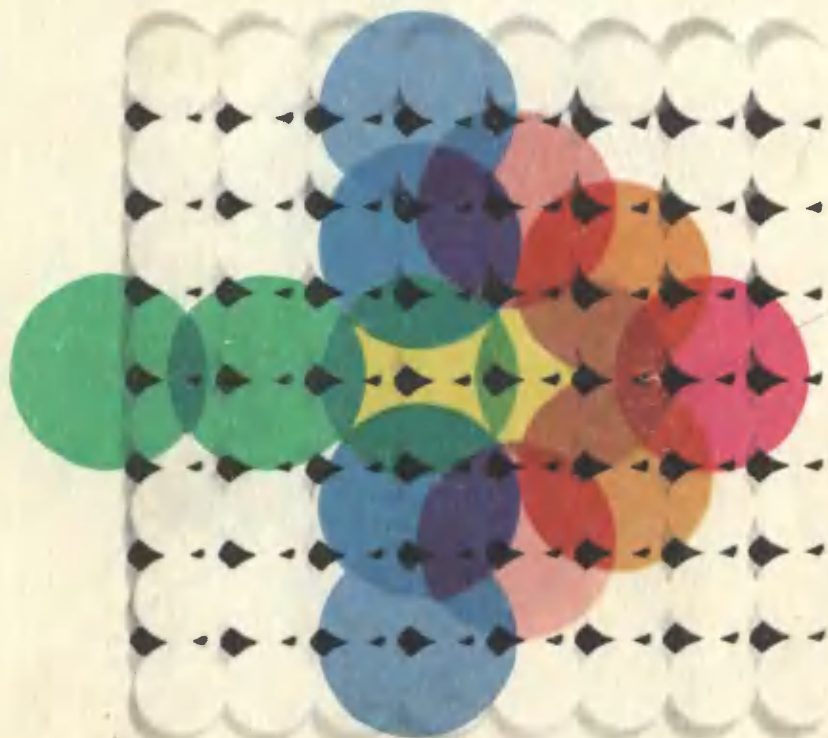


КИБЕРНЕТИКА



# **ЭКСПЕРТНЫЕ СИСТЕМЫ**

ПРИНЦИПЫ РАБОТЫ И ПРИМЕРЫ

# **ЭКСПЕРТНЫЕ СИСТЕМЫ**

## ПРИНЦИПЫ РАБОТЫ И ПРИМЕРЫ

# EXPERT SYSTEMS

*Principles and case studies*

EDITED BY

**Richard Forsyth**

*Polytechnic of North London*

LONDON NEW YORK

**Chapman and Hall**

● КИБЕРНЕТИКА ●

# ЭКСПЕРТНЫЕ СИСТЕМЫ

ПРИНЦИПЫ РАБОТЫ И ПРИМЕРЫ

Под редакцией Р. Форсайта

Перевод с английского С.И.Рудаковой  
под редакцией В.Л.Стефанюка



МОСКВА «РАДИО И СВЯЗЬ» 1987

ББК 32.973  
Э 41  
УДК 681.31\* 5/520

Редакция переводной литературы

Э 41      **Экспертные системы. Принципы работы и примеры:** Пер. с англ./А. Брукинг, П. Джонс, Ф. Кокс и др.; Под ред. Р. Форсайта. — М.: Радио и связь, 1987. — 224 с.: ил. (Кибернетика).

В книге коллектива специалистов под редакцией английского ученого приводятся основные сведения по созданию экспертных систем: общее описание и их архитектура, связь с японским проектом машин пятого поколения, способы построения машинных рассуждений исходя из нечеткой, неполной и противоречивой информации, совокупность методов сбора, обработки и использования знаний, вопросы обучения.

Для широкого круга инженерно-технических работников, интересующихся созданием прикладных интеллектуальных систем.

Э 2405000000-093 126-87  
046 (01)-87

32.973

© 1984 Chapman and Hall

© 1984 Tom Stonier (Chapter 13)

© Перевод на русский язык, предисловие редактора перевода, примечания, список литературы к предисловию редактора перевода. Издательство "Радио и связь", 1987

## ПРЕДИСЛОВИЕ РЕДАКТОРА ПЕРЕВОДА

Экспертные системы, пожалуй, наиболее значительное практическое достижение в области искусственного интеллекта, получили в настоящее время самую широкую известность. Многие ведущие специалисты в этой области уже с самого начала полагали, что искусственный интеллект должен представлять собой инженерную дисциплину, задачей которой является создание конструкции [1]. Именно такие практические цели преследовались в большинстве научных разработок. Даже разработка программ для игры в шахматы первоначально считалась ценной, главным образом потому, что они были удобны для изучения одной из часто возникающих в приложениях задач — создания эффективных методов перебора возможных вариантов с привлечением эвристических средств.

И все же основные усилия в области искусственного интеллекта приходились на поиск универсальных методов решения: ученые старались по возможности найти общие принципы, которые можно было бы применять, отвлекаясь от специфики конкретной предметной области. На этом пути были получены важные результаты и методы, которые в совокупности составляют фундамент искусственного интеллекта как научной дисциплины [2, 3].

К началу 80-х годов тенденция в области искусственного интеллекта к созданию прикладных разработок заметно усилилась. Хотя отчасти это явилось результатом общего ограничения на чисто фундаментальные исследования [4], основными причинами были особенности внутреннего развития дисциплины. Среди последних можно отметить определенный предел, который был достигнут в разработке интеллектуальных роботов, а также наметившийся застой в теоретической области искусственного интеллекта, нуждавшейся в притоке новых идей, идущих от реальных задач [5]. Немалую роль сыграло и то, что в области искусственного интеллекта накопился солидный багаж методов и инструментальных средств, которые не могли слишком долго "лежать без дела".

Все это привело к появлению экспертных систем — систем искусственного интеллекта, использующих знания из сравнительно узкой предметной области для решения возникающих в ней задач, причем так, как это делал бы эксперт-человек, т.е. в процессе диалога с заинтересованным лицом, поставляющим необходимые сведения по конкретному вопросу (см. [6, 7] и рис. 2.1 в настоящей книге).

Экспертные системы, в отличие от других ориентированных на практику программных комплексов искусственного интеллекта, имеют *массовое* применение, так как каждая экспертная система позволяет решить неограниченное число сложных для человека задач из соответствующей предметной области. Экспертная система может быть, по-видимому, построена для любой предметной области, о чем говорит следующий перечень уже решаемых экспертными системами классов задач: интерпретация, диагностика, пред-

сказание, проектирование, планирование, управление, наблюдение, исправление.

Нелегко предложить задачу, интересную для практики и не попадающую ни в одну из перечисленных категорий!

Накопление в искусственном интеллекте необходимой "критической массы" идей и методов обеспечило взрывной характер развития экспертных систем: труды самых представительных конференций, журналы по искусственному интеллекту и смежным дисциплинам буквально полны работами по экспертным системам [4—7].

За последние годы появился целый ряд книг, посвященных этим системам (см., например, [8]). Одной из последних работ является изданная в Великобритании книга под редакцией Р. Форсайта, перевод которой предлагается советскому читателю. Книга имеет достоинства, которые делают ее интересной достаточно широкому кругу читателей, и прежде всего — это популярная, доходчивая форма подачи идей из области экспертных систем.

Авторы глав, в совокупности составляющих книгу, в основном опираются на имеющийся у них опыт создания сложных программных и человеко-машинных комплексов. Они стараются не столько осветить состояние в области экспертных систем в целом, сколько описать действующую, как правило, несложную экспертную систему или соответствующий инструментальный пакет (см. гл. 9).

Надо отметить, что, несмотря на популярность, книга под редакцией Р. Форсайта содержит важные технические подробности построения экспертных систем (например, схемы пересчета вероятностных оценок, способы комбинирования свидетельств при нечетких рассуждениях), которым мало внимания уделено в других книгах, но которые являются важными при самостоятельном создании экспертной системы.

Одной из существенных особенностей книги является то большое внимание, которое уделяется в ней обучающимся системам. Этот традиционный для кибернетики предмет в последнее время был фактически забыт. Однако в экспертных системах без автоматического обучения на этапе как формирования базы знаний, так и ее пополнения и уточнения обойтись, как правило, трудно. Обучающимся машинам, без которых "нельзя создать действительно интеллектуальную систему" [5], посвящены три главы книги.

В книге обсуждаются возможности, которые открываются благодаря появлению экспертных систем, и дается оценка их влияния на общество в настоящем и будущем. Хотя при этом не все рассуждения выглядят достаточно глубокими, очевидно, что такой анализ экспертных систем, учитывая их массовый характер, совершенно необходим, особенно при оценке перспектив.

Предлагаемый перевод книги под редакцией Р. Форсайта будет встречен с большим интересом. Эта книга послужит для всех читателей прекрасным введением в область экспертных систем, которая сейчас является весьма популярной и на которую возлагаются самые большие надежды в будущем.

## ПРЕДИСЛОВИЕ

Экспертная система — это вычислительная система, в которую включены знания специалистов о некоторой конкретной проблемной области и которая в пределах этой области способна принимать экспертные решения. В рамках экспертных систем к настоящему моменту достигнуты успехи в таких областях, как медицинская диагностика, геологическая разведка, органическая химия и обнаружение неисправностей в электронном оборудовании.

Хотя обычно экспертные системы ориентированы на очень узкие предметные области, в случае применения их к реальным задачам получены весьма яркие достижения. Они и обусловили большой интерес к экспертным системам за пределами тех исследовательских лабораторий, в которых такие системы разрабатывались.

Экспертные системы положили начало развитию совокупности методов "инженерии знаний" (техники использования знаний), составляющих новый подход к созданию высокоэффективных программных систем. Этот новый подход несет революционные изменения.

## ЗАДАЧИ КНИГИ

Многие люди независимо от того, считают они себя специалистами в области компьютеров или нет, уверены, что в области информатики и вычислительной техники происходит что-то очень важное, и поэтому они хотят повысить свою подготовку настолько, чтобы быть в состоянии дать всему самостоятельную оценку, а если необходимо — и воспользоваться новыми методами.

Этому призвана помочь настоящая книга, в которой основные идеи экспертных систем объясняются для пользователей вычислительных машин, не знакомых с последними результатами научных исследований. Она написана как вводная книга справочного характера для тех, кто хочет узнать, как работают экспертные системы. Это не академическое издание, а практическое руководство, написанное людьми, активно работающими в этой области. Она призвана осветить события в одной новой, быстро меняющейся области информатики и вычислительной техники. После прочтения книги вы должны быть в состоянии начать работу над собственным проектом, опирающимся на инженерию знаний.

## ПЛАН КНИГИ

Книга имеет справочный характер и предназначена для тех, кто хочет знать, как работают экспертные системы. Она состоит из четырех частей.

В ч. I объясняются принципы, лежащие в основе систем, использующих знания, и описывается анатомия типичной экспертной системы. В ней экспертные системы освещаются также в историческом аспекте: рассматрива-



ется японский проект вычислительных машин пятого поколения и отмечается острая (а некоторые могут сказать раздраженная) реакция, которую этот проект вызвал на Западе.

В ч. II изучаются проблемы логического вывода, а именно как используются стратегии приближительного рассуждения, чтобы получить хорошую оценку истинности исходя из не вполне определенных данных и несовершенных правил вывода. Эта область, где создатели экспертных систем внесли новую идею в практику применения вычислительной техники: в системах, работа которых основывается на использовании знаний, не предполагается абсолютной безошибочности математического обеспечения. Удалите наугад одно правило из базы знаний экспертной системы — и качество ее работы (по всей вероятности) лишь слегка упадет. Удалите теперь наугад одно утверждение из обычной программы — и... страшно подумать, что произойдет!

Часть III посвящена самой инженерии знаний. Может ли знание специалиста быть представлено в компьютере, а если может, то как? Здесь изложение ведется на конкретных примерах, знакомящих читателя с тем, как эта задача работы со знанием решается с применением трех различных программных средств. Этими средствами являются: Пролог — язык программирования, основанный на исчислении предикатов, "Микроэксперт" — оболочка экспертной системы, содержащая стандартный механизм вывода и позволяющая кодировать знания определенным образом, и REVEAL — универсальная среда поддержки программирования со специальными средствами, облегчающими конструирование систем экспертного решения задач.

Заключительная часть затрагивает вопрос, касающийся обучающихся машин, который был весьма популярным в 50-х годах, затем стал пользоваться меньшим вниманием. Новая волна интереса к этому вопросу обусловлена потребностями экспертных систем. Качество работы экспертной системы существенным образом зависит от качества и диапазона знаний, включенных в ее базу знаний. Однако это знание с неба не падает: его приходится с большим трудом извлекать из информации, получаемой от специалистов, или читать между строк различных учебников. Это — узкое место "извлечения знаний". Одна из возможностей прорваться через это узкое место, по-видимому, заключается в использовании программ автоматической индукции, которые, исходя из совокупности примеров, сами по себе открывают новые понятия и правила.

Все это ведет к представлению о "массовом производстве" знаний в символической форме. Намечается развитие целой индустрии, в которой знания являются ценным продуктом. Последняя глава, написанная профессором Стониером, связана с исследованием социальной и экономической роли такого рода развития событий. В ней поднимается вопрос о том, как мы можем контролировать процесс "индустриализации знаний".

*Р. Форсайт*

## ЧАСТЬ I

### ОСНОВНЫЕ ИДЕИ

#### Глава 1

#### ФЕНОМЕН ЭКСПЕРТНЫХ СИСТЕМ

*Ричард Фортсайт*

В последнее время все чаще говорится об экспертных системах. Для одних они представляют "светлую надежду" информационной технологии, для других — больше напоминают современный вариант "желтой опасности", поскольку японцы могут сильно обогнать европейцев. Для остальных же — это просто очередной трюк, старое вино в новых бутылках.

Задача вводной главы — объяснить, что такое экспертные системы, и показать их в историческом аспекте. Для того чтобы объяснить феномен экспертных систем, мы должны отойти несколько назад и посмотреть на ту почву, которая их взрастила, — богатую область искусственного интеллекта (ИИ).

Искусственный интеллект имеет долгую и "пеструю" историю, начало которой можно отнести даже ко времени возникновения легенды о Пигмалионе. Постоянное восхищение человечества перед искусственными разумными существами сопровождает длинную историю, от которой подчас стынет кровь, касающуюся големов, говорящих голов и в особенности чудовища Франкенштейна\* [1, 5]. Мы пропустим эти первые 2000 лет искусственного интеллекта и начнем его историю с 50-х годов, когда фантазия благодаря цифровым вычислительным машинам стала приобретать реальность.

В табл. 1.1 кратко отражена история искусственного интеллекта в век компьютеров. Я сократил ее до четырех этапов — одного на десятилетие — и тем самым, конечно, решительно ее упростил. Тем не менее основные этапы истории ИИ сохранены.

В колонке "Парадигма" помещен тот ответ, который вы получили бы, если бы в то время спросили специалиста по искусственному интеллекту, чему посвящены исследования в этой области. В колонке "Исполнители" указаны одна-две основные фамилии ученых, которые, по-видимому, характеризуют дух искусственного интеллекта в этот период. (Под основными исполнителями в круглых скобках я поместил мыслителей, или теоре-

---

\* Чудовище которое выходит из-под контроля и губит своего создателя. — *Прим. ред.*

## Сжатая история искусственного интеллекта

Годы	Парадигма	Исполнители	Система
50-е	Нейронные сети	Розенблат (Винер, Маккалок)	PERCEPTRON
60-е	Эвристический поиск	Ньюэлл и Саймон (Шеннон, Тьюринг)	GPS
70-е	Представление знаний	Шортлифф (Минский, Маккарти)	MYCIN
80-е	Обучающиеся машины	Ленат (Сэмюэл, Холланд)	EURISKO

тиков, которые заложили основы исследований.) Наконец, в колонке "Система" указаны типичные системы (не обязательно самые лучшие), которые отражают тенденции или модные течения.

## 1.1. НЕЙРОННЫЕ СЕТИ

В 50-х годах исследователи в области искусственного интеллекта пытались строить разумные машины, имитируя мозг. Оглядываясь назад, не приходится удивляться, что эти попытки провалились, поскольку аппаратные средства были абсолютно непригодны для такого дела, не говоря уже о программных средствах.

Типичной системой, которую я привожу в таблице, является PERCEPTRON [8]. Она представляла собой самоорганизующийся автомат, который можно считать грубой моделью сетчатки глаза человека. Его можно было научить распознавать образы, но, как позже показали Минский и Пейперт, это был лишь ограниченный класс зрительных образов [7].

В то время имелся большой энтузиазм в отношении систем, подобных системе Розенблата, основанный на пионерских кибернетических представлениях Норберта Винера и Уоррена Маккалока об абстрактных нейронных сетях. Считалось, что если взять сильно связанную систему модельных нейронов, которой вначале ничего не известно, применить к ней программу тренировки из поощрений и наказаний, то в конце концов она будет делать все, что ни задумает ее создатель. При этом тактично обходился тот факт, что мозг человека содержит  $10^{10}$  нейронов, каждый из которых по сложности примерно соответствует, скажем, одному транзистору\*.

Этот неоправданный оптимизм испарился еще до того, как Минский и Пейперт провели свою теоретическую обширную работу, развенчавшую представление о системе PERCEPTRON. Просто экспериментальные результаты оказались недостаточно хорошими, так что воображением специалистов по искусственному интеллекту овладело новое увлечение.

\* Транзистор — это компьютер на одном кристалле, способный самостоятельно выполнять сравнительно небольшую программу. Из транзисторов можно составлять сеть для параллельной обработки данных, поскольку каждый из них располагает средствами общения с несколькими (обычно четырьмя) соседями. — *Прим. ред.*

## 1.2. ЭВРИСТИЧЕСКИЙ ПОИСК

На новые рубежи указали Аллен Ньюэлл и Герберт Саймон из Университета Карнеги-Меллона (США), работа которых завершилась созданием системы GPS – универсального решателя задач [2].

Центральным для их подхода было представление об эвристическом поиске. Они считали, что мышление человека основано на определенном сочетании простых задач манипулирования символами, таких как сравнение, поиск, модификация символа и т. п. – операций, которые могут выполняться компьютером. Решение задач они представляли себе как поиск (перебор) в пространстве возможных решений по эвристическим правилам, которые помогают направить поиск к искомой цели.

Они отбросили мысль о сети нейронов, указав, что даже задача конструирования нервной системы муравья, в которой участвует менее тысячи нейронов, выходит за пределы доступной технологии. (Справедливости ради отметим, что Маккалок, дедушка теории нейронных сетей [6], занимался достаточно ограниченной задачей, а именно зрительной системой лягушки, и внес важный вклад в развитие нейрофизиологии в этой области.)

Ньюэлл и Саймон начали с программы доказательства теорем, потом перешли к машинным шахматам. Затем они переключили свое внимание на поиск общих методов, которые могли бы быть применены к широкому спектру задач, пока не пришли к системе GPS.

Система GPS была универсальной в том отношении, что "не было конкретного указания, к какой области относится задача". Пользователь должен был задать "проблемную среду" в терминах объектов и тех операторов, которые к ним применимы.

Но эта универсальность относилась лишь к ограниченной области математических головоломок с относительно небольшим множеством состояний и хорошо очерченных формальных правил. Как и большинство ее современниц, система GPS функционировала в таком формализованном микромире, где возникающие проблемы (например, задача "Ханойская башня", задача о миссионерах и людоедах), с точки зрения людей, проблемами и не являются.

С технической точки зрения можно сказать, что процесс, известный как "поиск в глубину" и состоящий в последовательном разбиении задачи на подзадачи, пока не будет получена легко решаемая подзадача, является малоэффективным по той причине, что большое число тупиковых направлений подвергается весьма тщательному анализу. Впоследствии исследователи разработали более эффективные стратегии "поиска в ширину". (Правда, поиск в глубину продолжает жить, и довольно успешно, внутри бесчисленного множества интерпретаторов для языка Пролог!)

Тем не менее проект создания системы GPS выделяется среди других работ. Его авторы также стали жертвой неоправданного оптимизма, как в случае, когда Герберт Саймон в 1957 г. предсказал, что через десять лет компьютер станет чемпионом мира по шахматам, но в науке всегда находится место для чересчур смелых неоправдавшихся надежд. Парадоксаль-

но, но подчеркивая, что и мозг, и компьютеры представляют собой примеры универсальных устройств общего назначения, манипулирующих символами, эти авторы укрепили положение людей. В конце концов, очевидно, что люди запрограммированы совершенно гениально.

### 1.3. ЗНАНИЕ – СИЛА

Система GPS не могла решать реальные задачи. В 70-х годах группа ученых, возглавляемая Эдвардом Фейгенбаумом из Станфордского университета, начала искать средства, избавляющие от такого недостатка. Вместо того чтобы отыскивать очень эффективные и универсальные эвристики, они занялись сужением рассматриваемых вопросов. То, чем, по-видимому, располагает специалист – это набор разнообразных умений, т. е. большое число приемов и неформальных правил.

На этом пути и родилась экспертная система, выглядевшая почти как карикатура на специалиста-человека, который узнает все больше о все меньшем. Интерпретатор для масс-спектрограммы DENDRAL [3] является прототипом всех экспертных систем, но я здесь остановлюсь на его дочерней системе MYCIN [9], которая оказала большое влияние на развитие экспертных систем.

Система MYCIN – это компьютерная система, которая диагностирует бактериальные инфекции крови и дает предписания относительно лекарственной терапии. Она была родоначальником целой серии медико-диагностических машин, некоторые из которых используются в рутинной клинической практике. Например, PUFF – система для диагностики состояния легких, построенная по плану системы MYCIN, регулярным образом используется в Тихоокеанском медицинском центре вблизи Сан-Франциско.

Система MYCIN ввела в рассмотрение несколько характеристик, которые стали отличительной чертой экспертных систем. Во-первых, ее знание составляют сотни правил, подобных следующему:

- ЕСЛИ (1) инфекция представляет собой первичную бактеримию, и  
(2) место взятия культуры является стерильным, и  
(3) предполагается, что этот организм проник через желудочно-кишечный тракт

ТО можно предположить (0,7), что этот организм носит бактериальный характер.

Во-вторых, эти правила являются вероятностными. Шортлифф разработал схему, основанную на использовании "коэффициентов уверенности" (которые, строго говоря, вероятностями не являются) и позволяющую системе прийти, исходя из ненадежных данных, к правдоподобным заключениям. Система MYCIN и подобные ей системы являются достаточно робастными, т. е. приходят к правильным заключениям даже тогда, когда некоторые из исходных данных неверны или отсутствуют. Это свойство выполняется благодаря тому, что в таких системах используются методы, позволяющие собирать фрагментарную и возможно неточную информацию для того, чтобы

сделать хорошую оценку истинного положения вещей с использованием "нечеткой логики", коэффициентов уверенности или других способов описания степени правдоподобия.

В-третьих, система MYCIN может объяснить свой процесс рассуждения. Пользователь (врач, а не пациент) может допрашивать систему различным образом, спрашивая, почему она задала некоторый вопрос или как она, например, пришла к данному промежуточному решению. Это была одна из первых по-настоящему "дружественная" по отношению к пользователю система. Достигнутая степень дружественности на самом деле была прямым следствием подхода к программированию с использованием правил. Каждое правило – это практически независимая от других совокупность знаний, и, если пользователь хочет узнать, почему был задан определенный вопрос, система просто проигрывает у себя некоторый след, описывающий правила, приведшие к вопросу.

В четвертых, и самое главное, система MYCIN действительно работает. Она выполняет работу, которой люди обучаются годами. На самом деле, система MYCIN весьма ограничена в своих возможностях, даже по сравнению с вашим районным терапевтом, и тем не менее ее работа оставляет большое впечатление.

Имеется хорошо известный пример системы PROSPECTOR, другой экспертной системы с западного побережья США, работающей на этот раз в геологии. С помощью этой системы открыты ранее не известные запасы молибдена. Это пока еще самое начало, но деловые люди уже чувствуют возможность извлечения прибылей. Искусственный интеллект перестает быть чистой наукой.

#### 1.4. СЛЕДУЮЩИЙ ШАГ

Итак, где мы при этом оказались? Мы оказались в середине очередного раунда необузданного оптимизма, охватившего теперь не только замкнутую в себе область искусственного интеллекта, но и всю совокупность людей, занятых информатикой, вычислительной техникой и обработкой данных.

На этот раз таким магическим ингредиентом являются знания. Именно обширность и качество базы знаний и определяют успех экспертной системы. Но на этот раз энтузиазм может оказаться оправданным. Если это так, то знания (в машинно-приемлемой форме) могут стать "черным золотом" 90-х годов. А если это так, то те люди, которые хотят открыть самую мощную скважину, будут в числе самых первых, кто займется массовым производством знаний с помощью компьютера.

Это приводит нас к четвертой и последней строке табл. 1.1. Дуг Ленат (также из Станфорда) создал машинную обучающуюся систему EURISKO [4], которая автоматически улучшает и расширяет свой запас эвристических правил. Кроме того, что эта система выигрывала три года подряд в учебной военной игре (несмотря на то, что правила игры каждый раз менялись, чтобы помешать ей это сделать), она произвела переворот в области создания СБИС (сверхбольших интегральных схем), изобретая трехмерный узел типа И/ИЛИ.

Нет сомнения, что программы автоматической индукции, подобные созданным для системы EURISKO, являются важнейшим этапом в развитии искусственного интеллекта в последние десять лет и что машинное обучение — ключ к будущему. До сих пор перенесение умения специалиста-человека в машинную программу было утомительной и долгой процедурой. Знания — ценный товар, поэтому рано или поздно его начнут производить, и в больших количествах (см. также гл. 10).

Таким образом, пока экспертные системы проживают интеллектуальный капитал 70-х годов, искусственный интеллект 80-х перенес внимание в область проблем машинного обучения.

Вот мы и замкнули круг, а точнее, прошли по витку спирали в развитии искусственного интеллекта, поскольку машинное обучение и было той проблемой, которую кибернетики Винер, Маккалок и другие атаквали различными методами в 50-х годах, опираясь на более примитивную технологию тех лет.

## Глава 2

### АРХИТЕКТУРА ЭКСПЕРТНЫХ СИСТЕМ

*Ричард Форсайт*

Наиболее существенным в отношении экспертных систем является то, что они действуют чрезвычайно успешно. Уже существуют системы; превосходящие в медицинской диагностике искусных специалистов. То же самое наблюдается при интерпретации масс-спектрограмм, предсказании возможности заболевания зерновых культур, поиске полезных ископаемых и во многих других областях. Совершенно неожиданно люди и некоторые действительно крупные корпорации стали использовать программы искусственного интеллекта для своего обогащения. Экспертные системы наконец перечеркнули старое высказывание: если нечто работает, то это уже не искусственный интеллект!

В основе экспертной системы находится обширный запас знаний о конкретной проблемной области. В большинстве случаев эти знания организуются как некоторая совокупность правил, которые позволяют делать заключения на основе исходных данных или предположений.

Такой подход к конструированию систем, опирающийся на использование знаний, представляет собой революционное изменение с последствиями революционного характера, поскольку традиционное соотношение

Данные + Алгоритм = Программа

заменяется на новую архитектуру, основу которой составляют база знаний и "машина логического вывода", так что теперь мы имеем

Знания + Вывод = Система,

что похоже на предыдущее и все же отличается достаточно сильно, чтобы иметь серьезные последствия.

## 2.1. ЧЕРТЫ ЭКСПЕРТНЫХ СИСТЕМ

Что же в точности представляет из себя экспертная система? Специальная группа по экспертным системам Британского компьютерного сообщества предложила следующее формальное определение.

Экспертная система рассматривается как результат создания в компьютере основанной на знаниях компоненты, соответствующей навыку эксперта, в такой форме, которая позволяет системе дать разумный совет или принять разумное решение о функции обработки данных. Желательной дополнительной характеристикой, которую многие считают фундаментальной, является способность системы по требованию объяснить свою линию рассуждения в виде, непосредственно понятном тому, кто задал вопрос. Достижение таких свойств обеспечивается самим методом программирования с использованием правил.

Здесь сказано все, но лично я предпочитаю неформальное определение: "Экспертная система — это такой пакет программ, который вынуждает постановщика соответствующей телевизионной передачи терять всякую меру".

На самом деле вы, возможно, найдете более полезным следующий перечень характеристик экспертных систем:

1. Экспертная система ограничена определенной сферой экспертизы.
2. Она способна рассуждать при сомнительных данных.
3. Она способна объяснить цепочку рассуждений понятным способом.
4. Факты и механизм вывода четко отделены друг от друга.  
(Знания НЕ кодируются в дедуктивные процедуры.)
5. Она строится так, чтобы имелась возможность постепенного наращивания системы.
6. Чаще всего она основана на использовании правил.
7. На выходе она выдает *совет* — не таблицы из цифр, не красивые картинки на экране, а четкий совет.
8. Она экономически выгодна. (Это требование к ее работе.)

Самой серьезной проблемой до сих пор было получение знаний от специалиста в форме, пригодной для манипулирования на вычислительной машине.

## 2.2. КОМПОНЕНТЫ ЭКСПЕРТНОЙ СИСТЕМЫ

Мы уже говорили, что экспертная система содержит машину вывода и базу знаний. На самом деле у полностью оформленной экспертной системы имеются четыре существенные компоненты: 1. База знаний. 2. Машина вывода. 3. Модуль извлечения знаний. 4. Система объяснения (интерфейс).

Все четыре модуля, показанные на рис. 2.1, являются важными, и, хотя система, основанная на знаниях, может обойтись без одного-двух из них, истинно экспертная система обязана иметь их все.



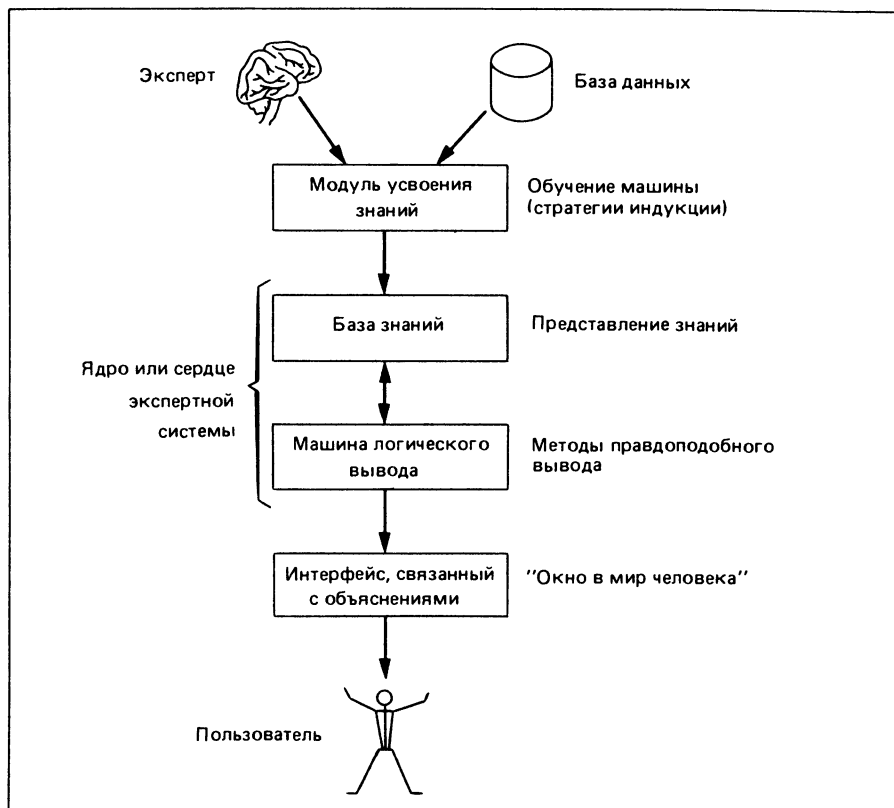


Рис. 2.1. Типичная экспертная система

### 2.3. БАЗА ЗНАНИЙ

База знаний содержит факты (или утверждения) и правила. Факты представляют собой краткосрочную информацию в том отношении, что они могут изменяться, например, в ходе консультации. Правила представляют более долговременную информацию о том, как порождать новые факты или гипотезы из того, что сейчас известно.

Чем такой подход отличается от обычной методики использования базы данных? Основное различие состоит в том, что база знаний обладает большими творческими возможностями. Факты в базе данных обычно пассивны: они либо там есть, либо их нет. База знаний, с другой стороны, активно пытается пополнить недостающую информацию.

Правила продукций являются предпочтительным средством отображения неформальных знаний. Такие правила имеют уже знакомый нам формат ЕСЛИ-ТО, например:

ЕСЛИ хозяева поля проиграли последнюю игру у себя дома И гости выиграли последнюю игру на своем поле,

ТО вероятность ничьей следует умножить на 1,075; вероятность выигрыша гостей умножается на 0,96.

Но следует помнить, что эти знания не воплощены в какую-то программу, а представляют собой данные для высокоуровневого интерпретатора, а именно машины вывода.

Правила продукций — не единственный способ представления знаний. Для этой цели в других системах использовались деревья решений (например, в системе ACLS), семантические сети (например, в системе PROSPECTOR) и исчисление предикатов. Поскольку одна из форм исчисления предикатов встроена в язык Пролог вместе с системой доказательства теорем, то это представление, похоже, завоевывает все большую популярность. На некотором глубоком уровне все типы представления знания должны быть эквивалентны между собой, но лучшей рекомендацией будет выбор самого простейшего способа из тех, которые дают возможность работать.

#### 2.4. МАШИНА ВЫВОДА

В области экспертных систем имеются некоторые разногласия между сторонниками "прямой цепочки рассуждений" и "обратной цепочки рассуждений" в качестве стратегии для логического вывода в целом. Прямая цепочка связана с рассуждениями, ведущимися от данных к гипотезам, тогда как обратная цепочка — с попыткой найти данные для доказательства или опровержения некоторой гипотезы. Чисто прямая цепочка рассуждений ведет к неуправляемому режиму задания вопросов в диалоге, тогда как обратная цепочка будет, как правило, приводить к настойчивому повторению вопросов, касающихся цели.

По этой причине наиболее удачные системы используют комбинации этих цепочек, а К. Нейлор недавно описал метод, известный как подход с оценкой правил, который сочетает в себе достоинства обеих цепочек [6]. Я называю его "косвенной цепочкой рассуждений" (см. также гл. 6).

Работает ли ваша процедура вывода в прямом или в обратном направлении, она будет иметь дело с ненадежными данными, и здесь начинается самое интересное. Слишком долго специалисты по вычислительной технике пытались втиснуть мягко очерченный мир, в котором мы на самом деле живем, в жесткие рамки строго очерченной памяти компьютера. И никогда это не проходило легко. Теперь же у нас появились средства работы с неопределенностью, т.е., иными словами, с реальным миром, а не с некоторой идеализированной абстракцией, в которую заставляли нас верить наши системы данных.

Действительно, существует слишком много способов работы с неуверенностью в данных! Имеется нечеткая логика, байесовская логика, коэффициенты уверенности, многозначная логика, если назвать только четыре

из них. Были испробованы все виды схем, и, странное дело, большинство из них, по-видимому, работает.

Мое объяснение такого положения состоит в том, что организация знаний играет большую роль, чем связанные с ними числовые значения. В большинстве баз знаний предусмотрена избыточность, позволяющая экспертной системе прийти к правильному заключению несколькими различными маршрутами. Числа, измеряющие степень доверия, служат лишь для тонкой настройки.

По этой причине можно выбирать любую меру ненадежности из тех, которые вас устраивают (см. также гл. 5 и 6).

Теперь давайте перейдем к вопросу извлечения знаний.

## 2.5. СТРАТЕГИИ ИНДУКЦИИ

Знания представляют собой редкий и дорогой ресурс. Как мы их получаем?

До настоящего времени самым узким местом в развитии экспертных систем было усвоение знаний в машинно-употребимой форме. Эксперты известны своей неспособностью объяснить, каким образом они приходят к определенным решениям, а объяснения, которые они дают, часто оказываются чисто внешними. Как бы мы могли формализовать их опыт?

Традиционный способ состоял в том, чтобы запереть на несколько месяцев высокооплачиваемого "домашнего эксперта" (или пару) вместе с высокооплачиваемым специалистом по технологии знаний, в течение которых они эффективно обговаривают закодированный вариант того, что знает специалист. На это нужно время и, очевидно, деньги.

Поэтому остро ощущается потребность автоматизировать процесс извлечения знаний. По моему мнению, программа EURISCO является первым предвестником нового поколения обучающихся машин [3].

Ленат — не единственный ученый, который активно работает в этой области. Михальским была сконструирована система, которая обучалась классификации болезней зерновых хозяйственных культур [4]. Алгоритм обучения понятиям, в котором дерево решения наращивается в результате анализа базы данных, содержащей примеры, был разработан Квинланом. Я сам написал программу [2], в которой для создания и усиления правил классификации используется дарвиновская схема естественного отбора. Эта программа названа BEAGLE (Biological Evolutionary Algorithm Generating Logical Expressions — Биологический эволюционный алгоритм, порождающий логические выражения).

Но удивительным в программе EURISCO является то, что используемый язык описаний (средство, в котором хранятся правила и понятия) оказался достаточно выразительным для того, чтобы в ней присутствовало в зачаточной форме самосознание, выражаемое "метаправилами". Это — система с высокой степенью интроспекции, тратящая массу времени на управление своим поведением, запоминая обнаруженные ею правила и применяющая их к самой себе.

Один из открытых ею результатов характеризует некоторый иронический момент. Так, она обнаружила, что обычно правила, поставляемые человеком, оказываются лучше ее собственных, поэтому она создала эвристику:

ЕСЛИ правило создано машиной  
ТО вычеркни его

К счастью, первым правилом, которое оказалось стертым системой EURISCO, было именно это правило!

(Более подробно вопросы усвоения знаний и их уточнения рассматриваются в гл. 10 — 12.)

## 2.6. ОКНО В МИР ЧЕЛОВЕКА

Четвертой важной компонентой экспертной системы является подсистема, обеспечивающая возможность объяснения с человеком,

Одним из самых замечательных свойств, присущих классическим экспертным системам, подобным системе MYCIN, является то внимание, которое было уделено интерфейсу с пользователем. В любой момент можно было спросить систему, почему была сделана такая дедукция или почему система задала такой вопрос пользователю. В системе, основанной на использовании правил, ответ обычно получается путем прослеживания еще раз тех шагов рассуждения, которые привели к данному вопросу или к данному заключению. Легкость, с которой это достигается, является одним из соображений в пользу систем, основанных на правилах.

Средства объяснений не следует рассматривать лишь как одну из возможных черт экспертной системы. Доналд Мичи (1982) и другие авторы указывали на обреченность систем, в которых не предусмотрено "когнитивное окно для человека", т. е. действия которых носят скрытый, необъяснимый характер.

Если мы хотим избежать трагических ситуаций типа несчастья в местечке "Трехмильный остров"\* или еще худших, в наших экспертных системах должны быть предусмотрены расспрашивание и инспекция. Короче, метод рассуждения, который не может быть объяснен человеку, является неудовлетворительным, *даже если с ним система работает лучше, чем специалист.*

## 2.7. КОМУ НУЖНЫ ЭКСПЕРТНЫЕ СИСТЕМЫ?

На этом этапе читатель может задать вопрос: "Нужна ли мне экспертная система?" Ответ зависит от типа задачи, которую вы хотите решить. В табл. 2.1 представлен контрольный список характеристик степени пригодности подхода с использованием знаний. Если имеющееся вами в виду при-

---

\* Имеется в виду известная история с атомным реактором в штате Пенсильвания, США, в 1979 г., когда неполадка в вакуумной системе реактора вызвала цепочку событий, которая грозила радиоактивным заражением окружающего района. При этом вся весьма изощренная система управления реактором полностью вышла из-под контроля человека, и даже специалист был не в состоянии сказать, где именно неисправность и что необходимо предпринять. — Прим. ред.

<i>Подходит</i>	<i>Не подходит</i>
Диагностика Нет установившейся теории Мало специалистов Данные "зашумлены"	Вычислительные задачи Существует подходящая формула Специалистов пруд пруди Известны точные факты

ложение больше относится к левой части таблицы, чем к правой, то следует всерьез рассмотреть перспективу использования экспертной системы.

Под диагностической задачей мы понимаем не только медицинский диагноз. К этим задачам относится любая область, в которой имеется множество возможных ответов, и трудность состоит в том, чтобы выбрать из них один верный или, по крайней мере, отбросить заведомо неверные. Сюда попадают многие задачи классификации и предсказания, например диагностика неисправностей в вычислительной машине. Как только установлено, что дело в интерфейсе с магнитным диском, а не в оперативной памяти компьютера, то можно считать, что основная трудность позади: ремонт — это рутинная процедура замены печатных плат.

Под областью, где нет твердо установленной теории, мы подразумевали вопросы типа законов о налогах, ремонта двигателей, предсказания погоды и многие медицинские вопросы. В них имеется слишком много переменных величин, затрудняющих создание полной и цельной теории, так что искусные практики опираются на свои познания и интуицию. Этим самым мы исключаем задачи, в которых можно ввести некоторую формулу, повернуть ручку и — готов ответ, как, например, в случае движения небесных тел, где законов ньютоновской механики вполне достаточно для управления полетом космического корабля.

Область с малым числом специалистов легко узнается по красноречивым симптомам высокой зарплаты, спросу на специалистов и очередям на курсы переквалификации. Ясно, что экономически оправдано механизировать навыки, на которые высок спрос (например, обработку данных, поступающих из скважины при поиске нефти), предпочитая их более привычным, например, распознаванию образов, на что способны даже голуби, и которые часто оказывается гораздо труднее механизировать, чем так называемые интеллектуальные навыки.

Наконец, если имеющаяся информация надежна и четко задана, то использование экспертной системы не рекомендуется. Если же доступные данные "замусорены", то экспертные системы — это как раз то, что вам нужно. Тогда начнет играть нечеткая, неточная или какая-нибудь "нечистая" логическая схема со смешным названием.

## 2.8. ВОПРОС ВЫБОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Имеется широко распространенное, но ошибочное мнение, что экспертные системы должны быть обязательно написаны на языке либо Лисп, либо Пролог.

