинения музыки // ДАН СССР. - 1960. - Т. 132, N 6. - С. 1283-1286.
- Зарипов, 1963 Зарипов Р.Х. Кибернетика и музыка. - М.: Знание, 1963. - 56 с.

- Зарилов,
1965 Зарилов Р.Х. О моделировании мелодий заданного
стиля на ЭВМ. // Проблемы кибернетики. - М.: На-
ука, 1965. - Вып. 15. - С. 157-200.
- Зарилов,
1966 Зарилов Р.Х. Об алгоритмизации решения задач по
гармонии и анализа гармонизации // ДАН СССР.
- 1966. - Т. 166, N 5. - С. 1069-1072.
- Зарилов,
1967 Зарилов Р.Х. Моделирование функций композитора и
музыковеда на ЭВМ // Проблемы моделирования пси-
хической деятельности. - Новосибирск: НГУ,
1967. - С.121-127.
- Зарилов,
1971 Зарилов Р.Х. Кибернетика и музыка. - М.: Наука,
1971. - 236 с.
- Зарилов,
1983 Зарилов Р.Х. Машинный поиск вариантов при моде-
лировании творческого процесса. - М.: Наука,
1983. - 232 с.
- Зарилов,
1984 Зарилов Р.Х. Построение частотных словарей музы-
кальных интонаций для анализа и моделирования
мелодий // Проблемы кибернетики. - М.: Наука,
1984. - Вып. 41. - С. 207-252.
- Зарилов,
1985 Зарилов Р.Х. Диалоговый режим в музыке на основе
интервально-метрической структуры интонации // Изв. АН СССР. Техн. кибернетика. - 1985. - N 5.
- С. 115-128.
- Зарилов,
1986 Зарилов Р.Х. Компьютер в исследовании и сочине-
нии музыки // Природа. - 1986. - N 8. - С.
59-69.
- Зарилов,
1987а Зарилов Р.Х. Алгоритмизация мелодий методом за-
имствованных интонаций // ДАН СССР. - Т. 296,
N 6. 1987. - С. 1306-1309.
- Зарилов,
1987б Зарилов Р.Х. Продукционная система в музыке // Изв. АН СССР. Техн. кибернетика. - 1987, N 2. -
С. 207-216.
- Зарилов,
1988 Зарилов Р.Х. Проблемы применения ЭВМ в музы-
кальной практике // ЭВМ и проблемы музыкальной
науки. - Новосибирск: НГК, 1988. -С. 83-109.
- Зенкин,
1984 Зенкин А.А. Применение ЭВМ на этапе построения
научных теорий. Некоторые подходы и результаты
// Автоматизация научных исследований. - М.:
МГУ, 1984. - С. 63-72.
- Зенкин,
1987 Зенкин А.А. Интерактивная компьютерная графика в
теоретических исследованиях: обобщение класси-
ческой теоремы Гильберта-Варинга // Изв. АН СССР.
Техн. кибернетика. - 1987, N 5. - С.
- Зрительное,
1975 Зрительное опознание и его нейрофизиологические
механизмы. - Л.: Наука, 1975. - 278 с.
- Зубов,
1985 Зубов А.В. Вероятностно-алгоритмическая модель
порождения текста (семантико-синтаксический ас-

- пект): Дис. ... докт. филол. наук. - Минск, 1985. - 370 с.
- Ивин,
1970 Ивин А.А. Логика времени // Неклассическая логика. - М.: Наука, 1970. - С. 124-190.
- Ильин и др.,
1982 Ильин Г.М., Игнатов В.Н., Король И.А. Задачи автоматического синтеза текстов // Переработка текстов методами инженерной лингвистики: Тез. докл. Всесоюз. конф. - Минск, 1982. - С. 90-92.
- Кайберг,
1978 Кайберг Г. Вероятность и индуктивная логика: Пер. с англ. - М.: Прогресс, 1978. - 374 с.
- Калиниченко,
1983 Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука, 1983. - 423 с.
- Кандрашина,
1983 Кандрашина Е.Ю. Время в представлении знаний. Соотношение между единичными событиями. - Препринт. - Новосибирск, 1983. - 25 с. -(ВЦ СО АН СССР).
- Кандрашина,
1988 Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А. Пространство и время в системах искусственного интеллекта. - М.: Наука, 1988.
- Кандрашина и др.,
1989 Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А. Представление знаний о пространстве и времени в системах искусственного интеллекта. - М.: Наука, 1989.
- Канович,
1985 Канович М.И. Эффективные логические алгоритмы анализа и синтеза зависимостей // ДАН СССР. - 1985. - 285, N 6. 1985. - С. 1301-1305.
- Канович,
1986 Канович М.И. Квазиномиальные алгоритмы распознавания выполнимости и выводимости препозициональных формул // ДАН СССР, - 1986. - 290, N 2 - С. 281-286.
- Канович,
1987 Канович М.И. Вычислительные задачи: семантика и логика. Логико-алгебраические конструкции. - Калинин: Калинин. ГУ, 1987. - С. 35-42.
- Караваев,
1983 Караваев Э.В. Основания временной логики. - Л.: ЛГУ, 1983. - 177 с.
- Карпилович,
1977 Карпилович Т.П. Алгоритмы порождения предложений естественного языка (обзор и анализ) // Романское и германское языкознание. Вып. 1: Вопросы экспериментальной фонетики и прикладной лингвистики. - Минск, 1977. - С. 205-218.
- Карпов,
1987 Карпов Ю.Г. Согласованность систем параллельных информационных процессов. // Изв. АН СССР. Техн. киберн. - 1987. - N 5. - С. 170-186.
- Касслер,
1975 Касслер М. МИР - простой язык программирования для поиска музыкальной информации // Моль А.,

- Фукс В., Касслер М. Искусство и ЭВМ. - М.: Мир, 1975. - С. 443-489.
- Катыс,**
1975 Катыс Г.П. Объемное и квазиобъемное представление информации. - М.: Энергия, 1975. - 368 с.
- Кац,**
1978 Кац Б.Г. О программе, сочиняющей стихи // Автоматика и телемеханика. - 1978. N 8. - С. 151-156.
- Кендел и др.,**
1960 Кендел М., Юл Дж. Э. Теория статистики.: Пер. с англ. - М.: Гостандарт. - 1960. - 401 с.
- Киквидзе и др.,**
1979 Киквидзе З.А., Ткемаладзе Н.Т. Об одном способе взвешивания элементов нечеткого множества // Сообщения АН ГССР. - 1979. - Т. 93, N 2. - С. 317-320.
- Кинбер,**
1977 Кинбер Е.Б. Об ускорении предельной идентификации путем изменения порядка вопросов // Elekt.Information svegarb.Kibern. (EIK). -1977. V.17. - P. 345-349.
- Кирчук и др.,**
1984 Кирчук Р.С. и др. Методы и средства оперативной цифровой обработки изображений // Автоматика. - 1984. - N 4 - С. 80-87.
- Классификация,**
1980 Классификация и кластер: Пер. с англ. - М.: Мир, 1980. - 389 с.
- Клини,**
1957 Клини С.К. Введение в метаматематику: Пер с англ. - М.: ИЛ, 1957. - 526 с.
- Клини,**
1976 Клини С. Математическая логика: Пер. с англ. - М.: Мир, 1976.
- Клоксин и др.,**
1987 Клоксин У., Меллиш К. Программирование на языке пролог: Пер. с англ. - М.: Мир, 1987. - 336 с.
- Ковалевский,**
1976 Ковалевский В.А. Методы оптимальных решений в распознавании изображений. - М.: Наука, 1976. - 328 с.
- Ковалевский и др.,**
1980 Ковалевский В.А., Гимельфарб Г.Л., Возиянов А.Ф. Оптические читающие автоматы. - Киев.: Техника, 1980. - 207 с.
- Колмогоров,**
1963 Колмогоров А.Н. К изучению ритмики Маяковского. // Вопросы языкознания. 1963, N 4.
- Колмогоров,**
1965а Колмогоров А.Н. Замечания по поводу анализа ритма "Стихов о советском паспорте" Маяковского. // Вопросы языкознания, 1965, N 3.
- Колмогоров,**
1965б Колмогоров А.Н. Три подхода к определению понятия "Количество информации". // Проблемы передачи информации, 1985, N 1.
- Колмогоров,**
1966 Колмогоров А.Н. О метре пушкинских "Песен западных славян". // Русская литература, 1966, N 1.

- Колмогоров и др., 1963 Колмогоров А.Н., Прохоров А.В. О дольнике современной русской поэзии. // Вопросы языкознания. - 1963. N. 6.
- Кондратов 1963 Кондратов А.М. Создание словарей поэтического языка и рифм посредством ЭВМ // Тез. симп. по комплексному изучению художественного творчества -Л.: СП РСФСР, 1963. - С. 26-27.
- Кондратов и др., 1984 Кондратов А.М., Зубов А.В. Программа "СКАЛЬД" - опыт моделирования поэтического творчества для ЭВМ // Кибернетика. - 1984. - N 5. - С. 86-88.
- Котик и др., 1985 Котик М.А., Емельянов А.М. Ошибки управления. - Таллин: Валгус, 1985. - 390 с.
- Краткий, 1985 Краткий психологический словарь. - М.: Просвещение, 1985. - 432 с.
- Ксенакис, 1986 Ксенакис Я. Музыка и наука // Курьер ЮНЕСКО. - Май 1986. - С. 4-9.
- Кузнецов, 1975 Кузнецов И.П. Некоторые механизмы порождения текстов описания // НТИ. Сер. 2. - 1975. - N 1. - С. 10-15.
- Кузнецов, 1986 Кузнецов В.Е. Математические построения в К-системах // Семантика и информатика. - 1986. - Вып. 27. - С. 62-81.
- Лавров, 1986 Лавров С.С. Архитектура баз знаний. Программное обеспечение вычислительных комплексов новой архитектуры / - Новосибирск: ВЦ СО АН СССР, 1986, 1986. - С. 3-13.
- Лакхем и др., 1975 Лакхем Д., Парк Д.М., Патерсон М.С. О формализованных машинных программах // Кибернетический сборник. Новая серия. - 1975. - Вып. 12. - С. 78-114.
- Ледли, 1966 Ледли Р.С. Программирование и использование цифровых вычислительных машин: Пер. с англ. М.: Мир, 1966. - 644 с.
- Ленат, 1975 Ленат Д. БИИНГИ: знание как система взаимодействующих экспертов // Тр. 1У междунар. конф. по искусственному интеллекту. - Тбилиси, 1975. - Т. 2. - С. 63-96.
- Ленат, 1984 Ленат Д.Б. Программное обеспечение искусственного интеллекта // В мире науки. - 1984. - N 11. - с. 117.
- Леонов, 1970 Леонов В.А. Трехмерная индикация. - М.: Наука, 1970.
- Леонтьева, 1981 Леонтьева Н.Н. Семантика связного текста и единицы информационного анализа // НТИ. Сер. 2. - 1981, N 1. - С. 21-29.
- Леонтьева, Леонтьева Н.Н. Автоматический перевод как пони-

- 1982 мание и реферирование // Прикладные и экспериментальные лингвистические процессоры. - Новосибирск: ВЦ СО АН СССР, 1982.
- Леонтьева, 1986 Леонтьева Н.Н. Об информационной системе словарей Машинного фонда русского языка // Машинный фонд русского языка. Идеи и суждения. - М.: Наука, 1986.
- Лефевр, 1973 Лефевр В.А. Конфликтующие структуры. - М.: Сов. радио, 1973. - 159 с.
- Линдсей и др., 1974 Линдсей П., Норман Д. Переработка информации у человека. - М.: Мир, 1974. - 550 с.
- Литвак и др., 1975 Литвак И.И., Ломов Б.Ф., Соловейчик И.Е. Основы построения аппаратуры отображения в автоматизированных системах. - М.: Сов. радио, 1975. - 350 с.
- Литвак и др., 1981 Литвак С.Р., Роосмаа Т.А., Салуверс м., Ыйм Х.Я. О распознавании гиперсобытий в системах понимания связного текста // Диалоговые системы и представление знаний: Тр. по искусственному интеллекту Тартус. ун-та. - Тарту, 1981. С. 56-70.
- Литвинцева и др., 1980 Литвинцева Л.В., Поспелов Д.А. Время в роботах и диалоговых системах // Вопросы кибернетики: Проблемы искусственного интеллекта. - 1980. N.5 - С. 61-70.
- Литвинцева, 1985 Литвинцева Л.В. Представление знаний о каузальных и временных зависимостях // Тр. школы-семинара "Семиотические аспекты формализации интеллектуальной деятельности". - Кутаиси, 1985. - С. 87-89.
- Литвинцева и др., 1986 Литвинцева Л.В., Дембовская В.Н. Время в естественном языке и в системах искусственного интеллекта // Сообщения по прикладной математике. - М.: ВЦ АН СССР, 1986. - 22 с.
- Лозовский, 1978 Лозовский В.С. Задание реляционной базы данных в виде мультисети и реализация поиска по образцу // Информационной и программное обеспечение систем ситуационного управления. - Препринт. - Киев, 1978. - С. 13-24. - (ИЖ АН УССР 78 - 14).
- Лозовский, 1979а Лозовский В.С. Ситуационная и дефиниторная систематика системы представления знаний // Кибернетика. - 1979. - N 2. - С. 98-101.
- Лозовский, 1979б Лозовский В.С. Иерархическое программирование // Всесоюз. конф. "Семиотические модели при управлении": Тез. докл. - М., 1979. - С. 5-7.
- Лозовский, 1982 Лозовский В.С. Экстенциональная база данных на основе семантических сетей. // Изв. АН СССР. Техн. кибернетика. - 1982. - N 5. - С. 23-42.
- Ломов Ломов Б.Ф. Человек и техника. Л.: ЛГУ. - 1975. -

- 1975 88 с.
- Дурия, 1973 Дурия А.Р. Основы нейропсихологии. - М.: Наука, 1973. - 374 с.
- Дурия, 1979 Дурия А.Р. Язык и сознание. - М.: МГУ, 1979. - 319 с.
- Мальцев, 1970 Мальцев А.А. Алгебраические системы. - М.: Наука, 1970. - 392 с.
- Манин, 1979 Манин Ю.И. Доказуемое и недоказуемое. - М.: Сов. радио, 1979. - 167 с.
- Марков, 1962 Марков А.А. О конструктивной математике. // Тр. матем. ин-та АН СССР. - 1962. - N. 67. - С. 8-14.
- Марков, 1982 Марков Ю.Г. Функциональный подход в современном научном познании. - Новосибирск: Наука, 1982. - 254 с.
- Марр, 1987 Марр Л. Зрение. Информационный подход к изучению представления и обработки зрительных образов: Пер. с англ. - М., Радио и связь, 1987. - 400 с.
- Мартин, 1980 Мартин Дж. Организация баз данных в вычислительных системах: Пер. с англ. М.: Мир, 1980. - 662 с.
- Мартirosян и др., 1977 Мартirosян А.А., Погосян Э.М. Об устойчивости сравнительных характеристик согласующих индукторов // Семиотика и информатика. - 1977. - Вып. 14. - С. 92-107.
- Мартirosян, 1986 Мартirosян А.А. О некоторых классах надежных алгоритмов индуктивного вывода // Математические вопросы кибернетики и вычислительной техники. - Ереван, 1986. - Т. XVII. - С. 13-38.
- Мартынов, 1984 Мартынов В.В. Универсальный семантический код: УСК-3. Минск: Наука и техника, 1984. - 132 с.
- Марчук и др., 1975 Марчук Ю.Н., Тихомиров Б.Д., Шербин В.И. Система машинного перевода с английского языка на русский // Машинный перевод и автоматизация информационных процессов. - М.: ВЦП, 1975.
- Маслов, 1964 Маслов С.Ю. Обратный метод установления выводимости в классическом исчислении предикатов // ДАН АН СССР. - 1964. - 159, N 1. - С. 17-20.
- Маслов, 1983 Маслов С.Ю. Асимметрия познавательных механизмов и ее следствия // Семиотика и информатика. - 1983. - Вып. 20. - С. 3-31.
- Маслов, 1986 Маслов С.Ю. Теория дедуктивных систем и ее применения. - М.: Радио и связь, 1986. - 133 с.
- Меерсон, 1986 Меерсон Я.А. Высшие зрительные функции. Зрительный гнозис. - Л.: Наука, 1986. - 163 с.

- Мелетиносский, 1958 Мелетиносский Е.М. Герой волшебной сказки. - М.: Изд-во восточной литературы, 1958. - 267 с.
- Мелетиносский, 1971 Мелетиносский Е.М. Еще раз о проблеме структурного описания волшебной сказки. // Ученые записки Тартуского IV. - Тарту, 1971. - Вып. 264. - С. 63-91.
- Мелетиносский, 1979 Мелетиносский Е.М. Палеоазиатский мифологический эпос. - М.: Наука, 1979. - 228 с.
- Мельчук, 1974 Мельчук И.А. Опыт теории лингвистических моделей. "Смысл - текст". - М.: Наука, 1974. - 314с.
- Микулич, 1985 Микулич Л.И. Проблемы создания экспертных систем // Ученые записки Тартуского IV. - Тарту, 1985. - Вып. 714. Теория и модели знаний (теория и практика создания систем искусственного интеллекта). - С. 87-114.
- Милль, 1900 Милль Д.С. Система логики силлогистической и индуктивной: Пер. с англ. - М.: Книжное дело, 1900.
- Минский и др., 1971 Минский М., Пейперт С. Перцептроны: Пер. с англ. - М.: Мир, 1971. - 261 с.
- Минский, 1979 Минский М., Фрейды для представления знаний: Пер. с англ. - М.: Энергия, 1979. - 151 с.
- Миц и др., 1963а Миц Г.Е., Пенъям Я.П., Тыгу Э.Х. Структурный синтез рекурсивных программ // Автоматический синтез программ. Таллин: Ин-т кибернетики АН ЭССР, 1983. - С. 58-70.
- Миц и др., 1983б Миц Г.Е., Тыгу Э.Х. Структурный синтез и неклассические логики // III конф. "Применение методов математической логики". - Таллин, 1983. - С. 52-60.
- Моделирование, 1987 Моделирование языковой деятельности в интеллектуальных системах. - М.: Наука, 1987. - 279 с.
- Моль, 1975 Моль А. Искусство и ЭВМ // Моль А., Фуко В., Касслер М. Искусство и ЭВМ: Пер. с англ. - М.: Мир, 1975. - С. 13-258.
- Налимов, 1974 Налимов В.В. Вероятностные модели языка. - М.: Наука, 1974. - 272 с.
- Нарасимхан, 1969 Нарасимхан Р. Лингвистический подход к распознаванию образов // Автоматический анализ сложных изображений: Пер. с англ. - М.: Мир, 1969. - С. 22-26.
- Нариньяни, 1979 Нариньяни А.С. Проект ЗАПСИБ - серия лингвистических процессоров для взаимодействия с базами данных // Вопросы разработки прикладных систем - Новосибирск : ВЦ СО АН СССР., 1979.
- Нариньяни и др., Нариньяни А.С., Яхно Т.Г. Продукционные системы

- 1984 // Представление знаний в человеко-машинных и робототехнических системах .-М.: ВЦ АН СССР, ВИНТИ, 1984. - С. 136-177.
- Нариньяни, 1986а Нариньяни А.С. Неопределенность в системе представления и обработки знаний // Изв. АН СССР. Техн. кибернетика. - 1986, - N 5. - С. 14-19.
- Нариньяни, 1986б Нариньяни А.С. Параллельность обработки знаний и технологии виртуальных машин // Разработка ЭВМ нового поколения: архитектура, программирование, интеллектуализация / ВЦ СО АН СССР, - Новосибирск, 1986.
- Нильсон, 1973 Нильсон Н. Искусственный интеллект. Методы поиска решений: Пер. с англ. - М.: Мир, 1973.- 270с.
- Нильсон, 1985 Нильсон Н. Принципы искусственного интеллекта: Пер. с англ. - М.: Радио и связь, 1985. - 373 с.
- Новик, 1974 Новик Е.С. Образ в восточно-славянской волшебной сказке. - Л.: Наука, 1974. - 253 с.
- Новик, 1975 Новик Е.С. Система персонажей русской волшебной сказки. // Типологические исследования по фольклору. - М.: Наука, 1975. - с. 214-245.
- Новик, 1983 Новик Е.С. Структура содержания текста // Лингвистические вопросы алгоритмической обработки. - М.: Наука, 1983.
- Ньюэлл и др., 1967 Ньюэлл А., Шоу Дж. и Саймон Г. Эмпирические исследования машин "Логик-теоретик": пример изучения эвристических // Вычислительные машины и мышление: Пер. с англ. - М.: Мир, 1967 .
- Обработка, 1979 Обработка изображений и цифровая фильтрация: Пер. с англ. - М.: Мир, 1979. - 318 с.
- Осипов, 1987 Осипов Г.С. О формировании модели для плохо структурированной предметной области // Изв. АН СССР. Техн. кибернетика. - 1987. - N 5. - С. 198-200.
- Осис, 1968 Осис Я.Я. Распознавание неисправностей сложных объектов диагностики с использованием теории размытых множеств // Кибернетика и диагностика. - Рига : РПИ , 1968. - С. 105-120.
- Павлидис, 1986 Павлидис Т. Алгоритмы машинной графики и обработки изображений. - М.: Радио и связь, 1986. - 398 с.
- Параллельная, 1985 Параллельная обработка информации. Т.2.Параллельные методы и средства распознавания образов. - Киев: Наукова думка, - 1985. - 280 с.
- Пильшиков, 1983 Пильшиков В.Н. Язык ПЛЭНЕР. - М.: Наука ,1983. - 208 с.
- Пилюгин и др., Пилюгин В.В., Сумароков Л.Н., Фролов К.В. Машин-

- 1985 ная графика и автоматизация научных исследований
// Вестник АН СССР. - 1985. - N 10. - С. 50-58.
- Погосян,
1977 Погосян Э.М. К теории автоматического синтеза
понятий // Семиотика и информатика. - 1977.
- Вып. 8. - С. 125-152.
- Погосян,
1983 Погосян Э.М. Адаптация комбинаторных алгоритмов.
- Ереван: Изд-во АН АрмССР, 1983. - 288 с.
- Подниекс,
1974 Подниекс К.М. Сравнение различных типов предель-
ного синтеза и прогнозирования функций // Учен.
записки Латв. ГУ. - Рига, 1974. - Т. 210. - С.
68-81.
- Подниекс,
1975 Подниекс К.С. Сравнение различных типов предель-
ного синтеза и прогнозирования. / Учен. записки
Латв.ГУ. - Рига. 1975. - Т. 233. - С. 35-44.
- Поля,
1970 Поля Д. Математическое открытие: Пер. с франц. -
М.: Наука. 1970. - 452 с.
- Поля,
1975 Поля Д. Математические и правдоподобные рассуж-
дения : Пер. с англ. - М.: Наука. 1975.
- Попов,
1982 Попов Э.В. Общение с ЭВМ на естественном языке.
- М.: Наука, 1982. - 360 с.
- Поспелов Г.,
1980 Поспелов Г.С. Системный анализ и искусственный
интеллект. - М.: АН СССР, 1980. - 47 с.
- Поспелов Г.и др.,
1986 Поспелов Г.С., Поспелов Д.А. Искусственный ин-
теллект - прикладные системы. - М.: Знание,
1985. - 48 с.
- Поспелов Г.,
1986 Поспелов Г.С. Что может компьютер // НТР. Проб-
лемы и решения. - 1986. - N 8. - 20 с.
- Поспелов Д.,
1966 Поспелов Д.А. Игры и автоматы. - М.-Л.: Энергия,
1966. - 134 с.
- Поспелов Д. и др.,
1967 Поспелов Д.А., Пушкин В.Н., Садовский В.Н. Эв-
ристическое программирование и эвристика как
наука // Вопросы философии. - 1967. - N 7. - С.
35-39.
- Поспелов Д.,
1972 Поспелов Д.А. Введение в теорию вычислительных
систем. - М.: Сов. радио, 1972. - 280 с.
- Поспелов Д. и др.,
1972 Поспелов Д.А., Пушкин В.Н. Мышление и автоматы.
- М.: Сов Радио, 1972. - 222 с.
- Поспелов Д.,
1975 Поспелов Д.А. Элементы аксиоматики временных от-
ношений // Вопросы кибернетики. -1975. -N 5 - С.
- 15-21.
- Поспелов и др.,
1980 Поспелов Д.А., Шустер В.А. Нормативное поведение
интеллектуальных систем // Учен. записки Тартус-
кого ГУ. - Тарту, 1980. - Вып. 551. Логико-
семантические вопросы искусственного интеллекта.