Полезность языка Лисп оправдывается его использованием в течение более 25 лет в исследованиях по искусственному интеллекту, но то, что он не находит коммерческого применения, случайностью не является. Он процветает в академической среде, присущей Массачусетскому технологическому институту, Станфордскому университету, Университету Карнеги-Меллона (США), поскольку создает условия для обмена идеями. Вы можете взять систему сопоставления образцов Джо Хекера, добавить систему доказательства теорем Майка Мак-Куэрти и подсоединить систему молодого Карла Уизкида для разбора фраз естественного языка. К этому времени вами проделано уже более половины работы (конечно, если эти системы стыкуются между собой).

Но за пределами подкультуры, связанной с университетским искусственным интеллектом, вы предоставлены сами себе. Язык Лисп, по сути, — это не более чем комбинация команд CAR, CDR, COND и CONS (плюс еще EVAL). Все интересные элементы языка являются результатом добавлений, которые следуют философии исследований по искусственному интеллекту, но которые, по существу, являются дополнительными пакетами программного обеспечения. Последние не стандартизованы, а в Великобритании большинство из них к тому же и недоступны. Так что получив Лисп, вы не так далеко уйдете.

Что касается языка Пролог, то мой совет — забудьте о нем. Это широко рекламируемый язык программирования, причем рекламируемый некоторыми как основа новой эры, грядущей в области развития матобеспечения машин. Если японцы и в самом деле примут его без радикальной модификации (а есть признаки того, что они этого не сделают), то Западный мир сможет спокойно отдыхать до конца этого века\*. Пожалуй, самое доброе, что можно сказать о языке Пролог, — это то, что он оказался впереди своего времени. А теперь перечислим несколько недобрых высказываний об этом языке:

1. Пролог содержит лишь весьма слабые средства защиты от незначительных ошибок в написании команд, которые могут приводить к фантастическим нежелательным последствиям в работе программ.

2. Пользователю необходимо понимать детали реализации встроенного механизма возвращения при написании программы, так что это не чисто "логическое программирование".

3. Порядок предложений существует для их значения — снова противоречие смыслу логического программирования.

4. Многие из встроенных предикатов дают побочные эффекты, вследствие чего этот язык не пригоден для параллельной обработки.

---

\* Имеется в виду так называемый "японский вызов", в рамках которого в Японии предлагается создание ЭВМ пятого поколения. В проектах машин нового типа предлагается в основном использовать языки высокого уровня Лисп и Пролог. Подробнее об этом проекте и аналогичных ему западных проектах см. в следующей главе. — *Прим. ред.*

5. Пролог "бесплатно" обеспечивает реляционную базу данных — большое достоинство языка; но беда заключается в том, что она располагается в главной памяти машины, из-за чего язык нуждается в большой машинной памяти.

6. Все в этой базе данных носит глобальный характер, в ней нет локальных фактов или модулей в привычном смысле.

7. Поиск в глубину введен там с самого начала, хотите вы этого или нет. Если вам потребуется что-то более изощренное, то вам придется "прорубать" через джунгли, отсекая ветви имеющегося поиска.

8. Пролог уже содержит множество нестандартных "улучшений", короче, Пролог — это "мечта лесоруба", поскольку лишь преданная этому языку элита в состоянии овладеть его элегантными суперсложностями. Заявление, что это язык логического программирования, просто не выдерживает серьезного анализа. Некоторые из моих друзей используют Пролог, но я и не приронусь к терминалу. Когда вы слышите, что какие-то школьники свободно им пользуются, вы можете быть уверены, что их обучали лишь подмножеству языка, в котором отсутствуют "усложняющие детали" (и, следовательно, это подмножество не годится для построения серьезных программ).

Как сказали Фейгенбаум и Маккордак: "Специалист по использованию знаний лишь в самую последнюю очередь доверит управление "автоматическому" процессу доказательства теорем, осуществляющему массивный поиск без контроля шаг за шагом со стороны знаний, находящихся в базе знаний" [1].

Если вам непременно необходимо использовать Пролог, то подождите, пока он будет "проглочен" языком Бейсик, этим питоном, пожирающим все, что лежит на его пути. Бейсик только что закончил "переваривать" язык Паскаль со всеми его управляющими структурами. После небольшой паузы и нескольких "отрыжек" он будет в состоянии "слопать" Пролог, так что мы увидим варианты языка Бейсик со встроенным механизмом доказательства теорем методом резолюции. Тогда наступит время подумывать о возможном переводе на Пролог. (Однако см. гл. 7.)

## 2.9. СДЕЛАЙ САМ

Самым правильным будет воспользоваться тем языком, который вы знаете, на машине, которая у вас есть. Одним из относительно дешевых решений будет приобретение оболочки экспертной системы (например, "Микроэксперт" или Nexus) и использование ее в качестве прототипа, пока вы не поймете, чего хотите. Затем можно будет написать вариант с использованием продукций на эффективных и компактных языках, подобных языкам Си, Фортран 77, Паскаль или даже Форт.

Нет необходимости дожидаться реализации мечты о создании машин пятого поколения. Если ваш компьютер не больше чем VAX, но и не меньше чем Sinclair QL, то он вполне годится. Машины типа IBM PC или Apple Macintosh, особенно снабженные жестким диском, будут самыми подходящими.

## 2.10. ЗАКЛЮЧЕНИЕ

Когда речь идет о системах, использующих знания, неведение ни в коем случае недопустимо. Никто в информационном бизнесе не может себе позволить занять роль выжидающего наблюдателя, ибо будущее уже началось.

## Глава 3

### ПРОЕКТ МАШИН ПЯТОГО ПОКОЛЕНИЯ

*Анни Брукинг*

#### 3.1. ВВЕДЕНИЕ

В конце 1978 г. Министерство внешней торговли и промышленности (МВТП) Японии поручило электротехнической лаборатории разработать проект вычислительных систем 90-х годов под названием "Пятое поколение". Предполагалось, что проект не должен конкурировать с направлением, выбранным фирмой IBM. Кроме решения чисто коммерческих задач проект должен был способствовать росту международного престижа Японии.

После тщательного продумывания характера различных комитетов было решено приступить к проекту в 1981 г. Формально к выполнению проекта приступили в 1982 г.

Под руководством Хадзаме Карацу был создан коллектив специалистов, в задачи которого входило определение функциональных требований к новой машине. Для этого они попытались спрогнозировать характер общества, которое должна будет обслуживать новая машина в следующих десятилетиях.

Проделав такую работу и сопоставив результаты с теми направлениями, в которых компьютеры уже используются, специалисты пришли к заключению, что на пути удовлетворения прогнозируемых "социальных нужд" существуют различные препятствия, которые могут быть устранены или обойдены благодаря применению новой машины. Среди задач будущего общества имеются такие:

1. Улучшение дел в областях с низкой производительностью.
2. Включение Японии в международное сотрудничество.
3. Учет нехватки энергии и истощения полезных ископаемых.
4. Высокий средний возраст и высокий уровень образования (образованное, но старое общество).
5. Создание информационного общества для людей.

Решение этих задач нового общества может показаться весьма напряженным вследствие слишком большой дистанции между людьми и существующими машинами. Поэтому функцией машины пятого поколения является ликвидация этого несоответствия.

Как сказал Х. Карацу: "...люди — существа мягкие. Поэтому много неприятностей может возникнуть у мягких человеческих существ при контакте с этими твердыми машинами. С этой точки зрения, машина пятого



поколения должна работать так, чтобы во всем соответствовать человеку, в противоположность сложившейся ситуации, когда человек должен осваивать машину, подлаживаться под нее и следовать ее правилам”.

Доктор Кадзухиро Фути (глава Токийского института вычислительной техники нового поколения (ICOT), центра по созданию ЭВМ пятого поколения) продолжает эту мысль: ”Много раз высказывалось неудовлетворение имеющимися компьютерами. Одна из претензий состоит в том, что сегодняшнюю технологию никак нельзя считать ”ручной”.

Затем он говорит, что причина того, что технология не является ”ручной” — отсутствие развитого человеко-машинного интерфейса (ЧМИ). Большинство согласится с тем, что графическая или языковая связь была бы предпочтительнее алфавитно-цифрового ввода-вывода. Однако один из факторов, усложняющих дело, состоит в том, что любое решение для требуемой задачи должно быть запрограммировано.

Если имеется возможность собрать информацию о предметных областях программы и те законы, которым подчиняются эти области, то от этого возможности системы по решению проблем могут сильно расшириться. Такие знания соответствуют знаниям, имеющимся у человека. Такая точка зрения соответствует тому отношению, которое давно сложилось в Великобритании и других странах у исследователей, работающих в области искусственного интеллекта. Для того чтобы машины оказались по-настоящему ”ручными”, необходимо иметь более естественный ЧМИ; на самом деле он должен походить на интерфейс, характерный для общения между людьми. В результате в японском проекте возникает множество трудных проблем, среди которых имитация функций сенсорного восприятия человека, таких как зрение, осязание, синтез речи и понимание речи. Для выполнения этих функций необходим гигантский запас соответствующих знаний. Простая обработка поступающих данных не обеспечила бы тот уровень сложности, который необходим для выполнения таких в высшей степени сложных функций. С этой целью в машинах пятого поколения предусматривается разработка базовой технологии как на уровне создания аппаратных средств, так и программного обеспечения.

## 3.2. ОБЗОР ПРОЕКТА

Проект машин пятого поколения охватывает весьма широкую область, поскольку его задачей является создание альтернативной архитектуры и программного обеспечения. В проекте было выделено семь исследовательских направлений (табл. 3.1), которые являются чрезвычайно сложными, и для их адекватного охвата потребуются тысячи человеко-лет.

Вычислительная система пятого поколения будет ориентирована на обработку знаний и будет располагать весьма развитыми возможностями логического вывода. Важнейшая ее черта должна состоять в том, чтобы используемый интерфейс был непосредственно рассчитан на человека. Программная моделирующая система по замыслу должна весьма эффективно использовать аппаратные средства и выполнять в основном метафункции для реше-

## Темы научно-конструкторских работ по вычислительной системе пятого поколения

<i>Основные прикладные системы</i>	1.1. Система машинного перевода 1.2. Вопросо-ответная система 1.3. Прикладная система понимания речи 1.4. Прикладные системы понимания изображений и рисунков 1.5. Прикладная система решения проблем
<i>Основные системы программного обеспечения</i>	2.1. Система управления базой знаний 2.2. Система решения проблем и логического вывода 2.3. Система интеллектуального интерфейса
<i>Новая развитая архитектура</i>	3.1. Машина логического программирования 3.2. Функциональная машина 3.3. Машина реляционной алгебры 3.4. Машина поддержки абстрактных типов данных 3.5. Машина потоков данных 3.6. Обновленная машина фон Неймана
<i>Архитектура систем с распределенными функциями</i>	4.1. Архитектура системы с распределенными функциями 4.2. Сетевая архитектура 4.3. Машина базы данных 4.4. Быстродействующая машина для численных расчетов 4.5. Высокоуровневая система человеко-машинного общения
<i>Технология СБИС</i>	5.1. Архитектура СБИС 5.2. Интеллектуальная система машинного проектирования СБИС
<i>Системотехника</i>	6.1. Интеллектуальная система программирования 6.2. Система проектирования баз знаний 6.3. Системотехника для архитектуры ЭВМ 6.4. Система базы данных и распределенной базы данных
<i>Техника обеспечения разработок</i>	7.1. Система обеспечения разработок

ния проблем, таких как понимание и синтез программ. Предполагается, что система будет в состоянии воспринимать естественный язык, фотографии, изображения и т. п.

Это будет, однако, не одна машина пятого поколения, а несколько, причем каждый модуль будет состоять из нескольких программных или аппаратных компонент, ориентированных на выполнение отдельных функций. В результате пользователь получит возможность сформировать конфигурацию речевой машины с интересующей его функцией или робота с развитым зрительным восприятием и т. д.

Однако успех в создании таких модулей в конечном счете зависит от качества информационной обработки знаний и характера ее использования. В этом отношении можно сказать, что успех проекта пятого поколения зависит от успешного применения методов из каждой проблемной области.

Аппаратное решение, обеспечивающее основу такого рода деятельности, не будет классической архитектурой машины фон Неймана. Уже проведены исследования по языкам, более приспособленным для программирования методов решения задач. В настоящее время Пролог является единственным хорошо известным языком обеспечения процедуры решения проблем. С учетом опыта сторонников применения логического программирования японские специалисты разработали новый язык (он будет называться NIMIKO), который более адекватно отражает логические концепции. Первоначально обработка будет осуществляться на последовательной машине PSI — персональной последовательной машине вывода. На более поздних этапах развития проекта должна быть создана параллельная машина. Предполагаемое быстроедействие такой машины будет измеряться числом логических выводов в секунду (ЛВС). Намечаемая производительность исчисляется в ГЛВС:

1 ГЛВС =  $10^9$  ЛВС;

1 ЛВС = между 100 и 100 000 миллионов машинных операций в секунду, что примерно в 30 000 превышает производительность процессора DEC KL-10, используемого сегодня.

Концептуальная структура машины пятого поколения, построенная "сверху-вниз", отражена в табл. 3.2. Как уже отмечалось, не все элементы будут включаться во все планируемые машины, но таблица позволяет почувствовать потенциальную сложность машины.

Для каждого из семи направлений был составлен план исследований и разработок, состоящий из трех этапов: начального, промежуточного и заключительного. Продолжительность начального этапа три года. Его задачами являются:

Создание высокопроизводительной персональной Пролог-машины.

Создание машины реляционной базы данных.

Проведение фундаментальных исследований.

Предполагалось, что первые результаты появятся в апреле 1985 г.

### 3.3. ДОСТИГНУТЫЕ РЕЗУЛЬТАТЫ

По характеру реализации проекта машин пятого поколения чрезвычайно трудно точно определить степень успеха на каждом этапе.

Официально ответственным за проект является лишь институт ICOT. Однако большинство исследований ведется не в этой организации. (В момент написания настоящей статьи в институте насчитывалось, вероятно, 30–40 исследователей.) Большинство исследовательских работ осуществляется фирмами, работающими над некоторыми конкретными этапами проекта, согласованными с МВТП. Среди них такие промышленные гиганты,

## Концептуальные элементы машины, расположенные в порядке их усложнения

Три слоя	
Язык пользователя (речь, естественный язык, рисунок)	
Понимание и синтез речи	
Знания (область языка и рисунков)	
Отклик на уровне промежуточной спецификации	
Интеллектуальные системы программирования	
Понимание проблем и выработка отклика	
Знания (проблемная область)	
Система базы знаний	
Синтез и оптимизация программ	
Модель машины знаний (представление знаний)	
Язык логического программирования	
Машина решения проблем и логического вывода	
Машина базы знаний	
Интерфейс с машинами четвертого поколения	
Машина для манипулирования символами	
Машина для численных расчетов	
Машина базы данных	

как FUJITSU, MITSUBISHI и NEC. Однако в отличие от других национальных и международных проектов эти фирмы не только сотрудничают, но и конкурируют. Это означает, что многие результаты исследований остаются секретами фирм. Таким образом, проект машины пятого поколения имеет несколько этапов, одни из которых являются открытыми, а другие — закрытыми для публики.

Одним таким видимым результатом является машина PSI, три экземпляра которой были построены фирмой MITSUBISHI. Первая машина была передана в институт ICOT в декабре 1983 г. и предназначалась для того, чтобы успеть закончить первый уровень программного обеспечения к моменту открытия второй конференции по машинам пятого поколения, состоявшейся в ноябре 1984 г. в Токио.

Достаточно перспективными представляются различные исследования, ведущиеся в области распознавания речи и ее синтеза. У японцев есть особый интерес к такого рода системам, поскольку их письменный язык является слишком сложным для реализации на вычислительной машине. Новые подходы к обработке слов, включая ввод FAX — фототелеграфный (факсимильный) ввод с высокой степенью разрешения, несомненно, серьезно отразятся на положении Японии на международном рынке.

#### 3.4. РЕАКЦИЯ ВЕЛИКОБРИТАНИИ

Япония предложила нескольким странам, включая Великобританию, направить в Японию группы ученых для обсуждения их участия в японской программе машин пятого поколения. В результате в Великобри-

тании был образован комитет, возглавляемый Джоном Алви из фирмы British Telecom. В группу из двенадцати человек вошли ученые и специалисты по технологии вычислительной техники, работающие в британской промышленности и правительственных учреждениях. Единственным приглашенным человеком из "академических кругов" оказался Роджер Нидхем — директор вычислительной лаборатории Кембриджского университета. Бросается в глаза отсутствие в этом комитете Дональда Мичи — пожалуй, наиболее известного британского специалиста по искусственному интеллекту, и Алекса Д. Агапьефа — председателя первой группы по экспертным системам Британского вычислительного общества. Только сейчас стало ясно, что ни один из членов этого комитета не имеет достаточного опыта в области искусственного интеллекта.

Тем не менее группа Алви посетила первую конференцию по машинам пятого поколения, на которой японцы объявили свою программу. Масштаб и связанность элементов программы воспринимались как главная конкурентная угроза. Кроме того, ожидалась реакция и со стороны США, что создаст дополнительную угрозу промышленности Великобритании, обеспечивающей выпуск информационной технологии.

После возвращения в Великобританию эта группа провела исследование тех областей, в которых возможно совместное проведение научных работ. Результаты были опубликованы в сентябре 1982 г. и получили название Отчета Алви.

В отчете выделены четыре ключевые области, в которых, по мнению комитета, необходимо добиться наибольшего технического прогресса, если в будущем планировать более широкое приложение информационной технологии. Такими областями являются:

Разработка программного обеспечения.

Сверхбольшие интегральные схемы (СБИС).

Интеллектуальные системы, основанные на знаниях (ИСОЗ).

Человеко-машинный интерфейс (ЧМИ).

Предполагается, что выдвинутая программа, целью которой является совершенствование информационной технологии, будет осуществляться на основе тесного взаимодействия промышленности с научно-исследовательскими организациями. Подобно проекту создания машин пятого поколения, программа является национальной, нацеленной на придание нового импульса британской промышленности, связанной с информационной технологией, чтобы в будущем она могла конкурировать с промышленностью Японии и США. Сам отчет был представлен в правительство Великобритании, которое проголосовало за выделение 350 млн. фунтов стерлингов на выполнение программы, рассчитанной на пять лет. Было назначено Правление Алви во главе с Брайаном Оукли из фирмы СВЕ, бывшим руководителем Научно-технологического исследовательского совета (SERC). Были назначены также четыре управляющих в каждой из четырех областей исследований.

По каждому из четырех направлений был проведен необходимый анализ с целью определения возможных областей развертывания исследовательских проектов:

1. Исследование архитектуры ИСОЗ.
2. Разработка программного обеспечения.
3. Стратегия в области СБИС.
4. Стратегия в области ЧМИ.

Результаты анализа, в которых имеются четкие указания на содержание и объем будущих проектов Алви, стали доступны широкой общественности, и промышленные и научные круги были приглашены для участия в их обсуждении. В частности, исследование по архитектуре ИСОЗ, которое было опубликовано первым, дает описание стратегии предполагаемых работ в Великобритании в области интеллектуальных систем, основанных на знаниях.

#### 3.4.1. Программа исследований и разработок

Эта программа рассчитана на 5—10 лет и охватывает пять основных видов деятельности, имеющих различные цели и сроки:

**Иллюстративные проекты типа "Покажи мне"**, позволяющие за короткое время продемонстрировать промышленные образцы техники ИСОЗ.

**Краткосрочные конструкторские проекты**, нацеленные на создание за 2—3 года предназначенных для продажи изделий, основаны на разработках, отталкивающихся от существующей технологии и поэтому не связанных с большим риском.

**Демонстрационные проекты** — научно-исследовательские и конструкторские разработки под контролем промышленности, направленные на построение полных прототипов возможных будущих систем, для создания которых необходим существенный прогресс в науке и технике.

**Исследовательские темы**, включающие более или менее продолжительные исследования, проводимые сотрудничающими коллективами и ориентированные на ограниченное число тщательно отобранных вопросов, которые с большой вероятностью являются основными для успешного построения развитых ИСОЗ.

**Фундаментальные исследования.** Прогресс в указанных частях программы исследований и разработок возможен лишь на базе сбалансированного запаса высококачественных поисковых исследований во всех важных аспектах ИСОЗ и связанных с ними вопросах, таких как техника разработки программного обеспечения и создание развитого машинного взаимодействия.

Приблизительно через год после публикации отчета были заключены первые контракты по изучению целесообразности демонстрационных проектов. Предполагается, что эти проекты должны контролироваться промышленностью. Все они являются крупномасштабными. В некоторых случаях консорциум включает более восьми организаций (всего до 30—40 человек). Однако они не должны заниматься фундаментальными исследованиями,

их задача — лишь выделить вызванные необходимостью потребности. В момент написания этих строк были заключены первые контракты по исследованию целесообразности. В то же время более мелкие проекты и фундаментальные исследования еще не получили одобрения, хотя в Правление Алви поступило множество предложений.

Одной интересной особенностью проектов Алви явилась высокая доля правительственного финансирования. Считается, что это необходимо для совместных работ, особенно если привлекаются малые фирмы. К тому же академические учреждения финансируются полностью — существенный ингредиент в момент урезания расходов на образование.

Как уже указывалось, в программе Алви предлагается четыре области проведения исследований и разработок. Ниже я приведу достигаемые при этом цели:

#### *(а) Техника программного обеспечения*

В Великобритании всегда были хорошие специалисты по программному обеспечению, и будет справедливо отметить, что Великобритания экспортирует большое число программ и делится опытом в разработках с другими странами.

Однако, несмотря на появление структурного программирования более десяти лет назад, Правление Алви пришло к выводу, что слишком большое доверие оказывается методам разработки *ad hoc*\*. Для того чтобы Великобритания стала мировым лидером в технике программного обеспечения, необходимо создать новые методы.

Исследовательская деятельность в области программного обеспечения концентрируется большей частью на разработке различных поколений интегрированных сред программной поддержки (ИСПП). Интересно, однако, заметить, что в Отчете Алви рекомендуется учредить центр по созданию мат-обеспечения и национальный центр контроля качества.

#### *(б) Человеко-машинный интерфейс*

Поскольку область человеко-машинного интерфейса остро нуждается в многообразных исследованиях, то программа ЧМИ чрезвычайно широка и включает:

Построение диалогов.

Модели вход-выход и их эффективность.

Идентификацию коммуникативных навыков людей с целью улучшения качества систем, предназначенных для консультирования и обучения. Когнитивная совместимость человека и системы — выделение в явном виде несоответствия между пользователем и системой.

Анализ поведения при решении проблем в сложных задачах.

Исследования степени, в которой необходимо, чтобы система имитировала человеческое поведение при решении задач.

Новые устройства ввода-вывода, такие как сканеры с высокой разрешающей способностью, и соответствующие устройства печати.

---

\* В данном случае — специально разрабатываемые в каждой ситуации приемы. — Прим.

Обработка речи и изображения, включая поиск признаков, распознавание и синтез речи.

### (в) ИСОЗ

Комитет считает исследования по интеллектуальным системам, основанным на знаниях, не только трудными, но и рассчитанными на длительный срок. Среди выделенных тем, требующих немедленного внимания, указаны такие:

Представление знаний.

Решение задач путем вывода.

Разработка функциональных языков.

Разработка функциональных машин.

Разработка "умных" баз данных.

Способы извлечения знаний.

Общение на естественном языке.

Построение прототипа ИСОЗ.

По-видимому, программа ИСОЗ сильно зависит от развития других областей, указанных в программе Алви, таких как ЧМИ, новое аппаратное обеспечение, которое, без сомнения, будет создано на базе успехов в области конструирования СБИС.

Однако предстоит провести еще большую теоретическую работу независимо от создания новой аппаратуры для быстрой параллельной обработки. В частности, извлечение знаний по-прежнему остается узким местом в создании экспертных систем, требующим особого внимания.

### (г) СБИС

Вопрос конструирования сверхбольших интегральных схем является весьма важным для перспективных методов обработки информации, поскольку СБИС позволяет на порядок увеличить число обменов сигналами между элементами по сравнению со связанной группой БИС, расположенных на одной печатной плате. Цель состоит в том, чтобы обеспечить Великобритании доступ к конкурентоспособным в международном масштабе СБИС.

Представляется, что существующие средства конструирования далеко не адекватны и не позволяют полностью реализовать потенциал, заложенный в СБИС. По этой причине предлагается проведение программы по исследованию вопросов проектирования с помощью компьютеров. Задача была сформулирована в виде некоторой совокупности средств, которые преобразуют описание системы в три различных результата:

1. Геометрическое задание для производства фотошаблона.
2. Тестовые программы.
3. Задание процесса.

Необходимо также располагать совершенными способами создания кремниевых приборов, причем кремниевая технология должна находиться на мировом уровне.



Особое внимание должно быть уделено следующим направлениям:

- развитию биполярной технологии;
- развитию МОП-технологии и созданию МОП-структур для осуществления цифроаналогового преобразования;
- исследованиям в 0,5-микронной области.

При этом потребуются одновременно решить целый ряд технических вопросов в области:

- электронно-лучевой микротехнологии рентгеновской литографии — прямого последовательного шагового экспонирования (ППШЭ) всей пластины;
- фотолитографии с ППШЭ для прецизионной обработки;
- ”сухого” травления с использованием плазмы и ионной бомбардировки вместо жидких химикалий;
- имплантации ионов;
- эпитаксиального выращивания кремния либо на сапфире, либо на другой изолирующей подложке;
- нанесения слоя металла или диэлектрика на поверхность кремния для создания контактов, межсоединений, формирования рисунка прибора и пассивации;
- разгонки примесей отжигом.

Задачи, поставленные перед исследовательскими программами на следующие пять лет, приведут к массовому выпуску одномикронных демонстрационных кристаллов, отвечающих потребностям программ ЧМИ и ИСОЗ.

Программа Алви явилась первым национальным откликом на проект машин пятого поколения. Структура сама по себе пока прорабатывается и потребуется, по-видимому, по крайней мере еще два года, прежде чем можно будет судить о результатах.

Тем временем Великобритания продолжает участвовать в гонке информационной технологии, но не под своим флагом, а под флагом Европейского экономического сообщества (ЕЭС) по программе ESPRIT (European Strategic Program for Research in Information Technology — Европейская стратегическая программа исследований информационной технологии).

### 3.5. РЕАКЦИЯ ЕВРОПЫ

В Брюсселе — штаб-квартире ЕЭС было принято решение организовать Комиссию, в обязанности которой входило бы нанимать коллективы специалистов для исследования и контроля различных тенденций в разнообразных областях, представляющих интерес для ЕЭС. В начале 80-х годов были примерно определены тенденции на 1984—1987 гг., в структуре которых содержалось шесть главных целевых направлений политики, проводимой сообществом:

- Повысить конкурентоспособность в области сельского хозяйства.
- Повысить конкурентоспособность в промышленности.

Улучшить характер использования энергетических ресурсов.

Усилить помощь развивающимся странам.

Улучшить условия работы и жизни.

Усовершенствовать использование научных и технических ресурсов общества.

На заседании Совета ЕЭС, состоявшемся в Брюсселе 28–29 июня 1982 г., информационная технология была выделена как требующая принятия срочных мер. Затем на встрече глав правительств в Версале в июне 1982 г. лидеры промышленно развитых стран пришли к соглашению о необходимости сотрудничества в новых видах технологии, в частности в информационной технологии.

В документе от 25 мая 1982 г. Комиссия представила Совету доказательства шаткости ситуации, сложившейся в Европе, и показала необходимость создания Европейской стратегической программы исследований и разработок по информационной технологии.

Затем Комиссия обратилась к основным фирмам, занимающимся информационной технологией, с призывом обсудить эту проблему. Эти фирмы стали известны как "большая дюжина" (табл. 3.3).

Каждая фирма направила представителей, которые вместе с Комиссией разработали программу ESPRIT. В некотором смысле можно сказать, что разработка программы ESPRIT не была непосредственной реакцией на план машин пятого поколения. Озабоченность европейских стран была обусловлена не просто японским вызовом, а чем-то, что потенциально может произойти завтра.

За последние несколько лет Европа попадает во все большую зависимость от японского производства интегральных схем и американского програм-

Т а б л и ц а 3.3

Список фирм, входящих в "большую дюжину"

AEG Nixdorf Siemens	ФРГ
Olivetti Stet	Италия
Bull CGE Thompson CSS	Франция
GEC ICL Plessey	Великобритания
Philips	Нидерланды

много обеспечения. В последнее время США стали проявлять большую осторожность в отношении экспорта математического обеспечения, ссылаясь на возможную "утечку" технологии в социалистические страны. В частности, были случаи, когда требовалось потратить около года, чтобы получить разрешение на экспорт некоммерческого программного обеспечения. В момент написания этих строк мы получили информацию, что официальный вариант системы UNIX будет снова доступен за пределами США.

Таким образом, эта "большая дюжина" была озабочена последствиями сокращения потока программного обеспечения из США в Европу, а также возможным осложнением в европейской промышленности, связанной с информационной технологией, если одновременно и Япония решит прекратить поставку в Европу своих интегральных схем.

На политическом уровне также высказывается точка зрения, что тот, кто выиграет в гонке информационной технологии, будет контролировать поток информации. А тот, кто контролирует поток информации, контролирует весь мир!

От всего этого может заболеть голова, но, как я уже отмечала, некоторые симптомы такого положения дел мы наблюдали в этом году. Столкнувшись с такого рода ситуацией, Совет ЕЭС принял решение создать специальное подразделение по информационной технологии, функцией которого является управление вырабатываемой программой. Был разработан план. Решено, что исследования должны вестись по следующим направлениям:

1. Развитая микроэлектроника.
2. Технология программного обеспечения.
3. Развитая обработка информации.
4. Автоматизация в учреждениях.
5. Производственные интегрированные системы, включающие компьютеры.
6. Системы обмена информацией.

С каждым таким направлением была связана группа специалистов, назначенная для выработки плана работ. Эта группа состояла из представителей "большой дюжины" и экспертов в данном направлении. Группы представили начальный рабочий план для первой части программы. Эта часть, известная как начальная фаза, рассчитана на один год и подводит к главной фазе, которая рассчитана на четыре года.

К начальной фазе приступили в июле 1983 г. Было одобрено около 35 контрактов, выполняемых объединениями и представленными в числе предложений на научные исследования в конце февраля 1983 г. Эти контракты довольно равномерно распределились по пяти указанным направлениям, дальнейшая детализация которых привела к формулировке 15 тем научных исследований:

1. Развитая микроэлектроника
  - 1.1. Развитие межсоединения для СБИС.
  - 1.2. Машинное проектирование схем высокого уровня интеграции.

2. **Техника программного обеспечения**
  - 2.1. Мобильная инструментальная среда.
  - 2.2. Формальная спецификация и систематическое построение программ.
  - 2.3. Системы создания и поддержки программного обеспечения.
3. **Развитая обработка информации**
  - 3.1. Развитые алгоритмы и структуры для обработки сигналов.
  - 3.2. Информационные системы управления знаниями.
  - 3.3. Система интерактивного запроса (вопросо-ответная система)
4. **Автоматизация в учреждениях**
  - 4.1. Функциональный анализ условий работы учреждения.
  - 4.2. Пользовательский интерфейс с многими средами для автоматизированных рабочих мест (АРМ).
  - 4.3. Локальная широкополосная система связи.
  - 4.4. Занесение в файлы и извлечение бесструктурной информации в учреждении.
5. **Производственные интегрированные системы, включающие компьютеры**
  - 5.1. Правила конструирования производственных интегрированных систем, включающих компьютеры.
  - 5.2. Интегрированные микроэлектронные подсистемы.
  - 5.3. Контроль процессов и производства, основанный на системах обработки образов в реальном времени.
6. **Система обмена информацией**

ЕЭС обеспечило не только руководство программой ESPRIT, но и необходимое финансирование. Для большинства проектов финансирование составляло 50% требуемой суммы. Такое финансирование затрудняло участие академических организаций, которые могли работать только по контракту (при этом они получали от своего партнера 100% требуемой суммы). Тем не менее в программе ESPRIT участвует много академических учреждений; в то же время имеются контракты, все участники которых работают в промышленности. Для участия в программе ESPRIT требуется, чтобы хотя бы два партнера работали в промышленности. Кроме того, по крайней мере два партнера должны проводить исследования в двух разных странах, входящих в ЕЭС.

На этапе начальной фазы была создана комиссия для детальной разработки рабочего плана на 1–5 лет (начальная фаза считалась нулевым годом). Разработанные планы по пяти направлениям были затем повсеместно опубликованы в Западной Европе в расчете на комментарии и дополнения. Проводились рабочие совещания, на которые приглашались специалисты из числа тех, кто внес наиболее существенный вклад. Такие пересмотренные рабочие планы были приняты в качестве согласованных направлений на следующие пять лет.

Хотя все это сильно затрудняет управление программой ESPRIT из-за гигантского объема информации, с которой должно справляться специальное подразделение (более 1000 предложений было получено только относительно рабочего плана для раздела "Развитая обработка информации"),

все-таки видно, что программа ESPRIT создана и, по-видимому, является весьма демократической программой, выполняемой в Европе и в интересах Европы.

Предложения в отношении главной фазы программы ESPRIT принимались с марта по май 1983 г. Предполагалось, что главная фаза должна начаться ранней осенью 1984 г. Нет надобности говорить о том, что все проекты начальной фазы должны перейти в главную фазу.

### 3.6. РЕАКЦИЯ США

Первое и самое важное в отношении реакции США — это то, что такой реакции нет, по крайней мере, нет национально-координируемого проекта.

Однако США весьма активны во всех областях исследований, относящихся к проекту создания машин пятого поколения. В частности, вокруг Массачусетского технологического института и Станфордского университета возникло несколько предприятий, ставящих своей целью развитие экспертных систем и приложений методов искусственного интеллекта к промышленным задачам.

В одном из обзоров приведен следующий список организаций, активно работающих в области систем, основанных на использовании знаний:

Университеты и институты	Университет Карнеги-Меллона Массачусетский технологический институт Станфордский университет и около десятка других
Некоммерческие организации	JPL Rand SRI
Государственные учреждения	NOSC (г. Сан-Диего, шт. Калифорния) NRL, Лаборатория искусственного интеллекта (г. Вашингтон, федеральный округ Колумбия)
Промышленные фирмы	Aids AMOCO BBN Bell Labs DEC Fairchild Hewlett Packard Hughes IBM Intelligenetics Jaycor Machine Intelligence Corp. Martin Marietta

До настоящего времени правительство было главным источником финансирования работ по экспертным системам. Недавно Управление перспективного планирования НИР Министерства обороны США предложило направление исследований по искусственному интеллекту, смыслом которого является применение методов искусственного интеллекта в военных задачах. Там есть проекты, включающие интеллектуального помощника пилота, интеллектуальные подводные системы и т. п.

На недавно прошедшем совещании в фирме SPL профессор Эвард Фейгенбаум (США) высказал мысль, что американцы относятся с подозрением к общенациональным проектам, выдвигаемым Вашингтоном, поэтому такого рода инициатива на Американском континенте вряд ли будет успешной. Возникнет ли такой национальный проект в США — будущее покажет.

Можно спорить о том, нужна ли такая программа. Американские научные учреждения уже работают совместно с промышленностью в течение какого-то времени. Одним блестящим примером успеха в этой области является партнерство фирмы DEC с Университетом Карнеги-Меллона, которое привело к созданию системы R1 (теперь эта система называется XCON), экспертной системы DEC для выбора конфигурации машин VAX и PDP. Эта система явилась первой из многих таких систем, постоянно используемых в настоящее время фирмой DEC.

Кроме того, похоже, что США довольно хорошо справляются с проблемой соотношения в проведении фундаментальных и прикладных исследований. Это привело к недавнему росту фирм, группирующихся вокруг таких университетов, как Станфордский, в которых выполняемые исследования взяты в качестве прототипов и доведены до уровня рыночного продукта, причем университетам предоставлена возможность спокойно продолжать фундаментальные исследования.

Активность в Кремниевой долине носит просто лихорадочный характер. Несколько фирм ведет работу над альтернативными архитектурами и технологиями СБИС. Пожалуй, самый известный лидер в этой области Г. Амдал планирует поставить на рынок свои полупроводниковые пластины через два года.

Фирмы Херох и Symbolics уже в течение нескольких лет поставляют на рынок АРМ для искусственного интеллекта, причем обе поставляют и разнообразное программное обеспечение, позволяющее быстро разрабатывать прототипы систем. Тем временем японцы пока еще разрабатывают операционную систему для персональной машины вывода PSI. Она не демонстрировалась до конца 1984 г.

Наконец, возникает вопрос о том, что думает такой гигант, как фирма IBM; ответить на него пока трудно. Нет сомнения, что фирма IBM проявля-

ет определенный интерес к исследованиям в области машин пятого поколения, но в чем именно — остается лишь догадываться. По одним таким неточным сведениям, фирма IBM располагает своим собственным институтом, подобным институту ICOT, который уже существует продолжительное время.

### 3.7. ВЫВОДЫ

Наиболее очевидный факт, который проявился благодаря игре в машины пятого поколения, состоит в том, что для игры не хватает людей. В этой области ощущается острый недостаток работников и весьма велик спрос на ведущих игроков.

Поэтому в настоящее время возникают такие задачи:

- а) подготовка следующего поколения исследователей;
- б) разработка специального курса;
- в) формулировка исследовательских проектов;
- г) привлечение различных финансирующих организаций;
- д) написание книг и научных статей;
- е) выступления на конференциях, семинарах и т. п.;
- ж) вовлечение промышленности;
- з) . . . и, разумеется, проведение научно-исследовательских работ.

Во времена, когда безработица носит массовый характер, представляется абсолютно ненормальным, что какая-то небольшая часть нашего общества должна так отчаянно много работать.

Учитывая, что такая ситуация характеризует все страны, вовлеченные в разработку машин пятого поколения, естественно полагать, что выиграет тот, кто располагает большим опытом и опирается на лучшую стратегию.

У европейцев есть программа ESPRIT. Нет сомнения, что она преследует похвальные цели. С другой стороны, проведение совместных межнациональных исследований является весьма трудным делом, требующим большой затраты времени на управление. Если бы сотрудничество было главной задачей программы ESPRIT, то можно было бы сказать, что она оказалась и в самом деле хорошей. Примером успеха может служить объединенный Исследовательский центр, который учредили в Мюнхене, не требуя дополнительного ассигнования, такие фирмы, как ICL, Siemens и Bull. Не будь программ ESPRIT, возможно, что такой центр никогда бы не возник.

Программа ESPRIT выглядит весьма справедливой. Комиссия настроена на ее успешную работу. Нет никакого сомнения, что Западная Европа занялась важным делом и наконец проявила большую заинтересованность.

Великобритания вся в делах. У нас есть проект Алви и программа ESPRIT. Наши эксперты в этой области работают на обе программы и поэтому, вероятно, заняты больше других в Западной Европе. Недавно профессор Э. Фейгенбаум (США) назвал Б. Оукли, главу Правления Алви, "администратором больницы".

Безусловно, в прошлом мы затягивали с процессом перехода наших идей из научно-исследовательской фазы в фазу рыночную. Вопрос сейчас

состоит в том, не опоздали ли мы из-за этого. Я думаю, что нет. Алви призывает к более тесному сотрудничеству промышленности и научных учреждений. Одним из возможных просчетов было бы допустить, чтобы все исследования направлялись и контролировались Комитетом Алви: мы обязаны оставить некоторые фонды и на чисто фундаментальные исследования. Одним из последних интересных событий является заявление, сделанное Б. Оукли о том, что мы намерены сотрудничать с Японией. До этого было не совсем ясно, следует нам или нет разговаривать с японцами, не говоря уже о работе с ними. Пожалуй, в мифе о том, что мы не можем создавать аппаратуру, а японцы не могут создавать программное обеспечение, на самом деле заключено какое-то рациональное зерно!

США всегда активно работали в этой области. Когда исследования по искусственному интеллекту были весьма эффективно приостановлены в Великобритании в результате появления отчета Лайтхилла в 1972 г., США настойчиво продвигались вперед, и некоторые наши лучшие ученые, такие как Майк Брейди, Пат Хейес и Дэвид Уоррен, чтобы что-то сделать, уехали в США. Тот факт, что финансирование в США в основном поступает от военных, означает, что никогда не будет недостатка в аппаратном обеспечении и другом оборудовании, которого так не хватает в Западной Европе.

Наконец, имеется японский проект создания машин пятого поколения. Нет сомнения, что замечание д-ра Фути о том, что компьютеры не являются "ручными", следует считать и правильным, и своевременным. В связи с популяризацией микрокомпьютеров и распространением их в учреждениях, школах и быту мы создаем общество, привычное к использованию плохо сконструированных инструментов. Наступило время кому-то провести функциональный анализ человеческих требований к вычислительным средствам. Инженерия знаний является первым шагом в этом процессе.

Достигнут ли японцы поставленных перед собой целей, в результате чего появятся машины, которые полностью изменят общее восприятие процесса вычислений, нам еще предстоит узнать. Если их ждет успех, то, без сомнения, в их руках окажется львиная доля западного бизнеса в области информационной технологии. С другой стороны, похоже, что Западный мир оказался не в состоянии сформулировать план такого рода, несмотря на то, что почти все необходимые фундаментальные исследования им были проделаны.

Если японцы намеревались увеличить свой национальный престиж с помощью этого проекта, то можно сказать, что этой цели они уже достигли. Западная Европа, Великобритания и США имеют свои проекты по созданию развитой информационной технологии в попытке умалить роль заявки Японии на лидерство в этой технологии.

В конце концов, наиболее интересным остается вопрос о том, удастся ли создать машину пятого поколения в результате этого соревнования или всеобщего сотрудничества. В заключение я хотела бы поблагодарить японцев за начатую ими игру, без чего многие из нас все еще продолжали бы ждать действий от своих партнеров, ничем не занимаясь вообще.