- Поспелов Д., 1981 Поспелов Д.А. Логико-лингвистические модели в системах управления. - М.: Энергоиздат, 1981. - 231 с.
- Поспелов Д., 1982 Поспелов Д.А. Фантазия или наука? - М.: Наука, 1982. - 224 с.
- Поспелов Д., 1983 Поспелов Д.А. О "человеческих" рассуждениях в интеллектуальных системах // Вопросы кибернетики. Логика рассуждений и ее моделирование. - М., 1983. - С. 5-37.
- Поспелов Д. и др., 1985 Поспелов Д.А., Сильдмие И.Я. Рольные структуры в представлении знаний и в диалоговых системах // Изв. АН СССР. Техн. кибернетика. - 1985. - N 5. - С. 83-89.
- Поспелов Д., 1986 Поспелов Д.А. Ситуационное управление: теория и практика. - М.: Наука. 1986. - 284 с.
- Поспелов И. и др., 1987 Поспелов И.Г., Поспелова Л.Я. Динамическое описание систем продукции и проверка непротиворечивости продукционных экспертных систем // Изв. АН СССР. Техн. кибернетика. - 1987. - N 1. - С. 184-192.
- Представление, 1984а Представление знаний в человеко-машинных и робототехнических системах. Том А. Фундаментальные исследования в области представления знаний. М.: ВЦ АН СССР, ВИНТИ. - 1984. - 261 с.
- Представление, 1984б Представление знаний в человеко-машинных и робототехнических системах. Том В. Инструментальные средства разработки систем, ориентированных на знания. - М.: ВЦ АН СССР, ВИНТИ, 1984. - 236 с.
- Представление, 1984в Представление знаний в человеко-машинных и робототехнических системах. Том С. Прикладные человеко-машинные системы, ориентированные на знания. - М.: ВЦ АН СССР, ВИНТИ, 1984. - 380 с.
- Представление, 1984г Представление знаний в человеко-машинных и робототехнических системах. Том Д. Фундаментальные и прикладные исследования в области робототехнических систем. - М.: ВЦ АН СССР, ВИНТИ, 1984. - 291 с.
- Преображенский, 1990 Преображенский А.Б. Состояние развития систем естественно-языкового общения. // Искусственный интеллект. Кн.1. Системы общения и экспертные системы. - М.: Радио и связь, 1990.
- Пропш, 1928 Пропш В.Я. Морфология сказки. - М.: Академия, 1928.
- Пропш, 1946 Пропш В.Я. Исторические корни волшебной сказки. - Л.: ЛГУ, 1946. 2-е изд. - Л.: ЛГУ, 1986.
- Пропш, Пропш В.Я. Морфология сказки. - М.: Наука, 1969.

- 1969 - 168 с.
- Претт, 1982 Претт У.К. Цифровая обработка изображений. - М.: Мир. 1982. - 31- с.
- Психология, 1978 Психология машинного зрения: Пер. с англ. - М.: Мир, 1978. - 344 с.
- Пушкин, 1965 Пушкин В.Н. Оперативное мышление в больших системах. - М.: Энергия, 1965. - 375 с.
- Растрингин и др., 1981 Растрингин Л.А., Эренштейн Р.Х. Метод коллективного распознавания. - М.: Энергоиздат, 1981. - 80 с. Библиотека по автоматике.- Вып. 615.
- Реальность, 1987 Реальность и прогнозы искусственного интеллекта: Пер. с англ. - М.: Мир, 1987. - 247 с.
- Розенблатт, 1965 Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга: Пер. с англ. - М.: Мир, 1965. - 480 с.
- Розенфельд, 1972 Розенфельд А. Распознавание и обработка изображений с помощью вычислительных машин. - М.: Наука, 1972. - 232 с.
- Рок, 1980 Рок И. Введение в зрительное восприятие. В 2-х т. - М.: Педагогика, 1980. - 311 с. 279 с.
- Рошияну, 1974 Рошияну Н. Традиционные формулы сказки. - М.: Наука, 1974. - 216 с.
- Русын, 1986 Русын В.П. Структурно-лингвистические методы распознавания изображений в реальном времени. - Киев.: Наукова думка, 1986. - 127 с.
- Севбо, 1969 Севбо И.П. Структура связанного текста и автоматизации реферирования. - М.: Наука, 1969. - 135 с.
- Севбо, 1981 Севбо И.П. Графическое представление синтаксических структур и стилистическая диагностика. - Киев: Наукова думка, 1981. - 192 с.
- Селиванова, 1987 Селиванова Н.В. Уровни понимания в общении и их моделирование в системах искусственного интеллекта. // Учен. записки Тартус. ГУ.-Тарту. - 1987. Вып. 751. Модели диалога в системах искусственного интеллекта. Труды по искусственному интеллекту. - С. 113-126.
- Селиванова, 1989 Селиванова Н.В. "Дружелюбный интерфейс" и закономерности естественного диалога. - Препринт. - М., 1989. 30 с. - (ВЦ АН СССР).
- Скороходько, 1968 Скороходько Э.Ф. Информационно-поисковая система БИТ. - Киев: Наукова думка, 1968. - 120 с.
- Соболев, 1963 Соболев С.Л. Да, это вполне серьезно! // Возможное и невозможное в кибернетике. - М.: Наука, 1963 - С. 82-88.

- Советский, 1986
Совпель, 1982
Советский энциклопедический словарь. М., 1986.
Совпель И.В. Вопросы реализации систем автоматического анализа текстов естественных языков. Дис. канд. физ.-мат. наук. - Минск, 1982. - 305 с.
- Солтон, 1979
Солтон Дж. Динамические библиотечно-информационные системы: Пер. с англ. - М.: Мир, - 1979. - 557 с.
- Стеблин-Каменский, 1978
Стеблин-Каменский М.И. Историческая поэтика. - Л.: ЛГУ, 1978. - 183 с.
- Стеблин-Каменский, 1979
Стеблин-Каменский М.И. Древнескандинавская литература. - М.: Высшая школа, 1979. - 191 с.
- Степанов, 1967
Степанов А.М. Эксперимент по моделированию структуры полифонической музыки строгого стиля // Эстетические очерки. - 1967. - Вып. 2. - С. 387-406.
- Тимохин, 1983
Тимохин В.И. Применение ЭВМ для решения задач распознавания образов. - Л.: ЛГУ, 1983. - 215 с.
- Тихомиров, 1969
Тихомиров О.К. Структура мыслительной деятельности человека. - М.: МГУ, 1969. - 304 с.
- Толанский, 1975
Толанский А. Зрительные иллюзии. - М.: Мир, 1975.
- Трактенброт и др. 1970
Трактенброт Б.А., Бардзинь Я.М. Конечные автоматы (поведение и синтез) - М.: Наука, 1970. - 400 с.
- Трактман, 1987
Трактман А.А. Параллельные алгоритмы и процессоры анализа сигналов. - М.: Наука, 1987.
- Ту и др., 1978
Ту Дж., Гонсалес Р. Принципы распознавания образов: Пер. с англ. - М.: Мир, 1978. - 411 с.
- Тыгу, 1984
Тыгу Э.Х. Концептуальное программирование. - М.: Наука, 1984. - 255 с.
- Тыгу и др., 1985
Тыгу Э.Х., Мацкин М.В., Пеньям Я.Э., Зомойс П.В. Объектно-ориентированный язык НУТ // Прикладная информатика. - 1985. - Вып. 2(9). - С. 45-67.
- Тюхтин, 1976
Тюхтин В.С. Теория автоматического опознавания и гносеология. - М.: Наука, 1976. - 192 с.
- Уинстон, 1978
Уинстон П.Г. Построение структурных описаний по примерам // Психология машинного зрения: Пер. с англ. - М.: Мир, 1978. - С. 185-248.
- Уинстон, 1980
Уинстон П. Искусственный интеллект: Пер. с англ. - М.: Мир, 1980. - 520 с.
- Ульман, 1983
Ульман Дж. Основы систем баз данных : Пер. с англ. - М.: Мир, 1983. - 335 с.
- Файкс и др.,
Файкс Р, Нильсон Н. Система STRIPS - новый

- 1973 подход к применению доказательства теорем при решении задач // Интегральные роботы. - М.: Мир, 1973. - Вып.1. - С. 382-403.
- Файн,
1982 Файн В.С. Новые аспекты машинного понимания сообщений // Изв. АН СССР. Техн. кибернетика. - 1982. - N 5. - С. 100-112.
- Федосов ,
1986 Федосов Е.А. Автоматизация проектирования сложных технических систем // Вестник АН СССР. -N 10. - 1986. - С.40-50.
- Фейгенбаум,
1973 Фейгенбаум Э.А. Искусственный интеллект: темы исследований во втором десятилетии развития // Кибернетический сборник: Новая серия. - М.: Мир, 1973. - Вып 10. - С. 171-203.
- Филиппов,
1983 Филиппов В.И. Реляционно-сетевая модель данных // Прикладная информатика. - М.: Финансы и статистика, 1983. - С. 147-170. - Вып. 2.
- Филиппов,
1985 Филиппов В.И. Теоретико-множественный подход к моделям данных // Материалы III Всесоюз. конф. "Банки данных". - Таллин, 1985. - С. 16-25.
- Финн и др.,
1981 Финн В.К., Забейало М.И. Логические проблемы автоматического формирования гипотез // Тез. IX Всесоюз. симп. по кибернетике. - М.: ВИНТИ, 1981. - Т.2. - С. 54-60.
- Финн,
1983 Финн В.К. О машинно-ориентированной формализации правдоподобных рассуждений в стиле Бекона - Милля // Семиотика и информатика. - М., 1983. - Вып. 20. - С. 35-101.
- Финн,
1984 Финн В.К. Индуктивные модели // Представление знаний в человеко-машинных и робототехнических системах. - М.: ВИНТИ, 1984. - Т. А. - С. 58-76.
- Финн,
1985 Финн В.К. Правдоподобные выводы и проблемы автоматического порождения теорий из фактов // Тез. докл. советско-финского colloquiuma по логике "Интенциональные логики и логическая структура теорий". - Тбилиси: Менциереба, 1985. - С. 156-158.
- Фоли и др.,
1985 Фоли Дж., Вен Дем А. Основы интерактивной машинной графики. Пер. с англ. - М.: Мир, 1985. - Кн.1 - 367 с. Кн.2 - 368 с.
- Фоменко,
1981 Фоменко А.Т. О наглядном изображении математических понятий // Химия и жизнь. - 1981. - N 11. - С. 84-89.
- Фомина,
1985 Фомина М.В. Методы последовательного построения иерархического представления состояний сложного объекта // Изв. АН СССР. Техн. кибернетика. - 1985. - N 6. - С. 193-201.
- Француз,
1966 Француз А.Г. Некоторые вопросы статистической теории опознавания образов. // Бионика. -

М.: Наука, 1966.

- Фрейвалд, 1974 Фрейвалд Р.В. Предельно вычислимые функции и функционалы // Учен. зап. Латв. ГУ. - Рига, 1974. - Т.210. - С.6-19.
- Фрейвалд, 1975 Фрейвалд Р.В. О сложности и оптимальности предельных вычислений. Учен. зап. Латв. ГУ. - Рига, 1975. - Т. 233. - С. 155-173.
- Фрейвалд и др., 1975 Фрейвалд Р.В., Бардзинь Я.М. Соотношения между прогнозируемостью и предельной синтезируемостью // Учен. зап. Латв. ГУ. - Рига, 1975. - Т. 233. - С. 26-34.
- Фу, 1977 Фу К.С. Структурные методы в распознавании образов: Пер. с англ. - М.: Мир, 1977. - 319 с.
- Фукунага, 1979 Фукунага К. Введение в статистическую теорию распознавания. - М.: Наука, 1979. - 368.
- Хазен, 1968 Хазен Э.М. Методы оптимальных статистических решений и задачи оптимального управления. - М.: Сов. Радио, 1968. - 256 с.
- Хант и др., 1970 Хант Э., Марин Дж., Стоун Ф. Моделирование процесса формирования понятий на вычислительной машине: Пер. с англ. - М.: Мир, 1970. - 301 с.
- Хант, 1978 Хант Э. Искусственный интеллект: Пер. с англ. - М.: Мир, 1978, - 558 с.
- Харалик, 1982 Харалик Р. Структурное распознавание образов: гомоморфизмы и размещения. - Кибернетический сборник: Пер. с англ. - 1982. - N 19. - С. 170-199.
- Хачатрян, 1987 Хачатрян А.Р. Анализ классических методов объединения свидетельств в экспертных системах // Изв. АН СССР. Техн. кибернетика. - 1987. - N 5. - С. 67-73.
- Хафман, 1986 Хафман И. Активная память. - М.: Прогресс. 1986. - 309 с.
- Хьюбел, 1974 Хьюбел Д. Зрительная кора мозга. Восприятие. Механизмы и модели: Пер. с англ. - М.: Мир, 1974. - 205 с.
- Хьюитт, 1987 Хьюитт К. Открытие системы // Реальность и прогнозы искусственного интеллекта: Пер. с англ. - М.: Мир, 1987. - С. 85-103.
- Цаленко, 1985 Цаленко М.Ш. Семантические и математические модели баз данных // Итоги науки и техники. Сер. Информатика. - М., ВИНТИ. - 1985. - Т. 9. - 280 с.
- Цивьян, 1975 Цивьян Т.В. К семантике пространственных элементов в волшебной сказке. // Типологические исследования по фольклору. - М.: Наука. 1975.

- Цикритис и др., 1985 Цикритис Д., Лоховски Ф. Модели данных. - М.: Финансы и статистика, 1985. - 343 с.
- Цифровое, 1980 Цифровое кодирование графики // ТИИЭР. - 1980. - 68. - 7. - 216 с.
- Цифровое, 1981 Цифровое кодирование телевизионных изображений. - М.: Радио и связь, 1981.
- Цыпкин, 1968 Цыпкин Я.З. Адаптация и обучение в автоматических системах. - М.: Наука. 1968. - 399 с.
- Цыпкин, 1970 Цыпкин Я.З. Основы теории обучающихся систем. - М.: Наука, 1970. - 252 с.
- Чень и др., 1983 Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем: Пер. с англ. - М.: Наука, 1983. - 358 с.
- Чепис и др., 1958 Чепис И.А., Яблонский С.В. Логические способы контроля электрических схем // Сб. статей по математической логике и ее применениям к некоторым вопросам кибернетики. Труды матем. ин-та им. Стеклова. -Т.51.- М.: АН СССР, 1958. -С. 270-360.
- Чесноков, 1984 Чесноков С.В. Силлогизмы в детерминальном анализе // Изв. АН СССР. Техн. кибернетика. - 1984. - N 5. - С. 55-83.
- Чукин, 1983 Чукин Ю.В. Структуры данных для представления изображений // Зарубежная радиоволноводная электроника. - 1983.- N 8. - С. 85-108.
- Шанин и др., 1965 Шанин Н.А., Давыдов Г.В. и др. Алгоритм машинного поиска естественного логического вывода в исчислении высказываний. - М., Л.: Наука, 1965.
- Шапиро, 1973 Шапиро С.И. От алгоритмов к суждениям. - М.: Сов. радио, 1973. - 288 с.
- Шенк, 1980 Шенк Р. Обработка концептуальной информации : Пер: с англ. - М.: Энергия, 1980.-С. 15-26.
- Шенк и др., 1987 Шенк Р., Хантер Л. Познать механизмы мышления // Реальность и прогнозы искусственного интеллекта.: Пер. с англ. - М.: Мир, 1987. - С. 15-26.
- Шер, 1978 Шер А.П. Согласование нечетких экспертных оценок и функции принадлежности в методе размытых множеств // Моделирование и исследование систем автоматического управления. - Владивосток: ДВНЦ АН СССР, 1978.
- Шереметьева, 1984 Шереметьева С.О. О формальном синтезе формулы изобретения // НТИ. Сер. 2. - 1984. - N 5. - С. 5-41.
- Шлезингер, 1976 Шлезингер М.И. Синтаксический анализ двумерных зрительных сигналов в условиях помех // Киберне-

тика. - 1976. - N 4. - С. 76-82.

- Штайнбух, 1968 Штайнбух К. Человек и автомат. - М.: Сов. радио, 1968. - 420 с.
- Шустер, 1983 Шустер В.А. Субъективные оценки словесных и фреймовых описаний поступков // Вопросы кибернетики. Логика рассуждений и ее моделирование. - 1983. - с. 103-136.
- ЭВМ, 1984 ЭВМ пятого поколения. Концепции, проблемы, перспективы - М.: Финансы и статистика, 1984. - 110 с.
- Элти и др., 1987 Элти Дж., Кумбс М. Экспертные системы: концепции и примеры: Пер. с англ. - М.: Финансы и статистика, 1987. - 192 с.
- Эндрю, 1985 Эндрю А. Искусственный интеллект: Пер. с англ. - М.: Мир, 1985. - 265 с.
- Эрлих, 1984 Эрлих А.И. Диалоговая система моделирования альтернатив и выбора решений в проектировании // Представление знаний в человеко-машинных и робототехнических системах. - Том С. Прикладные человеко-машинные системы, ориентированные на знания. - М.: ВЦ АН СССР, ВИНТИ, 1984. -С.209-220.
- Эшби, 1959 Эшби Р. Введение в кибернетику. - Пер. с англ. - М.: ИЛ, 1959. - 432 с.
- Яблонский, 1979 Яблонский С.В. Введение в дискретную математику. - М.: Наука, 1979. - 272 с.
- Япония, 1987 Япония. Проект "Человеческие границы" // НТР. Проблемы и решения - 1987. - N 9.
- Ярославский, 1979 Ярославский Л.П. Введение в цифровую обработку изображений. - М.: Сов. Радио, 1979. - 220 с.
- Anderson, 1983 Anderson R.John. The architecture of cognition. -Cambridge, Mass.: Harvard Univ. Press, 1983.-163p.
- Andrews, 1981 Andrews P.B.Theorem proving via general matching // J.Assoc. Computer Mach.-1981.-Vol.28, N 2. -P. 193-214.
- Angluin, 1980 a Angluin D. Inductive inference of formal languages from positive data // Information and Control.-1980. -N.45.-1980.-P.117-135.
- Angluin, 1980 b Angluin D. Finding patterns common to a set of strings // J. Comput. System Science. -1980. -N.21. -P.46-62.
- Angluin, 1982 Angluin D. Inference of reversible languages // J. ACM. -1982. -N.29. -P. 741-765.
- Angluin, 1983 Angluin D.,Smith C.H. Inductive inference: theory and methods // Comp. Surveys. -1983.

- Anshakov et al., 1987 Anshakov O.M., Skvortsov D.P., Finn V.K. On logical means of formalization of plausible inferences // Proc. VIII Congress of Logio, Methodology and Philosophy of Science. - Moscow. -1987. -P.39-42.
- Appelbaum et al. 1985 Appelbaum L., Ruspini E. ARIES: an approximate reasoning inference engine // Approximate reasoning in Expert Systems. -1985.
- Averkin et al., 1987 Averkin A.N., Tarasov V.B. The fussy modelling relation and its application in psychology and artificial Intelligence // Fussy Sets and Systems. -1987. -Vol.22. -P.3-24.
- Austin, 1962 Austin J.L. How to do things with words.- N.Y.: Oxford Univ. Press, 1962. -119 p.
- Barbaud, 1966 Barbaud P. Initiation a la composition musicale automatique. -Paris: Dumod, 1966.-160 p.
- Barbour, 1962 Barbour J.M. Comment on "Aid to music composition employing a random probability system" // J. Acoust. Soc. America. -1962. -Vol.34. -N.1. -P.128-129.
- Barnett, 1981 Barnett J. Computational methods for a mathematical theory of evidence// Proc.7-th IJCAI.-Vancouver, 1981. -P.868-875.
- Barton, 1985 Barton G., j-r. On the complexity of ID/LP parsing// Computational Linguistics. -1985. -Vol.11. -N.4. -P.205-218.
- Barwise, 1981 Barwise J., Perry J. Semantic innocence and uncompromising situations // Midwest Studies in Philosophy, VI.-Minneapolis: University of Minnesota Press, 1981. -P.387-404.
- Barwise et al., 1983 Barwise J., Perry. J. Situations and attitudes. -Cambridge (Mass.): Bradford Books, 1983.
- Bibel 1972 Bibel W. A Deductive solution for plan generation // New Generation Computing. -1986. - Vol.4, N.2. -P.115-132. Biermann A.W., Feldman J.A. A survey of results in grammatical inference // Frontiers of Pattern Recognition. -New York: Academic Press, 1972.
- Bibel, 1983 Bibel W. Matchings in matrices // CASM.-1983.-N. 26.-P.844. - 852.
- Biermann, 1972 Biermann A.W., Feldman J.A. On the synthesis of finite-state machines from samples of their behavior// IEEE Trans.-1972.V.-SMC-21.-N. -P.592-597.
- Biermann, 1978 Biermann A.W. The inference of regular LISP Programs from Examples // IEEE Trans. -1978.-

- Blum et al.,
1975 Blum L., Blum M. Toward a mathematical theory of inductive inference // Information and Control. -1975. -N.28. -P.125-155.
- Bobrow et al.,
1969 Bobrow D.G., Fraser B. An augmented state transition network analysis procedure // Proc. Joint Conf. on Artificial Intelligence.-Bedford (Mass.): Mitre Corp., 1969.-P.557-567.
- Bobrow,
1985 Bobrow D. (ed.). Special issue on non-monotonic logic // Artif. Intell.-1980.-Vol.13,N.1-2.
Bonissone P., Tong R. Editorial: reasoning with uncertainty in expert systems // J. Man-Mach. Stud.-1985.-Vol. 22, N.3.-P.241-250.
- Bose,
1984 Bose J. Personal construct theory and the transfer of the human expertise // Proc. National Conf. Artif. Intell.-Austin, Texas, 1984.
- Boyer,
1971 Boyer R.S. Looking: A restriction of resolution: Ph. D. dissertation.-Austin, University of Texas at Austin. -1971.
- Brayer et al.,
1977 Brayer J.M., Fu K.S. A Note on the K-tail method of free grammar inference // IEEE Trans. -1977. -SMC -7. - P. 293-300.
- Bremond,
1966 Bremond C. La logique des possibles naratifs // Communication.-1966.-N.8.
- Brooks et al.,
1957 Brooks F.R., Hopkins A.L., Neumann P.G., Wright W.V. An experiment in musical composition // IRE Trans.,IX, ES-6, -1957.-N.3.-P.175-182.
- Buchanan et al.,
1978 Buchanan G.B., Feigenbaum E.A. DENDRAL and META-DENDRAL, their applications' dimensions // Artif. Intell.-1978. -N.11.-P.5-24.
- Buchanan et al.,
1984 Buchanan B., Shortliffe E. (eds.) Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project, 1984.
- Bundy,
1985 Bundy A. Incidence calculus: a mechanism for probabilistic reasoning // J. Aut. Res. -1985. -Vol.1. - N.3.- P. 263-283.
- Bundy,
1986 Bundy A. Correctness criteria of some algorithms for uncertain reasoning using incidence calculus // J. of Aut. Reas.-1986.-Vol.2.,N.1. -P.109-126.
- Burt,
1982 Burt P.J. Fast filter transforms for image processing // Computer Graph. Image Process. -1982. -Vol.16, N.1. - P.20-51.
- Burton et al.,
1979 Burton R.R., Brown J.S. Towards a natural language capability for computer-assisted instruction // Proc. Instructional Systems Development. -N. Y.: Academic Press, 1979. -P.273-313.