### КАК МЫ СОБИРАЕМСЯ ОЦЕНИВАТЬ РАБОТУ ЭКСПЕРТНОЙ СИСТЕМЫ?

*Антони Стевенс*

Я решил интерпретировать этот вопрос двояко. Во-первых, предположение, что экспертные системы будут работать так, как от них требуется, наводит меня на размышления о возможных при этом социальных последствиях. Во-вторых, мы обсудим технические проблемы, связанные с постановкой эксперимента, позволяющего сравнить достоинства различных экспертов, будь то люди или технические системы. И наконец, я приведу некоторые соображения об экспертных системах, которые возникли у меня при взгляде на них несколько со стороны.

Если мы примем, что экспертные системы выполняют свое предназначение, то можно попытаться понять, как это отразится на нас.

Говоря о технике, весьма соблазнительно забыть о том политическом контексте, в котором она существует. Творческая мысль писателя при этом может быть настолько занята объяснением технических вопросов, что он способен забыть об их последствиях, которые могут и не быть во всем хорошими или во всем плохими. Но это возможно только для автора научно-фантастического произведения, который, как правило, ориентируется на описание последствий для "человека", а не для "трудящегося" или "руководящего" человека.

#### 4.1. ВЛИЯНИЕ ЭКСПЕРТНЫХ СИСТЕМ НА ОБЩЕСТВО

Каждому правительству следует руководствоваться конкретными (количественными) данными, осуществляя управление обществом. Так, если даются сведения, отражающие число рабочих, которые должны потерять работу, то это число может быть получено использованием некоторой цепочки рассуждений. Конечно, нам не известно, была ли на самом деле использована такая цепочка рассуждений, или это число просто согласуется с ней. Но мы можем быть уверены в том, что если у некоторого политического деятеля затребуют оправдания проводимой им политики, то ответ должен содержать нечто более серьезное в ее поддержку, чем просто соображения морального или политического характера.

Предположим, что некоторая администрация уделила большое внимание вопросам планирования. Общественность вправе попросить администрацию обосновать ее подход к решению поставленной задачи. Если она этого сделать не может, то у нас есть все основания думать, что эта администрация просто цинично отнеслась к той задаче, которую она сама себе поставила.

Предположим, что соответствующая информация общедоступна и что используемое администрацией рассуждение может быть внесено в вычислительную машину. Если его нельзя подходящим образом закодировать, то мы можем прийти к выводу, что оно нелогично и, следовательно, не стоит

нашего внимания. А если такое кодирование возможно, то что нам мешает создать экспертную систему, воплощающую такого рода рассуждение? Ясно, что существуют вечные технические трудности, связанные с созданием экспертных систем: туманные объяснения, опасность комбинаторного взрыва\* (основной пункт критики Джеймса Лайтхилла в адрес искусственного интеллекта, который, я уверен, будет заново обнаружен в этой области через год-два) и затруднения с приданием рассуждениям человека формы, пригодной для использования в какой-нибудь оболочке экспертной системы.

Однако при умозрительном рассмотрении возможного влияния экспертных систем необходимо полагаться на предположение, что эти трудности будут становиться менее существенными по мере развития техники. Мы предполагаем, что требуемая техника будет создана, и хотим предложить вариант того, как следует управлять обществом, в виде некоторой игры, в которой вы можете держать палец на кнопке "избыточность", пока уровень других переменных не станет приемлемым. Тогда "модель богатства" или что-то в этом роде станет игрой, в которую каждый сможет поиграть на домашнем компьютере.

Если администрация располагает знаниями и если экспертные системы способны делать то, что обещают их создатели, то такие знания в свое время будут размещены на одной интегральной схеме (ИС).

Нельзя предсказать, что будет говорить нам эта ИС. Наиболее вероятно, что такая ИС уведет нас в лабиринт соображений технического порядка. Состояние общества будет выглядеть как сложное вычисление с использованием электронных страниц, и каждый обращающийся к нему человек будет хорошо представлять себе условность тех предположений, на которых все это основано. Вместо чувства, что мы все понимаем о происходящем, мы получим уверенность, что ни у кого нет веских оснований для принятия определенных решений.

Если, следовательно, существуют экспертные системы, дающие экономические или политические рекомендации, то как следует судить об их достоинствах? Если мы говорим, что некоторая экспертная система получит больше очков, если вызовет радикальные изменения, значит ли это, что мы приветствуем экспертную систему, если она способствует росту анархии?

Такие же рассуждения приложимы и ко всем профессиям, где главным продаваемым товаром в настоящее время являются интеллект и опыт. Ясно, что эффект не будет всегда одинаков. Нам по-прежнему будут нужны оперирующие хирурги, тогда как потребность в специалистах по внутренним болезням значительно снизится\*\*.

---

\* Понятие "комбинаторный взрыв" используется для описания тех ситуаций, в которых число возможных состояний в многошаговой задаче растет экспоненциально с числом возможных вариантов в каждой точке выбора (см. [4]).

\*\* Напомним, что система INTERNIST, которая уже в течение ряда лет успешно эксплуатируется в клинике, способна диагностировать около 400 заболеваний внутренних органов человека, делая это на уровне хорошего врача. — *Прим. ред.*

Если экспертные системы будут такими, как об этом говорят, или хотя бы будут содержать строгие и формальные утверждения из областей, требующих профессионального опыта, то они радикально изменят соотношение знаний в нашем обществе и приведут к перераспределению ролей членов общества.

Приведем несколько кратких замечаний о возможном влиянии экспертных систем на нашу культурную жизнь. Я использую термин "культурная жизнь" в широком смысле, понимая под ним то, о чем мы говорим или чем мы наслаждаемся, когда не работаем. Обычно принято считать, что разговор людей — это обмен знаниями, характеризующими эрудицию участников беседы. Поэтому возникает вопрос, не приведет ли исчезновение понятия эрудиции в связи с широкой доступностью знаний к тому, что люди растеряются, не зная что нового сообщить собеседнику. Всегда приятно иметь в запасе что-то интересное для сообщения собеседнику, поэтому интересно знать, как вы себя почувствуете, если какая-то ИС окажется в этом отношении лучшим, чем вы, собеседником за обеденным столом?

Повсеместная и широкая доступность информации, затрудняющая сохранение оригинальности отдельных людей, напоминает нам представление о большой деревне. В значительной степени психология человека связана с тем, что, как он полагает, происходит у него в мозгу. Было бы любопытно увидеть, что случится, если у этого мозга появится достойный конкурент.

Наше отношение к компьютерам сильно изменилось за последние двадцать лет. Рассмотрим разницу между машиной HAL [3] и компьютером космического корабля в "Руководстве по туризму в Галактике на попутных" [1]. HAL — это компьютер, управляющий космическим кораблем в кинофильме "Космическая Одиссея: 2001 год"\* . Относительно него все было ясно и логично. Как зрители, мы легко приняли идею, что нечто столь ясное и логичное непременно уничтожит людей. У компьютера HAL была только одна программа, и она не могла не содержать ошибок. В случае космического туризма мы уже попадаем в эпоху существования программного генератора. Ни одна программа не является совершенной, но мы можем создать несколько ее вариантов (эти варианты соответствуют тем разным личностям, которыми становится бортовой компьютер) и можем посмотреть, в состоянии ли мы ужиться с недостатками каждого из них. Он не может нас убить — в самом худшем случае мы убедимся только, что данный вариант программ является неудачным.

Пожалуй, привлекательность робота по имени Марвин является признаком того, что ждет нас в будущем. Размер его мозга равен целой планете, и он постоянно находится в подавленном состоянии. Эти две характеристики, совмещенные в одной личности, возможно, станут более распространенным культурным стереотипом, чем-то вроде робота—аналога типа "человек ниоткуда".

---

\* Фильм режиссера Кубрика (США) с таким же названием демонстрировался в Москве на VI Международном кинофестивале (1969 г.). — Прим. ред.

Возможно, читатель, привыкший к чтению работ более технического характера, будет раздражен той легкостью, с которой я использую слова "культурный", "политический" и "экономический", и может задать вопрос, какими данными я располагаю в поддержку того, что я здесь говорю? Однако же в кругах специалистов по вычислительной технике были приняты термины, подобные терминам "эксперт" и "инженер знаний", которые используются в течение ряда лет без предварительного тщательного их анализа.

#### 4.2. ОЦЕНКА ЭКСПЕРТНОЙ СИСТЕМЫ

Теперь перейдем к оценке экспертной системы как создаваемого продукта. Выполняет ли программа то, что ей положено?

Когда мы занимаемся оценкой чего-либо, мы, естественно, хотим сравнить работу системы с конкурирующим или стандартным устройством. Хотя некоторые вопросы, связанные с таким сравнением, будут рассмотрены ниже, сначала остановимся на ситуации, когда в магазине имеются две похожие или почти похожие экспертные системы, ориентированные на решение одной и той же задачи. Скорее всего, мы остановимся на той, которая будет получена нами быстрее, хорошо сочетается со стоящим дома компьютером и которой легко пользоваться.

Зависит ли какой-либо из этих факторов, способных повлиять на наш выбор, от состояния исследований по экспертным системам? Ясно, что если мы не можем скомпоновать экспертную систему, тогда нам нечего продавать, но является ли создание экспертной системы как части поставляемого продукта настолько трудным? Быстродействие интегральных схем возрастает, а программное обеспечение становится все более доступным. Столь ли критична зависимость успеха продукта от времени отклика системы? Может оказаться, что мы потратим громадное количество денег на научные исследования по экспертным системам, чтобы оказаться на уровне японской разработки (которой, как некоторые считают, придается неоправданно большое значение) только для того, чтобы в результате обнаружить невозможность продавать свои разработки из-за неблагоприятных условий на рынке сбыта.

Возможно, нам лучше забыть об экспертных системах и сосредоточить усилия на создании дешевых и надежных автомобилей. Безусловно, рынок сбыта автомобилей поддается количественной оценке, но, по-видимому, никто не может или не проявляет желания оценить возможности сбыта экспертных систем в будущем\*. Ясно, что экспертные системы будут созданы, но будут ли они приносить доход?

Теперь предположим, что вы приобретаете экспертную систему, поскольку она позволяет более совершенным образом обнаружить залежи урана. Как предотвратить продажу поставщиком той же самой системы вашему конкуренту или, если он этого не сделает, как помешать конкуренту заказать свою собственную экспертную систему?

---

\* В периодическом международном журнале *Expert Systems* содержится обширная количественная информация о рынках сбыта экспертных систем. — *Прим. ред.*

Иными словами, возможна ситуация, когда у нас будут две относительно одинаковые экспертные системы, ищущие одни и те же урановые руды. Кто-нибудь выиграл? Это зависит от вашего отношения к тому, что является источником богатства. Если в мире имеется достаточное количество урановой руды, так что работа двух экспертных систем, охотящихся за ураном, приведет к тому, что будет обнаружено вдвое больше руды, то тогда экспертные системы действительно участвуют в умножении богатства. Если же, наоборот, мы находимся в ситуации, когда все экспертные системы ищут одно и то же, то разве хотели бы мы, чтобы ресурс истощился быстрее?

Это, очевидно, очень упрощенный взгляд на экономический эффект, порождаемый интеллектуальным продуктом, но мы коснулись полезного соображения, о котором не следует забывать при попытке создания жизнеспособного продукта: не ориентироваться на задачи, носящие разовый характер. Возможно, по этой причине экспертные системы достигли заметного успеха в медицине: всегда будут люди и у них всегда будут болезни.

Экспертные системы можно разделить на две группы в соответствии с тем, используются ли в них рассуждения, основанные на вероятностных соображениях. Если такие рассуждения не используются, то вопрос оценивания работы экспертной системы должен решаться весьма просто: мы либо согласимся, либо не согласимся с цепочкой рассуждений, поскольку результаты действий такого рода экспертных систем должны обладать силой строгой логики.

Множество разнообразных трудностей возникает, если экспертная система пользуется псевдовероятностными соображениями. В этих случаях мы не можем гарантировать, что заключения системы всегда истинны, поскольку отсутствует теоретическая схема, позволяющая убедиться, что получаемое нами заключение является наилучшим из всех возможных. Эта проблема ждет внимания со стороны нового математического гения.

Наука о статистике содержит множество технических результатов, обосновывающих преимущества какого-то одного метода оценки по сравнению с остальными конкурирующими методами. Эти результаты основываются на математических доводах, которые связаны с определенными предположениями относительно контекста, в котором производится оценка. Может быть, понятие наилучшей оценки является иллюзией или идеализацией, которая редко подвергается проверке, поскольку различные, но относительно хорошие практические методы всегда согласуются друг с другом. Тем не менее статистики, по-видимому, не могут не искать нечто, что, вне всяких сомнений, является наилучшим.

При уже укрепившихся в нашем сознании высоконаучных традициях мы видим, что техническая оценка экспертных систем для нас затруднительна. Это связано с тем, что с самого начала мы знаем, что изучаемое поведение в некотором отношении является несовершенным, а определить, что есть совершенство, — нелегкое дело. В условиях, при которых проводится статистическая оценка, всегда предполагается, что существует возможность для строгой проверки исходных математических предположений, а уверенность

в результате закрепляется с ростом числа экспериментов. Дело выглядит так, как если бы мы прокладывали путь к определенности, несмотря на то, что определенность фактически означает, что ничего неизвестно. В случае же экспертной системы или человека это совсем не так.

С самого начала нам следует признать, что работа экспертной системы не будет носить совершенный характер. Более того, мы не знаем, насколько далеки от совершенства. Если задача экспертной системы — достичь больше того, чем в течение многих лет занимались компетентные статистики, то система должна демонстрировать свою компетентность по большому числу возможных вопросов. Это означает, что наше понятие того, что представляет собой хорошая работа, должно быть представлено как сочетание способности выполнять различные задачи, некоторые из которых являются простыми, а некоторые — вовсе нет.

Как из этих различных оценок качества работы скомпоновать одну общую меру качества? Нам не избежать требования давать некоторую оценку по мере того, как нарастает необходимость придать какой-то формальный статус в обществе тому, что говорит нам экспертная система.

Куда приводят нас такие размышления? Создатели экспертных систем заявляют, что возможно написание программы, которая может по исходным сведениям выдать надежный совет способом, относительно которого любой статистик будет чувствовать себя весьма неуютно. (Некоторые из таких приемов, включая использование нечеткой логики, обсуждаются во второй части этой книги.) Почему, можем мы спросить, по этому пути еще раньше не пошли статистики? Имеется достаточное число статистиков, которые свободно оперируют и методами статистики, и вычислительными машинами, чтобы предположить, что недостаток понимания остановил их от использования методов нечеткой логики. Скорее всего, они не пошли по этому пути по той причине, что чувствовали, что он приведет их к ситуации, в которой они будут сравнивать между собой несовершенные результаты. Они были бы не в состоянии понять, что же им следует в такой ситуации измерять.

Для оценки качества экспертной системы предлагалось много методов. Если их описывать простыми словами, то в них учитывалось то, насколько успешно экспертная система приходит к верному решению.

Если нам нужно решить, насколько монета является несимметричной, мы начнем подбрасывать ее, сравнивая число выпадения герба с теоретическим ожиданием, и определим по полученным данным соответствующую вероятность. Аналогичная линия рассуждения позволила бы нам решить также, достаточно ли у нас информации, чтобы из того, что у некоторой монеты выпало гербов больше, чем у другой, прийти к выводу о ее несимметричности.

Для сравнения двух экспертных систем такой теоретической основы не существует. Если в результате некоторой модификации число раз, когда система дает верный ответ, увеличивается с 60 до 65 %, как нам узнать, является ли это увеличение значительным? Мы не в состоянии даже перечислить все возможные результаты работы такой программы (рассматри-

вая при этом цепочку рассуждений как часть этих результатов), так что не ясно, каким образом мы исключим возможность того, что такое улучшение носит случайный характер?

Если экспертная система, использующая вероятности, делает что-то новое, то она должна быть источником знаний, которые недоступны практикующему статистику. В таком случае — это другой путь к достижению истины. Если, однако, подход с позиции экспертных систем является просто "еретическим", то, пожалуй, мотивом служит не желание их создателей познать истину, а традиционные причины, которые ведут обычно к ереси: экспертные системы, возможно, выдвигаются определенной общественной группировкой, которой необходимо иметь свое лицо и защищать себя от других общественных группировок. Одним из путей такой защиты является разработка особой идеологии, которую можно использовать для противоборства с уже существующими.

Антагонизм, который я только что описал, между статистиками и специалистами по использованию знаний не обязательно носит конфликтный характер, когда каждый человек считается занимающим позицию либо одной, либо другой из сторон. Это просто антагонизм между двумя совокупностями представлений о том, что является хорошей основой при выработке советов или рекомендаций. Очевидно, что обе эти совокупности представлений могут уживаться в одном человеке. В понедельник он может решить, что поиск знаний следует осуществлять с максимальной осторожностью и придерживаться стиля статистика, а во вторник захочет и поверит, что можно неясное сделать очевидным, если несколько отрывочных мыслей сообщить некоторой программе, которая разрекламирована как способная усвоить эти мысли, и выработать знания. Я думаю, что эти два отношения не могут сосуществовать до бесконечности, и было бы интересно, какое из них останется в будущем.

Все больше появляется свидетельств тому, что те, кто работает с экспертными системами, считают, что производимые ими системы должны проходить ту же самую проверку, которой подвергаются другие продукты, такие как лекарства или инженерные сооружения. Будем надеяться, что читатели не будут больше оскорблены заявлениями типа: "Компьютер открывает залежи драгоценного металла стоимостью в столько-то миллионов". Это заявление банально, поскольку оно игнорирует затраты на добычу металла и вопрос, нужен ли такой металл. Оно напоминает мне радость таможенного чиновника, который, обнаружив на корабле чемодан, полный старой и сухой травы, заявляет, что конфисковано наркотиков на сумму в тысячи фунтов стерлингов. Эта та сумма, которая была бы получена, если бы этот товар достиг своего назначения. В чемодане же на корабле он стоит намного меньше.

Предположим, что эксперименты по оценке достоинств экспертных систем делаются по аналогии с тем, что известно о проведении клинических испытаний. В простейшем случае пусть сравниваются два медицинских препарата. Эти препараты случайным образом раздаются больным, и проводят-

ся сравнения степени выживаемости по двум группам. На этом основании формируется мнение, какой препарат лучше. (Замечания, которые идут сейчас, можно перенести и на другие области, в которых признается авторитет экспертов.)

В интересующем нас случае один препарат — это совет некоторой экспертной системы, а другой — совет специалиста-врача. Однако если продолжать эту аналогию, то возникает множество вопросов. Когда мы сравниваем два лечебных препарата, то принимаем ряд мер, чтобы быть уверенным в однородности препаратов. Возможно, что больные, получившие один и тот же препарат, подвергаются на самом деле различному лечению в силу некоторого неизвестного свойства процесса производства препарата, из-за которого препараты из одной упаковки имеют различные свойства. Однако такие возможные вариации не идут в сравнение с теми, которые возникают, когда лечение производится по совету доктора. Как получить какой-то эквивалент "упаковки" идентичных докторов? Если мы решим сравнивать работу системы с одним-единственным врачом, то на такого врача ложится большая ответственность. Если же мы выберем консилиум врачей, то на какой основе подбирать его членов? Если мы знаем, что должны отбирать метод лечения, которое получают больные, случайным образом, то не следует ли тот же принцип использовать и при выборе врачей?

Если экспертная система объявлена компетентной в некоторой специальной области заболеваний, то у нас есть выбор между оценкой работы системы по сравнению с врачом-специалистом или по сравнению с самым лучшим врачом, специализирующимся на этих заболеваниях.

Можно предположить, что поскольку система нацелена на врача, не являющегося специалистом, то он и должен участвовать в таком сравнении. Тогда мы должны задуматься над тем, что получается, если неспециалисту продали что-то, что заведомо хуже совета специалиста. С большой вероятностью такая система вскоре будет заброшена, поскольку никто не захочет оставаться посредственным врачом. Если же система так же хороша, как хороший специалист, тогда мы должны задать вопрос, собирается ли специалист способствовать ее совершенствованию и, более того, не знает ли он чего-то такого (если, конечно, это легко поддается кодированию), что следовало бы знать с самого начала?

Такого рода размышления заставляют нас думать, что оценивать достоинства следует только у такой экспертной системы, которая может конкурировать с самым лучшим специалистом, поскольку никто не будет проявлять интереса к экспертной системе, про которую известно, что она не самая лучшая.

Насколько реально думать о клиническом испытании, в рамках которого одного пациента будет лечить экспертная система, а другого — консилиум специалистов? Когда сравниваются два лечебных препарата, то жаловаться на доктора нельзя, по крайней мере в самом начале обследования, поскольку именно потому и ставился эксперимент, что доктор не знал свойств препарата. Но кто осмелится подвергнуть больного лечению, которое предложила экспертная система, когда оно расходится с существующим мнением?



Таким образом, наша область исследования сужается еще больше. Разумно сравнивать лишь те экспертные системы, которые конкурируют со специалистами, поскольку мы ожидаем, что к не самой лучшей системе рано или поздно перестанут обращаться. И кроме того, мы можем сравнивать экспертные системы лишь с тем, что о них думают эксперты, поскольку реально нельзя думать о том, чтобы на самом деле доверить им человеческие жизни. Можно предположить, что методология экспертных систем окажется наиболее выигрышной в некоторых безопасных областях, а это приведет к некоторому росту доверия к ним и в тех ситуациях, которые связаны с некоторой долей риска. Если оболочка для экспертной системы окажется черзвычайно полезной при разработке систем, контролирующих поток транспорта в городе, не проявится ли в этом ее какое-то внутреннее достоинство, позволяющее нам верить в нее и в другом контексте? Это весьма опосредованное размышление.

Возникло совершенно нелепое положение. С одной стороны, у нас есть мнения специалистов, а с другой — мы имеем заключения компьютера, о которых мы будем судить, опираясь на мнения специалистов.

Имеется прием, которым можно воспользоваться, чтобы получить ощущение большей объективности. Если мы смешаем мнения, поступившие от компьютера, с мнениями специалистов, а затем попросим специалистов высказаться о достоинстве каждого из них, не говоря им об их источнике, то мы можем надеяться на то, что получится более совершенное сравнение. Разочаровывает только тот факт, что в этой ситуации мы ничего не узнаем о лечении людей, а познакомимся лишь с тем, что происходит в умах специалистов. При таком методе мы обеспечиваем, чтобы видимая работа экспертной системы не превышала уровень специалистов. К тому же если экспертные системы, использующие вероятности, должны обладать какими-то новыми характеристиками, приводящими к новому знанию, то иногда они должны и удивлять специалистов.

Предположим, что все указанные трудности каким-то образом удалось преодолеть, эксперимент продуман, оценка достоинств получена и достигнуто соглашение о том, что конкретная экспертная система действует на определенном уровне компетентности. Этот уровень определен по отношению к компетентности человека. Если затем я куплю эту систему и начну ею пользоваться в моей каждодневной работе, то практически без усилий я начну усваивать содержащиеся в ней знания. Мое положение относительно системы изменится. Разумеется, возможен и обратный процесс. Когда я вернусь из продолжительного отпуска, система будет в выигрышном положении по сравнению со мной, поскольку к этому времени я забуду что-то из того, что ей известно.

Мне бы хотелось предложить читателю разумный критерий, но я боюсь, что, говоря об оценке работы экспертной системы, мы говорим о чем-то весьма эфемерном. Предлагая новую методологию программирования, люди, занятые экспертными системами, создали ситуацию, которая имеет многие характеристики "показного" мероприятия. Мы создали что-то,

работоспособность чего можно оценить лишь весьма грубо и полезность чего зависит главным образом от энтузиазма тех, кто принимал в этом участие. Является ли MYCIN живучей программой? Позвольте мне попросить читателя самому ответить на этот вопрос, а не принимать просто на веру то, что ему говорят.

#### 4.3. ЗАКЛЮЧИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

Имеется еще два аспекта феномена экспертных систем, которые необходимо обсудить. Первым аспектом является то, что необходимость в создании экспертных систем возникла как попытка специалистов по вычислительной технике создать что-то для других академических дисциплин.

В глубокой истории приложений вычислительных машин (т.е. 20 лет назад) достаточно было просто предложить проект с названием "Приложение анализа с использованием ЭВМ к предмету X", где предмет X мог быть чем-то весьма непонятным, вроде популяционной генетики или доисторического пещерного искусства.

У мировой научной общественности было твердое убеждение, что взаимное скрещивание дисциплин может многое дать, а вклад обработки данных при этом просто грандиозен.

С ходом времени такие попытки утратили свою привлекательность. Очень часто предлагаемая новая технология требовала, чтобы исследователь из области X применял статистические понятия случайных выборок и планирования эксперимента определенным систематическим образом. А как получить случайную выборку для группы охотников каменного века, когда вам неизвестно, сколько их было или даже где они были?

Я думаю, справедливо будет сказать, что специалисты по вычислительной технике в конечном счете мало обогатили наши знания. Они сделали возможным применение методов, созданных статистиками, причем иногда с большим успехом. Но при этом ими создана лишь машина, водить которую должен кто-то другой.

Таким образом, специалисты по вычислительной технике всегда чувствовали потребность показать, что существует оригинальный вклад, который они могут сделать в развитие науки. Это приводит к совершенно иному взгляду на их усилия. Пропагандисты экспертных систем зарабатывают на жизнь вычислениями, и они хотели бы закрепить этот заработок, захватив другие предметные области. Дело выглядит так, как если бы их перестало вдохновлять то, что они делают, и они хотели бы получить вдохновение, впитав ту жизнь, которая, по их мнению, существует в работах других.

Мое заключительное замечание напоминает то, что мог бы сказать антрополог о развитии направления экспертных систем как аспекте человеческого общества. Я сомневаюсь, что мы когда-нибудь найдем объяснение привлекательности для нас бестелесного интеллекта. Почему мы восхищаемся, когда видим, что машина думает подобно нам? Если мы проследим за мыслительной сферой жизни людей с давних времен, то увидим, что многое из того, что мы сегодня называли бы поддержкой принятия решений, пришло из

источников, которые воспринимались людьми как внешние силы, заключенные в некоторые сверхъестественные рамки. Психолог Дж. Джейнес в книге [2] подробно показывает, как люди постепенно перестали принимать совет от внешних "голосов" и других оракулов и стали сами решать, что им делать. Я вижу глубокую аналогию между древней ролью Дельфов и надеждами, которые есть у людей в отношении той роли, которую экспертные системы будут играть в нашей жизни.

Еще рано говорить, насколько здорово то, что происходит. Если же нет, то экспертные системы присоединятся ко многим изобретениям, попавшим на свалку истории, и у нас будет интересное занятие попытаться понять, как могло получиться, что так много в других отношениях талантливых людей стали верить, что что-то новое появилось на нашей Земле.

Я счастлив подтвердить, что многое извлек из обсуждения некоторых мыслей, содержащихся в этой главе, со следующими людьми (которые, естественно, не несут никакой ответственности за то, что мною написано).

1. Н. Барги, Факультет политических наук, Сорбонна.
2. Г. Эллиот, Факультет математики и вычислительной техники, Слау-Колледж.
3. Б. Джонсон, фирма ISTEEL.
4. Т. Джонсон, фирма OVUM.
5. А. Паркин, фирма Computer Techniques International.
6. Р. Раулэнд, компания Allen and Overy.

## ЧАСТЬ II

### ЛОГИЧЕСКИЙ ВЫВОД

#### Глава 5

#### СИСТЕМЫ НЕЧЕТКОГО РАССУЖДЕНИЯ

*Ричард Форсайт*

Одна из проблем, с которой сталкиваются все эксперты, будь то люди или машины, состоит в том, что в жизни ничто не имеет определенного характера, исключая смерть, но даже она может наступить неожиданно. Решение проблем в реальной жизни поэтому требует учета неопределенности. Для этого были предложены различные схемы (некоторые из них оказались вполне пригодными), в которых фрагментарная и ненадежная информация использовалась для получения оценки истины.

В действительности, одним из существенных достоинств экспертных систем является создание методов, позволяющих быть точными в отношении неточностей. В настоящей главе мы рассмотрим несколько способов рассуждения при наличии неопределенности, когда данные, относящиеся к задаче, или правила вывода (возможно, то и другое) не являются на 100% надежными.

##### 5.1. БЕГЛОЕ ВВЕДЕНИЕ В НЕЧЕТКУЮ ЛОГИКУ

Нечеткая логика (называемая также "логикой возможностей") была придумана Л. Заде, который распространил булеву логику на действительные числа [7]. В булевой алгебре 1 представляет истину, а 0 — ложь. То же имеет место и в нечеткой логике, но, кроме того, используются также все дроби между 0 и 1, чтобы указать на частичную истину. Так, запись  $p(\text{высокий}(X)) = 0,75$

говорит о том, что предположение "X — высокий", в некотором смысле на три четверти истинно. Точно так же оно на одну четверть ложно. Для комбинирования нецелочисленных значений истинности в нечеткой логике определяются эквиваленты операций И, ИЛИ и НЕ:

$$\begin{aligned} p1 \text{ И } p2 &= \min(p1, p2) \text{ (т. е. меньшее),} \\ p1 \text{ ИЛИ } p2 &= \max(p1, p2) \text{ (т. е. большее),} \\ \text{НЕ } p1 &= 1 - p1 \text{ (т. е. "обратное значение").} \end{aligned}$$

Таким образом, обрывочные сведения можно комбинировать на основе строгих и согласованных методов, поэтому нечеткая логика с успехом используется, например, в системе обеспечения принятия решений REVEAL.

Слабым моментом в применениях нечеткой логики является отображение или функция принадлежности. Предположим, что мне 35, насколько истинно предположение, что я старый? Равна ли эта величина 0,5, поскольку я прожил примерно полжизни? Или что-то вроде 0,4 или 0,6 будет более реалистичным?

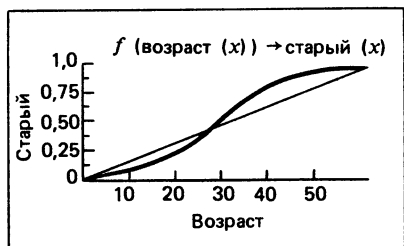


Рис. 5.1. Функция, связывающая возраст человека в годах с понятием "старый". (Но кто может решить, что предпочтительнее, прямая или кривая?)

позволяющие пользователю легко модифицировать различные функции отображения и экспериментально установить, насколько важна их точная форма. Конечно, нечеткая логика не сводится к тому, что мы здесь упоминаем, но мы ограничиваемся лишь введением.

Одной из проблем, которой все же требуется уделить внимание, прежде чем двигаться дальше, является проблема взвешивания отдельных сведений. Предположим, например, что мы располагаем некоторой совокупностью "нечетких" правил, касающихся предпочтений в политике:

#### Правило 1

ЕСЛИ X водит машину шведской фирмы "Вольво"  
И X читает газету "Гардиан"  
ТО X будет голосовать за социал-демократическую партию (СДП).

#### Правило 2

ЕСЛИ X не любит Маргарет Тэтчер  
И X не поддерживает идею членства Великобритании в Европейском экономическом сообществе  
ТО X будет голосовать за СДП.

Предположим, что мы своими собственными глазами видели X за рулем машины "Вольво" (определенность равна 1) и вполне уверены (0,8), что X обычно читает "Гардиан". Тогда условия, входящие в правило 1, имеют совместное значение степени истинности, равное 0,8, поскольку в случае логической функции И мы используем операцию  $\min$ .

Разумеется, само правило может быть ненадежным, но пока для простоты мы примем, что степень истинности заключения та же, что и степень истинности его условий, что дает нам 0,8 для утверждения "X будет голосовать за СДП".

Теперь предположим, что мы знаем, что X не любит Тэтчер в весьма умеренной степени (0,5) и не поддерживает продолжение членства Великобритании в ЕЭС (степень истинности равна 0,25), тогда степень истинности

Кто-то должен решить, какова форма графика, пример которого приведен на рис. 5.1. Но кто может сказать, что лучше: кривая, изображенная здесь, или прямая линия? Для предпочтения одной формы функции другой нет серьезных рациональных обоснований, поэтому в реальной задаче будут присутствовать десятки или сотни подобных функций, каждая из которых до некоторой степени является произвольной. Поэтому в практических системах, использующих нечеткую логику, таких как REVEAL (см. гл. 9), предусматриваются средства,

предложения "X будет голосовать за СДП" равна всего лишь 0,25 (меньшему из значений), если пользоваться правилом 2.

Теперь как же нам следует относиться к этим противоречивым значениям степени истинности (0,8 и 0,25), полученным для одного и того же предложения? Следует брать минимальное или максимальное или среднее из них или взять какую-то иную функцию от двух чисел?

Эта трудность возникает и в том случае, если два свидетельства не вступают в противоречие: два различных правила, указывающие в пользу одного и того же заключения, обычно будут подкреплять друг друга и усиливать доверие к заключению, давая более высокую степень истинности, чем среднее или даже максимальное.

Аналогично действие нескольких таких правил, указывающих в одном направлении, не может быть полностью компенсировано правилом, указывающим в обратном направлении.

Здесь мы переходим из собственно нечеткой логики в теорию свидетельств, которая выходит за рамки этого обзора. Я хочу лишь подчеркнуть, что использование нечеткой логики само по себе не решает проблем, связанных с взаимным взвешиванием раздельных (и, возможно, противоречивых) источников сведений.

## 5.2. ИСПОЛЬЗОВАНИЕ КОЭФФИЦИЕНТОВ УВЕРЕННОСТИ

Одним из многих вкладов, привнесенных системой MYCIN, является некоторое (хотя и не единственно возможное) решение проблемы комбинирования свидетельств [5].

Шортлифф разработал схему, основанную на так называемых коэффициентах уверенности, которые он ввел для измерения степени доверия к любому данному заключению, являющемуся результатом полученных к этому моменту свидетельств. Коэффициент уверенности — это разность между двумя мерами:

$$КУ[h : e] = МД[h : e] - МНД[h : e].$$

В этом выражении  $КУ[h : e]$  — уверенность в гипотезе  $h$  с учетом свидетельств  $e$ ;  $МД[h : e]$  — мера доверия  $h$  при заданном  $e$ , тогда как  $МНД[h : e]$  — мера недоверия гипотезе  $h$  при свидетельствах  $e$ .

КУ может изменяться от  $-1$  (абсолютная ложь) до  $+1$  (абсолютная истина), принимая также все промежуточные значения, причем 0 означает полное незнание. Значения же МД и МНД, с другой стороны, могут изменяться лишь от 0 до 1. Таким образом, КУ — это простой способ взвешивания свидетельств "за" и "против".

Заметим, что эта формула не позволяет отличить случай противоречащих свидетельств (и МД, и МНД обе велики) от случая недостаточной информации (и МД, и МНД обе малы), что иногда было бы полезно.

Заметим также, что ни КУ, ни МД, ни МНД не являются вероятностными мерами. МД и МНД подчиняются некоторым аксиомам теории вероятности,

но не являются выборками из какой-нибудь популяции, и, следовательно, им нельзя дать статистическую интерпретацию. Они просто позволяют упорядочить гипотезы в соответствии с той степенью обоснованности, которая у них есть.

### 5.3. ВЗВЕШИВАНИЕ СВИДЕТЕЛЬСТВ

Пока мы ничего не извлекли из нечеткой логики. На самом деле, в системе MYCIN операции  $\min$ ,  $\max$  и вычитание из единицы были использованы точно так же, как в нечеткой логике.

То, что было добавлено Шортлиффом, — это формула уточнения, по которой новую информацию можно было непосредственно сочетать со старыми результатами. Она применяется к мерам доверия и недоверия, связанным с каждым предположением. Формула для МД выглядит следующим образом:

$$\text{МД}[h : e1, e2] = \text{МД}[h : e1] + \text{МД}[h : e2] (1 - \text{МД}[h : e1]),$$

где запятая между  $e1$  и  $e2$  означает, что  $e2$  следует за  $e1$ . Аналогичным образом уточняются значения МНД.

Смысл формулы состоит в том, что эффект второго свидетельства ( $e2$ ) на гипотезу  $h$  при заданном свидетельстве  $e1$  сказывается в смещении МД в сторону полной определенности на расстояние, зависящее от второго свидетельства. Эта формула имеет два важных свойства:

1. Она симметрична в том смысле, что порядок  $e1$  и  $e2$  не существен.
2. По мере накопления подкрепляющих свидетельств МД (или МНД) движется к определенности.

Давайте возвратимся к нашим фиктивным политическим правилам, придав им трансатлантический оттенок:

#### *Правило 1*

ЕСЛИ     X водит "Фольксваген"  
И         X читает "Вашингтон пост"  
ТО        X будет голосовать за демократов

#### *Правило 2*

ЕСЛИ     X не любит Рональда Рейгана  
ИЛИ      X хочет, чтобы США убрались из Сальвадора  
ТО        X будет голосовать за демократов.

Примем, что значения МД для этого X таковы:

- 1а. X водит "Фольксваген"     0,8 }  
1б. X читает "Вашингтон пост"   0,75 } И  $\leq = > \min$   
2а. X не любит Рейгана         0,4 }  
2б. X хочет покинуть Сальвадор 0,6 } ИЛИ  $\leq = > \max$

Тогда гипотеза, что X голосует за демократов, поддерживается на уровне 0,75 правилом 1 и на уровне 0,6 правилом 2. Применяя приведенную формулу, получаем

$$\begin{aligned} \text{МД [демократы : правило 1, правило 2]} &= \text{МД [демократы : правило 1]} + \\ &+ \text{МД [демократы : правило 2]} (1 - \text{МД [демократы : правило 1]}) \\ &= 0,75 + 0,6 \cdot 0,25 \\ &= 0,9. \end{aligned}$$

Таким образом, объединенная мера доверия оказывается выше, чем при учете каждого свидетельства, взятого отдельно. Это согласуется с нашей интуицией, что несколько показывающих одно и то же направление свидетельств подкрепляют друг друга. Кроме того, можно поменять порядок правил 1 и 2, и на результате это не отразится.

Схема Шортлиффа допускает также возможность того, что, как и данные, правила могут быть ненадежными. Это позволяет описывать более широкий класс ситуаций.

Каждое правило снабжено "коэффициентом ослабления", числом от 0 до 1, показывающим надежность этого правила. Так, возвращаясь к нашим избирателям, мы могли бы сказать что-то вроде следующего:

*Правило 3 (надежность 0,64)*

ЕСЛИ X водит "Шевроле"  
И X читает Readers' Digest  
ТО X будет голосовать за республиканцев.

*Правило 4 (надежность 0,8)*

ЕСЛИ X любит бывших актеров  
ИЛИ X хочет, чтобы США оккупировали Никарагуа  
ТО X будет голосовать за республиканцев.

Здесь правило 4 вызывает больше доверия, чем правило 3. Если степени поддержки условий таковы:

3а. X водит "Шевроле"	0,88}
3б. X читает Readers' Digest	0,5} И $\leq$ $\geq$ min
4а. X любит бывших актеров	0,5}
4б. X хочет вторжения США в Никарагуа	0,7} ИЛИ $\leq$ $\geq$ max

то немодифицированная сила заключений будет равна 0,5 и 0,7, но эти МД следует умножить на ослабляющие коэффициенты 0,64 и 0,8, что дает 0,32 для правила 3 и 0,56 для правила 4. Применяя формулу уточнения Шортлиффа, получаем

$$\begin{aligned} \text{МД [республиканцы : п3, п4]} &= 0,32 + 0,56 \times 0,68 \\ &= 0,7008. \end{aligned}$$

Шортлифф предпринял попытку дать теоретическое обоснование этим методам, но с моей точки зрения, его соображения не слишком убедительны. Важным здесь является то обстоятельство, что такой набор приемов сослужил хорошую службу в системе MYCIN и последовавших за ней программах. Другими словами, эмпирические результаты достаточно ободрают, чтобы оправдывать дальнейшее их использование и изучение.



#### 5.4. КВИНТЭССЕНЦИЯ БАЙЕСОВСКОГО ПОДХОДА

Одним из исчислений неопределенностей, имеющим сравнительно прочные теоретические оправдания, является теория вероятностей.

Во многих системах, включая систему обнаружения полезных ископаемых PROSPECTOR [1], теорема Байеса используется в качестве нити, помогающей связать вместе информацию, поступающую из различных источников. Это правило позволяет вычислить относительное правдоподобие конкурирующих гипотез исходя из силы свидетельств. В основе правила лежит формула

$$\text{ОП}(H : E) = P(E : H) / P(E : H'),$$

где отношение правдоподобия (ОП) определяется как вероятность события или свидетельства  $E$  при условии заданной конкретной гипотезы  $H$ , деленная на вероятность этого свидетельства при условии ложности данной гипотезы ( $H'$ ). Так, если мы знаем вероятности свидетельства при заданной гипотезе и ее дополнение, то мы можем определить правдоподобие данной гипотезы в свете имеющегося свидетельства. Например, если мы знаем вероятность появления пятен сыпи среди больных ветряной оспой и вероятность появления сыпи среди больных, но не этой болезнью, то мы можем вычислить вероятность заболевания ветряной оспой, если есть пятна сыпи. Обычно легче так воспринимать приведенную формулу.

Отношение правдоподобия может быть использовано для уточнения шансов в пользу рассматриваемой гипотезы, если становится известно, что произошло событие  $E$ .

Нейлор К. (см. гл. 6) выражает формулу правила Байеса в вероятностных терминах. Для разнообразия, а также потому, что с вычислительной точки зрения часто удобнее работать с шансами, я дам это правило на языке шансов. Шансы в пользу чего-то ( $O$ ) и вероятность ( $P$ ) просто преобразуются друг в друга:

$$O = P / (1 - P); P = O / (1 + O).$$

Преобразование оценки "шансы против" ( $A$ ) в оценку "шансы за" также весьма просто:

$$O = 1/A.$$

Таким образом, 7 против 4 может быть выражено как 1,75 против 1, что соответствует 0,5714 (к 1) "за" (или 4 к 7 в пользу рассматриваемой гипотезы).

Байесовская схема уточнения может быть сведена к выражению

$$O'(H) = O(H) \times \text{ОП}(H : E),$$

где  $O(H)$  – *априорные* шансы в пользу  $H$ , а  $O'(H)$  – *результатирующие апостериорные* шансы, при условии наступления события  $E$ , в соответствии с отношением правдоподобия.

Тогда информация от различных источников знаний может комбинироваться простым умножением. При заданных априорных шансах для конку-

Таблица 5.1

Отношение к курению	Продолжительность жизни		Всего
	> 75 лет	75 лет или меньше	
Курящие (чел.)	20	33	53
Некурящие (чел.)	24	23	47
Всего	44	56	100

рирующих гипотез и событий (наблюдений), про которые известно, что они произошли, легко вычисляются апостериорные шансы, а по ним — и вероятности. Отношения правдоподобия получаются из простой двумерной таблицы, показывающей, насколько часто случается каждое событие при каждой из гипотез.

Ниже идет простой пример. Чтобы дать отдохнуть нашим мифическим избирателям, мы представили в табличной форме некие воображаемые данные, которые могли бы (именно могли бы) быть использованы для предсказания продолжительности жизни. Как бы там ни было, это — иллюстрация рассматриваемых принципов.

В приводимой табл. 5.1 содержатся данные по 100 умершим людям, из которых 44 умерли в возрасте за 75 лет, а остальным было 75 лет или меньше, причем указано, кто среди них был курильщиком, а кто нет. (Не забывают, что это просто пример, а не реальные данные.)

Априорные шансы в этой выборке из 100 случаев в пользу того, что человек проживет более 75 лет:

$$O(\text{Долгожитель}) = 44/56 = 11/14 = 0,7857,$$

а отношения правдоподобия

$$OP(\text{Долгожитель} : \text{Курящий}) = (20/44)/(33/56) = (20 \cdot 56)/(44 \cdot 33) = 320/363 = 0,8815;$$

$$OP(\text{Долгожитель} : \text{Некурящий}) = (24 \cdot 56)/(44 \cdot 23) = 336/253 = 1,3280.$$

Предположим далее для наших иллюстративных целей, что пол также принимается во внимание как еще одна переменная, имеющая отношение к долгожительству. Согласно табл. 5.2 отношение правдоподобия

для мужчин

$$OP(\text{Долгожитель} : \text{Мужчина}) = (24/44)/(36/56) = 0,8484;$$

для женщин

$$OP(\text{Долгожитель} : \text{Женщина}) = (20/44)/(20/56) = 1,2727.$$

Теперь, учитывая, что априорные шансы в пользу продолжительной жизни (свыше 75 лет) равны 11/14, мы можем вычислить апостериорные шансы того, что курящий мужчина проживет долгую жизнь, пользуясь выражением

Таблица 5.2

Пол	Продолжительность жизни		Всего
	> 75 лет	75 лет или меньше	
Женщины (чел.)	24	36	60
Мужчины (чел.)	20	20	40
Всего	44	56	100

$$O' (\text{Долгожитель}) = \text{ОП} (\text{Долгожитель} : \text{Куращий}) \times \\ \times \text{ОП} (\text{Долгожитель} : \text{Мужчина}) \times O (\text{Долгожитель}),$$

откуда

$$O' (\text{Долгожитель}) = 0,8815 \cdot 0,8484 \cdot 11/14,$$

что примерно равно 0,5876. Это значение соответствует вероятности 0,3701, тогда как начальная вероятность была 0,44. Таков результат учета двух довольно негативных факторов.

Этот же принцип допускает модификацию для работы более чем с двумя гипотезами (колонками таблицы) без особых затруднений.

Отношения правдоподобия всегда положительны. Хотя нуль и бесконечность могут встретиться, всегда есть возможность избежать этих значений, при необходимости слегка подправив данные. Тогда  $\text{ОП} > 1$  указывает на свидетельства в пользу гипотезы,  $\text{ОП} < 1$  — против нее, а  $\text{ОП} = 1$  говорит о том, что свидетельства не влияют на правдоподобие рассматриваемой гипотезы.

Множитель ОП показывает, насколько более вероятной становится данная гипотеза при наличии свидетельств, чем при их отсутствии.

Если свидетельства сами по себе вызывают сомнения, то не составляет труда построить масштабированное  $\text{ОП}'$ , такое как

$$\text{ОП}' = \text{ОП} \cdot \text{ВС} + (1 - \text{ВС}),$$

где ВС — вероятность того, что свидетельство надежно. Например, если свидетельство известно с вероятностью  $p = 0,8$ , то отношение правдоподобия, равное 1,2 (в пользу гипотезы), уменьшится до

$$1,2 \cdot 0,8 + 0,2 = 0,96 + 0,2 = 1,16,$$

тогда как отношение правдоподобия 0,5, значительно противоречащее гипотезе, увеличится до

$$0,5 \cdot 0,5 + 0,5 = 0,75,$$

т. е. в меньшей степени противоречит гипотезе.

Подводя итоги, отметим, что отношения правдоподобия дают два преимущества: во-первых, они допускают комбинирование нескольких независимых источников данных; во-вторых, их легко корректировать, если свидетельство ненадежно само по себе.

Здесь имеется, однако, замаскированное препятствие. Внимательный читатель, несомненно, споткнется о слово "независимый". Статистики просто придут в отчаяние, глядя на вычисления, связанные с учетом курящих мужчин в предыдущем рассуждении. В конце концов, может ведь быть и так, что просто число курящих мужчин больше, чем курящих женщин. Необходимо дополнительное исследование, чтобы оправдать возможность рассматривать данные о мужчинах/женщинах и курящих/некурящих как два независимых источника информации. А такое дополнительное рассмотрение не только потребует много времени, но и приведет к построению гораздо большей таблицы, с большим числом клеток, каждая из которых будет содержать слишком малую по объему выборку, чтобы можно было ей довериться.

В любой реальной совокупности указателей (индикаторов) почти неизбежно будет иметься какая-то степень ассоциации. Ее минимизация является важной практической задачей. Другими словами, было бы неразумно добавлять к нашему списку индикаторов такие данные, как бородатые/безбородые, поскольку вряд ли найдется хотя одна бородатая женщина. Точно так же сильный-кашель/отсутствие-кашля также не будет хорошей дополнительной переменной, поскольку среди курящих с большей вероятностью встретятся люди с хроническим кашлем. В последнем случае возникает опасность, что один и тот же эффект будет измерен дважды.

Приведенные примеры являются весьма очевидными случаями корреляций, однако могут быть более запутанные ситуации, так что в задачу программного обеспечения системы должны входить их отфильтровывание или соответствующая калибровка значений отношений правдоподобия, когда такие корреляции встречаются. Например, в пакете HULK (Helps Uncover Latent Knowledge — помогает выявить скрытое знание), предназначенном для использования правила Байеса весьма примитивным образом, предусмотрено применение энтропийной меры для предостережения пользователя от попытки привести в систему новое правило, которое в действительности лишь дублирует одно из прежде введенных. Это позволяет исключить наиболее распространенный случай неверного применения правила Байеса. Большинство других программ построения рассуждений также прибегает к проверкам подобного сорта.

## 5.5. БЕСПОДОБНЫЙ HULK

Пакет HULK существенным образом опирается на использование отношений правдоподобия [3] и состоит из двух модулей, работающих один за другим:

LOOK — логический организатор знаний;

LEAP — модуль оценки и предсказания отношений правдоподобия.

LOOK — программа проверки правил. Она дает возможность пользователю попробовать работу решающих правил. Новое правило вводится в виде булевого выражения, в котором используются имена переменных,

возникающих в задаче. Затем предлагаемое правило в программе LOOK используется для классификации каждой выборки из файла данных и пользователю сообщаются степень успешности классификации, выведенные отношения правдоподобия и, что наиболее важно, оценка эффективности в комбинации со всеми другими правилами, построенными к этому моменту. Только в случае, если общий счет улучшается, новое правило рекомендуется для включения в систему. Окончательное решение остается за пользователем, который и решает, сохранить правило или его следует отбросить.

Таким образом, LOOK можно считать системой программного обеспечения для фильтрации плохих правил путем применения их на специальной совокупности данных. Она не изобретает своих собственных правил, но позволяет создать совокупность правил, работающих сообща. Она реализует метод выдвижения предположений и последующей отбраковки, который лежит в основе любой науки. Пользователь строит догадки (с чем люди справляются очень хорошо), оставляя машине их проверку (к чему та приспособлена больше).

Если два правила в своей основе опираются на один и тот же фактор, то второе правило, будучи само по себе весьма продуктивным, не внесет новой информации и система воздержится от рекомендации о включении его в используемую совокупность правил.

Система LEAP обеспечивает достаточно жесткую экспертизу, состоящую в том, что правила, построенные с помощью системы LOOK, применяются для предсказания вероятностей ряда выборок, принадлежащих любому заданному классу. В отличие от обучающих данных, используемых в системе LOOK, нет необходимости заранее знать, какому классу принадлежат случаи, взятые в качестве тестовых данных в системе LEAP. Выборки ранжированы в порядке убывания вероятностей.

## 5.6. ЗАКЛЮЧЕНИЕ

Все методы неточного рассуждения, представленные здесь, — нечеткая логика, коэффициенты уверенности и байесовская логика — были проверены на практике. Их полезность была доказана в серьезных приложениях.

Но у них нашлись и свои критики. Некоторые возражают против изъятия из теории нечетких множеств "закона исключения третьего" на том основании, что со времен Аристотеля этот закон является основным законом формальной логики. Многие математики с неодобрением относятся к факторам уверенности Шортлиффа из-за кавалерийской лихости, с которой в этом подходе вносятся ничем не обоснованные корректировки, не вдаваясь в суть возникающих при этом проблем.

И теорема Байеса является уже в течение 200 лет центром разногласий между статистиками. В частности, оценка априорной вероятности может оказаться весьма серьезной трудностью. Так, если необходимо узнать априорную вероятность, скажем, рака желудка, то следует ли взять для этого долю всех больных, обратившихся к доктору по поводу болей в желудке,

или долю населения США с такого рода болезнью, или оценку доли мирового населения, или что-то еще? Часто истинное значение остается неизвестным, так что кто-то строит предположение. К счастью, чем больше накапливается свидетельств, тем менее важной становится оценка априорных шансов, но и это едва ли оправдывает использование первой же оценки, которая приходит в голову, как это делается в большинстве систем.

Быть точным в отношении неопределенности может оказаться задачей, у которой никогда не будет окончательного решения. Именно это сообщает теории вероятностей наибольшую привлекательность, которая, однако, является объектом непрекращающихся дебатов. Даже сам смысл невинных вероятностных схем, подобных "вероятность, что завтра будет дождь, равна 0,55" или "вероятность, что я вытяну туза из хорошо перетасованной колоды карт, равна  $1/13$ ", не может считаться выясненным до конца. Теоретики до сих пор спорят в отношении возможности частотной интерпретации неповторяющихся событий и по вопросу о связи рациональной уверенности и вероятности и по многим другим вопросам [2, 4, 6].

Оставляя в стороне теоретические возражения, отметим и чисто практические затруднения. Нечеткая логика, например, была предложена как одна из возможностей для выражения внутренние неточных лингвистических понятий, таких как "высокий" и "старый". Но, конечно, старый атлет (в возрасте 36 лет) намного моложе молодого пенсионера (возраста 55 лет), точно так же, как маленький слон больше крупной мыши. Контекст оказывается весьма важным.

Кроме того, возникает вопрос, связанный с приписыванием числовых значений субъективным суждениям. Во многих системах, следующих системе PROSPECTOR, пользователю предлагается отвечать, пользуясь шкалой от  $-5$  (определенно нет) до  $+5$  (абсолютная уверенность). Но предположим, что система идентификации деревьев спрашивает:

"По шкале от  $-5$  до  $+5$  насколько вы уверены, что эти листья имеют сильный острый запах?"

и получает в ответ нуль. Ясно, что пользователю ответ неизвестен, но ни к чему не обязывающий нуль мог бы иметь и весьма широкий спектр возможных значений:

1. Как профессиональный ботаник с многолетним опытом, я понюхал листья и после размышления заключил, что это нечто промежуточное между сильным острым запахом и полным отсутствием запаха.
2. Некоторые люди являются "цветослепыми", а что касается меня, то, я боюсь, что моему обонянию доверять нельзя.
3. Послушайте, но я всего лишь хочу сфотографировать их!
4. Сейчас зима, ради бога, какие могут быть листья?!
5. Что? Иди сам под этот проливной дождь и нюхай чертовы листья! Надеюсь, что это шутка?
6. "Острый". А какой это запах?
7. Наш кот считает это дерево границей своей теории, так что запах острый, но дерево в этом никак не повинно.

Неопределенность имеет множество сторон, которые оказывают различное влияние. Деревья — относительно знакомые нам объекты, а теперь с сочувствием отнесемся к отчаянию бедного пользователя, когда PROSPECTOR его спрашивает: "Не произошло ли превращения роговой обманки в биотит в результате замещения?"

Несмотря на все эти трудности, мы не можем тем не менее, уклониться от того, чтобы как-то работать с неопределенностью. Эвристические знания никогда нельзя считать выделенными и фиксированными. Если мы будем настаивать на использовании лишь бесспорных "фактов", мы выбросим все интересные задачи.

Грамотный специалист по использованию знаний обязательно столкнется с необходимостью встраивания в экспертную систему какой-то взвешивающей процедуры, возможно, основанной на одной из схем, приведенных в этой главе. Он должен также себе представлять, что любая процедура связана с некоторыми ограничениями. Это соображение говорит в пользу внесения в программу каких-то проверок на согласованность и других мер предосторожности, а не в пользу того, чтобы дожидаться прихода абсолютной определенности, т. е. смерти.

## Глава 6

### КАК ПОСТРОИТЬ МАШИНУ ВЫВОДА

*Крис Нейлор*

Что-то в этом заголовке создает ощущение некоторого скрытого диссонанса. Диссонанс заключается, по-видимому, в очевидном несоответствии между гладкостью фразы "машина вывода", принятой в кругах специалистов, и довольно прозаической монотонностью "Как построить . . ?".

Машина вывода — это прекрасно. Говоря об этом, можно почувствовать, что мы прикасаемся к самым современным представлениям о компьютерах. Но "как построить . . ?" — это что-то совсем другое. Дело в том, что все мы знаем, что мы-то будем иметь дело с теми же самыми старыми клавиатурой ввода и экраном терминала и что на машине будет использоваться столь привычный нам язык программирования, с которым мы работаем.

Средства, которыми мы располагаем, ни в коем случае не являются *плохими*. Просто это хорошо нам знакомые средства, и ничего в них нет исключительного. И тем не менее на их основе мы собираемся строить нашу машину вывода. Если так рассуждать, то ощущение чего-то исключительного, связанного с термином "машина" начинает понемногу стираться. Потому что с нашими средствами мы можем писать программы, и ничего больше, а отсюда наша сверхпрекрасная машина вывода становится программой, и только.

В этой главе рассмотрен подход, если он вообще может что-то дать, которым вы сможете воспользоваться, когда у вас появится настроение пора-

ботать со своим терминалом. Он не является даже отдаленным родственником волшебной палочки, обеспечивающей решение проблемы. Я хотел бы, чтобы это было не так, но за пределами некоторых забавных публикаций такого волшебного подхода просто не существует.

Однако после "холодного душа" давайте вернемся к как-то по-особенному звучащему представлению о машине вывода. Что это такое? Вопрос достаточно простой, но простой ответ на него получить не так-то легко. Поскольку почти все, что мы делаем, может быть подведено под общий термин "вывод", то ясно, что если бы мы располагали программой, которая делала бы за нас наш вывод, то можно было бы такую программу называть машиной вывода. Тогда это будет звучать сверхгладко. И мы могли бы воспользоваться таким конечным продуктом, чтобы поразить друзей и врагов (конечно, лишь его описанием, а не тем, как он на самом деле работает).

Но опять: что такое вывод?

Позвольте мне привести пример логического вывода. Я — большой бизнесмен и нанял на работу Фреда Блоггса и плачу ему один фунт стерлингов в час. На прошлой неделе он работал на меня в течение 20 ч. Следовательно, в эту пятницу я должен уплатить ему 20 фунтов. Это и есть вывод.

По существу, это то, что делает программа для составления платежной ведомости. И тот, кто начнет продавать программы для начисления платежей под видом машин вывода, семантически и будет прав, но я, честно говоря, воздержался бы от такой покупки у такого рода продавцов.

Но вернемся снова к примеру с платежной ведомостью, поскольку в связи с ним стоит сделать два интересных (хотя и очевидных) замечания. Первое состоит в том, что логический вывод, который мы делаем (относительно требования Фреда Блоггса об уплате ему 20 фунтов), был *точным*. Второе — вывод был сделан очень *легко*. И до настоящего момента эти черты характеризовали большую часть того, что было сделано в области вычислительных машин. Задачи, которые здесь ставились, в общем случае были задачами, имеющими точные решения, и практически по одной этой причине эти задачи легко решались. В некотором отношении именно точная определенность того, что является решением, позволяла программистам находить путь к решению и достигать его без больших затруднений.

Конечно же, я услышу тяжелые вздохи, когда скажу, что во всех современных программах точное решение строится легко. Я и сам, хотя бы в некоторой степени, вздыхаю, когда вспоминаю о некоторых трудностях, которые вызывала у меня работа над такими программами.

Рассмотрим, например, численный анализ. Имеется ли точное решение и легко ли оно достигается? В общем случае нет ни того, ни другого. Но, по крайней мере, вы можете купить книгу по численному анализу и воспользоваться ею. Эта книга, как правило, дает весьма тщательный анализ типов ошибок, которые могут встретиться в такого рода системах. Решение может быть неточным в математическом смысле, но оно является точным в том смысле, что, пользуясь данным методом, мы можем точно описать характер ошибок, которые нас ожидают.



Теперь обратимся к задаче другого типа — медицинской диагностике. Является ли она точной и легкой? Она *может* оказаться таковой. Возможно, врачи установят, что она является точной и легкой — в конце концов, они занимаются ею всю жизнь. Но как они ставят диагноз? Если их спросить, то почти наверняка получите какие-то обтекаемые ответы. Диагностика, по-видимому, основывается на любопытной смеси догадок, предчувствий, прямых наблюдений, практического опыта. . . Назовите что-то еще — они этим тоже пользуются! Здесь все настолько же тщательно перемешано, как в самой дьявольской смеси, которую вы только можете себе вообразить. И в то же самое время *мы*, безусловно, понимаем, что именно они делают. Мы можем не понимать медицины и, возможно, будем не в состоянии запрограммировать их деятельность на компьютере, но *мы*, безусловно, *понимаем*, какого типа действия совершает врач, когда он контактирует с больным. Происходит это от того, что *мы*, как и доктор, являемся людьми. И *мы* действуем точно таким же образом в наших разнообразных областях, в которых *мы* являемся экспертами.

Теперь, когда в нашем рассуждении возникло слово "эксперт", становится возможным увидеть, как вопрос о машинах логического вывода связан с экспертными системами.

Тип логического вывода, который нас интересует, не является простым, точным и легким. Как раз наоборот. Эксперты — люди, которые во все моменты, как правило, прибегают к *неточным* и *трудным* выводам. Именно такие логические выводы *мы* и будем строить, и наша проблема, в отличие от составления платежной ведомости, заключается в том, что *мы* даже не знаем, каковы исходные уравнения, не говоря уже о том, как запрограммировать их для получения решения.

Подумав над только что сказанным, я пришел к заключению, что здесь можно и остановиться, заявив, что область экспертных систем буквально начинена принципиальными трудностями и не остается никакой надежды у тех, кто решит идти дальше. Но все обстоит не настолько плохо, иначе те из вас, кто потратил значительную сумму денег на покупку этой книги, вполне резонно могут прийти к мысли, что следующее, что вы должны сделать, это как-то разделаться со мной.

Поэтому далее я предполагаю описать машину логического вывода (опять это непонятное словосочетание!), над которой я работал, и которая, похоже, дает неплохие результаты при решении определенных классов задач, и которую достаточно легко построить.

Я думаю, что, возможно, в дальнейшем то, что я должен сказать, покажется вам знакомым, и я надеюсь, что вы к этому отнесетесь с пониманием, ибо если некоторые идеи не окажутся знакомыми, то они останутся незамеченными и непонятыми. В особенности это верно, я думаю, в тех случаях, когда я буду использовать некоторые сведения из статистики.

## 6.1. МАШИНА ЛОГИЧЕСКОГО ВЫВОДА И БАЗА ЗНАНИЙ

При разработке большого числа экспертных систем вполне сознательно делались усилия разделить задачу на две части — машину вывода и базу знаний. Идея состояла в том, что машина логического вывода является универсальной думающей машиной, а база знаний — это то, над чем ей предстоит думать.

Первое, если хотите, является некоторым эквивалентом "необученного" мозга человека со встроенной способностью что-то делать. Второе — это сумма человеческого опыта в некоторой конкретной области знаний. Взяв их вместе, вы получите эквивалент эксперта-человека. Так, во всяком случае, гласит теория.

Во многих отношениях, конечно, это хорошая идея. В конце концов, предположим, что у вас просто имеется такая машина вывода и вы располагаете также базой знаний, применимой к области медицинской диагностики. Почему не воспользоваться машиной и тогда, когда вам необходим эксперт, скажем, по предсказанию погоды? Почему бы просто не отсоединить медицинскую часть и не подключить метеорологическую? Причина этого "почему бы" заключается в том, что машина вывода пока что не способна работать с двумя столь различными областями. Но попытаться мы можем. Более практичным соображением, с моей точки зрения, является то, что, даже если вы работаете над одной проблемой (такой, как медицинская диагностика), разбиение задачи на две части — базу знаний и машину вывода — облегчит ее рассмотрение. Это позволит вам один день потратить на выяснение того, почему не работает машина логического вывода, а другой — на то, почему что-то не так с базой знаний. Я нашел, что это намного лучше, чем потратить два дня на хлопоты с обеими компонентами, как было бы при использовании более привычной методики. И хотя это все может выглядеть как шутка, в действительности это совсем не смешно.

Очевидно, вы поймете, что сказанное ненамного ушло от мысли о том, что имеется программа и имеются данные. В самом деле, жизнь станет намного легче, если на время забыть о данных и, используя тестовые данные, потратить день на то, чтобы заставить хорошо работать саму программу.

Но факт состоит в том, что при работе с довольно расплывчатыми задачами часто имеется большая неопределенность в том, что именно считать программой, а что — данными. Эти вещи могут значительно перекрываться, и хитрость состоит в том, чтобы провести разделительную черту таким образом, чтобы как можно более значительная часть задачи попала бы в раздел данных (базу знаний), а не в программу (машину вывода). Полезным побочным результатом является возможность действительно понять, в какой мере можно создать такую универсальную машину логического вывода. В конце концов, если из программы удалось исключить все данные до последнего бита, то она и в самом деле может оказаться универсальной, и потому ее можно использовать с какой-то другой базой знаний.

Я думаю, что сейчас для того, чтобы сделать наше обобщение более конкретным, следует сказать что-то об одной определенной базе знаний.

База знаний — это не дедуктивная машина, которая вам была обещана, а это то, что должна "везти" подобная машина, поэтому формат, в котором она задается, должен быть точно таким же, как формат базы знаний и в другой области, если дедуктивной машине предстоит работать в этой области.

Все примеры, которые я буду приводить, относятся к области медицинской диагностики, хотя, я думаю, многие согласятся с тем, что точно так же можно сформулировать задачи из других областей и применять к ним нашу дедуктивную машину.

Так или иначе, в нашей конкретной базе знаний используется две разновидности формата для данных.

### 1. В первом формате хранятся знания о конкретном заболевании:

Название болезни,  $p$ , число применимых симптомов ( $j, p^+, p^-$ )

Первый элемент — название болезни. Второй — априорная вероятность того, что это заболевание имеется у взятого наудачу члена популяции. При байесовском подходе — это априорная вероятность рассматриваемого заболевания. Третий элемент — число симптомов, которые могут быть использованы либо как признаки этой болезни, либо как противоречащие ей признаки. Последним идет ряд троек (т.е. трехэлементных полей), соответствующих каждому из таких применимых симптомов. Первым элементом тройки является референтный номер симптома, который рассматривается в данный момент. Второй элемент тройки — вероятность, что этот симптом будет наблюдаться при условии, что у пациента имеется рассматриваемое заболевание. Третьим элементом тройки является вероятность того, что данный симптом будет наблюдаться при условии, что у пациента данной болезни нет.

### 2. Второй тип данных касается симптомов:

Номер симптома, название симптома, вопрос, который следует задать в отношении симптома

Здесь всего три поля. Первым идет номер симптома; это тот номер, который используется для ссылок на симптом в данных о заболевании, т.е. первый элемент тройки. Вторым является имя симптома. В третьем поле содержится вопрос, который можно будет задать о пользователе системы при попытке определить, проявляется ли этот конкретный симптом у данного пользователя.

Итак, пока все идет хорошо. Это та база знаний, которая и будет использоваться. В ней нет ничего особенного, но это, по-видимому, то, что нам нужно. Может быть, она станет несколько более понятной, если представить себе такое заболевание, как грипп. Итак, первый элемент указан. Какова вероятность того, что некоторое случайно выбранное лицо болеет гриппом? Ну скажем, одна тысячная, и мы указали  $p : p = 0,001$ . Каково число симптомов, которые могут указывать как в пользу гриппа, так и против него? Очень много, разумеется, но мы остановимся, скажем, на температуре и насморке — 2 симптома. Если у больного грипп, то у него, несомненно, есть

температура и, возможно, насморк. Если у него гриппа нет, то температура тем не менее может быть, хотя и с гораздо меньшей вероятностью. То же самое можно сказать и о насморке.

В результате мы приходим к чему-то вроде

ГРИПП, 0,001; 2; 1; 1; 0,01; 2; 0,9; 0,1

что достаточно хорошо суммирует информацию, которую вы можете получить о гриппе у любого врача. Однако правильнее было бы здесь говорить о конкретном враче, располагающем ограниченными познаниями в области таких заболеваний.

В соответствии с этим у нас имеются следующие элементы базы знаний, касающихся симптомов:

1. ТЕМПЕРАТУРА, ЕСТЬ ЛИ У ВАС ВЫСОКАЯ ТЕМПЕРАТУРА?

2. НАСМОРК, ЕСТЬ ЛИ У ВАС НАСМОРК?

Полученная база знаний представлена на рис. 6.1. Очевидно, ее можно легко и быстро модифицировать на этом этапе, чтобы уточнить некоторые положения, так что мы считаем вопрос создания базы знаний исчерпанным.

Однако нам остается решить главную задачу: как пользоваться такой базой знаний, т. е. как же выглядит дедуктивная машина?

На этом этапе я хотел бы разделить проблему на две части. Первая часть касается самого вывода или дедукции, а вторая — машины, которая реализует вывод. Такое разделение в нашем случае выглядит несколько странным, поскольку под "выводом" я понимаю практически ровно одно вычисление, тогда как под "машиной" я на самом деле подразумеваю систему, посредством которой мы будем определять, а точнее, система будет определять, в каком именно порядке делать логические выводы.

(1) Название болезни,  $p$ , число применимых симптомов ( $j, p^+, p^-$ )

(2) Номер симптома, название симптома, вопрос, который следует задать

Пример: ГРИПП, 0,001, 2, 1, 1, 0,01, 2, 0,9, 0,1 и связанные с ним два симптома:

1. ТЕМПЕРАТУРА, ЕСТЬ ЛИ У ВАС ВЫСОКАЯ ТЕМПЕРАТУРА?

2. НАСМОРК, ЕСТЬ ЛИ У ВАС НАСМОРК?

Итак, априорная вероятность, что некто болен гриппом, равна 0,001. С гриппом связано два симптома.

Первый симптом — высокая температура. Вероятность высокой температуры при гриппе равна 1.

Вероятность высокой температуры при отсутствии гриппа равна 0,01.

Второй симптом — насморк. Вероятность насморка при гриппе равна 0,9.

Вероятность насморка при отсутствии гриппа равна 0,1.

Рис. 6.1. Формат данных в базе знаний

## 6.2. ТЕОРЕМА БАЙЕСА

В сущности, используемый мною подход к построению логического вывода является байесовским. А для тех, кто еще этого не знает, я скажу, что Reverend Байес был английским священником, жившим в 18 веке, который все свое время отдавал изучению статистики.

Сразу же оговорюсь, что байесовский не является единственно возможным подходом к построению выводов. Имеются и другие, например методы классической статистики, а в последнее время — методы распознавания образов. Я не собираюсь углубляться в их рассмотрение, поскольку это непомерно расширило бы предмет настоящей главы. Лично я считаю байесовский подход чрезвычайно полезным и думаю, что читатель получит наибольшую пользу от чтения этой главы; если я сосредоточу внимание на одном методе, проследив достаточно глубоко его работу.

В сущности, теории Байеса опираются на предположение, что практически для любого положения имеется, какая бы малая она не была, априорная вероятность того, что данное положение истинно. Эта вероятность может быть очень малой, она может оказаться на самом деле нулевой. Но это не мешает нам вести вычисления так, как если бы существовала некоторая вероятность. Теперь, если имеется априорная вероятность некоторой гипотезы (болен ли пациент гриппом или является ли Земля плоской), то должны иметься некоторые данные, которые можно было бы привлечь для оправдания наших взглядов по этому вопросу (или, если хотите, наших убеждений). Если бы их не было, то процесс просто на этом и остановился бы и указанная априорная вероятность осталась бы навсегда неизменной. Однако при наличии относящихся к делу сведений мы можем модифицировать априорную вероятность так, чтобы получить уже апостериорную вероятность той же самой гипотезы с учетом поступивших новых данных.

Для тех, кому понятнее уравнения, приведем теорему Байеса:

$P(H)$  = априорная вероятность  $H$  при отсутствии каких-либо свидетельств  
 $P(H : E)$  = апостериорная вероятность  $H$  при наличии свидетельства  $E$ .  
По определению:

$$P(H : E) = \frac{P(H \& E)}{P(E)} \quad \text{и} \quad P(E : H) = \frac{P(E \& H)}{P(H)}$$

После некоторой перегруппировки получаем

$$P(H : E) = \frac{P(E : H) P(H)}{P(E)}$$

и

$$P(E) = P(E : H) P(H) + P(E : \text{не } H) P(\text{не } H)$$

Остается единственный вопрос: получили ли мы что-нибудь действительно полезное? Пожалуй да. Мы начали с величины  $P(H)$ , а хотели и получили  $P(H : E)$ .

Теперь подумаем о нашем пациенте, больном гриппом (или не больном, ведь в этом и состоит вопрос). Мы грубо представляем себе, какова априорная вероятность того, что пациенты болеют гриппом, т.е. мы знаем  $P(H)$ . Но мы хотим знать вероятность гриппа у данного, конкретного пациента.

Мы обнаруживаем, что у него жар (данные налицо), и хотим узнать  $P(H : E)$  — вероятность того, что у пациента грипп, при условии, что у него есть температура. Можно строить догадки, но это не очень простой вопрос. Намного легче ответить на вопрос противоположного свойства: если у больного грипп, какова вероятность, что у него высокая температура? Ответ получить легко — эта вероятность равна единице, а дальше формула Байеса подсказывает, что нужно сделать.

Но предварительно необходимо прояснить еще один момент, а именно выяснить значение знаменателя в уравнении, т.е.  $P(E)$  — вероятность самого свидетельства, в нашем случае вероятность, что у человека имеется высокая температура. Ведь если у каждого всегда повышенная температура, то тот факт, что у нашего пациента жар, ничего нам не объяснит. У нашего пациента жар, но является ли такое состояние в общем случае редким событием?

Для вычисления  $P(E)$  мы имеем:

$$P(E) = P(E : H) P(H) + P(E : \text{не } H) P(\text{не } H).$$

Другими словами, вероятность того, что у произвольного человека имеется высокая температура, равна вероятности температуры у больного гриппом, умноженной на вероятность гриппа у произвольно взятого человека, плюс вероятность высокой температуры у пациента не больного гриппом, умноженная на вероятность того, что данное лицо не болеет гриппом.

А теперь вы, возможно, заметили, насколько все это хорошо подходит под выбранное нами определение базы знаний:  $P(E : H)$  — это  $p^+$ , связанная с гриппом, а  $P(E : \text{не } H)$  — это  $p^-$ , так что мы получаем формулу

$$P(H : E) = \frac{p^+ p}{p^+ p + p^- (1 - p)}.$$

Здесь начинают проявляться достоинства байесовского метода. Первоначальным значением  $P(H)$ , которым мы воспользовались, была  $p$  (априорная вероятность), которая хранилась в нашей базе знаний. Но, вычислив новую величину  $P(H : E)$  (т.е. задав, скажем, вопрос о наличии у больного высокой температуры), мы теперь можем забыть первоначальную  $P(H)$  и воспользоваться значением  $P(H : E)$  как обновленным значением  $P(H)$ . Так что весь процесс можно повторять еще и еще, каждый раз несколько увеличивая или уменьшая вероятность заболевания, но каждый раз обращаясь к одной и той же формуле Байеса, подставляя в нее каждый раз новую априорную вероятность, получаемую из апостериорной вероятности, имевшейся на предыдущем шаге.

В конечном итоге, собрав все сведения, касающиеся всех гипотез, мы (или наша система) можем прийти к окончательному заключению, выведя, что верная гипотеза является истинной (конечно; если все идет гладко).

Все это выглядит хорошо, но в отношении этого метода можно высказать одно-два критических замечания, и я не буду откладывать их анализ. Первое касается вычисления  $P(E)$ . Представленное вычисление является точным и очень простым, т.е. оно является точным, если мы знаем  $P(E : \text{не } H)$ , а это значение не всегда может быть точно известно. Подумайте в качестве примера о вероятности того, что у кого-то высокая температура, если известно, что у этого человека нет гриппа. Как правило, мы будем знать очень точное значение для  $P(E : H)$ , вероятности высокой температуры при условии, что у больного — грипп, но не всегда для  $P(E : \text{не } H)$ .

Одна из возможностей обойти это затруднение заключается в использовании общей формулы для вероятности некоторого свидетельства

$$P(E) = \sum_{i=1}^n P(E : H_i) P(H_i)$$

с учетом того, что формула, использованная нами ранее, является частным случаем этого выражения.

Если  $P(H)$  постоянно уточняется в ходе работы, то  $P(E)$  теоретически также может уточняться с течением времени.

Это естественная мысль, поскольку формула гласит, что вероятность свидетельства  $E$  есть вероятность  $E$  при учете каждой данной ситуации, которая может повлечь за собой появление  $E$ , умноженная на вероятность такой ситуации. Эта формула учитывает все. Разумеется, если в нашей базе данных хранится каждая ситуация, с которой может быть связано появление высокой температуры, и указаны текущие вероятности таких ситуаций, то мы могли бы иметь точную оценку  $P(E)$ , которая будет постоянно обновляться по мере изменения  $P(H_i)$ , оставаясь всегда точной. Было бы очень неплохо работать таким образом, но трудность состоит в том, что наша система в некотором смысле всегда будет далека от совершенства. Скажем, такая болезнь, как ласская лихорадка, также вызывает жар, но если этой болезни нет в базе данных, тогда это приведет к соответствующей ошибке при вычислении  $P(E)$ , точно так же как и при формировании окончательного результата. Аналогично обстоит дело и для каждого элемента, который отсутствует в базе данных.

С практической точки зрения использование грубых оценок для  $P(E : H)$  и для  $P(E : \text{не } H)$  вполне допустимо, и если вас это беспокоит, то вы всегда можете утешить себя тем, что, используя более простой метод, вы экономите объем необходимых вычислений, благодаря чему ответ получается сегодня, а не на следующей неделе.

Второе критическое замечание, которое может быть выдвинуто, касается того, что в этом методе используется предположение о независимости переменных. С точки зрения теории это замечание очень серьезно. Предположим, что в нашей базе знаний имеется два симптома. Одним является "высокая температура", а вторым — "жар". Конечно, неразумно иметь оба эти

симптома, поскольку они означают одно и то же состояние. Эти элементы полностью коррелируют друг с другом. Но если оба они уже попали в базу знаний, тогда мы фактически включили одно свидетельство дважды и, следовательно, увеличили или уменьшили вероятность гриппа по сравнению с тем, что должно было быть, так что апостериорные вероятности будут ошибочными.

В каком-то небольшом масштабе подобная трудность не может не возникнуть просто потому, что большинство симптомов (или "показаний", если речь идет не о медицине, а о других областях) всегда в какой-то степени коррелируют друг с другом. И из этого положения нет никакого "теоретически приемлемого" выхода. Я говорю "теоретически приемлемого" по той причине, что эта проблема больше беспокоит статистиков, чем кого-либо еще. А причина заключается в следующем.

Если в конце процесса мы хотели бы иметь точные вероятности для каждой из рассматриваемых гипотез, тогда в отношении используемого метода можно было бы высказать множество критических замечаний. Но если нам достаточно знать порядок отношения вероятностей, то вся проблема, по-видимому, исчезнет. До тех пор пока информация в базе знаний одинаково ошибочна в отношении каждого ее элемента и числа свидетельств, имеющих для каждой гипотезы, примерно одно и то же, то и относительный порядок ошибок в целом примерно одинаков. Какие-то знаки после запятой могут оказаться ненадежными, но в целом, получаемая системой картина выглядит вполне убедительной. Может быть, это высказывание удастся сделать более сильным, если я на время перейду к рассмотрению вопроса о шансах.

Шансы и вероятности связаны следующей формулой:

$$O(H) = \frac{P(H)}{1 - P(H)},$$

так что для гипотезы с вероятностью 0,5 шансы равны.

Кроме того, некоторые отдают предпочтение вычислению байесовской суммы с использованием шансов, а не вероятностей только потому, что это оказывается проще для вычислений. Переходя к шансам, получаем из наших предыдущих формул

$$O(H : E) = \frac{P(E : H)}{P(E : \text{не } H)} O(H) = \frac{p^+}{p^-} O(H).$$

Таким образом,

$$\ln |O(H : E)| = \ln \left| \frac{p^+}{p^-} \right| + \ln |O(H)|.$$

Другими словами, шансы являются простыми функциями  $P(E : H)$  и  $P(E : \text{не } H)$ , и если перейти к логарифмам величин, а в базе знаний хранить логарифм отношения  $p^+/p^-$ , то все вычисления сведутся просто к суммированию



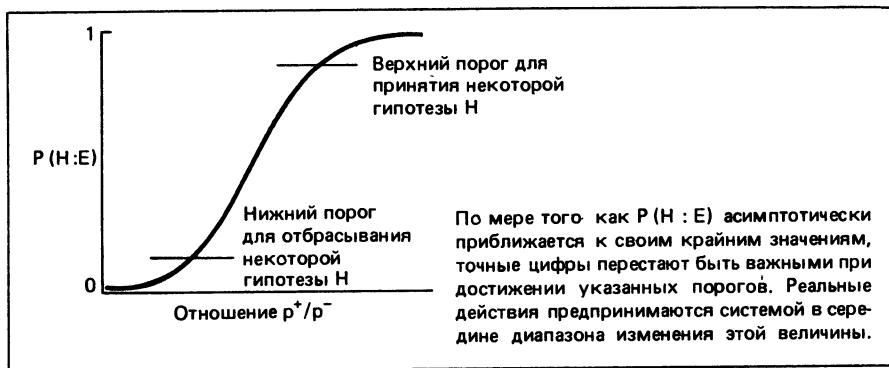


Рис. 6.2. Пороговые величины

У меня лично имеются некоторые возражения к использованию шансов. Главное из них сводится к тому, что крайние значения шансов равны "плюс" и "минус" бесконечности, тогда как для вероятностей это 0 и 1. Поскольку бесконечности находятся за пределами возможностей большинства вычислительных машин, то вместе с этими крайними точками теряется и очень ценная информация, поскольку события, происходящие с вероятностью 0 и 1, являются очень интересными и могут содействовать быстрому решению той задачи, которая анализируется в данный момент.

Отвлекаясь в еще большей степени, заметим, что поскольку шансы зависят линейно от отношения  $p^+/p^-$ , то, следовательно, величина  $P(H:E)$  при изменении этого отношения асимптотически стремится к 0 или 1 (рис. 6.2). Эта величина при надежных свидетельствах может на самом деле достигать приведенных крайних значений. Но до этого момента изменение указанной вероятности с приходом новых свидетельств происходит все в меньшей степени, что и позволяет смягчить влияние возможных корреляций элементов базы знаний. Может быть, что на один и тот же вопрос получено два идентичных ответа, но это едва ли отразится на окончательном результате, если мы будем придерживаться вероятностей и если у системы имеется достаточный запас других свидетельств. По существу, я считаю и, вероятно, многие другие, что восприятие вероятностных событий находится в хорошем соответствии с вероятностными мерами, а не с мерами типа шансов, которые оказываются менее предпочтительными.

Итак, мы бегло рассмотрели байесовские схемы и общие свойства процесса вывода, но, прежде чем двигаться дальше, мы должны остановиться на критериях, которые используются системой при выработке решения. Сейчас мы могли бы заставить систему, если бы этого захотели, напечатать на выходе все гипотезы с указанием ассоциированных с ними вероятностей. Такая система была бы слишком примитивной.

Обычно исследователи устанавливают два порога для вероятности гипотезы — верхний и нижний. Если вероятность превосходит верхний порог,

то гипотеза принимается как основа для возможного заключения. Если она оказывается ниже нижнего порога, то гипотеза отвергается как неправдоподобная. Благодаря этому система способна проявлять при выводе весьма разумное поведение. На практике такой метод жестко установленных порогов является открытым для разнообразной критики, и, чтобы показать, что я имею в виду, рассмотрим некоторую гипотезу, относительно которой в какой-то момент сразу стали известны некоторые свидетельства.

Предполагая, что все свидетельства независимы, можно вместо рассмотрения пошаговой процедуры, чем мы занимались раньше, просто произвести одно общее вычисление, чтобы найти величину  $P(H : \text{все относящиеся к делу поддерживающие данные})$ .

В своей основе такое вычисление носит тот же характер, что и раньше, но теперь вместо  $P(E : H)$  мы возьмем  $P(\text{все поддерживающие данные} : H)$ , которая представляет собой произведение всех  $P(E : H)$  для данной гипотезы.

Поступая так, мы получаем максимально достижимую для этой гипотезы апостериорную вероятность, которая соответствует случаю, когда каждый элемент данных, имеющийся в базе знаний в данный момент, говорит в пользу гипотезы.

Точно так же мы можем вычислить и минимально возможную вероятность для такой гипотезы, предполагая, что все свидетельства, имеющиеся в базе знаний, говорят против гипотезы.