- Cappelli et al., 1984 Cappelli A., Ferrary G., Moretti L., Prodanof I. A framework for integrating syntax and semantics // Computational Models of Natural Language Processing. 1984.
- Carnap, 1950 Carnap R. The logical foundations of probability. -Chicago, 1950.
- Case et al., 1983 Case J., Smith C. Comparison of identifications criteria for machine inductive inference // Theor. Comput. Scien.-1983.-N.25.-P.193-220.
- Chafe, 1976 Chafe W.L. Givenness, contrastiveness, definiteness, subjects, topics, and point of view // Subject and Topic. - N.Y.: Academic Press, 1976. -P.27-55. Рус.Пер.: Чейф У. Данное, контрастивность, определенность, подлежащее, тематика и точка зрения. // Новое в зарубежной лингвистике.-М. Прогресс, 1981. -С.277-316.
- Chang et al., 1971 Chang C.L., Slagle J.R. Admissible and optimal algorithm for searching AND/OR graphs // Artif. Intel. -1971.-Vol.2.
- Chomsky, 1957 Chomsky N. Syntactic structures.- The Hague: Gravenhage, Mouton, 1957.
- Chomsky, 1970 Chomsky N. Remarks on nominalization // Readings in English Transformational Grammar.-Boston, 1970. -P.184-221.
- Church, 1936 Church A. An unsolvable problem of elementary number theory // Amer.J.Math.-1936. -58. -P. 345-363.
- Civanlar et al., 1986 Civanlar M.R. Trussel H.J. Constructing membership functions using statistical data// Fuzzy Sets and Systems. -1986. -Vol.18, N.1. -P.32-40.
- Clema et al., 1985 Clema J.K., Werling R. Expert systems for real time applications, NAECON 1985 // IEEE.-1985. National Aerospace and Electronic Conference. -P. 1322-1329.
- Clocksinn et al., 1981 Clocksin W.F., Mellish C.S. Programming in Prolog. -N. Y.: Springer Verlag, 1981.- 279 p.
- Codd, 1970 Codd E.F. Relationship model of data for large shared data banks // CACM.- 1970.-Vol.13, N.6. -P.377-387.
- Codd, 1972 Codd E.F. Relationship completeness of database sublanguages // Data base systems. -N.Y.: Prentice Hall, 1972.
- Codd, 1979 Codd E.F. Extending the database relational model to capture more meaning // ACM Trans. on Database systems. -1979.-Vol.4, N.4.-P.397-434.
- Codd, Codd E.F. Data models in data base management //

- 1981 Proc. Workshop on Data Abstraction, Data Bases and Conceptual Modelling. SIGPLAN Notes.-1981. -Vol.16, N.1.-P.112-114.
- Cohen, 1977 Cohen B.L. A powerful and efficient structural pattern recognition systems // Artif. Intel. -1977.- Vol.9,N.3.
- Cohen, 1985 Cohen B.L. Heuristic reasoning about uncertainty: an artificial intelligence approach, N.Y.-1985.
- Colmerauer, 1971 Colmerauer A. Les systems-Q ou un formalisme pour analyser et synthetiser des phrases sur ordinateur // TADM 71.-Univ. Montreal, 1971. -P. 1-45.
- Corlett et al., 1986 Corlett R., Todd S. A Monte-Carlo approach to uncertain inference // Artif. Intel. and its Applications.-1986. -P. 127-137.
- Cook et al., 1976 Cook C.M., Rosenfeld A., Aronson A.R. Grammatical inference by hill-climbing // Information Science. -1976. -N.10.-P.59-80.
- Daley, 1977 Daley R. On the inference of optimal descriptions // Theor. Comput. Scien.-1977.-N.4.-P. 301-319.
- Daley et al., 1983 Daley R., Smith. C. On the complexity of inductive inference (Tech. Rep. N.83-4).- Dept. Comput. Scien., University of Pittsburgh, 1983.
- Davis et al., 1960 Davis M., Putnam H. A computing procedure for quantification theory // J. Assoc. Comput. Mach.-1960.- 7.- P.201-215.
- Davis et al., 1975 Davis R., King J. An overview of production systems, Stanford AI Lab. Memo AIM-271, CSD Rep. NSTAN-CS-75-524, Stanford University, 1975.-38 p.
- Davis et al., 1982 Davis R., Lenat D. Knowledge-Based systems in artificial intelligence. - N. Y.: McGraw-Hill, 1982.
- Dempster, 1967 Dempster A. Upper and lower probabilities induced by a multivalued mapping // Ann. Math. Stat.-1967. -Vol. 38.- P.325-339.
- Diederich et al., 1987 Diederich J., Ruhman I., May.M. KRITON: a Knowledge-acquisition tool for expert systems // Int. J. Man-Machine Studies. -1987.- Vol. 26,N. 1.-P.29-40.
- Dietterich, 1979 Dietterich T.G. The methodology of knowledge layers for inducing descriptions of sequentially ordered events. Master's thesis. - Urbana: Univ. of Illinois, 1979.
- Dietterich, 1981 Dietterich T.G., Michalski R.S. Inductive learning of structural descriptions: evaluation criteria and comparative review of selective methods // Artif. Intel. -1981.- 16 P.257-294.

- Dyer
1981 Dyer M.G. Integration, unification, reconstruction, modification: an eternal parsing braid // Proc. 7-th IJCAI.-1981.-P.37-42.
- Dyer,
1982 Dyer M.G. In-depth understanding // Research Report 219.-Yale Univ. Dep. Comp. Science, 1982.
- Escher,
1959 Escher M.C. Grafiek en Tekeningen. - Zwolle (Holl.), 1959.
- Feigenbaum,
1971 Feigenbaum E. On generality and problem solving: a case study using the DENDRAL program // Machine Intell.- Edinburgh Univ Press. -1971.-N.6. -P.181-191.
- Ferrary,
1986 Ferrary G. A Two-level dialogue system // Proc. 11-th International Conference on Computational Linguistics (COLING-86). -Munich, 1986. -P.42-45.
- Fikes et al.,
1971 Fikes R.E., Nilsson N.J. STRIPS: a new approach to the Application of the theorem proving to problem solving // Proc. 2nd Inter. Conf. on Artif. Intell. -London, 1971. -P.608-619.
Рус. пер.: Файкс Р., Нильсон Н. система СТРИПС - новый подход к применению методов доказательств в теореме при решении задач // Сб. Интегральные роботы.-М.: Мир, 1973. -С.382-403.
- Fikes et al.,
1972 Fikes R.E., Hart P.E., Nilsson N.J. Some new directions in robot problem solving // Machine Intelligence.-Edinburgh Univ. Press, 1972. -Vol. 7.-P. 405-430.
Рус.пер.: Файкс Р., Нильсон Н. Новые направления в автоматическом решении задач роботом. -М.: Мир, 1975. -С. 406-439.
- Fillmore,
1968 Fillmore C.J. The case for case // Universals in Linguistic Theory. N. Y. : Holt, Rinehart and Winston. -1968.
- Fillmore,
1971 Fillmore C.J. On a fully developed system of linguistic description // Feasibility Study of Fully Automatic High Quality Translation. -Tech. Report RADCO- TR-71-295.-Austin: Univ. of Texas.-1971. - P.77-94.
- Fillmore,
1977 Fillmore C.J. The case for case reopened // Syntax and Semantics. Vol. 8. Grammatical Relations.-N. Y.: Academic Press, 1977.
- Fountain,
1986 Fountain T.J. Array architecture for iconic and symbolic image processing // Proc. 8-th Int. Conf. on Pattern Recognition.- IEEE Computer Soc. Press, 1986.-. 24-33.
- Freivald et al.,
1979 Freivald R.V., Wichagen R. Inductive inference with additional information // Elekt. Information verarb. Kybern.-1979.-N.15.-P.179-185.
- Fu,
1979 Fu K.S. Recent advances in syntactic pattern recognition // Proc. 4-th Joint Conf. Pattern

- Recognition, Kyoto, 1978. - N. Y., 1979.-P.13-18.
- Fum et al., 1984 Fum D., Tasso C. A propositional language for text Representation // Computational Models of Natural Language Processing.-N. Y.: Elsevier Science Publishing Comp.(North Holland), 1984.-P.121-150.
- Gagliardo, 1979 Gagliardo E. Introduzione all' edizione italiana // Zaripov R.Ch. Musica con il calcolatore.. Padova: Muzzio, 1979. -P. 9-18.
- Galil, 1975 Galil Z. The complexity of resolution procedures for theorem proving in the propositional calculus.- Tech. Report 75-239.- Dept. of Comp. Scien., Cornell Univ., Ithaca, 1975.
- Gallaire et al., 1978 Gallaire H., Minker J. (eds.) Logic and data bases. N. Y.: Plenum Press, 1978. -458 p.
- Gasdar, 1982 Gasdar G. Phrase Structure Grammar// The nature of syntactic representation .-Dordrecht: D.Reidel.- 1982.
- Gasdar, 1983 Gasdar G. Phrase structure grammars and natural languages // Proc. IJCAI.-1983.-P.556-565.
- Gill, 1963 Gill S. A technique for the composition of music in a computer // Comput. J. -1963.-6.-N.2. -P.129-133.
- Gillmore, 1960 Gillmore P.C. A proof method for quantification theory: its justification and realisation // IBM J. Res. Development. - 1960. - P.28-35.
- Gold, 1967 Gold E.M. Language identification in the limit// Information and Control.-1967.-N.10.-P.447-474.
- Gold, 1978 Gold .M. Complexity of automation identification from given data// Information and Control.-1978. -N.37.-P.302-320.
- Goodman, 1987 Goodman B.A. Reference identification and reference identification failures // Computational Linguistics -1987.-Vol.12,N 4. Рус.пер.: Гудман Б.А. Идентификация референта и связанные с ней коммуникативные неудачи//Новое в зарубежной лингвистике: Вып.XXIV. Компьютерная лингвистика.-М.: Прогресс, 1989.- С.209-259.
- Gordon et al., 1985 Gordon J., Shortliffe E. A method for managing evidential reasoning in a hierarchical hypothesis Space // Artif. Intell.-1985.-Vol. 26, N.3. -P.323-357.
- Granlund et al., 1983 Granlund G.H., Knuysson H. Contrast of structured and homogeneous representation // Physical and Biological Processing of Images.-Berlin: Springer, 1983.
- Granlund et al., Granlund G.H., Arvidsson J.B. Computer architec-

- 1985 tures for image processing // Proc. 4-th Scandinavian Conf. on Image Analysis, Trondheim, 1985.
- Green,
1969 Green C. Theorem proving by resolution as a basis for question-answering systems // Mach. Intell. - V.4. - 1969. - P.183-205.
- Grice,
1973 Grice H.P. Logic and conversation // Syntax and Semantics, 3. Speech Acts. -N.Y., 1973.
- Grinberger
et al., 1968 Grinberger M., Jones M. On-line incremental Simulation // Simulation programming languages. Proc. IFIP Working Conf.-Amsterdam: North Holland, 1968. -P.13-30.
- Grishman,
1973 Grishman R. Implementation of the string parser of English // Natural Language Processing. Courant Computer Science Symposium 8.-N.Y.: Algorithmics Press, 1973.- P.89-110.
- Grosz,
1977 Grosz B.J. The representation and use of focus in a system for understanding dialogues // Proc. 5-th IJCAI. Cambridge, Ma., 1977.-P.67-76.
- Grosz,
1980 Grosz B.J. Utterance and objective: issues in natural language communication // AI Magazine. -1980.-Vol.1.-P.11-20.
- Guida et al.,
1975 Guida G., Mandrioli D., Paci A., Somalvico M. A formal framework for an unitary approach to the theory of problem-solving // Proc. IIASA Workshop on AI Question-Answering Systems. -Luxemburg, 1975.-P.140-172.
- Guida et al.,
1976 Guida G., Somalvico M. Semantics in problem representation and search // Infor. Proc. Letters. -1976. -Vol.5, N.4.
- Gunzenhauser
1963 Gunzenhauser R. Zur Synthese von Texten mit Hilfe Programmin - gesterwriten Zifferrechen - anlagen. // Mathematiktechnik, Wirtschaft, H.1., 1963.
- Haase-Rapoport
et al., 1984 Haase-Rapoport G., Pospelov D.A., Semenova Y.T. Generation of fairy tales // Computers and Artificial Intelligence. -Vol.3, 1984.-P.281-304.
- Hajek et al.,
1982 Hajek P. Havranek T. GUHA-80 -An application of artificial intelligence to data analysis // Computers and Artificial Intelligence. -1982. - Vol.1, N.2. -P. 107-134.
- Haralick,
1977 Haralick R.M. The pattern recognition, arrangement, and theory of covers // IEEE Conf. on Image Processing.-Troy, 1977. IEEE, N.4.
- Havass,
1964 Havass M. A simulation of musical composition synthetically composed folk-music // Computational Linguistics 3.-Budapest, 1964.-P. 107-127.

- Havranek et al., 1977 Havranek T., Chyba M., Pokorný D. Processing sociological data by the GUHA-method // Int. J. Man-Machine Studies.-1977.-Vol. 9. -P.439-447.
- Hayes-Roth et al., 1978 Hayes-Roth F., McDermot J. An interference matching technique for inducing abstractions // Communications of the ACM.-1978.-N.21.-401-411.
- Hendrix, 1975 Hendrix G. Expanding the utility of semantic networks through partitioning // Proc. IJCAI-IV.-1975.-Vol.1.-P.115-121.
- Herbrand, 1930 Herbrand J. Recherches sur la theorie de la demonstration // Travaux de la Societe des Sciences et des Lettres de Varsovie. -1930.- N. 33. -P.128.
- Hewitt, 1971 Hewitt C. PLANNER: A language for manipulating models and proving theorems in a robot // Memo 68, AI Lab., MIT, Cambridge, Mass., 1971.
- Higginbotham, 1984 Higginbotham J. English is not a context-free language // Linguistic Inquiry.-1984.-Vol. 15.-P. 119-126.
- Hiller, 1957 Hiller L.A., Isaacson L.M. Illiac suite for string quartet // New Music Edition 30.-1957. -N.3.-P.18.
- Hiller, 1959 Hiller L.A., Isaacson L.M. Experimental music. N. Y.: McGraw-Hill Book Company, 1959.-197 p.
- Hiller, 1970 Hiller L.A. Music composed with computer. A historical survey // The computer and music. - Ithaca and London: Cornell Univ. Press, 1970. -P.42-96.
- Hintikka, 1965 Hintikka J. Towards a theory of inductive generalization // Proc. intern. Congr. of Logic, Methodology and Philosophy of Science.- Amsterdam, 1965. -P. 274-283.
- Hirst, 1981 Hirst G. Anaphora in natural language understanding: a survey // Lecture notes in Computer Science. -119.- Berlin: Springer-Verlag, 1981.
- Hisdal, 1986 Hisdal E. Infinite - valued logic based on two-valued logic and probability. Part 1.1. Difficulties with present-day fuzzy-set theory and their resolution in the TEE model // Int. J. Man - Machine Studies. -1986.-Vol. 25, N.1 - P.89-111.
- Horning, 1969 Horning J.J. A study of agrammatical inference: Ph. D. Dissertation, Computer Science Dept., Stanford Univ., Stanford, Calif., 1969.
- Hubel et al., 1962 Hubel D.H., Wiesel T.N. Receptive fields, binocular interaction and function and functional structure in cat's visual cortex // J. Physiol.-1962. -Vol.160, N. 1. -P.106-154.

- Jacoby, 1982 Jacoby B.E. On Database logics // J ACM.-1982. -Vol.29, N. 2.-P.310-332.
- Jantke, 1978 Jantke K.P. Universal methods for identification of total recursive functions // Rostocker Math. Kolloquium .-1978. N.10.-P.63-69.
- Jantke, 1986 Jantke K.P. An algebraic framework for inductive program synthesis // Lecture Notes in Computer Science.-1986.-Vol.215.P.101-122.
- Jantke et al., 1981 Jantke K.P., Beick H.R. Combining postulates of naturalness in inductive inference // Electron. Informations-Verarb. Kybern.-1981.-N.17.-P. 465-484.
- Johnson, 1984 Johnson F. Fun and games. // IBM Personal Computer Textbook.- Berokly: California Press,1984.
- Jounaud et al., 1977 Jounaud J.R., Guiho G. Program synthesis from examples for a simple class of non-loop functions.//Technical rep., Lab. de Resherche en Informatique , Univ. Paris-Sud, Paris, 1977.
- Kahn et al., 1984 Kahn G., McDermott J. The MUD system // Proc. IEEE Conf. AI,Appl, Denver,Cal, 1984.
- Kahn et al., 1985 Kahn G. Nowlan S., McDermott J. Strategies for knowledge acquisition // IEEE Trans. Pattern Analisis and Machine Intelligence.- 1985.-Vol. 7., N.5.
- Kanade, 1981 Kanade T. Recovery of the three-dimensional shape of an object from a single view // Artif.Intell.- 1981.- Vol. 17.-P.409-460.
- Kanal, 1986 Kanal L., Lemmer J. (eds.) Uncertainty in artificial intelligence. - 1986.
- Kanovich, 1987 Kanovich M.I. Lossless calculi as a tool to reduce the search for analysis-and-synthesis algorithms in the knowledge base systems // 8 Международный конгресс по логике, методологии и философии науки, М.: Наука, 1987.-Том 1.-141-142.
- Kaplan et al., 1982 Kapan R., Bresnan J. Lexical-functional grammar: a formal system for grammatical representation // the Mental Representation of Grammatical Relations.- Cambridge, Mass.: MIT Press,1982.
- Katz et al., 1964 Katz J., Fodor J. (eds.) The structure of a semantic theory // The Structure of Language. -N.Y.: Prentice-Hall, Englewood, Cliffs, 1964.
- Kelly, 1955 Kelly G. The psychology of personal constructs. -N.Y.: Norton,1955.
- Kender, 1980 Kender J.R. Shape from texture // Doctoral Dissertation.-Carnegie-Mellon Univ., Computer Science Dep. -1980.

- Kickert, 1978 Kickert W.J.M. Fuzzy theories on decision - making (a critical review).-London, 1978.-182 p.
- Kim et al., 1983 Kim J., Pearl J. A computational model for combined counsel and diagnostic reasoning in inference systems // Proc. 8-th IJCAI.-Karlsruhe, 1983.-P.190-193.
- Klein et al., 1956 Klein M., Bolitho D. Syncoption by automation // Pasadena, Cal.: Burroughs Corporation, 1956.-Aug.-P.2-3.
- Klette et al., 1980 Klette R., Wiehagen R. Research in the theory of inductive inference by GDR mathematicians - a survey. // Information Science -1980.-N.22.-P.149-169.
- Kodratoff, 1979 Kodratoff Y. A class of functioned synthesized from a finite umber of examples and a LISP program scheme // Int.J Comp. and Inf. Science. -1979. -N.8. -P.489-521.
- Konolige, 1985 Konolige K. A deduction mode of belief. -N.-Y., 1985.
- Kowalsky et al., 1971 Kowalski R., Kuehner D. Linear resolution with selection function // Artif. Intell.-1971.-2 -P.227-260.
- Kowalsky, 1974 Kowalski R. Predicate logic as programming language // Proc. IFIP Congress. - 1974. - P. 569-574.
- Kowalsky, 1979 Kowalski R. Algorithm = Logic + control //ACM . -1979.- Vol. 22, N.7.-P.424-436.
- Krippendorf 1980 Krippendorf K. Content analysis. An introduction to its methodology.-Beverly Hills: Sage, 1980.
- Kugel, 1977 Kugel P. Induction, pure and simple // Information and Control. -1977.-N.35.-P.276-337.
- Kuno et al., 1963 Kuno S., Oettinger A.G. Multipe syntactic analyzer Information Processing -62- Amsterdam: North-Holland Publishing Co., 1963. -128-133.
- Kupper, 1967 Kupper H. Computer und Musikwissenschaft //IBM Nachr.-1967.-Vol.17, N.180.-S.297-303.
- Kyburg, 1987 Kyburg H. Bayesian and non-Bayesian evidential updating // Artif. Intell. -1987. -Vol.31, N.3. -P.271-293.
- Lange, 1986 Lange S. A program syntheses algorithm exemplified // Lecture Notes in Computer Science.-1986.-P.185-193.
- Lehnert et al., 1981a Lehnert W.G., Black J.B., Reiser B.J. Summarizing narratives// Proc. 7-th IJCAI.-Vancouver, 1981. -P.184-189.

- Lehnert et al.,
1981b Lehnert W.G. Dyer M.G., Johnson P.N., Yang C.J.,
Harley S. BORIS - an experiment in in-depth
understanding of narratives // Artif. Intell.
-1981.-Vol. 20,N.1.-P.15-62.
- Lenat,
1976 Lenat D. AM: an artificial intelligence approach
to discovery in mathematics as heuristic search
// Comp. Science Dept. Rep. STAN-CS-76-570.
Stanford Univ. -Stanford, Calif.,1976.
- Lenat et al.,
1978 Lenat D.,Harris G. Designing a rule system that
searches for scientific discovery // Pattern
Directed Inference Systems.-N.Y.: Academic
Press,1978.-P.25-51.
- Lenat,
1984 Lenat D. Automated theorem formation in
mathematics // Contemporary math.-1984. -
N.29.-P.287-314.
- Lettvin et al.,
1959 Lettvin J.Y.et al .What the frog's eye tells the
frog's brain // Proc.IRE.-1959.-Vol.47
-P.1940-1951.
- Levi et al.,
1976 Levi G., Sirovich F. Generalized AND/OR Graphs //
Artif. Intell.-1976.-Vol. 7.-P.243-259.
- Levine,
1982 Levine B. The use of free derivatives and a
sample support parameter for inferring tree
systems // IEEE Trans. Pattern Anal. Mach.
Intel.-1982.-P.25-34.
- Lincoln,
1970 Lincoln H.B. (ed.) The computer and music.
-Ithaca, London: Corn. Univ. Press, 1970. -354 p.
- Loveland,
1968 Loveland D.W. Mechanical theorem proving by model
elimination // J. Assoc. Comput. Mach.
-1968.-15.-P.236-251.
- Loveland,
1970 Loveland D.W. A linear format for resolution //
Proc. IRIA Symp. Automatic Demonstration,
Versailles, France, 1968. - N.Y.: Springer,
1970.-P.147-162.
- Luckham,
1970 Luckham D. Refinement theorems in resolution
theory // Proc. IRIA Symp. Automatic
Demonstration, Versailles, France, 1968.- N.Y.:
Springer, 1970.-P.163-190.
- Ludvova,
1975 Ludvova J. Matematicke metody v hudebni analyze.
K muzikologicke aplikaci teorie informace a
teorie mmozin. -Praha: Supraphon,1975.-136 o.
- Lutz,
1960 Lutz Th. Veber ein Programm zur Erzeugung
stochastischelogsitistischenTexten//
Grundlagenstudium aus Kybernetik und
Geistwissenschaft, Heft1,1960. -S.11-16.
- Maier,
1983 Maier D. The theory of relational Databases.
Computer Science Press,1983.-637 p.