Ясно, что такие значения будут разными для разных гипотез. На самом деле, мы можем установить, что для некоторых гипотез максимально возможная вероятность равна, скажем, 0,5. Так что если мы установим жесткие верхний и нижний пороги равными 0,9 и 0,6, то эти конкретные гипотезы всегда будут отбрасываться как маловероятные.

Однако предположим, что гипотеза состояла в том, что у больного — рак легких. Ясно, что, используя обычный терминал ЭВМ, нельзя с помощью

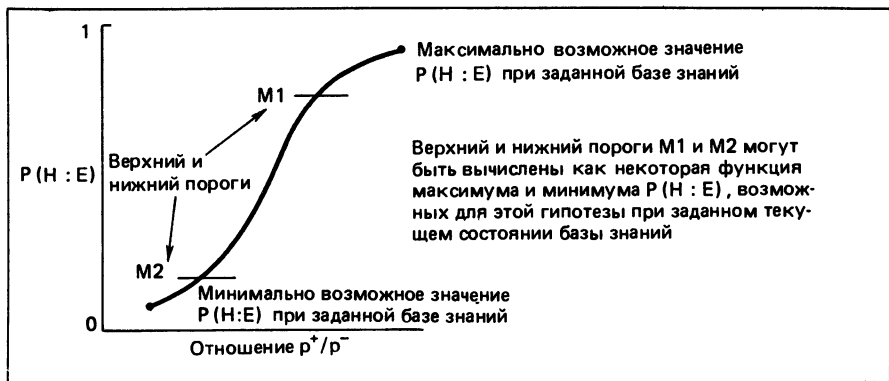


Рис. 6.3

компьютера диагностировать рак легких. Необходимо, как минимум, иметь еще рентгеновский снимок. Так что все, что вы хотите от машины, — это мнение системы о конкретной гипотезе в той степени, в какой это возможно при имеющихся у систем знаниях. Иными словами, вы хотите, чтобы система сказала, насколько она уверена в этой гипотезе, исходя из ограничения, что ни рентгеновский снимок, ни прямые наблюдения ей недоступны.

В таком случае можно установить верхний порог,  $M_1$ , который может составлять 0,9 максимального, достижимого для рассматриваемой гипотезы при заданной базе знаний. Аналогично нижний порог,  $M_2$ , может исчисляться как половина от этого максимума или как минимально возможное значение вероятности, умноженное на некоторый коэффициент. При таком подходе верхний и нижний пороги становятся специфическими для каждой конкретной гипотезы (рис. 6.3).

### 6.3. НЕОПРЕДЕЛЕННОСТЬ, ЗАКЛЮЧЕННАЯ В РЕАКЦИИ ПОЛЬЗОВАТЕЛЕЙ

Может быть, кто-то подумал, что мы приближаемся к концу нашего анализа вопросов вывода. Однако следует добавить еще учет неопределенности, заключенной в реакциях пользователей.

В идеале, когда система обращается за конкретной информацией, пользователь дает прямой ответ на вопрос. Но может быть так, что пользователь не знает, какой ответ ему дать. Пример тому, который лично мне импонирует, встречается в работе прекрасной экспертной системы PROSPECTOR, которая в ходе процесса поиска полезных ископаемых в какой-то момент задает пользователю вопрос: "Не произошло ли превращения роговой обманки в биотит в результате замещения?" Я весьма сочувствую тому, кто не вполне уверен, что ответить на такой вопрос.

Подобная трудность может встретиться и в других областях. Когда задается такой вопрос: "Есть ли у вас сильный кашель?", то необходимо дать пользователю возможность отвечать, пользуясь, скажем, одиннадцатибальной шкалой, на которой +5 означает "Да", -5 означает "Нет", а 0 — "Я не знаю", тогда как все остальные варианты ответа располагаются где-то в промежуточных точках шкалы.

В этом случае вычисления протекают в значительной степени так же, как раньше, но с заменой  $P(H : E)$  на  $P(H : R)$ , вычисляемую по формуле

$$P(H : R) = P(H : E) P(E : R) + P(H : \text{не } E) P(\text{не } E : R),$$

где  $R$  представляет собой ответ пользователя. Другими словами, чтобы допустить неопределенность, система должна допустить некоторую вероятность  $P(H : E)$  и некоторую вероятность  $P(H : \text{не } E)$ , величины которых зависят от той степени, в которой ответ пользователя поддерживает или опровергает конкретный элемент данных (рис. 6.4).

Ясно, что если этот ответ нулевой, то  $P(H : R) = P(H)$ , т.е. ничего не меняется, а при других ответах неопределенность отражается путем кусочно-линейной интерполяции между  $P(H : E)$  и  $P(H : \text{не } E)$ .

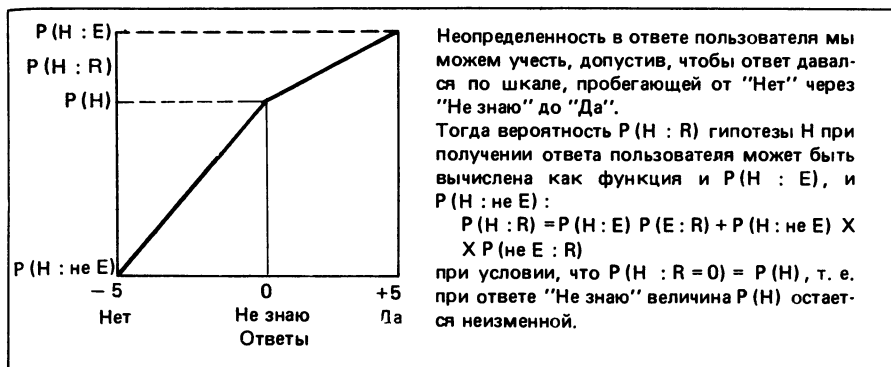


Рис. 6.4. Учет неопределенности в ответе пользователя

На этом в общих чертах заканчивается рассмотрение вопросов дедукции в экспертной системе.

#### 6.4. МАШИНА

Итак, мы подошли к проблеме машины, т.е. того средства, которое будет использоваться в системе путешествия среди возможных дедуктивных выводов. Напрашивается вопрос, а так ли это важно? В конце концов, система могла бы взять список вопросов и последовательно их прорабатывать.

Таким образом поступить можно — это так называемая прямая цепочка рассуждений. При такой направляемой данными стратегии задается вопрос, берется ответ пользователя, затем делается тот вывод, который возможен при наличных данных. После этого задается следующий вопрос, и так далее, пока все гипотезы не будут окончательно подтверждены либо опровергнуты.

В сущности, при использовании такого рода подхода возникает две проблемы. Во-первых, процедура может оказаться слишком долгой. Некоторые данные, очевидно, могут оказаться более общезначимы, чем другие, и оставлять их получение на самый конец, что может произойти в силу совершенно случайных обстоятельств, — значит затянуть весь процесс получения ответа.

Рассмотрим два вопроса: "Вам не здоровится? Есть ли у Вас утолщение пальцев руки?" Ясно, что первый вопрос является очень хорошим, если задать его в самом начале сеанса диагностики, поскольку отрицательный ответ позволяет сразу отбросить множество заболеваний, тогда как положительный ответ увеличит вероятности целого ряда других заболеваний. Что касается второго вопроса, то положительный ответ часто говорит о присутствии некоторых легочных заболеваний, но даже он не является четким свидетельством в пользу того или иного заболевания, тогда как отрицательный ответ вообще отмечает очень небольшое число случаев.

Вторым недостатком использования прямой цепочки рассуждений является то, что для пользователя система выглядит бесцельной: выстреливается

серия из наугад выбранных вопросов, а затем система либо останавливается, либо отказывается выдать диагноз.

Конечно, если бы первая проблема (задание "хорошего" вопроса) была решена, то и эта вторая проблема была бы не столь важной, поскольку пользователь по крайней мере видел бы, что система пытается установить диагноз.

Другим часто используемым методом является построение обратной цепочки рассуждений. В этом случае система сначала выбирает некоторую гипотезу из базы знаний, затем она осматривается, пытаясь найти свидетельства, которые ей нужны, чтобы либо оставить, либо отбросить эту гипотезу, и задает пользователю соответствующий вопрос.

По своему поведению такая система выглядит более целенаправленной, потому что она всегда в состоянии сказать, что "у нее на уме" в соответствии с текущей теорией, которая определяется рассматриваемой в этот момент гипотезой. Но такой подход в исходной своей форме страдает рядом недостатков. Главным образом они являются следствием того, что возникает проблема, в каком именно порядке гипотезы выбираются для анализа, а также в каком порядке следует отыскивать поддерживающие эти гипотезы данные.

Предположим, что первым в базе знаний идет аллергическое заболевание. Строя как обычно обратную цепочку, система решает сначала рассмотреть эту гипотезу. Глядя на возможные "показания" для аллергического заболевания, система обнаруживает, что его симптомы часто связаны с излишним потреблением чая. Или, по крайней мере, различные данные, связанные с чрезмерным употреблением чая, являются первыми в базе знаний системы.

Итак, пользователь обращается к своему доктору, и первым делом он слышит: "Много ли вы пьете чая"? Система преследует определенную цель, но для пользователя ее поведение может показаться даже оскорбительным. Система напоминает мне в этом случае медика, от которого я однажды страдал, потому что он был убежден, что все люди делятся на два лагеря — те, у кого воспалены железы, и те, у кого они здоровы.

## 6.5. ПОДХОД С ЦЕНАМИ СВИДЕТЕЛЬСТВ

Я предложил другой подход к общей проблеме: какой вопрос следует задать в следующую очередь. В этом подходе вводятся оценки свидетельств, и его было предложено называть также "косвенной цепочкой рассуждений". В нем не используются ни прямая, ни обратная цепочки рассуждений, которые ориентируются на учет гипотез. Наш подход ориентируется на свидетельства. В сущности, он сводится к тому, что каждому элементу свидетельства приписывается цена, отражающая его роль в процессе вывода, и в первую очередь задается тот вопрос, для которого цена оказывается наибольшей.

Интуитивно этот метод представляется вполне разумным, поскольку теперь система задает те вопросы, которые ей кажутся важными, постоянно

$$\text{ЦС} = \sum_{i=1}^n |P(H_i : E) - P(H_i : \text{не } E)|$$

Таким образом, цена свидетельства вычисляется для каждого свидетельства  $E$  как сумма максимальных изменений вероятностей, которые могут произойти во всех  $n$  гипотезах, которым это свидетельство приложимо. Первым всегда задается тот вопрос, у которого такая цена оказывается наибольшей.

Рис. 6.5. Цены свидетельств ("косвенная цепочка рассуждений")

но модифицируя свои представления по мере поступления ответов. Похоже, что именно так себя ведут специалисты.

Проблема, конечно, состоит в том, как выделить важный вопрос. В простейшем случае можно вычислить цену каждого свидетельства (ЦС) как полную сумму максимальных изменений вероятностей по всем гипотезам, имеющимся в базе знаний, к которым такое свидетельство может привести (рис. 6.5).

Цена свидетельства вычисляется для каждого свидетельства  $E$  из базы знаний. Затем система может найти максимальную ЦС и потребовать информацию для соответствующего свидетельства.

Ясно, что такие цены свидетельств не остаются неизменными. По мере того как постоянно уточняются апостериорные вероятности  $P(H_i : E)$ , они будут приводить к непрерывному изменению цен свидетельств. Например, если в ходе диалога какое-то конкретное множество гипотез окажется почти полностью "уничтоженным", то остается мало шансов на то, что их вероятности будут существенно изменяться, а следовательно, применимое к ним свидетельство станет менее важным при последующих пересчетах цен. В результате уменьшится вероятность того, что такие вопросы теперь будут задаваться. И наоборот, по мере того как некоторые гипотезы становятся более вероятными, возрастают шансы на то, что их вероятности будут изменяться под воздействием оставшихся релевантных свидетельств. В результате для этих свидетельств цены будут более высокими, поэтому соответствующие им вопросы будут заданы с большей вероятностью. Дело выглядит так, как будто бы система располагает некоторой мерой "внимания", уделяемого в основном тем гипотезам, относительно которых имеются наибольшие шансы их прояснения в каждый данный момент.

Другое большое преимущество описываемого метода — относительная простота реализации, что является немаловажным.

Возможны, конечно, варианты построения оценки для свидетельств. Например, мы могли бы взять не сами ЦС, а их квадраты, подчеркнув тем

самым те данные, которые дают наибольшие изменения вероятностей для некоторых гипотез за счет тех элементов, которые создают лишь небольшие изменения вероятностей, но на большом числе гипотез. Разумеется, именно в случае больших изменений возникает возможность быстрого принятия или опровержения гипотезы и сокращения объема дальнейшей обработки.

Другим вариантом могла бы быть сумма квадратов отклонений от среднего значения ЦС. Этим снова преследуется цель выделить более четкую структуру в решаемой задаче, еще больше подчеркивая те элементы, которые могут дать большие изменения в некоторых гипотезах в противовес многим малым изменениям в большом числе гипотез, которые могли бы отклоняться влево и вправо от среднего, не приводя ни к какому определенному и быстрому решению.

Еще одним уточнением метода может служить "взвешивание" цен свидетельств, при котором больший вес получают те свидетельства, которые оказались применимыми к гипотезам, имеющим отношение к последнему завершенному свидетельству.

Этим можно избежать "нервного" поведения системы, состоящего в следующем. Предположим, что система работает и в настоящее время рассматриваются две группы гипотез — "группа А" и "группа Б". Найдено наибольшее значение правила, задан вопрос, и оказалось, что свидетельство в большей степени приложимо к гипотезам из группы А. Однако следующий вопрос, который необходимо задать в соответствии со следующим наибольшим значением свидетельства, хотя и с весьма малым запасом, но оказывается приложимым главным образом к гипотезам из группы Б. Тогда следующим идет вопрос для группы Б, затем — для группы А, и т.д.

Все это носит случайный характер, и различие в ценах свидетельств, которое вызывает такую "нервность", может быть весьма малым. Но для пользователя все будет выглядеть так, как будто система весьма странным образом перепрыгивает с одного предмета на другой. Поэтому необходимо сделать систему несколько более инерционной, чтобы заставить ее задержаться в некоторой точке на какое-то время, что создаст больший эффект "внимания" со стороны системы.

Степень, в которой подобные усовершенствования метода будут целесообразны на практике, зависит от того, какое внимание предполагается уделить программированию и какие дополнительные расходы по обработке данных признаны допустимыми. В сущности, простой подход с ценами свидетельств "работает" очень хорошо и не приводит к излишнему замедлению работы системы.

На самом деле, самое большое затруднение вызывает оценка каждого из подходов, поскольку на практике совсем не ясно, какую последовательность задаваемых вопросов следует считать хорошей, а какую — плохой. Конечно, из ряда вон выходящую последовательность заметит каждый. Но гораздо труднее оценить достоинства последовательности, которая достаточно хороша по сравнению с той, которая теоретически могла бы оказаться и лучше данной.

Эта проблема — проблема оценки работающей системы — отнюдь не является тривиальной. Если вы располагаете экспертной системой вывода, которая способна собрать 100 свидетельств, каждое из которых измеряется по одиннадцатибальной шкале, то возможно 1100 различных путей, по которым ее может повести пользователь.

Для каждого из 1100 таких путей имеется 100! (факториал) различных способов упорядочить вопросы. При числах такого гигантского порядка никакой компьютер не будет в состоянии вычислить, сколькими различными способами можно повести дело, не говоря уже об оценке каждого такого способа. И тем не менее система такого рода вполне помещается в компьютер Apple II или Sinclair Spectrum.

Итак, вы выдумываете свой собственный метод и весьма грубо оцениваете, что он дает. И если он не разваливается на ваших глазах, то, по-видимому, этот метод вполне хорош.

## 6.6. ВЫРАБОТКА ЗАКЛЮЧЕНИЯ

В сущности, система работает, она задает хорошие вопросы, адаптируя свои гипотезы с учетом получаемых системой ответов. Она может продолжать свою работу до тех пор, пока у нее не останется больше вопросов, и остановиться. Но она может прийти также и к некоторому заключению. Она может сказать: "Я думаю, что в данном случае — это такое-то заболевание и, возможно, вам следует также учесть и другие элементы. . ."

Первым следует рассмотреть *наиболее вероятный результат* системы. В простейшем случае это может быть результат, имеющий максимальную вероятность, но здесь следует помнить все, что было сказано о максималь-

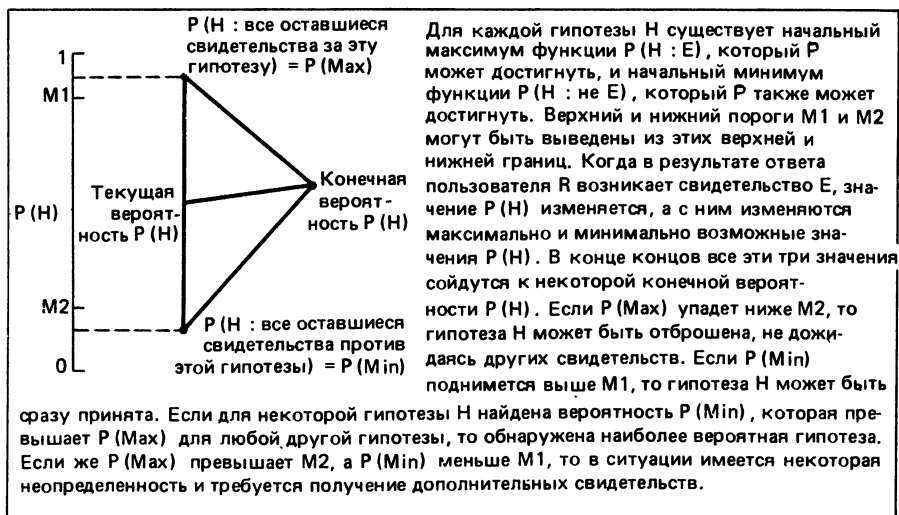


Рис. 6.6. Минимумы, максимумы и построение заключений



ной и минимальной вероятностях для каждой гипотезы, зависящих от информации, хранящейся в базе знаний в данный момент.

С любой гипотезой связано пять величин, каждая из которых является вероятностью и открывает большие возможности для появления различных недоразумений. Тем не менее приведем их на рис. 6.6:

- $P(H)$  — текущая оцениваемая вероятность того, что данная гипотеза является истинной.
- $P(Max)$  — текущая максимальная вероятность для данной гипотезы, которая могла бы быть достигнута, если все остающиеся свидетельства были бы в ее пользу.
- $P(Min)$  — текущая минимальная вероятность для данной гипотезы, которая могла бы быть достигнута, если все остающиеся свидетельства были бы против этой гипотезы.
- $M1$  — критерий верхнего порога для принятия конкретной гипотезы, вычисляемый как некоторая пропорциональная  $P(Max)$  величина еще до того, как начался сбор свидетельств.
- $M2$  — критерий нижнего порога для отказа от конкретной гипотезы, пропорциональный либо  $P(Max)$ , либо  $P(Min)$ , вычисляемый еще до того, как начался сбор свидетельств.

Таким образом, считается, что найден *наиболее вероятный результат*, если существует некоторая гипотеза,  $P(Min)$  которой больше, чем  $P(Max)$  для любой другой гипотезы. Другими словами, если даже еще не все свидетельства были собраны, может представиться возможность сказать, какой результат является наиболее вероятным, поскольку никакой другой вариант не может опровергнуть данную гипотезу. Это соображение может помочь прервать общение с пользователем ранее обычного.

Следующим классом заключений являются *вероятные заключения*, т.е. те гипотезы, для которых  $P(Min)$  больше, чем  $M1$ , т.е. для них минимально достижимая вероятность превышает верхний порог.

Следующим классом являются *неопределенные заключения*. Это те гипотезы, для которых  $P(Min)$  меньше, чем  $M1$ , но  $P(Max)$  больше, чем  $M2$ . Другими словами, они не превзошли верхний порог (хотя это могло бы произойти при поступлении новых данных), но они, вероятно, будут превышать нижний порог, поэтому их следует принимать во внимание и впредь. Подобные гипотезы, безусловно, могут вынудить систему продолжить задавать вопросы пользователю с тем, чтобы снять возникшую неоднозначность результата.

В самом низу нашего списка мы имеем случай, когда невозможен никакой вывод, не считая того тривиального заключения, что ни одна из гипотез не является истинной. При этом наибольшая  $M(Max)$  меньше, чем  $M2$ , т.е. нижний порог. В общем случае некоторая неопределенность существует еще тогда, когда  $P(Max)$  не равно  $P(Min)$ , но равно  $P(H)$ . Но стоит ли заниматься разрешением такой неопределенности, зависит от того, насколько далеко различные гипотезы могут переходить за пороговые значения  $M1$  и  $M2$ , что определяется сравнением  $P(Max)$  и  $P(Min)$  с  $M1$  и  $M2$ .

Теперь я перейду к моему заключению, дав общее изложение описанного мною процесса:

1. Для каждой гипотезы найти априорную вероятность. Это делается путем просмотра базы знаний и извлечения  $p$  для каждой из гипотез (см. рис. 6.1). Эта величина берется в качестве  $P(H)$ , которая в дальнейшем будет уточняться.
2. Для каждого свидетельства найти его цену. Это делается путем просмотра базы знаний и извлечения переменных  $(j, p^+, p^-)$ , где  $j$  — "номер свидетельства", а  $p^+$  и  $p^-$  — соответственно  $P(E : H)$  и  $P(E : \text{не } H)$  для этого свидетельства. Затем вычисляется ЦС для каждого свидетельства в соответствии с формулой для ЦС.
3. Из всех ЦС найти наибольшее, указывающее на то свидетельство, которое может вызвать наибольшее изменение вероятности по всем рассматриваемым гипотезам.
4. Расспросить пользователя по поводу этого свидетельства, используя вопрос, хранимый в базе знаний. Ответ пользователя может быть дан на шкале от  $-5$  до  $+5$ ; обозначим его символом  $R$ .
5. При заданном  $R$  сделать пересчет для всех гипотез, в которых упоминалось это свидетельство, с тем, чтобы установить  $P(H : R)$  (см. рис. 6.4.).
6. Сделать пересчет цен для всех свидетельств с тем, чтобы учесть изменения в вероятностях, которые произошли в результате получения реакции пользователя.
7. Вычислить те минимальные и максимальные величины, которые еще достижимы для каждой гипотезы (см. рис. 6.6).
8. Найти наибольший из возможных минимумов для этих гипотез.
9. Проверить, существует ли гипотеза, максимально возможное значение которой превышает максимум из этих минимумов. Если существует такой максимум, который превышает наибольший минимум, то вернуться к пункту 3 и задать другой вопрос. Если такого нет, то существует наиболее вероятный результат, который и следует выдать пользователю.
10. Вызвать из системы резюмирующую программу, в ходе которой она сообщает пользователю (возможно, снабдив все звуками фанфар) подробное описание всех выводов, которые были проделаны системой.

И наконец, никогда не рассказывайте конечному пользователю, как была написана эта дедуктивная машина. Если вы и расскажите ему, то он вам не поверит, поскольку, как и большинство людей, он думает, что сделать вывод очень легко. И разумеется, он будет прав (до тех пор, пока вы не захотите использовать для этого компьютер).

#### ПРИЛОЖЕНИЕ: ПРИМЕР БАЗЫ ЗНАНИЙ

Предположим, что вы хотите иметь экспертную систему, которая могла бы играть роль вашего личного механика по ремонту автомашин. Это пример очень небольшой

базы знаний, рассчитанной на обеспечение советом в области ремонта автомашин с использованием методов, описанных выше:

#### СЕВШИЙ АККУМУЛЯТОР

0,1; 5; 1; 0; 0,99; 2; 0,7; 0,05; 4; 0,2; 0,5;

5; 0; 0,99; 6,1; 0,01

#### НЕТ БЕНЗИНА

0,05; 2; 2; 1; 0,01; 6; 0,9; 0,02

#### ОТСЫРЕЛО ЗАЖИГАНИЕ

0,01; 3; 3; 0,9; 0,1; 4; 0,25; 0,5; 6; 0,9; 0,02

#### ЗАМАСЛЕНЫ СВЕЧИ

0,01; 2; 4; 0,01; 0,5; 6; 0,9; 0,02

#### 1, ФАРЫ ГОРЯТ, ГОРЯТ ЛИ ФАРЫ?

#### 2, УКАЗАТЕЛЬ БЕНЗИНА НА НУЛЕ, НЕ ПОКАЗЫВАЕТ ЛИ ПРИБОР УРОВНЯ БЕНЗИНА СЛИШКОМ МАЛОЕ ЗНАЧЕНИЕ?

#### 3, АВТОМАШИНА ОТСЫРЕЛА, НЕ СТОЯЛА ЛИ АВТОМАШИНА ДОЛГО ПОД ДОЖДЕМ?

#### 4, АВТОМАШИНА НЕДАВНО ПРОШЛА ТЕХОБСЛУЖИВАНИЕ, ПРОХОДИЛА ЛИ НЕДАВНО ЭТА АВТОМАШИНА ТЕХОБСЛУЖИВАНИЕ?

#### 5, СТАРТЕР ПРОВОРАЧИВАЕТСЯ, ПРОВОРАЧИВАЕТСЯ ЛИ СТАРТЕР АВТОМАШИНЫ?

#### 6, АВТОМАШИНА НЕ ЗАВОДИТСЯ, АВТОМАШИНА НЕ ЗАВЕДЕТСЯ?

Здесь первая группа элементов — это гипотеза, например гипотеза, что батарея села. Вторая группа содержит свидетельства, например информацию о том, что фары горят.

Поскольку я не считаю себя прекрасным механиком, то приводимые мною вероятности могут несколько отклоняться от истинных значений, но, не обращая на это внимание, рассмотрим первую гипотезу о том, что сел аккумулятор. Априорную вероятность того, что на любой данной автомашине сел аккумулятор, возьмем, скажем, равной 0,1, т. е. считается, что в норме у каждой десятой автомашины севший аккумулятор.

Мы прикинули, что имеется пять различных свидетельств, которые могут быть использованы для того, чтобы установить, исправна или нет аккумуляторная батарея на данной автомашине. Первое свидетельство идет под номером 1 (фары горят). Если батарея села, то вероятность наблюдения, что фары горят, равна нулю. С другой стороны, если батарея исправна, то  $P(E : \text{не } H)$  — вероятность того, что фары горят, равна, скажем, 0,99 (т. е. почти наверняка фары будут работать, если не считать той возможности, что перегорела лампочка или сгорел предохранитель).

Второе свидетельство идет под номером 2. Указатель бензина на нуле. Поскольку этот прибор работает на электричестве, то он тоже может иметь отношение к состоянию аккумулятора. Правда, это не очень важно, потому что для работы указателя требуется очень небольшой ток. Таким образом, мы прикидываем, что вероятность наблюдать низкое показание прибора уровня бензина, если батарея села, равна 0,7, а вероятность низкого показания прибора при исправной батарее равна 0,05. (Низкое показание может быть и по другой причине, например потому, что нет бензина!) Заметим, что приведенная вероятность соответствует априорной вероятности гипотезы "нет бензина", поскольку это единственно возможная иная причина того, что указатель бензина близок к нулю. Точно так же можно и дальше просмотреть список приложимых к этой гипотезе свидетельств. Если автомашина недавно прошла техобслуживание, то вероятность того, что аккумулятор сел, существенно меньше; это полностью исключено, если стартер проворачивается; автомашина, безусловно, не заведется, если аккумулятор сел (вероятность равна 1), хотя она может не завестись и при исправном аккумуляторе (вероятность, скажем, 0,01).

Т а б л и ц а 6.1

Свидетельство	Цена свидетельства
Фары горят	0,9174
Указатель бензина на нуле	1,4151
Автомашина отсырела	0,0822
Автомашина недавно прошла техобслуживание	0,1376
Стартер проворачивается	0,9174
Автомашина не заводится	2,3807

Следующая гипотеза "нет бензина" зависит от таких свидетельств, как "указатель бензина на нуле" и "автомашина не заводится". Аналогично гипотеза "отсырело зажигание" зависит от свидетельств, что "автомашина отсырела", "автомашина недавно прошла техобслуживание" и "автомашина не заводится". Замасленные свечи определяют такими свидетельствами, как "автомашина недавно прошла техобслуживание" и "автомашина не заводится" (первое говорит против такой возможности, а второе свидетельствует за нее). Эти вероятности могут быть взяты из опыта или основаны на более строгих соображениях, но окончательное оправдание могут дать хорошие результаты работы всей системы.

Обратимся к рис. 6.5, содержащему соотношению для вычисления цен правил. Если применить его к нашему примеру, то мы получим результаты, сведенные в табл. 6.1.

Каждый раз при запуске система всегда будет задавать вопрос: "Заводится ли автомашина?" После этого в зависимости от ответа будут уточнены цены свидетельств. Но одного взгляда на табл. 6.1 достаточно, чтобы убедиться, что после свидетельства "автомашина не заводится" систему больше всего интересует вопрос о том, показывает ли что-то указатель бензина и проворачивается ли стартер, что выглядит вполне разумным.

Для примера, если ответом на вопрос: "Автомашина не заведется?" — было твердое "да", то система вычислит заново цены свидетельств (табл. 6.2).

Так что следующий вопрос будет касаться указателя уровня бензина. Но обратите внимание, как подскочили значения всех оставшихся цен свидетельств, когда была получена новость, что ваша машина не заводится. Внезапно механик по ремонту автомашин стал интересоваться тем, что прежде его почти не интересовало.

С другой стороны, если вы ответили "нет" на вопрос: "Автомашина не заведется?" — (т.е. машина в исправности), то значения всех цен свидетельств решительно падают. Наш механик отбрасывает все четыре гипотезы, которые он рассматривал, и приходит к заключению, что с машиной все в порядке, не задавая больше никаких вопросов.

Т а б л и ц а 6.2

Свидетельство	Цена свидетельства
Фары горят	0,9991
Указатель бензина на нуле	1,2135
Автомашина отсырела	0,7554
Автомашина недавно прошла техобслуживание	0,8153
Стартер проворачивается	0,9991
Автомашина не заводится	0

### ЧАСТЬ III

## ИНЖЕНЕРИЯ ЗНАНИЙ

### Глава 7

## ТЕХНИКА ИСПОЛЬЗОВАНИЯ ЗНАНИЙ НА ЯЗЫКЕ ПРОЛОГ

*Масауд Яздани*

### 7.1. ВВЕДЕНИЕ: ОТКУДА ПРИШЛИ ЭКСПЕРТНЫЕ СИСТЕМЫ?

Последние 25 лет дисциплина искусственного интеллекта (ИИ) в основном развивается в области, располагающейся на карте различных наук где-то между психологией и вычислительными науками. Это произошло благодаря тому, что исследователи были заняты проблемой, как заставить компьютеры решать задачи, для которых, если бы их решали люди, потребовался бы интеллект [12].

Вначале работы по ИИ носили исключительно исследовательский характер. Ученые прежде всего были заинтересованы в поиске возможных путей достижения интеллектуального поведения технических систем. Следовательно, на том этапе работы по искусственному интеллекту были ориентированы не на область психологии (хотя неоднократно утверждалось и нечто обратное). Если психологическую науку интересовало то, как на самом деле действуют люди, то специалисты по искусственному интеллекту пытались перебрать те способы, которыми то же самое можно сделать на компьютере.

Еще на исследовательском этапе ИИ стал источником теорий, интересных для психологов. Их внимание, в свою очередь, повлияло на искусственный интеллект и породило специальный интерес к области моделирования процесса познания, в которой от возможных искусственных интеллектуальных устройств дополнительно требуется, чтобы они действовали так же, как люди.

Другим существенным фактором оказалась позиция организаций, финансирующих работы по искусственному интеллекту. В США исследовательский этап искусственного интеллекта был обеспечен щедрой поддержкой, а сегодня от этой области ждут естественной отдачи. В Великобритании, где благодаря Отчету Лайтхилла [10] финансирование исследовательских работ в области искусственного интеллекта было весьма ограниченным, произошло определенное изменение. В Отчете Алви [1] предлагается весьма щедрое финансирование тех работ по искусственному интеллекту, от которых можно ожидать конкретных результатов в не столь отдаленном будущем. Роль этого второго фактора сказалась в том, что в искусственном интеллекте стало уделяться больше внимания построению полезных вычислительных систем.

## 7.2. ЧТО ТАКОЕ СИСТЕМЫ, ОСНОВАННЫЕ НА ЗНАНИЯХ?

На начальном этапе исследований по искусственному интеллекту [18] возникло всеобщее убеждение, что за интеллектуальным поведением людей скрываются их знания об окружающем мире. Попытки человека добиться интеллектуального поведения от компьютера ограничены доступными сведениями о тех общих знаниях, которыми располагают люди. Другие дисциплины, такие как психология, оказались не в состоянии предоставить ясное описание тех общих знаний, которые проявляются в поведении людей. Вследствие этого определенное внимание было сосредоточено на такой форме поведения, которую правильнее всего было бы назвать интеллектуальной. Речь идет о знаниях, которые имеются у специалиста-профессионала. Такие области уже достаточно хорошо очерчены и документально описаны в специальных журналах. Законодательство, юриспруденция, геология, механика — это те виды знания, которые легко выделяются из всего многообразия знаний, используемых человеком. Именно в этих областях знаний действуют системы, названные в Отчете Алви интеллектуальными системами, основанными на знаниях:

Интеллектуальная система, основанная на знаниях, представляет собой такую систему, в которой с помощью логического вывода знания применяются к решению поставленных задач. Можно ожидать, что такие системы в состоянии работать со знаниями, относящимися ко многим областям человеческого мышления и человеческой активности: от медицинской диагностики до сложного инженерного проектирования, от нефтяной промышленности до сельского хозяйства, от выбора военной стратегии до выдачи юридической консультации.

[Подразд. 4.4.1 Отчета Алви.]

Такая попытка работать лишь с небольшой частью знаний человека уменьшает сложность задачи, с которой сталкивается специалист по искусственному интеллекту, и позволяет создать работоспособную систему за разумный отрезок времени, оставляя при этом главные проблемы ИИ в виде долгосрочных целей проводимых исследований. И тем не менее независимо от того, насколько мал выбранный участок, имеется много вопросов, которые необходимо разрешить:

1. Каким образом с самого начала усваиваются эти знания?
2. В каком виде их представлять?
3. Как реализовать выбранные виды представления знаний на конкретных аппаратурных средствах?
4. Как обеспечивается доступ к знаниям внутри системы?
5. Каким образом знания применяются к конкретной ситуации?
6. Каким образом знания модифицируются в свете накапливаемого опыта?
7. Как обеспечивается поддержание уровня знаний, а во многих случаях — и отбрасывание не относящегося к делу знания?

Этот список вопросов можно было бы продолжать до бесконечности.

### 7.3. КАК ВЫГЛЯДИТ ЭКСПЕРТНАЯ СИСТЕМА?

Прежде чем рассматривать вопрос о том, как самим построить экспертную систему, обратимся к примеру системы такого рода. В справочнике по искусственному интеллекту [5] так описывается наиболее широко известная экспертная система MYCIN:

Система MYCIN [14] была разработана для оказания консультационной помощи при диагностике и лечении инфекционных заболеваний. Такая помощь часто необходима в условиях больницы, поскольку лечащий врач может быть неспециалистом по инфекционным заболеваниям, проявляющимся после госпитализации, например инфекции, поражающей кардиологического больного, перенесшего операцию на сердце. Временной фактор лишь осложняет проблему. Анализ (крови или мочи и т.п.), взятый у пациента, может дать некоторые указания на рост числа микроорганизмов за 12 ч, но обычно для надежной идентификации требуется от 24 до 48 и более часов. Поэтому врач часто при отсутствии полной информации должен решать, начинать ли ему лечение и какие медикаменты использовать, если лечение необходимо. И тот и другой вопрос может оказаться очень сложным.

Приведем отрывок из протокола общения врача с системой MYCIN:

- Назовите имя пациента.  
Джон Доу.
- Мужчина или женщина?  
Мужчина.
- Возраст?  
Ему 55.
- Были ли выделены культуры с положительными реакциями, указывающие на общий класс инфекции?  
Да.
- Каков тип этой инфекции?  
Первичная бактеримия.
- Когда были отмечены первые симптомы?  
5 мая 1975.
- Давайте назовем самую последнюю культуру, дающую положительную реакцию, КУЛЬТУРА-1. Откуда была взята КУЛЬТУРА-1?  
Из крови.
- Когда?  
9 мая 1975.
- Давайте назовем первый значительный микроорганизм из культуры крови МИКРООРГАНИЗМ-1. Знаете ли вы, что представляет собой МИКРООРГАНИЗМ-1?  
Нет.
- Является ли МИКРООРГАНИЗМ-1 палочкой, колбочкой или чем-то еще?  
Палочкой.
- Какова окраска по Граму микроорганизма МИКРООРГАНИЗМ-1?  
Грамотрицательная.

- Были ли найдены дающие положительные реакции культуры в местах, не связанных с очагами инфекции, относительно которых запрашивается консультация?

Нет.

- Давал ли рост МИКРООРГАНИЗМ-1 в аэробных условиях?

Да.

- Давал ли рост МИКРООРГАНИЗМ-1 в анаэробных условиях?

Да.

- Каков предполагаемый путь проникновения микроорганизма ОРГАНИЗМ-1?

Желудочно-кишечный тракт.

·  
·  
·

Когда в системе MYCIN складывается ощущение достаточно хорошего понимания ситуации, задается короткая цепочка вопросов, подготавливающих к рекомендациям по курсу лечения:

- Моя рекомендация будет основана на убеждении, что ОРГАНИЗМ-1 может представлять собой: 1 – *Pseudomonas-aeruginosa*, 2 – *Klebsiella-pneumoniae*, 3 – *E. Coli*, 4 – *Bacteroides-fragilis*, 5 – *Enterobacter*, 6 – *Proteus-non-mirabilis*. Пользуясь шкалой от 0 до 4, где большие числа соответствуют большей серьезности заболевания, как бы вы оценили степень заболевания больного?

3.

- Имеется ли у больного клинически заметная аллергическая реакция на какой-нибудь противобактериальный препарат?

Нет.

- Какова масса больного?

70 кг.

- В первую очередь я рекомендовала бы следующее: чтобы охватить случаи 1, 2, 3, 5 и 6, давайте больному в течение 10 дней гентамицин дозами по 119 мг (1,7 мг/кг). Измените дозировку в случае заболевания почек. А чтобы охватить случай 4, давайте больному в течение 14 дней клиндамицин дозами по 595 мг (8,5 мг/кг). Если обнаружатся симптомы диареи или других желудочно-кишечных заболеваний, сделайте проверку на pseudo-membranous colitis.
- Хотите ли вы знать, что я рекомендовала бы в качестве второго варианта?

На рис. 7.1 изображена поверхностная точка зрения на то, как работает система.

Программы искусственного интеллекта вообще и экспертные системы в частности отличаются от традиционных вычислительных пакетов тем, как эти программы организованы. С традиционных позиций программа представляет собой процедуру и данные. Однако экспертные системы "все могут быть описаны как системы, направляемые базами данных" [11]. На самом деле, в базе данных имеется три различных модуля:

1. Факты о признаках конкретной ситуации.



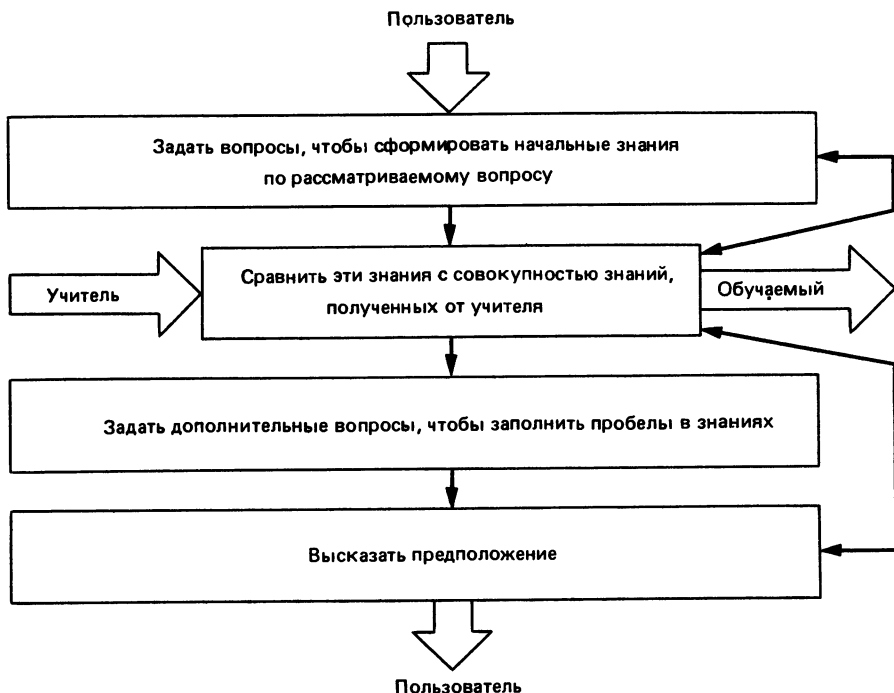


Рис. 7.1. Общее представление о работе экспертной системы

2. Правила, относящиеся к области экспертизы.
3. Правила работы с правилами 2 или "машина вывода".

В большинстве случаев различие между этими модулями носит концептуальный характер и все они могут располагаться в одной базе данных.

#### 7.4. АРХИТЕКТУРА ЭКСПЕРТНЫХ СИСТЕМ

Если компьютерной программе предстоит выполнять задачу эксперта, то она нуждается в большом объеме знаний, позволяющих решать сложные проблемы подобно тому, как это делает человек. Чтобы избежать при этом абсолютной путаницы, система должна быть тщательно организована. В общем случае знания разделяются на три типа:

**1. Фактические (декларативные) знания.** Этот вид знаний представляет информацию о конкретном случае, обычно собираемую посредством диалога с пользователем, который указывает, какие факты следует считать справедливыми в настоящее время. Важно то, как представлена эта информация, поскольку сама структура представления также информативна. Структуру представления следует выбирать в зависимости от содержания знаний.