- Masnio et al., 1983 Masnio S., McDermott Y., Sobel A. Decision-making in time-critical situation //Proc. of the 8th IJCAI.-1983.-Vol.1.-P.233-235.
- McCluskey, 1956 McCluskey E.J. Minimization of boolean functions // BellSyst.Techn. J.-1956.-35.-P.1417-1444.
- McDermott et al., 1972 McDermott D.V., Sussman G.J. The CONNIVER // Reference Manual MIT AI, Memo N.259.-Cambridge, Mass., 1972.-P.91.
- McKeown, 1985 McKeown K.R. Text generation.-Cambridge, Ma: Cambridge Univ. Press, 1985.-P.250.
- Meehan, 1980 Meehan J.R. The meta novel. Writing stories by computer.-N.Y., L.:Garland, 1980.-P.119-171. (Outstanding diss. on computer science).
- Michalsky 1973 Michalsky R.S. AQVAL/1 - Computer implementation of a variable valued logic system VL-1 and examples of its application to pattern recognition //Proc. of the First Int. Joint Conf.on Pattern Recognition.-Washington:D.C., 1973.-P.3-17.
- Michalsky et al., 1977 Michalski R.S., Negri P. An experiment on inductive learning in chess and games // Machine Intelligence. -1977.-N.8.-P.175-192.
- Michalsky, 1980 Michalski R.S. Pattern recognition as rule-guided inductive inference //IEEE Trans. on Pattern Analysis and Machine Intelligence.-Vol. PAMI-2, N.4. -1980.-P.349-361.
- Michalsky et al., 1981 Michalski R.S., Stepp R., Diday E. A recent advance in data analysis: clustering objects into classes characterizes by conjunctive concepts //Progress in Pattern Recognition. Vol.1. -Amsterdam : North-Holland, 1981.
- Michalsky, 1983 Michalski R.A. A theory and methodology of inductive learning // Artif. Intell. -1983. -Vol.20., N-2.-P.11-161.
- Michalsky et al., 1983 Michalski R.S., Carbonell J.G., Mitchell T.M. Machine learning. An Artificial Intelligence Approach // Palo Alto:Tiada Publishing Co., 1983. - 572 p.
- Minsky, 1974 Minsky M. A framework for representation //AI Memo N.306, MIT Cambridge, 1974.-82p. Рус.пер: Минский М. Фреймы для представления знаний.-М.:Энергия, 1979.-151 с.
- Minton, 1985 Minton S. Selectively generalizing plans for problem-solving // Proc. 9th IJCAI.-1985. Vol.1. -P.596-599.
- Montague, 1970a Montague R. Universal grammar //Theoria. 36. -Part 3. 1970.-P.373-398.
- Montague, Montague R. English as a formal language //

- 1970b Visentini et al (eds). Ling. nella Societae nella Teonica. Milan,1970.
- Montague, 1974 Montague R. The proper treatment of quantification in ordinary English // Thomason R.H. (ed.) Formal Phisophy.- New Haven: Yale Univ. Press, 1974. -P.247-270.
- Montgomery, 1972 Montgomery C.A. Linguistics and Information Science // J. ASIS.-1972.-Vol.23,N.-3.-P.195-219.
- Morgan, 1971 Morgan C.G. Hypotheses generation by machine // Artif. Intell.-1971.- Vol. 2.-P.179-187.
- Morik, 1987 Morik K. Acquiring domain models // Int.J.Man-Machine Studies. -1987.-Vol. 26,N.1.-P.93-104.
- Mostovsky, 1957 Mostovsky A. On a generalization of Quantifiers // Fundamenta Mathematica. -1957. -Vol.44. -P.12-36.
- Multyresolution, 1984 Multiresolution image processing and analysis/ Rosenfeld A., (ed.). Berlin: Springer, 1984.
- Narin'yan, 1984 Narin'yan A.S. Towards an integral model of language competence // Computational Models of Natural Language Processing.- Amsterdam: North-Holland, 1984. -P.275-295.
- Newell et al., 1961 Newell A., Simon H.A. GPS -a program that simulated human thought//Lernende Automaten.-Oldenburg, Munchen,1961.S.109-124. Рус. пер.: Ньюэлл А., Саймон Г. GPS - программа, моделирующая процесс человеческого мышления // Вычислительные машины и мышление. -М.:Мир, 1967. -С.283-301.
- Newell et al., 1972 Newell A., Simon H.A. Human Problem Solving. -Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1972.
- Nguyen et al., 1985 Nguyen H., Goodman I. Uncertainty models of knowledge-based systems . -1985.
- Nilsson, 1986 Nilsson N. Probabilistic logic // Artif. Intell. -1986.-Vol.28,N.1.-P.71-87.
- Nix, 1983 Nix R. Editing by Examples.Ph.D.dissertation, Computer Science Dep.,Yale University.-New Haven, Connecticut,1983.
- Norwich et al., 1984 Norwich A.M.,Turksen J.W. A model for theme asurement of membership and the consequences of its empirical implementation // Fussy Sets and Systems.-1984.-Vol 12.
- Obermeier et al., 1985 Obermeier K.K., de Hilster D. DIID - data-independant interface for data-bases: the AI perspective. // SPIE.-Vol.548.Applications of Artificial Intelligence 11. -1985.-P.220-226.
- Olson et al., Olson H.F., Belar H. Aid to music composition

- 1961 employing a random probability system // J. Acoust. Soc. America. -Vol.33, N.9. -1961. -P. 1163-1170.
- Osherson et al., 1982 Osherson D., Wienstein S. Criteria of language learning // Information and Control.-1982.-N.52. -P.123-138.
- Osipov et al., 1987 Osipov G.S., Churchev E.P., Golubev S.A., Gukova S.M., Komarov S.I. Expert systems tools for badly structured fields // Artif. Intell. and Information Control Systems of Robots.- Amsterdam: North-Holland, 1987.-P.393-397.
- Padberg, 1964 Padberg M.H.A. Computer-composed canon and free fugue. Diss. abstr.-St. Louis Univ.-1964.-26 p.
- PDIS, 1978 PDIS - Pattern-directed inference systems.-N.Y.: Academic Press, 1978.-658p.
- Pearl, 1986 Pearl J. Fusion, propagation, and structuring in belief networks // Artif. Intell.-1986.-Vol.29, N.3.-P.241-288.
- Petriok, 1981 Petriok S. Field-testing the transformational question answering (TQA) system // Proc. 19th Annual Conf. ACL, 1981.-P.35-36.
- Plaisted, 1981 Plaisted D.A. Theorem proving with abstraction // Artif. Intell. -1981.-Vol.16.-P.47-108.
- Plotkin, 1970 Plotkin G.D. A note on inductive generalization // Machine Intell.- 1970.-N.5.-P.153-163.
- Plotkin, 1971 Plotkin G.D. A further note on inductive generalization // Machine Intell.- 1971.- Vol. 6.-P.101-124.
- Pocorny, 1980 Pocorny D. Knowledge acquisition by the GUHA method // Int. J. Policy Anal.Inform. Syst.-1980.-N.4. -P.379-399.
- Pospelov, 1986 Pospelov D. Models of human communication: dialogue with computer //Int.J. General Systems. Methodology Applications. Education.-1986.-Vol.12, N.4.- P.333-338.
- Postal et al., 1984 Postal P.M. Langendoen D.T. English and the class of context-free languages.// Computational Linguistics. -1984.Vol.10, N.3-4.-P.177-181.
- Prade, 1985 Prade H. A computational approach to approximate and plausible reasoning with applications to expert systems // IEEE Trans. on Pattern Analysis and Mach. Intell. -1985. -Vol. PAMI-7, N.2. -P.260-283.
- Prior, 1984 Prior A.N. Time and modality.-Oxford,1957. Pullum G.K. On two recent attempts show that English is

- not a CFL // Computational Linguistics -1984.
-Vol.10, N.3-4.-P.182-186.
- Quillian, 1968 Quillian M.R. Semantic memory // Semantic Information Proc.- Cambridge, Mass.: MIT Press, 1968.-P.227-270.
- Quinlan, 1979 Quinlan J.R. Discovery rules from large collections of examples: a case study // Expert Systems in the Micro Electronic Age.- Edinburgh: Edinburgh Univ. Press, 1979.
- Quinlan, 1983 Quinlan J.R. INFERNO: a causations approach to uncertain inference // The computer J.-1983. -Vol.26,N.3.-P.255-269.
- Quinlan, 1985 Quinlan J. Internal consistency in plausible reasoning systems // New Generation Computing. -1985. -Vol.3,N.2.-P.157-180.
- Rabin, 1974 Rabin M.O. Theoretical impediments to artificial intelligence // Proc.IFIP Cong.-Amsterdam: North-Holland, 1974.-P.615-619.
- Ragade et al., 1977 Ragade R.K. Gupta M.M. Fuzzy sets theory: introduction // Fuzzy automata and decision process.- Amsterdam: North-Holland,1977.- P - 105-131.
- Reuter, 1978 Reiter R.On closed world data bases // Logic and Data Bases.-N.Y.:Plenum Press,1978.-P.55-76.
- Reynolds, 1970 Reynolds J.C. Transformational systems and the algebraic structure of atomic formulas // Machine Intelligence-1970.-N.5.-P.135-151.
- Reibeck, 1975 Reisbeck C.K. Conceptual analysis // Conceptual Information Processing.-Amsterdam: North-Holland, 1975.
- Robinson, 1965 Robinson J.A. A machine oriented logic based on the resolution principle // J. ACM.-1965. -N.12.-P.23-41.
Рус.пер.: Машинно-ориентированная логика, основанная на принципе резолюций. Кибернетический сборник, N.7, нов. серия.-М.Мир,1970.
- Rosen, 1967 Rosen B.K. Context-sensitive syntax analysis // Mathematical Linguistics and Automatic Translation. -Rep.NSF-18.-Cambridge, Mass., 1967.-P.1-46.
- Rosenthal et al., 1984 Rosenthal D.A., Bajacsy R. Visual and conceptual hierarchy: a paradigm for studies of recognition strategies// IEEE Trans.-1984.-Vol.PAMI-6, N.13.-P.319-325.
- Saaty, 1974 Saaty T.L. Measurement the fuzziness of sets// J.of Cybernetics.-1974.-N.4.- P. 53-61.
- Sacerdoti, Sacerdoti E.D.Planning in a hierarchy of

- 1974 abstraction spaces //Artif. Intell.-1974.-Vol.5.
-P.115-135.
- Sager, Sager N. Syntactic Analysis of Natural Language
1967 // Advances in Computers.-N.Y.: Academic Press,
1967.-Vol.8.-P.153-188.
- Sager, Sager N. The string parser for scientific
1973 literature // Natural Language Processing.
Courant Computer Science Symp.8.-N.Y.:
Algorithmic Press, 1973.-P.61-88.
- Sager, Sager N. Natural language information processing:
1981 a computer grammar of English and its applica-
tions. Reading, Mass.: Addison-Wesley, 1981.
- Salzberg, Salzberg S. Heuristics for inductive learning //
1985 Proc. 9th IJCAI.-1985.-Vol.1.-P.603-609.
- Samuel, Samuel A.L. Some studies in machine learning
1959 using the game of checkers//IBM J. Res.
Development. -1959.-Vol.3,N.3.
- Samuel, Samuel A.L. Some studies in machine learning
1967 using the game of checkers II // IBM J. Res.
Development.-1967.-Vol.11,N.6.
- Sanchez, Sanchez E. Inverses of fuzzy relations.
1979 Application to possibility distribution and
medical diagnosis // Fuzzy Sets and Systems.
-1979.-Vol.2.
- Scala, Scala H.J. On many-valued logics, fuzzy sets and
1978 systems. -1978.-Vol.1.
- Schank, Schank R. Conceptual information processing.
1975 -Amsterdam: North-Holland Publ.Co., 1975.
Рус. пер.: Шенк Р. Обработка концептуальной ин-
формации. М.:Энергия, 1980.-360 с.
- Schank et al., Schank R.C., Abelson R.P. Scripts, plans and
1975 knowledge // Advanced Papers of the 4th
IJCAI.Tbilisi.-1975. -Vol.1. -P.151-157.
- Schank, Schank R.C. Dynamic memory. -Cambridge,Mass.:
1982 Cambridge Univ.Press, 1982.
- Schubert, Schubert L.K. Extending the expressive power of
1975 semantic networks // Proc. 4-th IJCAI, 1975.-V.1.
-P. 158-164.
- Searl, Searl J.R. Indirect speech acts // Syntax and
1973 semantics.-N.Y.: Academic Press.-1973.-V.3.
-P.59-82.
- Selfridge et al., Selfridge O G , Neisser V. Pattern recognition by
1960 machine // Scientific American.-1960.-Vol.203,
N.3.-P.60-68.
- Senko, Senko M.E. A query-maintenance for data indepen-
1980 dent accessing model 11 // Inf. System.-1980.