**2. Процедурные знания.** Эти знания обычно собираются заранее путем опроса специалиста в данной предметной области и составляют ядро базы знаний, Они используются также в блоке рассуждения системы, позволяя выводить следствия. Процедурные знания дают возможность при необходимости генерировать и факты. Таким образом фактические и процедурные знания тесно переплетаются. Кроме того, в ходе работы системы приходится принимать решения, какие из таких правил следует использовать.

**3. Управляющие знания.** В системе должен быть предусмотрен некоторый набор стратегий, чтобы можно было рассматривать альтернативные возможности в ходе работы, переход при неудаче от одной стратегии к другой.

Системы продукции представляют собой конкретный метод организации программ в виде совокупностей трех групп, указанных выше. Каждая группа стоит особняком, поэтому такую систему легко понять и с ней легко работать.

(1) → База данных (БД).

(2) → Список порождающих правил (ПП).

(3) → Метод выбора, какое порождающее правило следует применять при данном состоянии базы данных.

Каждое порождающее правило (т.е. продукция) имеет форму ЕСЛИ (условие), ТО (действие) или, возможно, форму ЕСЛИ (условие), ТО (действие 1), В ПРОТИВНОМ СЛУЧАЕ (действие 2). При этом процедура сопоставления с образцом определяет, является ли данное правило применимым вообще.

Условная часть продукции сопоставляется с имеющейся базой данных. Если процесс сопоставления заканчивается успешно, то всем переменным, входящим в образец, т.е. переменным из условной части некоторого порождающего правила, придаются определенные значения из БД, которые обеспечивают сопоставимость. В дальнейшем эти значения будут использоваться в части, соответствующей действию того же самого порождающего правила. Эта часть может быть произвольной процедурой, которая запускается только при условии успешного сопоставления и если данная продукция была выбрана управляющим механизмом. Все изменения могут быть осуществлены посредством использования двух элементарных функций — добавления и удаления, например (добавить элемент2 БД), (удалить элемент1 БД). В начале работы системы запускается некоторое простое правило, которое обеспечивает сбор информации о рассматриваемом случае и вносит ее в базу данных. В дальнейшем система действует по простому принципу: "выбери продукцию p — примени соответствующие действия". Централизованная база данных является средоточием всей активности системы, причем программист должен ввести в систему потенциально весьма сложные стратегии управления, которые позволят избежать столь естественной неупорядоченности в работе системы, поведут ее по верному пути к успешному решению поставленной задачи. Самый распространенный способ как "запуск" системы продукции, так и "управления" ею состоит в использовании механизмов, названных "демонами".

Существует три возможные разновидности демонов, а именно "если-добавлено", "если-удалено" и "если-необходимо". Демоны запускаются в работу путем добавления их в базу данных. Как правило, демоны вида "если-добавлено" будут использоваться для проведения прямых логических дедукций, осуществляемых автоматически, если стали известны некоторые дополнительные факты. Демоны вида "если-удалено" запускаются функцией удаления. Эти демоны представляют собой по смыслу полную противоположность демонам "если-добавлено", но работают в значительной степени точно таким же образом. Как правило, демоны "если-удалено" будут использоваться для удаления из БД тех явных дедукций, которые были сделаны демонами "если-добавлено", как только те вещи, которые вызвали эти дедукции, были удалены из базы данных (что необходимо для поддержания согласованности базы данных). Демоны "если-необходимо" используются для проведения дедукции в обратном направлении.

Большую часть базы данных составляют правила, которые запускаются путем сопоставления с соответствующими признаками, характеризующими данную задачу, и которые могут добавляться, модифицироваться или изыматься пользователем. Базу данных такого специального типа обычно называют базой знаний.

Для взаимодействия пользователя с экспертной системой существует три режима, в отличие от единственного режима взаимодействия (получение ответов на поставленные задачи), типичного для более привычных вычислительных систем:

1. Получение решений поставленных задач -- пользователь как потребитель.
2. Усовершенствование или наращивание знаний системы -- пользователь как учитель.
3. Извлечение пользы для человека из базы знаний -- пользователь как ученик.

Не каждый специалист способен закодировать имеющиеся у него знания в виде явных правил, которые можно передать в компьютер. Более того, некоторые из них не дают даже себе отчета, каким именно правилам они следуют. Специалист по технике использования знаний ("инженер знаний") и является тем лицом, которое устраняет указанную трудность и кодирует знания эксперта-человека во множество правил для экспертной системы.

## 7.5. УЗКИЕ МЕСТА И "МИННЫЕ ПОЛЯ"

Главным узким местом, сдерживающим сегодня повсеместное распространение экспертных систем, является процесс "извлечения" знаний специалиста, осуществляемый инженером знаний. В настоящее время это происходит каждый раз по-разному, по своей сущности -- это процесс типа метода "проб и ошибок". Тем не менее стали вырисовываться методологические подходы к этой проблеме, а на горизонте уже виднеется автоматическая индукция знаний.

Несмотря на наличие узкого места извлечения знаний, один простой алгоритм, основанный на методе "проб и ошибок", был с успехом применен

в нескольких реализациях. Он может быть охарактеризован наличием следующих трех этапов:

1. Структурирование предметной области путем построения хорошей модели процесса решения задач, осуществляемого специалистом-человеком.
2. Создание работающей модели-прототипа экспертной системы, с пониманием того, что сначала она не будет хорошей, но что исправить конкретные ошибки будет легче и таким путем можно больше узнать о характере решаемой задачи.
3. Следование бесконечному циклу из "проверки", "отладки" и "усовершенствования", пока либо система не станет работать удовлетворительно, либо понимание задачи станет настолько глубоким, что можно будет приступить к созданию нового варианта.

Такая методология оказалась популярной среди некоторых исследователей (более подробные сведения можно найти в работе [18]) в силу ряда причин:

Специалисты находят, что легче критиковать работающую систему, чем сформулировать, что им необходимо.

Инженеры находят, что легче вносить оговорки в уже существующие правила, например "если А, то В, если только не С" (этого делать не следует, потому что. . .).

Наличие прототипа не дает остыть интересу со стороны людей-специалистов.

В общем случае полезно заставить систему работать как можно скорее.

Кроме того, на пути тех, кто только приступает к занятиям экспертными системами, возникают трудности, вызванные необходимостью сделать выбор средств, доступных конструктору таких систем [4]. В настоящее время доступны следующие средства, которые мы расположим в порядке убывания содержащихся в них заранее скомпонованных элементов экспертных систем:

Оболочки экспертных систем (такие как EMYCIN, SAGE, REVEAL, "Микроэксперт" и т. д.).

Специальные программные среды (типа POPLOG и т. п.).

Символьные языки (Пролог, Лисп).

Алгоритмические языки (Си, Фортран, Бейсик).

Поскольку имеется не так уже много работающих экспертных систем, то большинству приходится самим разрабатывать экспертные системы, если возникает в них необходимость.

1. Оболочки экспертных систем (ОЭС) оказываются весьма удобными, если они пригодны, но во многих случаях они оказываются совершенно бесполезными, являясь лишь отправными вариантами, на которых можно многому научиться. Обычно ОЭС строились в результате анализа уже существующей системы, такой как MYCIN, и создания на этой основе оболочки типа Empty-MYCIN. При этом решения возникающих задач оказываются ограниченными конструктивными подходами, в наибольшей степени соответствующими классам задач, на которые ориентировались первоначальные варианты экспертных систем.
2. Специальные программные среды, такие как POPLOG, дают меньше готовых решений, но предлагают широкий выбор средств [16]. Например,

в системе POPLOG не только предусмотрены символьные языки Пролог, POP-II в рамках одной программной среды, но имеется также экранный редактор, ориентированный на решаемую задачу, а также ряд частично готовых модулей, собранных в библиотеке. Такие системы, однако, доступны только на крупных вычислительных машинах и являются довольно дорогими.

3. Наиболее разумно для ознакомления с тематикой использовать такие языки, как Лисп и Пролог, если ставится лишь ознакомительная цель и есть возможность заняться изучением языков и программированием. Однако этот путь не обеспечивает того высокого соотношения между затратами и результатами, которое достигается в случае 2, а с другой стороны, не достигается столь эффективной реализации системы, как в случае 4, который рассматривается ниже.
4. Несмотря на упорное намерение некоторых лиц представить дело иначе, надо сказать, что алгоритмические языки, такие как Си или Бейсик, также могут быть использованы для построения экспертных систем. Для архитектур, ориентированных на системы продукции, эти языки могут выступать в качестве языков реализации. При этом конструктор должен быть уверен в принципиальной пригодности "машин логического вывода" и систем продукции, чтобы составлять программы с заданными характеристиками, типичными для экспертных систем. Использование таких языков позволяет создать наиболее эффективные реализации систем, возможные на заданной вычислительной машине.

Наконец, следует очертить предметную область и убедиться, что техника экспертных систем и искусственного интеллекта для нее годится. В настоящее время "хорошей" считается любая диагностическая задача, связанная со знаниями из узкой предметной области, тогда как всякая прикладная область, связанная с творческим рассуждением или рассуждением здравого смысла, относящаяся к широкой предметной области, является "плохой". Вот примеры случаев, в которых применение этих систем вызывает затруднения:

Мнения экспертов часто расходятся.

Используются сложные стратегии рассуждений.

Знания включают пространственные и временные соотношения.

Людям требуется продолжительное время для решения возникающих задач.

Имеется слишком много объектов и чересчур многое зависит от понятий, опирающихся на обычный здравый смысл.

## 7.6. ДРУГИЕ ОБЛАСТИ ПРИЛОЖЕНИЙ

Имеется целый ряд областей применения экспертных систем, которые обычно не указываются в числе приложений, дающих быструю коммерческую отдачу, но которые в будущем сулят большие перспективы. Первой из них является область, где экспертная система обеспечивает работу с некоторым другим пакетом программ. Такие системы часто называют

также "интеллектуальным интерфейсом". Так, пользователь сложного статистического пакета, взаимодействуя с экспертной системой, может получить совет о том, как пользоваться системой. Такая экспертная система может также оказаться интеллектуальным посредником между подобным вычислительным пакетом и неопытным пользователем.

Существует большое число других способов применения экспертных систем в качестве некоторой добавки к существующему пакету программ (например, реляционной базе данных), которые позволяют выводить новую информацию, выделяя в поступающих данных типичные формы отношений.

Наконец, весьма перспективным является использование экспертных систем для создания обучающих компьютерных систем, действующих на новых принципах. Одним из новых способов обучения таким предметам, как физика или медицина, явилось бы создание интеллектуальной программы, которая вела бы себя как опытный физик или медик [13]. Обучающийся при этом получил бы возможность следить за теми знаниями и теми рассуждениями, которые имеются в программе, обучаясь в среде, которая разительно отличается от той, которая была доступна ему прежде. Будущие врачи могли бы просто следить за тем, как решает свои задачи система MYCIN. Эта возможность обусловлена тем, что система MYCIN способна пояснять на естественном языке, что она делает, процесс принятия решений в этой системе подобен тому, который следует усвоить учащемуся, а медицинская информация в системе MYCIN представлена в форме, понятной человеку. На самом деле роль системы MYCIN, которую она сыграла в качестве системы обучения, была наиболее важной, но осталась в тени из-за ярких свойств, проявленных системой в качестве медицинского консультанта.

Другое, несколько отличное применение экспертных систем состоит в хранении знаний о некоторой предметной области, которыми располагает преподаватель, в виде интеллектуальных обучающих систем [15]. До настоящего времени пакеты обучающих программ, основанные на использование компьютера, в основном реализовали принцип "учи и практикуйся". Однако, опираясь на методологию экспертных систем, мы можем создать диагностические системы, которые были бы в состоянии строить модель учащегося, так же как учитель-человек. Следующие ошибочные вычитание и сложение:

$$\begin{array}{r} 170 \\ - 93 \\ \hline 187 \end{array} \quad \begin{array}{r} 33 \\ + 179 \\ \hline 102 \end{array}$$

не закончатся тем, что на экране будет напечатано: "Неверно, вы что-то упустили". Интеллектуальная обучающая система в состоянии точно обнаружить, что именно забыл ученик или чем он не умеет пользоваться. Используя методы экспертных систем, мы разработали весьма сложные пакеты обучения, которые могут работать даже на микрокомпьютерах [2]. В области образования открываются весьма большие возможности.

## 7.7. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОЛОГ В КАЧЕСТВЕ СИСТЕМЫ ПРОДУКЦИЙ

Для того чтобы проиллюстрировать мысли, представленные в настоящей главе, мы воспользуемся языком Пролог для разработки одной игрушечной экспертной системы [7]. Заинтересованный читатель может обратиться также к работе [6], где показано, что язык Пролог является удобным средством разработки экспертной системы. В этой работе авторы не только идут существенно дальше нашего игрушечного примера, но и до конца остаются верными свойственному языку Пролог "логическому" взгляду на мир. Здесь же мы воспользуемся языком Пролог, считая его, как и в работе [17], системой продукций:

1. Пролог дает нам базу данных, в которой можно расположить правило продукции в виде:

заключение ЕСЛИ	предусловие1
И	предусловие2

В "логической" интерпретации языка Пролог это означает:

Чтобы доказать заключение
доказать предусловие1 и
доказать предусловие2

Для этого в языке Пролог используется весьма простой синтаксис, в котором ":-" обозначает ЕСЛИ, а "," И.

2. Пролог позволяет хранить также и факты в форме правил, показанных выше, но у которых нет предусловий.
3. Управляющий механизм, обеспечиваемый системой Пролог, известен как поиск в глубину или рассуждение в обратном направлении. Он аналогичен также демонам "если-необходимо", описанным ранее. Процесс управления также зависит и от порядка, в котором в базе данных расположены факты и правила. Однако в системе Пролог не требуется какой-либо нумерации правил или фактов в базе данных. (Ниже мы нумеруем строки для удобства описания основных идей.)

Игрушечная экспертная система, предназначенная для различения животных (как в работе [18]) и записанная на языке Пролог с использованием правил, показанных в структуре на рис. 7.2, имеет следующий вид:

- |       |                                   |
|-------|-----------------------------------|
| 1.1.0 | это – (обезьяна) :-               |
| 1.1.1 | это – (млекопитающее),            |
| 2     | это – (хищник),                   |
| 3     | имеет (рыжевато-коричневый-цвет), |
| 4     | имеет (темные-пятна).             |
| 1.2.0 | это – (тигр) :-                   |
| 1.2.1 | это – (млекопитающее),            |
| 2     | это – (хищник),                   |
| 3     | имеет (рыжевато-коричневый-цвет), |
| 4     | имеет (черные-полосы).            |
| 1.3.0 | это – (жираф) :-                  |

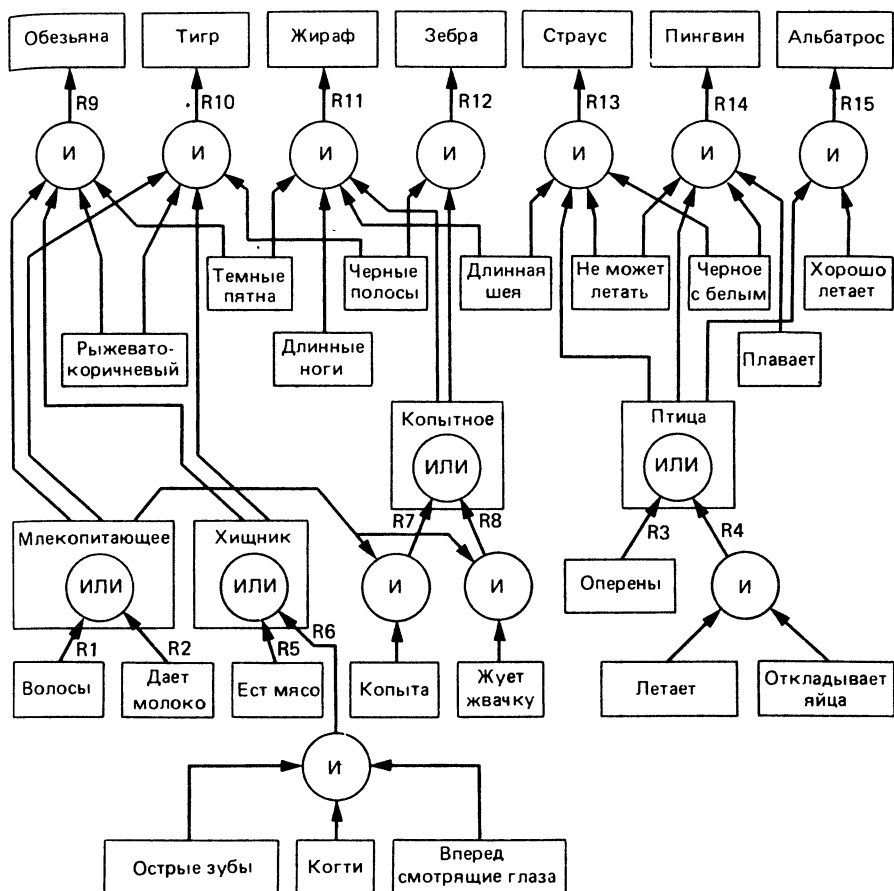


Рис. 7.2. Пример сети для простой системы, основанной на правилах. В прямоугольниках размещаются утверждения, кружочки представляют способы их комбинирования, а метки R1, R2 и т. д. указывают номер правил

- 1.3.1 это – (копытное),
- 2 имеет (длинную-шею),
- 3 имеет (длинные-ноги),
- 4 имеет (темные-пятна).
- 1.4.0 это – (зебра) :–
- 1.4.1 это – (копытное),
- 2 имеет (черные-полосы).
- 1.5.0 это – (страус) :–
- 1.5.1 это – (птица),
- 2 (не-летает),



- 3 имеет (длинную-шею),
- 4 имеет (длинные-ноги),
- 5 имеет черно-белый-цвет.
- 1.6.0 это – (пингвин) :–
- 1.6.1 это – (птица),
- 2 (не-летает),
- 3 (плавает),
- 4 имеет черно-белый-цвет.
- 1.7.0 это – (альбатрос) :–
- 1.7.1 это – (птица),
- 2 (летает-хорошо)

Нам необходимо также следующее:

- 2.1.0 это – (млекопитающее) :–
- 2.1.1 имеет (волосы).
- 2.2.0 это – (млекопитающее) :–
- 2.2.1 (дает-молоко).
- 2.3.0 это – (птица) :–
- 2.3.1 имеет (перья).
- 2.4.0 это – (птица) :–
- 2.4.1 (летает),
- 2 (откладывает-яйца).
- 2.5.0 это – (хищник) :–
- 2.5.1 (ест-мясо).
- 2.6.0 это – (хищник) :–
- 2.6.1 имеет (острые-зубы),
- 2 имеет (когти),
- 3 имеет (глаза-смотрящие-вперед).
- 2.7.0 это – (копытное) :–
- 2.7.1 это – (млекопитающее),
- 2 имеет (копыта).
- 2.8.0 это – (копытное) :–
- 2.8.1 это – (млекопитающее),
- 2 (жует-жвачку).

Как мы отмечали, в базе данных, основанной на системах продукций, вначале нет никаких фактов о текущей ситуации. Поэтому необходимо включить простой подраздел в нашу программу, позволяющий в ходе беседы получить информацию от пользователя:

- 3.1.0 имеет (Y) :–
- 3.1.1 написать ('имеет ли оно'),
- 2 написать (Y), pl,
- 3 прочесть (Ответ),
- 4 положительный (Ответ),
- 5 внести (Имеет (Y)).
- 4.1.0 (Y) :–
- 4.1.1 написать ('умеет ли оно'),
- 2 написать (Y), pl,
- 3 прочесть (Ответ),
- 4 положительный (Ответ),

- 5                    внести ((Y)).
- 5.1.0            положительный (да)

Наконец, нам нужно определение всей задачи в целом:

- 6.1.0            запуск:—
- 6.1.1            это — (Y),
- 2                написать ('я думаю, что это'),
- 3                написать (Y), pl, pl.
- 6.2.0            запуск:—
- написать ('я не знаю такое животное'), pl.

Все, что теперь нужно сделать пользователю, — это набрать на терминале команду **запуск**, а остальное сделает сама система.

## 7.8. КАК ПРОЛОГ ОБРАБАТЫВАЕТ ОПИСАННУЮ ПРОГРАММУ

В силу механизма управления, присущего системе Пролог, результатом такого действия будет поиск в базе данных смысла команды **запуск**. Он содержится в строке 6.1.0. Для того чтобы выполнить такую команду пользователя, система Пролог должна доказать то, что появляется в строках 6.1.1 — 6.1.3. Система начинает с первой из них. Это приводит к другому поиску в базе данных, начинающемуся со строки 1.1.0, где Y принимает значение "объезьяна", и управление передается строке 1.1.1, чтобы доказать "это — млекопитающее". Другой поиск в базе данных ведет нас к строке 2.1.0, причем на этот раз значением Y становится "млекопитающее", что и является предметом поиска в настоящий момент. У элемента 2.1.0 всего лишь одно предусловие "имеет (волосы)". Следующий этап поиска в базе данных ведет к строке 3.1.0. Предусловие 3.1.1 удовлетворяется посредством встроенного оператора, что приводит к тому, что на терминале пользователя появляются слова "имеет ли оно". Предусловие 3.1.2 ведет к тому, что там же печатается слово "волосы" и курсор переходит на новую строку. Предусловие 3.1.3 также удовлетворяется с помощью встроенного в систему оператора, который читает то, что печатает на терминале пользователь, и подставляет это в качестве значения переменной "Ответ". (Заметим, что переменные величины в языке Пролог всегда начинаются с прописной буквы, а постоянные — со строчной, значения же для переменных устанавливаются лишь в пределах действия одного правила.)

Теперь у пользователя на терминале появился первый вопрос: "имеет ли оно волосы". Давайте для начала ответим согласием и скажем "да". Благодаря этому переменная "Ответ" получит значение "да". Предусловие 3.1.4 снова приводит к поиску в базе данных, пока не будет обнаружено 5.1.0, которое безусловно выполняется, и никаких других предусловий не возникает. В предусловии 3.1.5 используется другой встроенный оператор, который вносит новый факт в базу данных, размещая его перед другими подобными фактами. В нашем случае "имеет (волосы)" будет добавлено непосредственно перед строкой 5.1.0. Система таким образом установила первый факт, касающийся обсуждаемого животного.

Теперь можно вернуться назад и посмотреть, какие предусловия на более высоком уровне остались непроверенными. Управляющий механизм системы Пролог поместил все такие невыполненные предусловия в стек, который обрабатывается по схеме "последним вошел — первым вышел", что приводит нас к удовлетворению 2.1.0, откуда мы возвращаемся назад к 1.1.1. Теперь мы готовы проверить 1.1.2.

Предусловие 1.1.2 ведет к 2.5.0 и такой же цепочке шагов, которые мы проделали выше, ведущей к тому, что на экране терминала будет напечатано "ест ли оно мясо". Предположим, что на этот раз мы ответили отрицательно, сказав "нет". Это приведет к поиску в базе данных "положительный (нет)". В базе данных не найдется такого, что системой Пролог рассматривается как неудача, так что 4.1.0 также будет считаться неудачей вследствие того, что неудачей закончилось доказательство одного из его предусловий. Следовательно, и предусловие 2.5.0 заканчивается неудачей.

Следует заметить, однако, что когда некоторое предусловие заканчивается неудачей и происходит "возвращение назад" к тем предусловиям, которые его вызывали, то при наличии других альтернатив они также будут подвергнуты проверке. Следовательно, в нашем случае будет проверяться 2.6.0, что приведет к другому вопросу "имеет ли оно острые зубы".

Предположим, что мы снова говорим "нет". Это приведет к неудаче 1.1.2. Пролог попытается проверить еще одну альтернативу для 1.2.1, прежде чем отказаться от 1.2.0. Новый вариант 1.2.1 приведет к 2.2.0, что, в свою очередь, приведет к вопросу "дает ли оно молоко".

Пусть мы сказали "да", что привело к успеху 1.2.1. Это означает, что мы готовы двигаться дальше. Система Пролог реализует такое соображение: "Может быть, к настоящему моменту ситуация изменилась, так что можно ожидать успеха". Проверка 1.2.2 приведет к тому, что опять появится вопрос, который мы уже однажды задавали: "ест ли оно мясо".

Это происходит потому, что наша программа довольно проста и не запоминает отрицательные факты, так как она запоминает положительные. В приложении А содержится программа, лишенная этого и некоторых других недостатков настоящей программы. Тем не менее попробуем эту альтернативу и, чтобы не противоречить самим себе, скажем снова "нет". Это приводит к неудаче для 1.2.2. Поскольку теперь все альтернативы для 1.2.1 опробованы, то и 1.1.0 терпит неудачу. Таким образом, все, что нам удалось сделать, — это доказать, что обсуждаемое животное не является обезьяной, и при этом мы узнали, что животное имеет волосы!

Вы уже догадались, что теперь мы попытаемся доказать, что это тигр и т.д. в том порядке, в котором заданы правила, пока не будет достигнут успех, зависящий от того, как будут чередоваться положительные и отрицательные ответы пользователя.

## 7.9. ДАЛЬНЕЙШИЕ ОБОБЩЕНИЯ

В этой главе мы представили краткий обзор вопросов, связанных с построением экспертных систем. Чтобы проиллюстрировать способ

достижения необходимого решения, мы использовали задачи различения животных. Предложенная для этого "программа" на языке Пролог достаточно простая для наших иллюстративных целей, далека от совершенства. Если вы попытаетесь обратиться к этой программе при широком наборе данных, то столкнетесь с тем, что она будет продолжать задавать вопросы, на которые вы уже давали ответы. Одна из причин этого состоит в том, что программа не запоминает сообщенные ей факты отрицательного характера.

В приложении А дается усовершенствованная программа, снимающая часть затруднений благодаря применению более сложных структур для представления фактов, а также использованию встроенного нелогического оператора *cut* языка Пролог, называемого "отсечение", который обеспечивает *подрезание* дерева поиска, что приводит к более эффективному поиску путей решения.

Использованная здесь программа, однако, содержит лишь одну из характеристик экспертных систем вообще, а именно "вопросы, порождаемые в процессе вывода". Это означает, что система не запрашивает информацию случайным образом, а запрашивает лишь то, что ей необходимо для осуществления процесса рассуждения. На самом деле некоторое дальнейшее обобщение позволяет пользователю задать вопрос: "Почему сделан некоторый запрос?" На это система будет в состоянии представить рассматриваемое ею правило и причину запроса.

Кларк и Маккейб (1982) опубликовали работу учебного характера [6], касающуюся характеристик типа "объяснение поведения", которые могут быть добавлены к системе, если использовать Пролог как базовую конструкцию. Харди также несколько расширил синтаксис языка Пролог [9], чтобы обеспечить успешную работу в более сложном примере, в котором правила записываются более естественным образом:

#### *Правило 100*

ЕСЛИ свойство — волосатый

ТО группа — млекопитающие ку 800

В этом примере ку обозначает коэффициент уверенности, благодаря которому в систему вносится в определенной степени вероятностное рассуждение.

Синтаксис языка Пролог на самом деле легко поддается расширению путем определения новых операторов, так что правила, представленные в вышеуказанном виде, трактуются системой, как и правила в форме, принятой в нашем примере об узнавании животного:

:- оп (975, xfy, :),  
оп (725, xfy, ку),  
оп (948, xfy, потому что),  
оп (800, xfy, и),  
оп (750, xfy, или),  
оп (959, fx, если),  
оп (949, xfy, тогда).

Пользуясь такой системой, мы можем пометить правило, которое используем, так что в дальнейшем сможем задать конкретный вопрос. Вызов правила тогда производился бы так:

вызвать (П) :—

П : если Р, то С к<sub>у</sub> N

( $N < 200$ ; заметить (С имеет к<sub>у</sub>, равное N, потому что [П])).

Отсюда, если коэффициент уверенности меньше 200, это правило игнорируется, в противном случае (""; " обозначает логический оператор ИЛИ) мы отмечаем заключение и номер правила, которое к нему привело. Тогда предикат "почему" проанализирует базу данных с целью установить, каков номер текущего отмеченного правила, а потом выдать его пользователю в качестве оправдания заданного тому вопроса:

почему:—

написать ("Ваш ответ поможет мне в работе с правилом"),  
текущее (ПРАВИЛО),  
показать (ПРАВИЛО).

Если мы будем продолжать добавлять в нашу систему новые свойства таким образом, то в конце концов мы придем к чему-то очень похожему на оболочку для экспертной системы, но на этот раз полученную путем обобщения системы Пролог. Однако, может быть, было бы лучше сразу воспользоваться какой-то из оболочек для экспертных систем. Кроме того, что выбор адекватного средства будет различным для каждого приложения, следует отметить, что вопрос о выборе средства выходит за рамки нашей вводной главы. Однако во многих случаях Пролог может оказаться хорошей основой при разработке дешевого прототипа системы до перехода к созданию эффективной системы.

Ниже мы приводим неполный перечень характеристик, которые должны быть привнесены в экспертную систему либо путем последовательного добавления (как это делалось в настоящей главе), либо с самого начала:

1. Запрос данных, порождаемый ходом вывода.
2. Отчетность (объяснение поведения).
3. Прозрачность (обучающая компонента).
4. Автоматическое пополнение совокупности правил.
5. Серьезное обучение.
6. Вероятностное рассуждение (неуверенность в фактах, правилах и управлении).

## ПРИЛОЖЕНИЕ А

Улучшенный вариант программы классификации животных

это — (обезьяна) :—

это — (млекопитающее),

это — (хищник),

положительный (имеет, рыжевато-коричневый-цвет).

положительный (имеет, темные-пятна).

это – (тигр) :–  
     это – (млекопитающее),  
     это – (хищник),  
         положительный (имеет, рыжевато-коричневый-цвет),  
         положительный (имеет, черные-полосы).  
 это – (жираф) :–  
     это – (копытное),  
         положительный (имеет, длинную-шею),  
         положительный (имеет, длинные-ноги),  
         положительный (имеет, темные-пятна).  
 это – (зебра) :–  
     это – (копытное),  
         положительный (имеет, черные-полосы).  
 это – (страус) :–  
     это – (птица),  
         положительный (имеет, длинную-шею),  
         положительный (имеет, длинные-ноги),  
         положительный (имеет, черно-белый-цвет).  
 это – (пингвин) :–  
     это – (птица),  
         отрицательный (умеет, летать),  
         положительный (умеет, плавать),  
         положительный (имеет, черно-белый-цвет).  
 это – (альбатрос) :–  
     это – (птица),  
         положительный (умеет, летать-хорошо).  
 это – (млекопитающее) :–  
         положительный (имеет, волосы).  
 это – (млекопитающее) :–  
         положительный (дает-молоко).  
 это – (птица) :–  
         положительный (имеет, оперенье).  
 это – (птица) :–  
         положительный (умеет, летать),  
         положительный (откладывает-яйца);  
 это – (хищник) :–  
         положительный (умеет, ест-мясо).  
 это – (хищник) :–  
         положительный (имеет, острые-зубы),  
         положительный (имеет, вперед-смотрящие глаза),  
 это – (копытное) :–  
     это – (млекопитающее),  
         положительный (имеет, копыта).  
 это – (копытное) :–  
     это – (млекопитающее),  
         положительный (умеет, жевать).  
 положительный (X, Y) :–  
     спросить (X, Y, Ответ),  
     да-проверка (Ответ).

отрицательный (X, Y) :–

спросить (X, Y, Ответ),

нет-проверка (Ответ).

да-проверка (да).

нет-проверка (нет).

спросить (X, Y, Ответ) :–

написать (X),

написать (это),

написать (Y),

прочесть (Ответ),

запомнить (Ответ, X, Y).

запомнить (да, X, Y) :–

добавить (положительный (X, Y)).

запомнить (нет, X, Y) :–

добавить (отрицательный (X, Y)).

добавить (X) :–

запуск:- внести (X:!).

это- (Y),

написать ('я думаю что это-'),

написать (Y), nl, nl.

## ПРИЛОЖЕНИЕ Б

### Упражнение для читателей (составлено Л. Фордом)

Напишите программу для экспертной системы, имеющей следующие правила:

- 1 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 9 ТО это – "бейсбол"
- 2 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 13 ТО это – "клубный регби"
- 3 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 15, мяч разрешается посылать вперед, ТО это – "шотландский футбол"
- 4 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 15, мяч НЕ разрешается посылать вперед, ТО это "регби"
- 5 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 11, бита НЕ используется, ТО это – "хоккей"
- 6 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 11, используется бита, ТО это – "крикет"
- 7 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 11, мяч круглый, мяч НЕ твердый, ТО это – "европейский футбол"
- 8 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 11, мяч НЕ круглый, ТО это – "американский футбол"
- 9 ЕСЛИ игра в мяч, две команды с числом игроков в команде более 2, число игроков в команде = 5, ТО это – "баскетбол"
- 10 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, мяч сплошной, НЕ играется в помещении, мяч круглый, НЕ используется деревянный молоток, ТО это – "гольф"
- 11 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, используется ракетка, ТО это – "настольный теннис"
- 12 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, НЕ играется в помещении, используется деревянный молоток, ТО это – "крокет"

- 13 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, НЕ играет-ся в помещении, ТО это – "теннис"
- 14 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, играет-ся в помещении, используется ракетка, мяч НЕ твердый, ТО это – "сквош"
- 15 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, мяч твер-дый, НЕ круглый, ТО это – "кегли"
- 16 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, мяч сплош-ной, круглый, играет-ся в помещении, ТО это – "бильярд"
- 17 ЕСЛИ игра в мяч, НЕ две команды с числом игроков в команде более 2, мяч НЕ твердый, бита НЕ используется, ракетка НЕ используется, ТО это – "пелота"\*
- 18 ЕСЛИ НЕ игра в мяч, используется ракетка, ТО это – "бадминтон"
- 19 ЕСЛИ НЕ игра в мяч, НЕ играет-ся сидя, играет-ся в помещении, ракетка НЕ исполь-зуется, то это "стрелки"\*\*\*
- 20 ЕСЛИ НЕ игра в мяч, играет-ся сидя, используются карты, доска НЕ используется, ТО это – "бридж"
- 21 ЕСЛИ НЕ игра в мяч, играет-ся сидя, карты НЕ используются и используется доска, ТО это – "шашки, шахматы или что-то в этом роде"

## Глава 8

### КАК МЫ СТРОИМ СИСТЕМУ "МИКРОЭКСПЕРТ"

*Фил Кокс*

Как-то в 1981 г. мы небольшой группой решили разработать и реализовать оболочку экспертной системы (ОЭС). В этой главе я хочу остано-виться на том опыте, который мы приобрели в процессе работы. Я по-пытаюсь объяснить те технические решения, которые были нами выбраны, и привести доводы в пользу выбранных вариантов.

Я надеюсь, что такая повествовательная форма позволит нам рассмот-реть некоторые проблемы, возникающие в ходе проектирования ОЭС, а также даст потенциальным пользователям ощущение того, что может быть сделано, а что нет, если воспользоваться теми ОЭС, которые существуют в настоящее время.

Разумеется, когда мы реализовали нашу ОЭС в виде программы и какое-то время пользовались ею, то быстро в ней разочаровались и подумали, что в следующий раз сделаем что-то получше. Потребовалось много времени, прежде чем мы сумели уяснить свои представления о том, какой должна быть следующая версия системы, но в конце концов она была написана и находится на заключительной стадии испытания. Однако здесь мы в ос-новном остановимся на системе "Микроэксперт". Нам необходимо получить достаточно опыта в использовании системы "Эксперт II", прежде чем мы бу-дем в состоянии беспристрастно оценить достоинства нашего нового варианта

---

\* Испано-американская игра в мяч. – *Прим. ред.*

\*\* Старинная английская игра, состоящая в метании небольших стрелок в размечен-ную мишень в форме круга. – *Прим. ред.*



решения, но я полагаю, что "Эксперт II" — это дальнейшее развитие того направления, по которому шло создание системы "Микроэксперт".

Прежде всего нам пришлось принять некоторые решения, касающиеся размеров системы. Следует ли нам ориентироваться на создание большой системы, рассчитанной на использование большой универсальной вычислительной машины, или же на что-то, что будет работать на микропроцессоре? Мы решили создать систему на основе микропроцессора. Наш выбор диктовался несколькими причинами, первой из которых была доступность вычислительных ресурсов. У нас было несколько микропроцессоров, но не было универсальной вычислительной машины, которая была слишком дорогой. Второй причиной был тот тип приложений, на который мы ориентировались. Во многих приложениях, где требуется дать определенный совет, несомненным достоинством является использование машины, которую можно положить в багажник автомашины. Не все экспертные системы являются интерактивными, т. е. требуют диалога с пользователем, но подавляющее их большинство работает все же в интерактивном режиме, и часто очень важно бывает иметь возможность использовать систему непосредственно на месте. Третьей причиной была наша подготовка в качестве программистов, так сказать представление нашей группы о самой себе. Мы считаем себя инженерами-практиками, а не учеными-теоретиками. Мы чувствуем свою силу в способности написать хорошо продуманные программы, требующие минимума ресурсов, а не в создании чего-то концептуально совершенно нового.

Если посмотреть на те экспертные системы, которые уже работают или находятся в стадии разработки, то поражает как разнообразие тех приложений, на которые системы ориентируются, так и разнообразие общих подходов к созданию систем. Отсюда нашей следующей задачей явился выбор типа системы. Мы решили создать систему, которая относится к разряду языков-советчиков. Такая система наполняется правилами, объясняющими, как выполнять некоторые конкретные функции, специалистом по знаниям — инженером знаний. На этапе применения пользователь, имеющий некоторую проблему, обращается к этой системе, а последняя пытается дать ему совет. Весь процесс аналогичен тому, как ведет себя врач, когда вы входите к нему в кабинет: он задает вам вопросы, касающиеся ваших симптомов, и на основании полученных ответов пытается поставить диагноз.

Я думаю, что на тот момент у нас и не было слишком большого выбора, так как мы уже приняли решение создать систему, не ориентированную ни на какое конкретное приложение. По-видимому, язык советов — это тот тип экспертных систем, который способен охватить самый широкий спектр приложений.

Затем нам пришлось заняться выбором языка программирования для нашей системы. Первая часть этого выбора приходилась на границу между языками искусственного интеллекта, такими как Лисп или Пролог, и обычными алгоритмическими языками. Хотя на более крупных машинах универсального пользования существуют эффективные реализации как языка Лисп, так и языка Пролог, на малых вычислительных машинах они оказываются

чересчур расточительными в отношении использования и памяти, и процессорного времени. Сила языков искусственного интеллекта заключена и в их гибкости, что делает их чрезвычайно удобными при экспериментальном программировании, и в том, что они существенно сокращают объем работ, необходимых при создании большой сложной программы. Мы, однако, намеревались создать систему, которая работала бы возможно эффективнее на небольшом микропроцессоре, так что мы с определенным сожалением решили остановиться на алгоритмическом языке. Было решено использовать систему UCSD PASCAL: PASCAL потому, что мы нуждались в языке, обеспечивающем рекурсивные вычисления и имеющем неплохую систему для работы с неупорядоченным массивом, а UCSD PASCAL потому, что этот язык имеется у весьма широкого спектра вычислительных машин, а по коммерческим соображениям мы хотели бы достигнуть максимально возможной машинной независимости системы.

На язык советов можно посмотреть как на частный случай некоторого специализированного языка программирования. Язык советов — это язык, на котором вы создаете экспертную систему в значительной степени аналогично тому, как на языке Симула вы бы стали писать программы для моделирования. При его разработке многие элементы выбираются точно так же, как и при разработке любого другого языка программирования. Большинство языков искусственного интеллекта являются инкрементными, что означает возможность запускать части программы в работу сразу же после их написания, как это делается, например, в языке Бейсик. Большинство первых экспертных программ также были инкрементными: правила в модели можно было добавлять в любое время. Мы решили, однако, пойти своим путем и создать систему, у которой будет отдельный компилятор правил, так что если вы хотите изменить свою модель, то сначала вам придется отредактировать файл-источник, содержащий правила, а затем заново скомпилировать вашу модель точно так же, как вам пришлось поступить при использовании таких языков программирования, как Фортран или Алгол. Мы поступили так по трем причинам. Во-первых, мы полагали, что это позволяет при фиксированном объеме памяти машины строить более крупные модели. Во-вторых, мы предвидели случаи, когда было бы нецелесообразно давать эту модель пользователю и позволять ему изменять ее, когда ему захочется. В таком режиме компиляции можно было бы избежать этого, просто не дав ему файл-источник или не предоставив в его распоряжение компилятор. И наконец, нас беспокоили те затруднения, с которыми может столкнуться пользователь при отладке своих правил. Дело в том, что мы хотели создать язык, в котором типы данных использовались бы возможно полнее, а инкрементные языки по необходимости допускают весьма слабую проверку типов.

Разумеется, мы не в вакууме приступали к разработке системы такого вида. Конструктор всегда находится под влиянием предшествующих систем, созданных для тех же приложений другими авторами. Мы прочли все, что было доступно по предыдущим разработкам экспертных систем, и в

конце концов решили принять в качестве модели для нашей системы систему PROSPECTOR [1]. В частности, мы заимствовали из нее метод логического вывода с использованием правила Байеса, а также точку зрения авторов на то, что совокупности правил следует рассматривать как деревья, которые удобно изображать на рисунках.

У системы PROSPECTOR имеются два свойства, которые мы не реализовали. Первое заключается в том, что система в начале очередного сеанса с пользователем просит его назвать список найденных им минералов. Этот список сопоставляется с некоторой таблицей, дающей синонимы для распространенных минералов, вырабатываются стандартные имена, которые используются для ответа на некоторые вопросы, возникающие в модели, так что сеанс получает некоторое разумное начало. В то же время мы подумали, что это свойство слишком тесно связано с конкретным приложением и не допускает обобщения, пригодного для использования в нашей не зависящей от контекста оболочке. Однако сейчас я не уверен, что это было правильное решение. Весьма распространена ситуация, когда потребитель какой-то модели перед началом сеанса что-то ей хочет сообщить. У него должна быть какая-то проблема, иначе он не обратился бы к этой модели. Было бы целесообразно включить такое свойство в будущие системы, однако реализовать его будет непросто.

Вторым свойством системы PROSPECTOR, которое мы не стали использовать, является семантическая сеть. Семантическая сеть дает возможность составителю модели указать, что, например, гранит, периодит и базальт являются вулканическими породами. Если в ходе сеанса система узнает, что имеется базальт, то без дополнительных распросов она узнает, что имеются породы вулканического происхождения. С другой стороны, если модели было сообщено, что вулканических пород нет, то система автоматически узнает, что гранит, периодит и базальт отсутствуют. Подобного эффекта можно достигнуть, не включая такое свойство непосредственно в язык советов, и мы решили обойтись без него в нашей первой системе. Однако это свойство может значительно облегчить построение моделей, поэтому оно остается первым кандидатом на использование в оболочках, которые еще будут создаваться.

На этом этапе были приняты основные решения относительно системы: создать экспертную систему, воспользовавшись языком UCSD PASCAL, реализовать ее на микропроцессоре, создать отдельные компилятор с языка правил и рабочую систему и взять в качестве прототипа систему PROSPECTOR. Прежде чем рассказать, как эти решения были реализованы, я хотел бы остановиться на том, как эта система, будучи реализованной, выглядит с точки зрения пользователя.

Система "Микроэксперт" состоит из двух программ: компилятора с языка правил, названного EXPCOMP, и рабочей системы, названной RUNEXPT. При создании своей модели пользователь сначала пишет правила на языке правил системы "Микроэксперт". В дальнейшем я более подробно останавлиюсь на этом языке, а сейчас достаточно сказать, что это очень про-

той язык: каждый прямоугольник на представленных в этой главе рисунках соответствует одному утверждению в этом языке.

Правила вводятся в микропроцессор и заносятся на магнитный диск с использованием стандартного системного редактора системы UCSD, образуя тем самым исходный файл обычным образом. Этот исходный файл затем компилируется посредством компилятора EXPCOMP. Компилятор создает не один объектный файл, как это принято, а два выходных файла. Первый выходной файл — объектный содержит структуру модели. В отличие от большинства компиляторов, создающих выполнимый код, эта структура имеет вид таблиц, которые могут быть загружены в основную память машины и использоваться рабочей системой. Второй выходной файл, создаваемый компилятором, — файл сообщений содержит все строки текста из исходного файла, записанные в коде ASCII. В экспертных системах обычно содержится очень много текстовой информации, и было бы нецелесообразно хранить ее всю в основной памяти, так как это наложило бы серьезные ограничения на размеры моделей, которые можно было бы обрабатывать на микропроцессоре.

Компилятор создает также распечатку, которую можно выводить либо на экран терминала, либо на принтер. В этой распечатке прежде всего содержится первая копия исходного файла, строки которого нумеруются, за которой идут сообщения об обнаруженных ошибках. Затем приводятся список всех переменных в программе и отдельный список целевых переменных с необходимыми перекрестными ссылками.

Компилятор проверяет типы переменных, например проверяет, чтобы к вероятностям не прибавлялись никакие числа. Он убеждается в том, что все утверждения могут быть достигнуты по крайней мере из одного целевого утверждения и что концевые вершины всех деревьев представляют собой вопросы. Затем выполняется проверка на отсутствие в модели циклических параметров. Всякая модель, которая компилируется без ошибок, будет работать и не "зависнет", т. е. большинство ошибок, которые возможны при обычном программировании, не смогут пройти через компилятор. Это не значит, однако, что мы гарантируем корректность модели. Логика того, что было написано пользователем, может оказаться тем не менее неверной.

Я не хочу входить в подробности внутренних операций нашего компилятора. Это весьма обычный компилятор, предназначенный для весьма небольшого и синтаксически очень простого языка. Для тех, кого интересует этот вопрос, отмечу, что синтаксис полностью определяется контекстом и что разбор фразы компилятором производится по методу, известному как рекурсивный спуск. Во многих книгах по языку Паскаль можно найти примеры компиляторов такого сорта.

После того как пользователь получает скомпилированную модель, наступает этап ее использования. Для этого пользователь запускает программу RUNEXPT, которая сначала запрашивает его о названии модели, которой он хочет воспользоваться. Когда такое имя задано, то программа читает

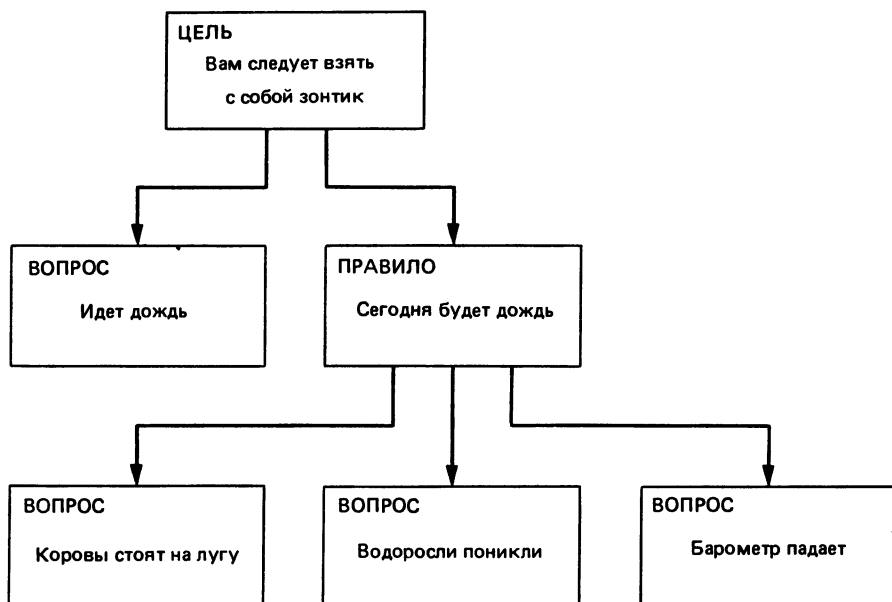


Рис. 8.1. Примеры базы правил

объектный файл для этой модели и начинается сеанс вопросов и ответов. Характер этого сеанса лучше всего объяснять, показав, как выглядит база правил, к чему мы и переходим.

На рис. 8.1 представлена структурная схема небольшой базы правил. В этом примере имеется лишь одна цель, заключенная в самый верхний прямоугольник. Целью служит: "Вам следует взять с собой зонт." Другими словами, цель модели — помочь пользователю решить, взять ли ему с собой зонт, для чего модель задает пользователю вопросы. От верхнего прямоугольника вниз отходят две стрелки к прямоугольникам, расположенным на втором уровне. Это — "Идет дождь" и "Сегодня будет дождь", то есть решение взять с собой зонт определенным образом зависит от того, идет ли дождь сейчас, или будет ли он идти сегодня. Заметьте, что на этом этапе мы не определили точно, какова эта зависимость, мы просто констатировали факт зависимости.

Если вы обратитесь к прямоугольнику на рис. 8.1 с пометкой: "Сегодня будет дождь", то увидите, что от него отходят три стрелки к трем другим прямоугольникам. В этих прямоугольниках содержатся слова: "Коровы стоят на лугу", "Водоросли поникли", "Барометр падает", так что истинность этого утверждения зависит каким-то образом от истинности последних трех высказываний.

У других четырех утверждений отсутствуют какие-либо выходящие из них стрелки. В модели нет никакого другого способа вывести истин-

ность или ложность этих утверждений, иначе как спросив пользователя. Следовательно, эти четыре утверждения представляют собой вопросы и помечены соответствующим образом на рис. 8.1. Утверждение: "Сегодня будет дождь" — не является ни целью, ни вопросом. Поэтому мы называем его правилом.

Итак, сделаем выводы: прямоугольники на наших рисунках представляют собой гипотезы модели и являются либо целями, либо правилами, либо вопросами. Сверху в дереве должны располагаться цели, в середине его могут также располагаться цели, если это необходимо пользователю. У цели имеется два атрибута. Первый — это то, что вы хотите попросить систему доказать, а второй — утверждение, которое будет выдано системой в качестве окончательного отчета. Концевые вершины деревьев обязательно должны представлять собой вопросы. Все другие прямоугольники называются правилами.

Мы уже говорили, что истинность или ложность каждого прямоугольника на рисунке (исключая прямоугольники-вопросы) некоторым образом выводится из истинности или ложности прямоугольников, на которые делается ссылка. Мы пока что ничего не сказали, как это происходит. В первых, следует отметить, что этот вывод можно производить несколькими способами, так что с каждым прямоугольником связывается некоторый оператор. Если в системе используется булева логика, то этими операторами служат булевы операторы И, ИЛИ и НЕ. Но в системе предусмотрен учет неопределенностей. Иными словами, вклад каждого фактора в некоторую гипотезу обычно не будет просто истинностью или ложью, а с ними будет связана некоторая вероятность.

На рис. 8.2 показана та же самая модель, что и на рис. 8.1, но развернутая так, чтобы отобразить операторы и типы вопросов, к обсуждению чего я сейчас перехожу.

Оператор НЕ реализовать проще всего. Если А есть не В, то вероятность А равна единице минус вероятность В:

ЕСЛИ  $A = \text{НЕ } B$

ТО  $\text{Pr}A = 1 - \text{Pr}B$ .

Имеется множество различных способов задания нечетких операторов типов И и ИЛИ. Однако всякий согласится, что независимо от них такие операторы в пределе должны вести себя в точности так же, как обычные булевы операторы (когда вероятности определяющих факторов стремятся к нулю или единице). При реализации системы "Микроэксперт" мы воспользовались определениями из системы PROSPECTOR. Оператор И определяется как минимум. Это означает, что в выражении: "А есть В и С и D" — при вероятностях А, В, С и D, обозначенных как  $\text{Pr}A$ ,  $\text{Pr}B$ ,  $\text{Pr}C$ ,  $\text{Pr}D$ ,  $\text{Pr}A$  является минимумом из величин  $\text{Pr}A$ ,  $\text{Pr}C$ ,  $\text{Pr}D$ .

Аналогично оператор ИЛИ определяется как максимум. В выражении: "А есть В ИЛИ С ИЛИ D" —  $\text{Pr}A$  является максимумом из величин  $\text{Pr}A$ ,  $\text{Pr}C$ ,  $\text{Pr}D$ .

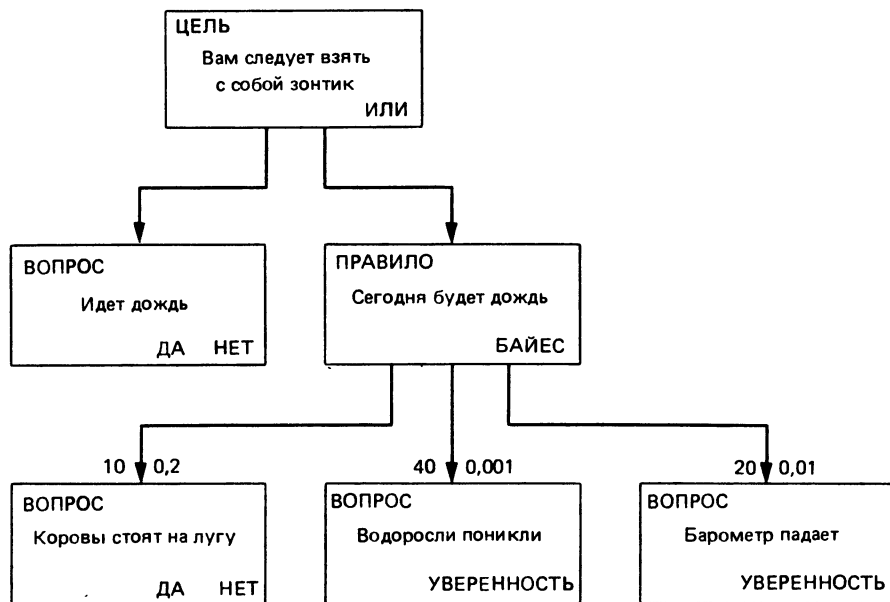


Рис. 8.2. Типы используемых операторов и вопросов

Теперь следует сказать, что функции И и ИЛИ могут быть реализованы многими способами, например функцию И можно было определять так, чтобы вероятности утверждений, вносящих в нее свой вклад, перемножались между собой. Отсутствует общее соглашение по этому вопросу. Однако, как уже отмечалось, эти операторы "на краях" должны себя вести как булевы операторы. Кроме того, представляется логичным, чтобы пара используемых факторов подчинялась теореме Муавра, утверждающей, что

$$A \text{ И } B = \text{НЕ} ((\text{НЕ } A) \text{ ИЛИ } (\text{НЕ } B))$$

Однако операции Max и Min этой теореме не подчиняются. Вы видите, что математические вопросы таких систем еще нуждаются в дальнейшем исследовании.

Кроме этих трех расплывчатых логических операций необходима еще одна. Мы хотим иметь возможность сказать что-то вроде: "А зависит от В и С, но В является более важным, чем С". Для такого рода операции мы прибегаем к так называемому байесовскому оператору. С математической точки зрения этот оператор основывается на теореме Байеса. Правило с байесовским оператором выглядит следующим образом:

*Правило свинка (байесовское) "У больного – свинка"*

Жар ЛД 10 ЛН 0,01

Сыпь ЛД 20 ЛН 0,001

Априорно 0,25

Правило гласит, что свинка может быть выведена на основании наличия сыпи и жара. Последняя строка в этом примере: "Априорно 0,25" — означает, что в 25% случаев, рассматриваемых этой моделью, заранее ожидается заболевание свинкой. Иными словами, если по некоторым причинам мы не можем проверить, есть ли у больного сыпь и жар, вероятность заболевания свинкой равна у него 0,25. Это число является *априорной* вероятностью заболевания свинкой. Первым этапом вычисления вероятности заболевания свинкой является превращение этой априорной вероятности в *априорные шансы*. Вероятность 0,25 соответствует 1/3 шанса, т. е. один шанс за эту болезнь, а три — против.

С каждым вносящим свой вклад фактором связаны два числа, известные как ЛД и ЛН. Здесь ЛД означает логическую достаточность, а ЛН — логическую необходимость. В приведенном примере ЛД для жара равна 10, ЛН — 0,01. Если мы знаем, что жар у больного имеется, то умножаем *априорный* шанс заболевания свинкой на коэффициент ЛД. В данном случае это даст нам текущий, или *апостериорный*, шанс, равный  $1/3 \cdot 10 = 10/3$ .

Если известно, что имеется также и сыпь, то умножаем априорный шанс сыпи на ЛД для сыпи, равную 20, что дает *апостериорный* шанс при учете и жара, и сыпи, равный  $1/3 \cdot 10 \cdot 20$ .

Если же мы знаем, что некоторый симптом отсутствует, то умножаем априорный шанс на соответствующий коэффициент ЛН. Если неизвестно, присутствует некоторый симптом или нет, то мы умножаем на единицу, т. е. шанс не изменяется.

Приведенное в качестве примера правило может быть так переведено на обычный язык. Если нет никаких данных, то вероятность заболевания свинкой равна 0,25. Явное присутствие жара делает свинку в 10 раз более вероятной\*. Явное отсутствие жара делает заболевание свинкой в 100 раз менее вероятным. Явное присутствие сыпи делает заболевание свинкой в 20 раз более вероятным, а четкое ее отсутствие — в 1000 раз менее вероятным.

Соответствующие этим случаям результаты приведены в табл. 8.1.

Т а б л и ц а 8.1

Жар	Сыпь	Шансы	Вероятность
Да	Да	$1/3 \cdot 10 \cdot 20 = 66,6$	0,98
Да	Не знаю	$1/3 \cdot 10 = 3,33$	0,77
Да	Нет	$1/3 \cdot 10 \cdot 0,001 = 0,0033$	0,003
Не знаю	Да	$1/3 \cdot 20 = 6,67$	0,86
Не знаю	Не знаю	$1/3 = 0,33$	0,25
Не знаю	Нет	$1/3 \cdot 0,001 = 0,0003$	0,0003
Нет	Да	$1/3 \cdot 0,01 \cdot 20 = 0,066$	0,06
Нет	Не знаю	$1/3 \cdot 0,01 = 0,0033$	0,003
Нет	Нет	$1/3 \cdot 0,01 \cdot 0,001 = 0,000003$	0,000003

\* В этом абзаце речь идет, конечно, не о вероятностях, а о шансах. — Прим. ред.



Не только вопросы могут выступать в качестве факторов, вносящих свой вклад в байесовское правило. В качестве них могут выступать также и сами правила. Возникает вопрос: как мы собираемся интерпретировать вероятности этих правил в терминах "да", "нет" и "не знаю"? "Да", очевидно, эквивалентно вероятности, равной 1, "нет" — вероятности равной 0, а "не знаю" интерпретируется как вероятность, равная *априорной* вероятности. Однако остается проблемой, что делать, если эта вероятность не совпадает ни с одной из этих трех величин.

Предположим, например, что вероятность некоторого фактора больше, чем априорная вероятность, т.е., скажем, где-то между "не знаю" и "да". Здесь для нас открываются две возможности. Первая — вычислить результирующие вероятности для случаев "не знаю" и "да" и произвести интерполяцию, взяв некоторый промежуточный результат. Вторая — использовать при умножении коэффициент, который лежит где-то между единицей и ЛД. На практике мы выбрали второй путь и используем линейную интерполяцию для коэффициентов ЛД и ЛН смотря по тому, что нужно. Однако теоретического оправдания в пользу какого-то одного подхода нет. Остается даже неясным из всех тех публикаций по системе PROSPECTOR, которые я просмотрел, какой из подходов принят в этой системе. Метод, который был выбран для использования нами, приведет к устройству, которое склонно давать более определенные результаты по сравнению с другими возможными подходами.

Если снова обратиться к рис. 8.2, то можно видеть, что там концевые вершины деревьев названы вопросами. Модель не располагает никакими средствами вычисления вероятностей этих утверждений, так что она должна спросить о них пользователя. Пользователя просят дать оценку вероятности того, что это утверждение истинно. Если требовать от пользователя сразу задать саму вероятность, то употребление такой системы может представить некоторые трудности. Однако в большинстве случаев естественно ожидать, что пользователь в состоянии в ответ на вопрос дать ответ "да", "нет", "не знаю". Этот ответ переводится системой в вероятность, где "да" соответствует 1, "нет" — 0, а "не знаю" соответствует априорной вероятности, задаваемой конструктором. Во втором классе вопросов, применяемых системой, пользователь в своем ответе использует шкалу от +5 до -5. На этой шкале +5 эквивалентно "да" (или вероятности 1), -5 эквивалентно "нет" (или нулевой вероятности), а 0 эквивалентно "не знаю" (т.е. априорной вероятности). Промежуточные значения обозначают степени уверенности в истинности или ложности утверждения вопроса. Например, ответ 4 может интерпретироваться как "очень вероятно", -2 как "вероятно нет", и т.д. Такие промежуточные значения система путем линейной интерполяции переводит в вероятности.

Существуют определенные трудности при интерпретации смысла вопросов такого типа, когда ответы отличаются от простых "да" или "нет". Например, предположим, что система спрашивает: "У вас замаслены свечи зажигания?", а пользователь отвечает: "Не знаю". Этот ответ можно понимать

по-разному. Например: "Я — опытный механик и, по моему мнению, степень замасленности свечей зажигания как раз носит промежуточный характер" или "Я раньше никогда не смотрел на свечи и не знаю, какая степень замасленности считается замасленностью", или "Я не могу найти свечной ключ", или даже "А что такое свеча зажигания?"

Для преодоления этой трудности существует несколько путей. Первый состоит в том, что ответ: "Не знаю" — означает, что система не может получить никакой информации о степени загрязненности свечи зажигания. Следовательно, система ничего не делает и не играет роли причина, по которой она не может получить эту информацию. Второй путь состоит в различении возможных интерпретаций ответа путем задания новых вопросов. Однако при этом подходе надо проявлять определенную осторожность, так как такая возможность может сделать систему весьма утомительной для пользователя.

Экспертные системы работают и дают результаты, несмотря на подобные нерешенные проблемы. Похоже, что сила экспертных систем обусловлена избыточностью правил и что они могут работать, несмотря на то, что используемая в них математика весьма примитивна, что многие задаваемые ими вопросы являются двусмысленными, а во многих моделях используемые правила являются сомнительными. Однако когда некоторые из этих проблем получают решение, то можно ожидать, что экспертные системы заработают еще лучше.

В том виде, в каком мы описали систему, она опирается исключительно на вероятности. Однако во многих моделях существуют такие участки, где наиболее естественно задавать вопросы, ответы на которые могут выражаться в виде числа.

Например, вопрос: "Является ли больной старым" — во многих обстоятельствах двусмыслен. В нем предполагается, что пользователь знает, что считается "старым" для целей, поставленных в модели. Более точным был бы вопрос: "Введите возраст больного в годах". Поэтому в системе предусмотрен один класс вопросов, называемых числовыми, на которые можно отвечать указанием числа.

При включении в систему числовых вопросов необходимо предусмотреть определенный объем арифметических операций с получаемыми системой числами. В системе имеются правила типа Плюс, Минус, Умножить и Разделить. У нас могут быть правила следующего вида:

*Правило разница-в-возрасте* "Различие в их возрасте"

Вычесть возраст-мужчины возраст-женщины

Было бы весьма утомительным строить сложные вычисления, опираясь на столь элементарную систему, но обычно в этом и не возникает необходимости. Когда я перейду к обсуждению внешних функций, вы увидите, что существует и другой путь рассмотрения подобных случаев. Наша система различает вероятности и числа. Она рассматривает эти объекты как объекты разных типов, так что компилятор не даст вам возможности совершать арифметические действия с вероятностями.

Для того чтобы хорошо сочетать численные вопросы с системой, которая исходно предназначена для рассуждений в терминах вероятностей, нужен какой-то способ преобразования чисел в вероятности некоторым контролируемым образом. Для этого имеется два типа правил. Они называются "Диапазон" и "Модуль". В математике эти типы операторов носят название функций принадлежности для нечетких множеств.

Предположим, что в нашей модели имеется вопрос: "Дайте возраст больного в годах", нам требуется утверждение, говорящее, что "больной является старым" и что в данный момент мы определили "старый" следующим образом: всякий моложе 60 лет безусловно не стар, старше 70 — является старым, а в отношении лиц между 60 и 70 годами мы не уверены. Можно было бы написать два следующих правила:

*Правило старый-больной* "Больной является старым"

Диапазон возраст-больного 60 70

*Вопрос возраст-больного* "Возраст больного в годах"

Число 0 120

Числа 0 и 120 в этом примере приведены для проверки диапазона, в котором должен находиться ответ на этот вопрос, чтобы ответ имел ценность. Например, если пользователь в ответ на вопрос дал число 150, то система откажется принять его и вопрос еще раз появится на экране дисплея.

Применяется следующая процедура: если возраст-больного меньше 60, то вероятность старый-больной равна нулю, если значение переменной возраст-больного равно 70 или больше, то вероятность старый-больной равна единице, если же значение переменной возраст-больного лежит между 60 и 70, то вероятность старый-больной находится путем линейной интерполяции\*. Например, если значение возраст-больного равно 65, то вероятность для старый-больной принимает значение 0,5.

Модуль является аналогичной функцией, два параметра которой соответствуют среднему значению и допуску. Например, предположим, что температура некоторой химической реакции должна быть равна 80 градусам, причем допускается отклонение плюс-минус 5 градусов. Иногда можно написать такое правило:

*Правило верная-температура* "Температура верная"

Модуль Показание-температуры 80 5

Вычисления будут производиться следующим образом:

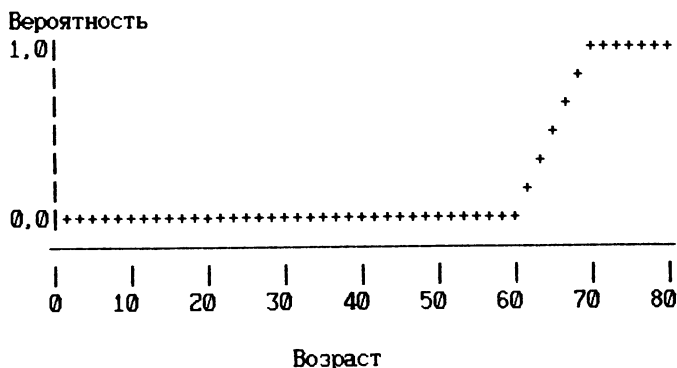
Если показание равно 80, то вероятность верная-температура равна 1. Если показание меньше 75 или больше 85, то вероятность верная-температура равна 0.

Для каждой промежуточной величины показания верная-температура вероятность находится в результате линейной интерполяции.

---

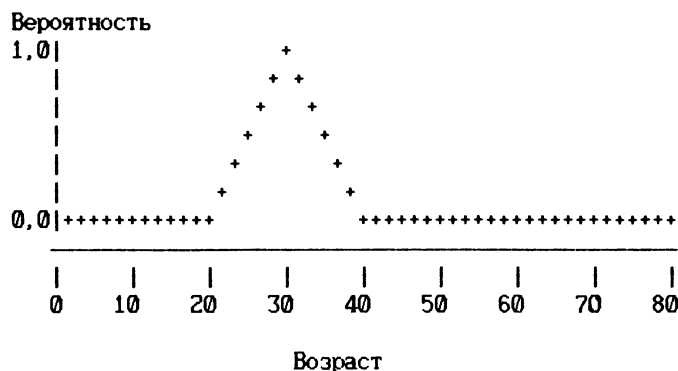
\* Читатель уже заметил, видимо, что здесь автор под вероятностью понимает функцию принадлежности из теории нечетких множеств Л. Заде. — *Прим. ред.*

### Диапазон 60,70 Возраст



а)

### Модуль 30,10 Возраст



б)

Рис. 8.3. Графики функций принадлежности типа "диапазон" (а) и "модуль" (б)

На рис. 8.3 приведены примеры соответствующих графиков. Здесь графически функция модуля отображается в виде пирамиды. Можно было бы возразить: а не лучше ли воспользоваться в этом случае гладкой колоколообразной кривой? Мы решили, однако, что для небольшой системы дополнительные вычисления, связанные с этим, перевесят те преимущества, которые могут быть получены, благодаря использованию более сложной кривой.

Имеется возможность строить более сложные функции принадлежности, используя диапазон и модуль как базовые строительные блоки. Я приведу

один часто используемый пример. Предположим, мы хотим решить, относится ли данное лицо к молодежной группе. Для нашего случая мы скажем, что возрасты от 13 до 19 безусловно являются молодежными, моложе десяти лет — безусловно нет. Двадцать два года и старше также не относятся к молодежи, а в отношении возрастов 11, 12, 20 и 21 имеются сомнения. Можно записать следующие правила:

*Правило молодежи* "Относится к молодежи"

И достаточно-взрослый не-слишком-старый

*Правило достаточно-старый* "Достаточно взрослый, чтобы относиться к молодежи"

Диапазон возраст 10 13

*Правило не-слишком-старый* "Не слишком старый, чтобы относиться к молодежи"

Не слишком-старый

*Правило слишком-старый* "Слишком старый, чтобы относиться к молодежи"

Диапазон возраст 19 22

*Вопрос возраст* "Каков возраст клиента в годах?"

Число 0 120

Теперь мы описали все функции, которые необходимы для логического определения нашей модели. Имеется еще один вид информации, которую мы должны уметь записывать в нашей базе правил и которая сделает нашу систему жизнеспособной. Мы должны иметь определенные средства управления порядком задания вопросов, а также иметь средства указывать, что при определенных обстоятельствах некоторые вопросы следует считать не относящимися к делу.

Рассмотрим сначала порядок задания вопросов. Существуют обстоятельства, при которых пользователю было бы очень неудобно, чтобы система задавала вопросы в том порядке, который представляется ей удобным. Предположим, например, что наша система содержит два вопроса: "Не замаслены ли у вас свечи зажигания?" и "Нет ли на свечах зажигания белого налета?" Если оба вопроса будут заданы так, что между ними будут вклиниваться другие вопросы, то появляется определенная вероятность, что пользователю придется дважды вывинчивать свечи зажигания из двигателя автомашины, и вряд ли это придется ему по душе. Поэтому мы должны иметь возможность заставить систему задавать эти вопросы вместе. Существует множество примеров, когда необходимо контролировать порядок вопросов просто для того, чтобы заставить систему вести беседу более привычным для человека образом. Пользователи-люди будут явно раздосадованы, если система станет перескакивать с одного вопроса на другой внешне случайным образом, поэтому часто необходимо вводить здесь определенный порядок.

Второе требование заключается в том, чтобы иметь возможность указать системе, что некоторые вопросы уместны лишь при определенных обстоятельствах. Как пример предположим, что в модели содержится три воп-

роса: (1) "Каков пол больного?", (2) "Каков возраст больного?" (3) "Сколько было беременностей?" Очевидно, что если больной оказался мужского пола, тогда число беременностей равно нулю. Вряд ли в этом случае система будет выглядеть для пользователя разумной, если тем не менее она задаст вопрос о том, сколько было беременностей. Кроме такой логической состоятельности существуют и другие причины, по которым хотелось бы, чтобы система была в состоянии управлять процессом задания вопросов.

Во-первых, имеется проблема реализации эвристик. Например, специалист по рассматриваемому вопросу может сказать: "Я считаю, что при таких-то обстоятельствах лучше исследовать возможность А прежде, чем рассматривать возможность В". Во-вторых, нужно, чтобы система выглядела достаточно вежливой. Слишком легко неумышленно создать модели, которые так ведут беседу с пользователем, что она выглядит грубой или невнимательной по отношению к нему. Например, предположим, что наша модель приводит к следующему диалогу:

Система: "Вы замужем?"

Пользователь: "Нет"

Система: "Сколько раз вы были беременны?"

Ясно, что в этом случае система выглядела бы весьма неотесанной.

Если вы посмотрите на рис. 8.4, то увидите, что он отличается от рис. 8.2 добавлением прямоугольника, в котором содержатся слова: "Есть ли у Вас

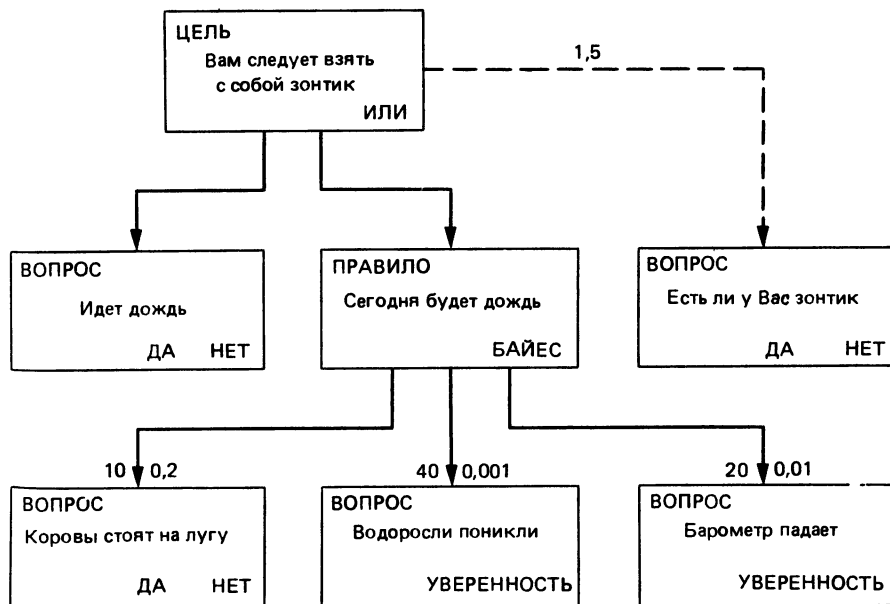


Рис. 8.4. Использование блокирующего фактора в базе правил на рис. 8.1, 8.2

зонтик"; этот прямоугольник подсоединяется к целевому посредством штриховой стрелки, с пометкой: "Вам следует взять с собой зонтик". Около этой стрелки указаны два числа 1,5. Они означают, что, прежде чем будет предпринята какая-либо попытка выяснить, следует ли вам брать с собой зонтик, необходимо ответить на вопрос: "Есть ли у Вас зонтик". Ответ должен лежать в диапазоне 1 5, означающем положительное утверждение. Это записывается на нашем языке правил с использованием ключевого слова "блокировка" и получило название блокирующего фактора. Целевое правило модели, изображенной на рис. 8.4, записывается так:

*Цель взять-зонтик* "Вам следует взять с собой зонтик"

Блокировка иметь-зонтик 1 5

Или дождь будет-идти

Этот простой механизм позволит осуществить все виды контроля порядка задания вопросов, описанные выше. Если вам просто необходимо ограничить порядок частей модели, можете записать, например

Блокировка возраст —5 5

При использовании блокировки необходимо принять во внимание еще один фактор. Сама по себе блокировка влияет на порядок, в котором задаются вопросы в этой модели, и на то, будут ли они вообще заданы. Она не влияет на вычисляемые вероятности. В последнем примере модель сначала задаст вопрос: "Есть ли у Вас зонтик". Если ответ будет отрицательным, то система больше не задаст никаких вопросов. Это как раз то, чего мы добивались, но система даст ответ: "Не знаю" — целевому утверждению: "Вам следует взять с собой зонтик". Это произойдет потому, что значение этого целевого утверждения определено, как "Идет дождь" или "Сегодня будет дождь", а поскольку мы не рассматривали ни одну из этих гипотез, то обе они имеют значение "Не знаю". Если вы хотите, а это, по-видимому, так, чтобы ответом было "Нет", в этих обстоятельствах это должно быть оговорено явно. Вот пример, в котором такая ситуация будет обработана правильно:

*Цель взять-зонтик* "Вам следует взять с собой зонтик"

Блокировка иметь-зонтик 1 5

И иметь-зонтик нуждаться-в-зонтике

*Правило нуждаться-в-зонтике* "Вам нужен будет зонтик"

Или идет-дождь сегодня-будет-дождь

Можно видеть, что вопрос "Иметь-зонтик" включен в цель "Взять-зонтик" и как блокирующий фактор, и как часть логического вывода, так что при негативном ответе на "Иметь-зонтик" он подавит любой дальнейший анализ гипотезы "Взять-зонтик" и обеспечит, чтобы у этой цели был результат "Нет".

Важно понимать, что использование блокирующего утверждения блокирует некоторый путь прохождения модели, а не вопросы, относящиеся к этому пути. Обратимся к следующему примеру:

*Цель "А" "..."*

Блокировка В 1 5

И С D

*Цель Е "..."*

ИЛИ С F

В этом примере система начнет с изучения цели А. Цель А заблокирована на В, так что сначала будет изучаться В. Предположим, что для В получен результат —3. Система не будет заниматься изучением С или D. Она перейдет к исследованию второй цели — цели Е. При изучении Е система будет изучать С, несмотря на то, что С заблокирована В в цели А. Как вы можете убедиться, заблокированным оказался лишь путь к С через цель А, а не само С. С другой стороны, у D нет альтернативного пути, оно может быть достигнуто лишь через А. Следовательно, D никогда не будет рассматриваться.

Теперь мы рассмотрели все характеристики языка советов. Однако на практике мы обнаружили, что время от времени конструктор модели выдвигает некоторые уникальные требования, которые не могут быть удовлетворены в том варианте системы, который мы описали. Поэтому мы включили в систему средство, позволяющее пользователю писать то, что известно под названием внешних функций или внешних вопросов. Они представляют собой процедуры, составляемые конструктором модели обычно на языке Паскаль, которые подсоединяются к системе на этапе ее работы.

Сначала я опишу, как выглядит внешний "вопрос": система не высвечивает сообщение на дисплее и из ответа пользователя не получает его значение, а вызывает процедуру на языке Паскаль, которая и дает вероятность или число, помещаемое системой в качестве текущего значения для этого вопроса. Внешние вопросы могут использоваться многими способами. Они могут употребляться для переговоров с каким-то блоком оборудования или для обеспечения интерфейса с некоторой базой данных.

Внешняя функция представляет собой правило, содержащее определенную пользователем процедуру для вычисления текущего значения, исходя из значений влияющих на него факторов. Например, в некоторых обстоятельствах пользователь может решить, что принятая в системе процедура вычисления нечеткого И как минимума из вероятностей влияющих факторов — это не то, что ему требуется. Он, возможно, предпочитает вычислить И как произведение вероятностей факторов. Это легко можно сделать с помощью внешней функции.

Внешние функции могут быть как вероятностными, так и числовыми. Мы могли бы при необходимости таким образом реализовать тригонометрические функции. Возникает также возможность для реализации специализированных функций принадлежности для нечетких множеств, например упомянутой выше колоколообразной кривой.

В качестве другого примера предположим, что в модели имеется вопрос, который читается следующим образом:

Из какого металла сделаны трубы теплообменника?

Напечатать 1 если Медь



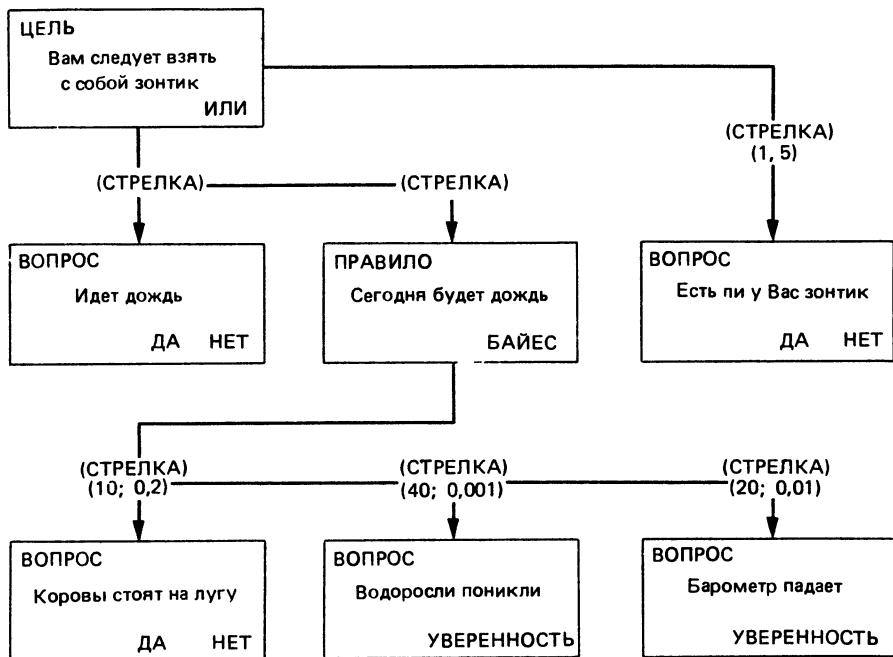


Рис. 8.5. Фактическое представление примера базы правил в памяти ЭВМ

- 2 если Никель
- 3 если Специальный сплав
- 4 если Бронза

И кроме того, нам нужно выяснить теплопроводность труб теплообменника. Тогда можно написать внешнюю функцию, которая будет воспринимать то число, которое вводит пользователь, отвечая на вопрос, и выдавать теплопроводность путем обращения к соответствующей таблице.

Теперь, когда вы знаете, как выглядит база правил в форме источника, я могу описать работу системы на этапе ее запуска. Для понимания принципа работы программы системы необходимо знать, как расположены знания в ее основной памяти. Память, взятая из неупорядоченного массива языка Паскаль, состоит из двух типов записей, прямоугольников и стрелок. На рис. 8.5 показан пример с зонтиком, так как он физически расположен в машине. Все типы записей имеют фиксированную длину, и, как вы помните, все текстовые строки в источнике хранятся в отдельном файле на диске и загружаются в основную память лишь тогда, когда их выводят на дисплей. Записи для прямоугольников содержат лишь указатели, идущие к текстовому файлу.

Имеется также массив указателей, ведущих ко всем целям модели. Этим обеспечивается начальный доступ к деревьям, которые образуют модель.

В каждой записи для прямоугольника имеется два действительных числа — текущая вероятность и априорная вероятность гипотезы, которую этот прямоугольник представляет. Кроме того, там есть два указателя. Первый из них указывает на начало цепочки из записей для стрелок, которые связывают его с влияющими на него факторами. Второй указывает на другую цепочку записей для стрелок, которые, в свою очередь, указывают на блокирующие факторы для этого правила. В записи имеются также два числа, указывающие на ассоциируемый текст из текстового файла для этой модели, на некоторые булевы флаги, отражающие текущее состояние прямоугольника, например "блокирован", "получает ответ" и т.д.

Каждая запись для стрелки имеет два указателя. Первый указывает на следующую за ней запись для стрелки в той цепочке, к которой она относится. Другой указатель показывает на влияющий или блокирующий фактор. В записи для стрелок также отведено место паре чисел. В них могут содержаться константы ЛД и ЛН, а если требуется — то константы области блокировки, которые ассоциированы с этой стрелкой.

Метод работы состоит в просмотре дерева, пока не встретится вопрос, на который нет ответа. Вопрос выдается на дисплей, а ответ поступает в поле текущей вероятности. Затем вероятности всех прямоугольников на дереве пересчитываются. Этот процесс продолжается до тех пор, пока не будет прерван пользователем или пока все вопросы не окажутся исчерпаны.

Имеется целый ряд вспомогательных функций, которые способна выполнять система. Они срабатывают, когда пользователь вместо прямого ответа на вопрос набирает на терминале контрольный символ. Все контрольные символы составлены из букв латинского алфавита, за исключением запроса типа "помощь", для чего используется вопросительный знак. При обращении за помощью на экране дисплея появится список всех управляющих кодов.

Я не буду подробно рассматривать все эти коды, а остановлюсь на некоторых наиболее нестандартных возможностях. Имеется целый ряд обслуживающих операций, таких как "Закончить" и "Вернуться" (т.е. повторить предыдущий вопрос). Обычно система рассматривает цели по очереди в том порядке, в котором они встречаются в списке целей. Однако каждый раз при смене цели система информирует об этом пользователя и дает ему возможность сделать свой выбор. Имеется контрольный код, который позволяет отбросить текущую цель и вернуть систему на этап выбора цели, давая возможность пользователю при желании самому выбрать следующую цель.

Имеются средства объяснения. Объяснение сводится к просмотру пути от текущего вопроса назад к цели, из которой отправлялся поиск, приведший к данному вопросу. Результаты трассировки на дисплей выводятся в виде обычного текста. Например:

Текущий вопрос состоит в том, поникли водоросли или нет.