- Serra,
1982 Serra J. Image analysis and mathematical morphology. -N.Y.: Academic Press, 1982.-687p.
- Shafer,
1976 Shafer G. A mathematical theory of evidence. N.Y., 1976.
- Shafer et al.,
1985 Shafer G., Tversky A. Languages and Designs for probability in judgments // Cognitive Science. -1985.-Vol.9, N.3.-P.309-399.
- Shannon,
1950, 1953 Shannon C.E. Computer and creativity // Proc. First London Symp. on Information Theory. -London, 1950 (Reprinted, 1953).
- Shapiro,
1981 Shapiro E. Inductive inference of theories from facts // Tech. Rep. 192 Dept. Comput. Scien., Yale Univ. -New Haven, Conn. -1981.
- Shaw,
1985 Shaw D.E. Non-von's applicability to three AI tasks areas // Proc. 9th IJCAI. -1985. -Vol.1. -P.61-72.
- Shenoy et al.,
1986 Shenoy P., Shafer G. Preparing belief functions with local computations // IEEE Expert. -1986. -Vol.1, N.3.-P.43-52.
- Shinohara,
1982 Shinohara T. Polynomial time inference of extended regular pattern languages // Lecture Notes in Computer Science. -1982.-N.147.-P.115-127.
- Shinohara,
1986 Shinohara T. Some problems on inductive inference from positive data // Lecture Notes in Computer Sci. -1986.-N.215.-P.41-58.
- Shortliffe
et al., 1975 Shortliffe E., Buchanan B. A model of inexact reasoning in medicine // Math. Biosciences. -1975.-Vol.23.-P.351-379.
- Science,
1986 Skienice C. An overview of machine learning in computer chess // ICCA J. -1986.-N.1.-P.20-28.
- Slagle,
1967 Slagle J.R. Automatic theorem proving with renamable and semantic resolution // J. Assoc. Comput. Mach. -1967.-N.14.-P.687-697.
- Sloan et al.,
1979 Sloan K.R., Tanimoto S.L. Progressive refinement of raster images // IEEE Trans. in Comput. -1979. -Vol. C-28, N.11.-P.871-875.
- Smith,
1982 Smith D.R. A survey of the synthesis of LISP programs from examples // Automatic Program Construction Techniques. N.Y.: MacMillan, 1982.
- Smith,
1985 Smith E. Cognitive psychology (Correspondent's report) // Artif. Intell. 1985.-N.25.-P.247-253.
- Solomonoff,
1964 Solomonoff R.J. A formal theory of inductive inference // Information and Control. -1964.-N.7. -P.1-22, 224-254.

- Soloway et al., 1977 Soloway E.M., Riseman E.M. Levels of pattern descriptions in learning // Papers 5th IJCAI. -Cambridge, Mass.-1977.-P.801-811.
- Sorrels, 1985 Sorrels M.E. A time-constrained inference strategy for real-time expert systems // NAECON-85 (1985 IEEE National Aerospace and Electronic Conference).-P.1336-1341.
- Sperry, 1973 Sperry R. Lateral specialization of cerebral function in the surgically separated hemispheres // The Psychology of Thinking.-N.Y., 1973.-P.126-141.
- Stefic, 1985 Stefic M. Strategic computing at DARPA: overview and assessment // ACM.-1985.-Vol.28, N.7. -P. 690-704.
- Stickel, 1967 Stickel G. Monte-Carlo text // Kunst aus dem Computer. -Stuttgart: Verlag Neelosky, 1967.
- Stolfo et al., 1982 Stolfo S.Y., Shaw D.E. DADO: a tree-structured machine architecture for Production Systems // Proc.2nd Conf.on the American Association for AI.-1982.-Aug.-P.242-246.
- Su et al., 1969 Su S.Y.W., Harper K.E. A detected random paragraph generation// Int. Conf. Comput. Linguistics (COLING), Sanga-Saby, 1969.-Preprint, 1969.-N.13.
- Summers, 1977 Summers P.D. A methodology for LISP program construction from examples // J.ACM.-1977.-N.24.-P.161-175.
- Tanimoto et al., 1975 Tanimoto S.L., Pavlidis T. A hierarchical data structure for picture processing // Comput.Graph. Image Process.-1975.-Vol.4, N.1.-P.104-119.
- Thole et al., 1979 Thole U., Zimmerman H.J., Zysno P. On the operations for the intersection of fuzzy sets // Fuzzy Sets and Systems.-1979.-Vol.2.-P.102-124.
- Thomason, 1980 Thomason R.H. A model Theory for Propositional Attributes // LTR.-1980.-N.4.-P.47-70.
- Tong et al., 1985 Tong R., Shapiro D. Experimental investigations of uncertainty in a rule-based system for information retrieval // Int. J. Man-Mach. Studies.-1985.-Vol.22, N.-3. -P.265-282.
- Tuma, 1970 Tuma S. Evaluation of cystoretrographical symptoms of enuresis by a computer // Prakticky lekar.-1970.-10.- P. 378-380.
- Turing, 1936 Turing A.M. On computable numbers, with an application to the Entscheidungs problem // Proc. London Math. Soc.-1936.-Vol.-42, N.2. -P.230-265.
- Uhr et al., 1982 Uhr L., Schmitt L., Hanrahan P. Cone / pyramid perception program for arrays and networks // Multicomputers and Image Processing Algorithms

- and Programs.-N.Y.: Academic Press, 1982.
-P.179-191.
- Vanderburg
et al., 1975 Vanderburg G.L., Minker J. State-space,
problem-reduction and theorem proving - some
relationships // CACM.-1975.-Vol.18,N.2.
-P.102-115.
- Vere,
1980 Vere S.A. Multilevel counter-factuals for
generalizations of relational concepts and
productions // Artif. Intell. 1980. N.14.
-P.139-164.
- Walker,
1973 Walker D.E. Automated language processing //
Annual Review of Information Sci. and Technology,
V.8.-Washington D.C.: American Soc. for
Information Science,1973.
- Waltz,
1978 Waltz D.L. PLANES: an English language question
answering system for a large relational database
// CACM. 1978.-Vol.21,N.7.-P.526-539.
- Watanabe,
1985 Watanabe S. Pattern recognition: human and
mechanical.-N.Y.: Jihn Wiley and Sons,Inc.,
1985.-592p.
- Waterman et al.,
1971 Waterman D.A., Newel A. Protocol analysis as a
task for artificial intelligence.-1971.
-P.285-328.
- Waterman et al.,
1973 Waterman D.A., Newel A. FASII: an interactive
task free version of an automatic protocol
analysis system // Proc. 4th IJCAI.-1973.
-P.431-445.
- Weihagen,
1978 Weihagen R. Characterization problems in the
theory of inductive inference // Lecture Notes in
Comp.Sci.-1978.-N.62.-P.494-508.
- Wierzbicka,
1969 Wierzbicka A. Dociekania semantyczne. Wroclaw,
1969.
- Wilensky,
1978 Wilensky R. Why John married Mary: understanding
stories involving recurring goals // Cognitive
Sci. -1978.-Vol.2.-P.235-266.
- Wilensky,
1983 Wilensky R. Story grammars versus story points //
The Behavioral and Brain Sciences.-1983.
-N.6.-P.579-623.
- Wilkins,
1982 Wilkins D. Using knowledge to control tree
searching // Artif. Intell. 1982.-N.18. -P.1-15.
- Wilks,
1872 Wilks Y.A. Grammar, meaning, and the machine
analysis of language. -London: Routledge and
Kegan Paul,1972.
- Wilks,
1977 Wilks Y.A. Frames, scripts, stories and fantasies
// Pragmatics.-1977.-N.2.
- Winograd,
Winograd T. Procedures as representation for data

- 1971 in computer programs for understanding natural language. -Ph. D. thesis.-Cambridge, Mass., MIT Rep. MAC-TR-84, 1971.
- Winograd, 1972 Winograd T. Understanding natural language. -N.Y.:Academic Press, 1972.
- Wong et al., 1978 Wong R.Y., Hall E.L. Sequential hierarchical scene matching // IEEE Trans.-1978.-Vol.C.-27, N.4.-P.359-366.
- Woods, 1970 Woods W.A. Transition network grammars for natural language analysis // CACM.-1970. -Vol.13, N.10.-P.591-606.
Рус.пер.: Вудс В.А. Сетевые грамматики для анализа естественных языков // Кибернетический сборник, НС, вып. 13.-М.:Мир, 1976.-С.120-158.
- Woods, 1973 Woods W. Progress in natural language understanding: an application to lunar geology // AFIPS Conf. Proc.42, 1973.-P.441-450.
- Woods, 1980 Woods W. Cascaded ATN grammars // American J. Computational Linguistics.-1980.-Vol.6, N.1. -P.1-15.
- Wos et al., 1965 Wos L., Robinson G.A., Carson D.F. Efficiency and completeness of the set of support strategy in theorem proving // J. Assoc. Comput. Mach. - 12. - 1965. -P. 536-541.
- Yager, 1982 Yager R.R. Level sets for membership evaluation of fuzzy subsets // Fuzzy Sets and Possibility Theory-N.Y.: Pergamon Press, 1982.
- Yasdani 1982 Yasdani M. How to write a story // Proc. of the 1982 European Conf. on AI, Orsay, France, 1982. -P.259-260.
- Zadeh, 1973 Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning. -N.Y.: Elsevier P.C, 1973.
- Zadeh, 1975 Zadeh L.A. Calculus of fuzzy restrictions // Fuzzy Sets and their Applications to Cognitive and Decision Processes.-N.Y.:Academic Press, 1975.
- Zadeh, 1986 Zadeh L. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination // The AI Magazine.-1986. -Vol.7.-P.85-90.
- Zaripov, 1969 Zaripov R.Ch. Cybernetics and music // Perspectives of new music. -V.7. -N 2. -1969. - P. 115-154.
- Zaripov, 1979 Zaripov R.Ch. Musica con il calcolatore. Le regole matematiche della composizione.-Padova: Muzzio, 1979.-169 p.

Оглавление

Предисловие	5
Глава 1. Представление знаний	7
1.1. Данные и знания (Д. А. Поспелов)	7
1.2. Логические модели (Г. С. Плещевич)	14
1.3. Сетевые модели (В. С. Лозовский)	28
1.4. Продукционные модели (Д. А. Поспелов)	49
1.5. Сценарии (Л. В. Литвинцева)	56
1.6. Ленымы (Е. Ю. Кандрашина)	60
Глава 2. Методы работы со знаниями	65
2.1. Приобретение и формализация знаний (А. Н. Аверкин, А. Ф. Блишун, Т. А. Гаврилова, Г. С. Осипов)	65
2.2. Пополнение знаний (Л. В. Литвинцева, Д. А. Поспелов)	76
2.3. Обобщение и классификация знаний (В. Н. Вагин, Н. П. Викторова)	82
2.4. Дедуктивный вывод на знаниях (В. Н. Вагин)	89
2.5. Неточный вывод на знаниях (А. Р. Хачатрян)	105
Глава 3. Общение	110
3.1. Уровни понимания (Д. А. Поспелов)	110
3.2. Понимание текстов на естественном языке (И. В. Совпель)	115
3.3. Синтез связанных текстов (А. В. Зубов, Н. В. Руссова)	121
3.4. Машинное творчество (М. Г. Гааза-Рапопорт, Р. Х. Зарипов)	126
3.5. Когнитивная компьютерная графика (А. А. Зенкин)	137
3.6. Компьютерные игры (Г. Г. Гнездилова, О. А. Гончаров, Г. В. Сенин)	143
Глава 4. Распознавание образов и анализ изображений (Ю. И. Журавлев, И. Б. Гуревич)	149
4.1. Проблема распознавания	149
4.2. Математическая теория распознавания образов	153
4.3. Распознавание изображений	174
4.4. Алгоритмические базы знаний	189
Глава 5. Восприятие и первичная обработка зрительной информации	191
5.1. Зрительное восприятие человека и машинное зрение (В. В. Александров, Н. Д. Горский)	191
5.2. Восприятие двумерных изображений (Н. Д. Горский)	196
5.3. Восприятие трехмерных сцен (В. С. Шнейдеров)	201
Глава 6. Обучение (А. А. Мартиросян, Э. М. Погосян)	206
6.1. Модели обучения	206
6.2. Обучение по примерам	208
6.3. Обучающиеся системы	216
Глава 7. Планирование решения задач	231
7.1. Планирование действий (А. Н. Аверкин, Е. И. Ефимов)	231
7.2. Планирование при синтезе программ (М. И. Канович, Г. Е. Минц)	243
7.3. Поступки и поведение (Д. А. Поспелов)	251
Список литературы (составитель Н. В. Руссова)	257