Истинность утверждения, что водоросли поникли, существенно подкрепляет утверждение, что будет дождь, а ложность существенно противоречит ему.

“Сегодня будет дождь” — является одним из факторов, который должен быть истинен для утверждения: “Вам следует взять с собой зонтик”. “Вам следует взять с собой зонтик” — является текущей целью.

Такого рода средства объяснения включены в большинство экспертных систем, и о них много сказано в литературе. Они призваны сделать систему более дружелюбной по отношению к пользователю и более понятной. Хотя на демонстрациях эти объяснения выглядят довольно хорошо, мы нашли, что пользователи быстро от них устают и считают объяснения такого сорта неглубокими. Для этого имеются две причины. Одна из них состоит в том, что когда объяснения дает человек, то он пропускает самые очевидные шаги, поясняя их только в том случае, если у слушателя обнаруживается затруднение в отслеживании материала. Другая причина состоит в том, что, по-видимому, люди не всегда думают в терминах правил. Хочется получить не такое объяснение: “У меня есть правило, которое говорит, что если ваши водоросли поникли, то дождь весьма вероятен”. Спрашивающий хотел бы знать, почему компьютер думает, что поникшие водоросли должны указывать на дождь? Однако то средство, которое имеется, весьма важно для отладки модели, и оно направлено на удовлетворение той потребности, которая реально возникает у пользователя. Все мы вынуждены как-то двигаться дальше, не дожидаясь, пока эта потребность получит полностью удовлетворительное решение.

Имеются контрольные коды, которые позволяют пользователю в любой момент изучить какую-то цель или всю их совокупность, правила или вопросы. Они позволяют также вывести на дисплей текущее состояние любого прямоугольника модели и его текущее значение. Это очень полезно при отлаживании модели.

Имеется также средство, с помощью которого можно в графической форме показать, насколько близко система подошла к решению. В начале сеанса единственное, что можно с определенностью сказать относительно некоторой целевой гипотезы, — это то, что ее вероятность находится между 0 и 1. По мере того как получаются ответы на вопросы, возникающие по пути к этой цели, верхняя и нижняя границы для вероятностей будут изменяться. Нижняя граница будет расти, а верхняя — падать. Когда будут получены ответы на все уместные вопросы, то верхняя и нижняя границы сольются, дав общее значение вероятности для этой цели.

Соответствующая контрольная комбинация позволяет вывести на экран верхнюю и нижнюю границы для вероятностей всех целей в виде набора вертикальных отрезков типа гистограммы. В большинстве случаев пользователь не пытается найти точные значения вероятностей для всех целей в модели. Как правило, ему хочется знать, какие утверждения являются истинными с наибольшей вероятностью или какой образ действия имеет большую вероятность успеха. Время от времени пользователь может смотреть на эту “гистограмму” и решать, в какой момент он достиг удовлетворяющей его точности.

У нас накопился двухлетний опыт работы с системой "Микроэксперт", и мы видели большое число моделей, созданных на этой системе нами и другими специалистами. Мы теперь в состоянии сделать определенные выводы относительно слабости и силы нашего подхода, знаем, в чем его можно было бы усовершенствовать и для каких задач он наиболее пригоден.

Мы установили, что "Микроэксперт" является очень хорошей системой для тех, кто хочет начать создавать экспертную систему. Используемые в ней концепции просты и доступны для понимания. Она достаточно эффективна, чтобы пользователь мог создавать свои первые модели. Постепенно становится общепринятым, что работу над экспертными системами следует начинать с чего-то очень простого. Этот вводный уровень системой "Микроэксперт" обеспечивается достаточно хорошо.

Мы обнаружили, однако, что во многих серьезных задачах содержится большое число правил, и было бы полезно разработать более эффективный и выразительный язык правил для таких более требовательных приложений. Соображения здесь подобны тем, которыми обмениваются сторонники языков Бейсик и Паскаль. Язык Бейсик легко изучить, и он очень хорош, пока вы не захотите написать очень большую программу. Когда же вы начинаете создавать большие программы, то обнаруживаете, что дополнительные средства языка Паскаль стоят того, чтобы тратить время на их изучение и усвоение.

Требования, которые различные пользователи предъявляют к человеко-машинному интерфейсу экспертной системы, варьируются достаточно широко. Мы прикладываем много усилий для развития набора средств по организации ввода и вывода в нашей следующей системе. Они должны позволять составителю модели форматировать экран любым удобным ему способом и выдавать отчет в любом необходимом ему формате как на экран, так и на устройство печати.

Мы работаем также над способами, которые дадут возможность разработчику моделей определить общую стратегию создания моделей. Мы хотим предоставить в его распоряжение средства, позволяющие писать: "Когда вы узнаете, что из А, В и С является более вероятным, то прервать эту линию рассуждения и перейти к следующим действиям".

Как говорилось выше, мы не удовлетворены имеющимися средствами объяснения. Для обеспечения более совершенного объяснения могут быть использованы различные средства. Однако, как мы сказали, источником слабости объяснений является тот факт, что информация, которая необходима в модели для выполнения ее первичной задачи дедукции или диагностики, недостаточна для создания хорошего объяснения. Создание хорошего объяснения связано с дополнительной нагрузкой на пользователя системы по занесению в модель дополнительной информации. Мы же можем лишь предоставить ему для этого средства и контролировать их использование. Пользователю решать, обращаться ли ему к этим средствам.

Было бы очень полезно иметь более развитые средства для объединения экспертной системы с другими программами. В частности, было бы неплохо

обращаться к экспертной системе из другой программы или иметь лучшие средства контакта с базой данных.

Экспертные системы работают достаточно хорошо, несмотря на шаткость того математического фундамента, на котором они создаются. Однако если удастся создать более надежный математический фундамент, то, вероятно, информацию в базе правил удастся использовать более разумно и экономно и создатель моделей сможет выразить более тонкие отношения между гипотезами в его моделях. Это — область, на которую должны быть нацелены научные исследования.

В заключение отметим, что работа над системой "Микроэксперт" многому научила нас. Мы пришли к скромной, но, в общем, работоспособной системе, а применение этой системы к различным задачам как нами, так и другими людьми позволило многое выяснить относительно требований, которые потенциальные пользователи предъявляют к экспертным системам. Мы надеемся, что полученный опыт позволит нам сделать следующую систему намного лучшей.

## Глава 9

### REVEAL — ВЫЧИСЛИТЕЛЬНАЯ СРЕДА ДЛЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ

*Питер Джонс*

Перед разработчиком системы с использованием знаний, предназначенной для определенной прикладной задачи, открывается несколько возможных направлений. Существует целый ряд специализированных языков программирования, ориентированных на применение в приложениях искусственного интеллекта. Среди них Лисп — язык программирования, наиболее популярный в США, обеспечивший значительные успехи в технологии разработки систем искусственного интеллекта, главным образом по причине его широкой распространенности в университетской среде.

В Европе более широкую поддержку имеют язык Пролог, первоначально разработанный в Марсельском университете и получивший развитие в Эдинбургском университете, и другие, ориентированные на искусственный интеллект средства, такие как POP-2 и POPLOG. Пролог, являющийся подмножеством языка программирования, основанного на исчислении предикатов, привлек большое внимание в последние годы вследствие выбора этого языка в качестве первого варианта логического языка программирования в японском проекте создания ЭВМ пятого поколения, возглавляемого институтом ICOT.

При использовании любого из этих языков разработчику необходимо создавать полную инфраструктуру для интересующего приложения с самого начала, хотя следует отметить, что развитая вычислительная среда для языка Лисп, имеющаяся, например, в специально для этого созданных Лисп-маши-

нах, обеспечивает весьма широким набором библиотечных программных модулей. Создания такой же серьезной поддержки следует ожидать и на автоматизированном рабочем месте (АРМ) с языком Пролог как первый результат японской инициативы по созданию ЭВМ пятого поколения.

Однако ясно, что конструирование с нуля в некотором исходном языке программирования будет весьма дорогим упражнением в отношении как затрачиваемых средств, так и необходимого для этого времени. По этой причине были разработаны специальные инструментальные средства, находящиеся на более высоком уровне абстракции по сравнению с машинными языками. Их можно разбить на две группы. С одной стороны, это языки "представления знаний" более высокого уровня, такие как ROSIE и OPS-5, а с другой — так называемые оболочки для экспертных систем, например EMYCIN, AL/X, LOOPS, UNITS и SAGE.

Языки представления знаний сами по себе, как правило, реализованы на языке Лисп. Вследствие уже этого факта они представляются менее подходящими для реальных приложений, носящих характер "промышленного продукта", поскольку их применение вызовет большую потерю времени работы системы. Часть исследователей в США считает, что указанным языкам, (и, конечно, языку Лисп) должна быть отведена роль некоторого исходного средства, позволяющего создать черновые варианты экспертных систем, которые для повышения рабочей эффективности системы в дальнейшем будут переписаны на таких языках, как, например, Бейсик или Фортран.

Если говорить об оболочках экспертных систем, то они позволяют работать на системном уровне, а не на уровне программирования. Они содержат невидимые для пользователя структуры данных и управляющую стратегию, необходимые для реализации каждого конкретного приложения. Благодаря этому объем программирования становится минимальным, но за это приходится платить потерей гибкости такого рода оболочки. Если планируемое приложение не относится к классу задач, который имелся в виду при создании оболочки, то могут возникнуть самые серьезные трудности при попытке уложить задачу в рамки архитектуры, которая, в сущности, предопределяет ответ.

Общий недостаток, присущий всем приведенным выше подходам, состоит в том, что указанные вычислительные средства рассчитаны исключительно на специалистов по использованию знаний. Существует множество приложений, где для решения задачи наряду со средствами, относящимися к технологии знаний, необходимы также и более традиционные средства численного программирования, работы с данными, средства создания таблиц и отчетов, графические средства и т.п. Языки искусственного интеллекта и оболочки не слишком удобны для этих задач.

На такие приложения нацелена система REVEAL. В этой системе нашли применение как методы использования знаний, так и методы поддержки процесса решения. Если не рассматривать приложения, в которых используются знания, то систему REVEAL можно считать генератором сис-

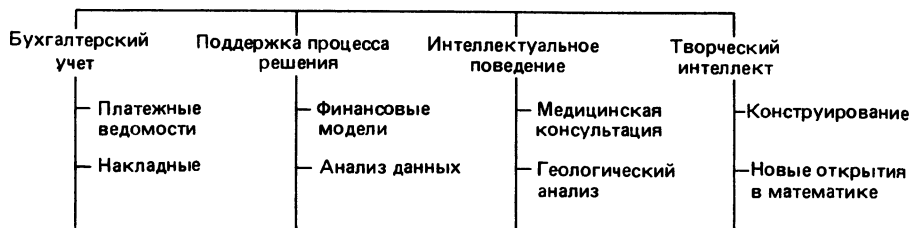


Рис. 9.1. Классификация областей применения вычислительных машин

тем поддержки решений, а если отвлечься от систем, требующих поддержки решений, то ее можно считать средством для создания экспертных систем. Но главным достоинством системы REVEAL является совместное использование этих методов, что позволяет рассматривать приложения, где требуется и то и другое.

Если попытаться предложить классификацию различных приложений вычислительных машин, то можно получить что-то похожее на классификацию, представленную на рис. 9.1. Слева на нем мы видим большую группу приложений, относящихся к бухгалтерскому учету. Это – самые непосредственные применения вычислительной техники, исторически рассчитанные на пакетный режим обработки. Они реализуются на одном из коммерческих языков, например на языке Кобол или ПЛ/1, а конечный пользователь знает о них лишь по готовым выходным результатам.

Вторая группа на приведенной схеме называется "поддержка процесса решения" и относится к системам, которые появились после создания интерактивных режимов вычислений в конце 60-х годов. Они характеризуются двумя моментами. Доступ к вычислительной машине через терминал означает, что конечный пользователь находится в контакте с самой программой, не ограничиваясь только результатами ее работы. При этом для облегчения задачи программирования конечному пользователю предоставляются средства, позволяющие воспользоваться элементарными типами данных. Такие типы данных могут варьироваться от простых векторов, составленных из чисел, как во многих системах поддержки экономических решений, и многомерных структур данных, которые имеются в более современных и более мощных системах, вплоть до парадигмы системы VISICALC, в которой структура данных в виде развернутой электронной страницы означает, что программа как таковая снова исчезает. К этой же группе относятся средства математического программирования для исследования операций, динамическое программирование, моделирование, стохастическое программирование и т.д., хотя правильнее было бы их относить к сфере деятельности специалистов, а не конечного пользователя.

Третья группа на схеме носит название "интеллектуальное поведение". Она включает в себя экспертные системы, основывающиеся на использовании знаний, которые были созданы и применены за последнее десятилетие. В качестве примеров можно взять систему MYCIN (на самом деле это могла

бы быть любая из десятка хорошо документированных экспертных медицинских систем) и систему PROSPECTOR для геологического анализа, разработанную в Станфордском исследовательском институте (США).

Последняя группа, названная здесь "творческий интеллект", в сущности, относится к переднему краю исследований, ведущихся в области искусственного интеллекта. Два приведенных на схеме примера относятся к работе, связанной с именем профессора Д. Лената, начинавшего в Университете Карнеги-Меллона, а впоследствии работавшего в Станфордском университете. Диссертация Лената была посвящена построению системы АМ (Artificial Mathematician — Искусственный математик). После внесения в эту систему множества основных дочисловых математических понятий ей была дана установка на выдвижение интересных математических идей, выполняя которую система открыла заново большую часть разделов теории чисел. Дальнейшее развитие структуры системы АМ, состоявшее во внесении в нее самомодифицирующихся эвристик, привело к созданию системы EURISKO, которая была использована в целом ряде областей для изучения систем понятий. В частности, это привело к разработке ("открытию") новых типов трехмерных компонент для сверхбольших интегральных схем.

Система REVEAL ориентирована на две центральные группы в приведенной классификации. С одной стороны, масштабные коммерческие применения достаточно хорошо обеспечены множеством существующих языков и программных систем. С другой стороны, высокие требования к вычислительным мощностям большинства приложений исследовательского характера из области искусственного интеллекта означают малую вероятность того, что они будут коммерчески полезными, пока в следующем поколении вычислительных систем не будут снижены затраты до уровня, приемлемого для включения в производственный бюджет.

Таким образом, существо проекта системы REVEAL состоит в добавлении к системной базе поддержки решения некоторых свойств языков, используемых специалистами по искусственному интеллекту. Среди них — возможность осуществлять логическую дедукцию или вывод и возможность работы с абстрактными типами данных (т.е. символьное программирование). В основу архитектуры такого рода системы положены результаты трех направлений исследований, которые активно проводились в течение последних 15–20 лет. Первое направление — разработка эффективных систем моделирования и систем поддержки решения. Второе — появление и развитие архитектуры экспертных систем. А третье, связывающее первые два, — это теория нечетких множеств.

#### 9.1. СИСТЕМА REVEAL КАК СРЕДСТВО ПОДДЕРЖКИ ПРОЦЕССА ПРИНЯТИЯ РЕШЕНИЯ

Профессор Р. Спрейг высказал мысль, что о возможностях любой системы поддержки процесса решения следует судить по тому, насколько хорошо в ней организовано управление логикой, управление данными и управление диалогом с пользователем [1].



Рассмотрим сначала систему REVEAL с точки зрения управления логикой. В этой системе имеется свой собственный язык программирования, предназначенный для самого широкого применения. С одной стороны, он может быть использован в очень простых числовых применениях финансового и другого характера. На рис. 9.2 приведена иллюстрация его использования при создании модели финансового документа о доходах и расходах, связанных с некоторым конкретным продуктом.

На этом уровне использования в нем с трудом можно узнать стандартный язык программирования. Он оказывается весьма удобным для конечного пользователя, который может непосредственно реализовать на нем некоторые простые дескриптивные модели. Однако этот язык содержит также все структуры данных и необходимые управляющие структуры, естественные для законченных языков высокого уровня, и по этой причине он годится для использования профессиональными программистами и аналитиками исследования операций, которым необходим развитый алгоритмический язык программирования. На рис. 9.3 приводится пример использования некоторых из этих высокоуровневых свойств языка для реализации одного алгоритма построения магических квадратов нечетного порядка.

Все модели (или программы), построенные на языке REVEAL, допускают выполнение в нескольких вариантах, выбор которых устанавливается командами языка. Кроме простых команд выполнения и печати в режиме анализа зависимости могут многократно выполняться определенные логические шаги. Этим обеспечивается выявление однопараметрической и многопараметрической зависимостей. Таким образом, одной командой можно задать целый набор возможных сценариев. Особенностью системы REVEAL является то, что это потенциально гигантское множество возможностей просматривается в ходе выполнения программы, причем на выход системой выдаются лишь результаты, получаемые для класса случаев, удовлетворяющих требуемому критерию. Поскольку сама установка такого критерия может выполняться динамически по определенному алгоритму, то возникает возможность реализовать на этом языке весьма сложные алгоритмы поиска.

На рис. 9.4 дан простой пример работы программы восстановления зависимости для случая простой финансовой модели, показанной на рис. 9.2. Иллюстрируется также возможность поиска цели или интерактивного поиска, при котором может быть вычислено требуемое значение входной переменной, обеспечивающее некоторое желаемое значение для выходной переменной. Анализ зависимости и поиск цели могут производиться одновременно, если воспользоваться режимом анализа устойчивости. В этом режиме некоторой переменной придаются значения из определенного набора и в каждом случае вычисляется такое значение другой переменной, при котором значение третьей переменной достаточно мало отклоняется от целевого значения.

Всякое серьезное моделирование неизбежно связано с накоплением данных. В системе REVEAL предусмотрены внутренние средства для работы с целостными данными. Все файловые объекты системы REVEAL

```

1 : ! Пример модели доходов и расходов
2 : ! =====
3 :
4 :     стоимость=складывается (базисн.стоимость,инфляц.стоимости,12)
5 :     издержки=складывается (базисн.издержки,инфляц.издержек,12)
6 :
7 :     разм.прибыли=стоимость-издержки
8 :     вал.стоимость=год.объем*стоимость
9 :     доход=год.объем*разм.прибыли
10 :    чист.доход=доход-фикс.издержки
11 :
12 :    доход%сбыт=чист.доход/вал.стоимость*100

```

### Планирование доходов и расходов

	Январь 1983	Февраль 1983	Март 1983	Апрель 1983	Май 1983	Июнь 1983
Продажная стоим.на ед.	10.00	10.12	10.24	10.36	10.48	10.60
Прямые издержки на ед.	7.00	7.10	7.20	7.30	7.40	7.50
Размер прибыли на ед.	3.00	3.02	3.04	3.06	3.08	3.10
Плановый объем сбыта	100	115	110	130	135	140
Валовая прибыль	\$300	\$347	\$334	\$398	\$416	\$434
Фиксированные издержки	\$150	\$150	\$150	\$160	\$165	\$170
Чист. доход от операции	\$150	\$197	\$184	\$238	\$251	\$264
	=====	=====	=====	=====	=====	=====
Доход в % к вал.стоим.	15%	17%	16%	18%	18%	18%

#### Основные предположения:

Инфляция продажной цены 15% в год  
Инфляция прямых затрат 18% в год

Рис. 9.2. Простая модель доходов и расходов

# РЕЖИМ >выполнение

```
..генератор магических квадратов
..введите размер квадрата >11
..магич.число равно 671
```

56	69	82	95	108	121	2	15	28	41	54
55	57	70	83	96	109	111	3	16	29	42
43	45	58	71	84	97	110	112	4	17	30
31	44	46	59	72	85	98	100	113	5	18
19	32	34	47	60	73	86	99	101	114	6
7	20	33	35	48	61	74	87	89	102	115
116	8	21	23	36	49	62	75	88	90	103
104	117	9	22	24	37	50	63	76	78	91
92	105	118	10	12	25	38	51	64	77	79
80	93	106	119	11	13	26	39	52	65	67
68	81	94	107	120	1	14	27	40	53	66

# РЕЖИМ >инспектирование

```
1 :   печать('')
2 :   печать('..генератор магических квадратов')
3 :   запросить('..введите размер квадрата ')
4 :   iах=ответ
5 :   если iах меньше 1 то идти.к непол:
6 :   если iах больше 19 то идти.к велико:
7 :   если целое(iах/2)*2 равно iах то идти.к нечетн:
8 :
9 :   печать('..магич.число равно ',(1+iах*iах)/2*iах как 'xxxx')
10 :  печать('')
11 :
12 :  повторять i=(1,iах)
13 :    повторять j=(1,iах)
14 :      k=j-i+целое(iах-1)/2
15 :      l=2*j-i
16 :      если k больше.равно iах то k=k-iах
17 :      иначе выполнить
18 :        если k меньше 0 то k=k+iах
19 :        закончить
20 :      если l больше iах то l=l-iах
21 :      иначе выполнить
22 :        если l меньше.равно 0 то l=l+iах
23 :        закончить
24 :      печать(k*iах+l как 'xxxx')
25 :    стопповтор
26 :  печать('')
27 :  стопповтор
28 :  печать('')
29 :
30 :  выход
31 :
32 :  непол: печать('—сторона квадрата должна быть положительной')
33 :  выход
34 :  велико: печать('—максимальная сторона квадрата равна 19')
35 :  выход
36 :  нечетн: печать('—у квадрата должно быть нечетное число строк')
37 :  выход
```

Рис. 9.3. Программа для системы REVEAL

РЕЖИМ >зависимость  
 ВАРЬИРУЕМ : >инфляц.стоимости 12% 20% 2%  
 ВАРЬИРУЕМ : >инфляц.издержек 14% 18% 1%  
 ВАРЬИРУЕМ : >  
 .. 25 СЛУЧАЕВ; ПРОДОЛЖАТЬ? >нет  
 .. РЕДАКТОР? >да  
 .. ВВЕДИТЕ УСЛОВИЯ ПРЕКРАЩЕНИЯ ВЫДАЧИ  
 > если элемент(чист.доход,6) меньше 300 то конец  
 ЗАПОМИНАТЬ? >нет

ИНФЛЯЦ. СТОИМОСТИ	ТЕКУЩЕЕ ЗНАЧЕНИЕ	0,20
ИНФЛЯЦ. ИЗДЕРЖЕК	ТЕКУЩЕЕ ЗНАЧЕНИЕ	0,14

	Январь 1983	Февраль 1983	Март 1983	Апрель 1983	Май 1983	Июнь 1983
ЧИСТ. ДОХОД	150,00	203,77	196,94	260,32	282,40	305,51

ИНФЛЯЦ. СТОИМОСТИ	ТЕКУЩЕЕ ЗНАЧЕНИЕ	0,20
ИНФЛЯЦ. ИЗДЕРЖЕК	ТЕКУЩЕЕ ЗНАЧЕНИЕ	0,15

	Январь 1983	Февраль 1983	Март 1983	Апрель 1983	Май 1983	Июнь 1983
ЧИСТ. ДОХОД	150,00	203,18	195,79	258,27	279,52	301,73

РЕЖИМ >найти инфляц.стоимости  
 .. ЗАДАНИЕ : >чист.доход 350  
 ЗАПОМИНАТЬ ? >нет

ПЕРЕМЕННАЯ	ЗАДАНИЕ
0,20	290,52
0,28	332,30
0,31	349,04
0,32	349,74
0,32	349,99
0,32	350,00

Рис. 9.4. Анализ зависимости (две переменные после редактирования)

хранятся в одном главном системном файле, называемом базой данных системы REVEAL. Эта база данных организована по реляционной схеме. Таким образом, сложными множествами данных, связанными с одной или несколькими программами, можно манипулировать на уровне команд языка, так что пользователь или разработчик не сталкивается с проблемами работы с файлами и записями. Можно создавать объединения и проекции отношений, можно сравнивать и перемещать логические компоненты различных отношений.

Кроме работы с внутренней базой данных в системе REVEAL предусмотрены все средства ввода данных и их вывода на внешний файл. Таким образом, внешние данные могут быть взяты из какого-то другого приложения или другой базы данных, а данные, созданные в системе REVEAL, могут быть переданы в другие прикладные задачи, выполняемые на той же самой или на других машинах.

Управление диалогом предусматривает использование полного набора средств человеко-машинного контакта. Что касается системы REVEAL, то в первую очередь необходимо отметить, что система полностью замкнута, располагает своим редактором, транслятором, подсистемой работы с данными и с файлами и подсистемой создания отчетов. Благодаря этому пользователь полностью изолирован от операционной системы, установленной на данной машине, и создается гарантия того, что во всех случаях использования системы REVEAL на разных машинах ее поведение будет в точности одинаковым.

Множество команд языка реализовано в виде множества атомарных (элементарных) команд, которое может пополняться. Такой подход дает два преимущества. Во-первых, неопытный пользователь может начать продуктивную работу с системой REVEAL, располагая вначале примерно десятком из более чем ста команд, имеющихся в системе, и обращаясь к дополнительным возможностям по мере того, как у него появляется в этом потребность. Во-вторых, атомарный характер команд системы REVEAL допускает использование интерпретатора. Команды можно группировать, создавая командный файл, который можно затем вызывать по имени, что, безусловно, удобно для объединения часто используемых средств системы. Кроме того, в системе предусмотрена возможность использования переменных в командном файле, значения которых определяются в ходе его выполнения. Это позволяет создателю системы сформировать полный словарь команд, соответствующий потребностям конечного пользователя. При этом конечный пользователь оказывается изолированным не только от операционной системы, используемой вычислительной машиной, но и от самой системы REVEAL. Например, с помощью системы REVEAL был рассмотрен ряд прикладных задач, в которых имена команд были переведены на иностранный язык, причем необходимости в модификации самой системы REVEAL при этом не возникло.

Можно создать интерпретаторы для команд, допускающие вложенность и рекурсию. Хотя в контексте просто организованной системы поддержки

процесса принятия решений редко возникает необходимость в рекурсии, последняя может оказаться существенной при построении машины логического вывода в приложениях, нуждающихся в методах инженерии знаний. Таким образом, с учетом этой возможности интерпретатор команд системы REVEAL реализует достаточно развитый самостоятельный язык программирования. Команды позволяют обратиться к любому языковому средству системы REVEAL, включая уровень программирования этой системы.

## 9.2. НЕЧЕТКИЕ МНОЖЕСТВА В СИСТЕМЕ REVEAL

Кроме обычных элементарных типов данных, таких как действительное число, скаляр и вектор, целое число, булева переменная, символ и строка, система распознает еще тип данных — *нечеткое множество*, что сделано по двум причинам. С одной стороны, благодаря этому расширяются возможности языка моделирования, с другой стороны, этот тип данных лежит в основе формальной схемы, используемой в системе REVEAL для представления знаний. Но сначала остановимся на том, что такое нечеткое множество.

В общепринятом смысле множество задается своими элементами. Так, мы имеем множество всех наручных часов или множество всех людей, рост которых превышает 1 м 80 см. Членство в некотором множестве определяется в соответствии с двузначной логикой, т.е. *истинно* или *ложно*. Так, всякий предмет либо представляет собой наручные часы, либо нет, а рост всякого человека либо превышает 1 м 80 см, либо нет.

Однако имеется много других возможных видов логики, а именно трехзначная логика, многозначная логика, бесконечнозначная логика и т.д. Один класс бесконечнозначных логик получил название *нечеткая логика*. В нем допускается возможность того, чтобы истинность любого высказывания принимала любое значение в интервале от истинно до ложно; значения концов этого интервала обычно полагаются (надо признать, произвольно) равными соответственно 0 и 1. Множества, соответствующие такой логике в смысле определения принадлежности к ним, задаются уже не столь жестко, как в случае классических множеств. Теперь допускается, что объекты могут лишь частично принадлежать некоторому множеству. Поэтому такие объекты получили название нечетких множеств.

Хотя указания на понятия, аналогичные тому, что теперь мы понимаем под нечеткими множествами, можно найти в работе Пуанкаре столетней давности, основателем современного подхода к нечетким множествам является профессор Л. Заде из Калифорнийского университета (Беркли, США). Он ввел понятие нечеткости в своей основополагающей работе, опубликованной в 1965 г. Его интерес в то время был обусловлен областью систем управления, где его беспокойство вызывало то, что чисто математический под-

ход к управлению исключал возможность использования опыта и знаний, накопленных инженерами на практике. Это беспокойство выразилось в формулировке Заде принципа несовместимости:

По мере роста сложности системы постепенно падает наша способность делать точные и в то же время значащие утверждения относительно ее поведения, пока не будет достигнут порог, за которым точность и значимость (или релевантность) становятся почти что взаимоисключающими характеристиками.

В сущности, Заде указывал на то, что человеческие существа способны упоминать или обсуждать многие важные вопросы, не прибегая даже к отдаленным попыткам придать им математическую строгость. На самом деле, основной тезис Заде состоит в том, что познание большей частью связано с присваиванием меток более или менее нечетким понятием и построением последующих рассуждений с использованием этих нечетких объектов. Таким образом, существует естественная связь между нечеткими множествами, описываемыми как математические объекты, и лингвистическими переменными, регулярно используемыми людьми на этапах познания и обсуждения.

Там, где раньше мы могли оперировать понятием множества людей ростом выше 1 м 80 см, используя лингвистические переменные, теперь можно оперировать понятием множества высоких людей. У этого множества нет жесткой границы, но мы все очень хорошо знаем, что означает слово "высокий", и в состоянии обсуждать такие множества, прибегая к ним в разговоре, неявно подразумевая, что при определении высокого человека в любом контексте присутствует элемент субъективности.

На рис. 9.5 представлен пример нечеткого множества, построенного системой REVEAL. Это множество описывает понятие "высокий", как оно представляется в данный момент лицу, определяющему это понятие. Таким образом, механизм создания и выдачи элементарного данного "высокий", представляемого в виде нечеткого множества, весьма прост.

В приведенном примере нечеткое множество задавалось в некоторой области определения, нижняя и верхняя границы которой равны соответственно 1 м 20 см и 2 м 20 см. Эта область представляет собой тот диапазон, в котором мы можем ожидать обсуждения роста взрослых людей. Все нечеткие множества определяются на такого рода области. В нашем случае характеристическая функция нечеткого множества "высокий" была получена с помощью встроенной функции РОСТ, для которой нужно задать три параметра, такие как значение в области определения с нулевой степенью принадлежности этому множеству, значение в точке, в которой степень принадлежности равна 50%, и значение, начиная с которого принадлежность — стопроцентная.

Имеется целый набор столь же простых функций, пригодных для задания нечетких множеств. Если в стандартном наборе отсутствует некоторая необходимая функция, то пользователь может написать ее, пользуясь языком системы REVEAL, и затем применить для построения нужных ему нечетких множеств. А если же нечеткое множество нельзя представить в виде функции, то его характеристическую функцию (степень принадлежности) можно задать в числовой форме.

Возникает естественный вопрос: а кто определяет эти нечеткие множества и в какой степени можно на них полагаться? Нечеткие множества представляют собой данные. Любое моделирование на вычислительной машине опирается на введенные данные. (А использование экспертных систем, безусловно, является моделированием, хотя такая их характеристика редко встречается в литературе.) Таким образом, нечеткие множества представляют собой конкретный способ внесения субъективной компоненты информации, поставляемой создателями системы. Они составляют четко очерченную форму восприятия, образующую среду взаимного контакта между кооперирующими создателями и пользователями системы. Одной из общепризнанных трудностей при конструировании любой системы, использующей знания, является проблема работы со знанием, получаемым из различных источников. Словарь определителей (*нечетких множеств*), надстраиваемый над контекстом решаемой задачи в силу архитектуры, присущей системе REVEAL, позволяет легко и результативно сравнивать вклад различных суждений в решение задачи.

Нечеткие множества играют роль прилагательных, т.е. определителей, в естественном языке. В теории нечетких множеств и в системе REVEAL наречиями, т.е. определителям определителей, таким как "очень", "весьма", "ниже", "около" и т.д., отводится большое место. Функциональные операторы над нечеткими множествами, представляющие такие определители второго уровня (известные в литературе как "перегородки" <hedges>), доступны как в виде определенных заготовок, так и могут определяться самим пользователем. Таким образом, определители создаются из слов, уже имеющихся в словаре. Например, если определитель "низкий" уже имеется, то определитель "умеренный", означающий "весьма низкий, но не очень низкий", можно задавать, используя в точности указанную цепочку слов.

Для работы с нечеткими множествами в системе имеется полное исчисление нечеткой логики. Теоретико-множественные операции взятия отрицания, объединения, пересечения и импликации удастся определить согласованным образом, причем так, что все классические теоретико-множественные тождества, такие как законы Моргана, кроме закона исключения третьего, переносятся на это исчисление. В частном случае применения исчисления нечеткой логики при нулевой степени нечеткости можно убедиться, что эти операции сводятся к классическим операциям теории множеств.

В системе REVEAL нечеткие множества хранятся в специальном поименованном файле, называемом словарем. Любой словарь может быть ассоциирован с контекстом текущей задачи так, что его элементы могут использоваться наряду с другими средствами программирования и управления. В качестве простого примера рассмотрим рис. 9.6. Здесь имеется некоторое отношение, содержащее данные, касающиеся группы промышленных фирм. Такие атрибуты, как отраслевая классификация, объем продаж, доход и число работающих, хранящиеся в памяти, дают возможность как в системе REVEAL, так и во многих других системах извлечь некоторые фактические отношения, соответствующие, например, такому предикату, как "объем про-



РЕЖИМ >определить высокий

```
.. НИЖНЯЯ ГРАНИЦА : >1,20
.. ВЕРХНЯЯ ГРАНИЦА : >2,20
.. ВЗЯТЬ ФУНКЦИЮ : >рост(1,60, 1,80, 2,00)
.. ПОСТРОЕНО
```

РЕЖИМ >начертить высокий

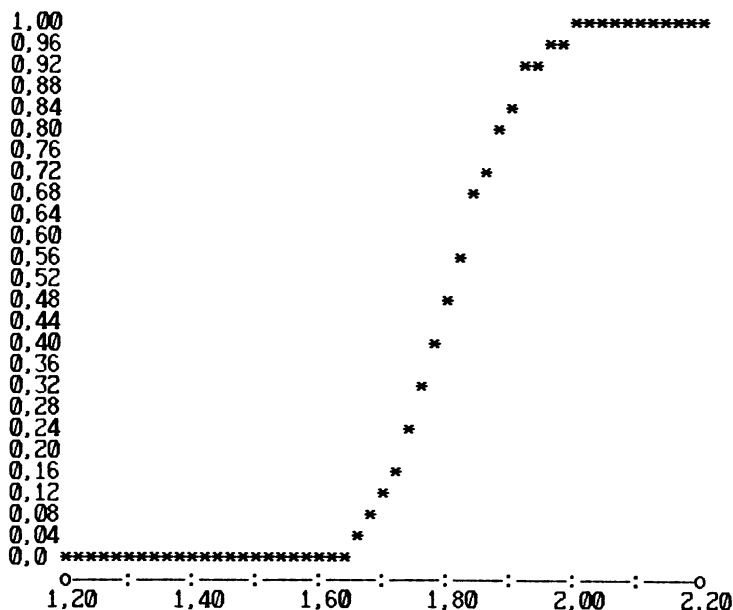


Рис. 9.5. Построение нечеткого множества

дажи больше 800". В этом случае из нашей небольшой базы данных будет извлечено множество фирм, удовлетворяющих этому требованию. Но в системе REVEAL, кроме того, имеется возможность создать отношение, соответствующее нечеткому предикату, как показано на рисунке. При этом из базы данных извлекается нечеткое множество, в котором для каждого случая будет указана степень истинности в соответствии с избранным нечетким предикатом. Это означает, что запрос можно сформулировать так, чтобы он лучше соответствовал изначальной цели запроса (ибо заведомо пользователь не имеет в виду то, что 800 будет жесткой границей), а кроме того, он получает из базы данных более осмысленную информацию.

Поскольку нечеткое множество понимается как такое отображение объекта познания, при котором создается информация, допускающая обработку на вычислительной машине, то мы имеем дело со знанием. Теория нечетких множеств предоставляет нам метод отражения существенно субъ-

РЕЖИМ >отношение

	A	B	C	D	E	F	G	H	I
КЛАССИФИКАЦИЯ	6	6	12	6	12	6	12	12	6
СБЫТ	200	1100	1000	800	500	1200	1300	100	400
ЗАНЯТО. ТЫСЯЧ	3	10	11	9	8	14	12	2	6
ДОХОД	40	110	160	120	55	115	200	20	10

РЕЖИМ >отношение : сбыт больше 800

	B	C	F	G
КЛАССИФИКАЦИЯ	6	12	6	12
СБЫТ	1100	1000	1200	1300
ЗАНЯТО. ТЫСЯЧ	10	11	14	12
ДОХОД	110	160	115	200

РЕЖИМ >отношение : сбыт высокий или доход/сбыт приемлемо

	F	B	C	D	E	F	G	H
КЛАССИФИКАЦИЯ	6	6	12	6	12	6	12	12
СБЫТ	200	1100	1000	800	500	1200	1300	100
ЗАНЯТО. ТЫСЯЧ	3	10	11	9	8	14	12	2
ДОХОД	40	110	160	120	55	115	200	20
ИСТИННОСТЬ	13	49	27	4	1	72	87	13

РЕЖИМ >отношение : сбыт высокий и доход/сбыт приемлемо

	C	D	G
КЛАССИФИКАЦИЯ	12	6	12
СБЫТ	1000	800	1300
ЗАНЯТО. ТЫСЯЧ	11	9	12
ДОХОД	160	120	200
ИСТИННОСТЬ	5	3	4

Рис. 9.6. Нечеткие отношения

ективной, расплывчатой и неоднозначной природы человеческого познания и человеческого рассуждения. Последнее соображение естественно подводит нас к возможности использования системы REVEAL в технологии применения знаний или инженерии знаний.

### 9.3. СИСТЕМА REVEAL И ИНЖЕНЕРИЯ ЗНАНИЙ

Представляется, что всякая экспертная система имеет по меньшей мере две компоненты — базу знаний и логическую машину. Во многих системах, созданных с использованием языков искусственного интеллекта, например с применением языка ЛИСП, это отличие имеется главным образом на концептуальном уровне. Все написано в кодах языка-источника, и для нетренированного глаза нет заметного отличия между знанием и управлением. Это различие подчеркивается в самой архитектуре системы REVEAL.

Для представления знаний применялись многие типы формальных схем. Среди них — процедурные представления, логика предикатов, байесовские

и семантические сети и в особенности представления с использованием продукционных правил.

Исходным механизмом, предусмотренным в системе REVEAL, являются нечеткие правила продукционного типа. Имеется понятие "политика". Политика – это совокупность нулевого или большего числа продукционных правил, хранящаяся в базе данных как некоторый поименованный файловый объект. Такие политики могут применяться в контексте текущей задачи под контролем пользователя или программы. Именно эта структура программного управления и образует логическую машину. Таким образом, имеется вполне определенное различие между знанием и управлением. Кроме того, использование таких совокупностей правил осуществляется средствами стандартного языка программирования системы REVEAL. Благодаря этому правила могут применяться в любом необходимом месте наряду с другими числовыми и нечисловыми программами, которые диктуются решаемой задачей.

Рассмотрим пример, показанный на рис. 9.7. Здесь приведены данные, представляющие результаты, полученные некоторой группой кандидатов, в ходе экзаменов по определенным предметам. Здесь наша цель – привести небольшую задачу, имитирующую работу специалиста по профессиям. Такой специалист по результатам экзаменов строит выводы относительно пригодности кандидатов для работы в различных вакантных областях.

На рисунке 9.7 показано также, как такая программа выполняется. На рис. 9.8 приведена таблица данных, показывающих силу различных рекомендаций, которые делаются, а также то множество нечетких правил, которые составляют знания этого специалиста по выбору профессий. На рис. 9.9 приведен пример совокупности нечетких множеств, представляющих понятия "хороший", "слабый", "блестящий" и т.д., использованные в приведенных правилах.

Следует остановиться на двух обстоятельствах. Во-первых, к множеству использованных правил имеется непосредственный доступ через редактор системы REVEAL, поэтому конечный пользователь, в данном случае гипотетический консультант по выбору профессий, имеет возможность непосредственно модифицировать или пополнять это множество. Во-вторых, разумеется, язык, используемый для представления правил, является формальным языком, элементы которого близки элементам языка Фортран или Бейсик. Но, по крайней мере внешне, он близок к формату естественного языка и конструировался так, чтобы быть естественным в употреблении. Системный редактор автоматически проверяет все новые и модифицируемые правила с тем, чтобы были выполнены все необходимые требования, характерные для языка продукции, так что система не воспринимает правила, которые не поддаются дальнейшей обработке.

Применение данного множества правил происходит по команде, содержащейся в программе системы REVEAL, и соответствующая управляющая программа для нашего примера приводится на рис. 9.10. В каком-то смысле этот пример задуман как иллюстрация определенной парадигмы. Три компо-

РЕЖИМ >вход. данные

	СМИТ	БРАУН	ДЕЙВИС	АРНОЛЬД	ДЖОНСОН	F
ФИЗИКА	0,90	0,60	0,53	0,84	0,80	
ХИМИЯ	0,70	0,70	0,64	0,58	0,79	
МАТЕМ	0,95	0,65	0,81	0,64	0,80	
БИОЛОГИЯ	0,60	0,40	0,76	0,88	0,72	
АНГЛИЙСКИЙ	0,10	0,60	0,63	0,25	0,39	
ФРАНЦ	0,05	0,10	0,29	0,71	0,60	
ЭКОНОМИКА	0,75	0,20	0,82	0,74	0,78	
ИСТОРИЯ	0,45	0,80	0,79	0,83	0,72	
ГЕОГРАФИЯ	0,30	0,70	0,75	0,68	0,59	
ИСКУССТВА	0,60	0,05	0,20	0,75	0,40	

РЕЖИМ >выполнение

---

..Имя кандидата : СМИТ

..Рекомендуемая специальность : ИССЛЕДОВАНИЯ

---

..Имя кандидата : БРАУН

..Рекомендуемая специальность : ПРЕПОДАВАНИЕ

..Альтернативная специальность : АДМИНИСТРИРОВАНИЕ

---

..Имя кандидата : ДЕЙВИС

..Рекомендуемая специальность : ПРЕПОДАВАНИЕ

..Альтернативная специальность : СЧЕТОВОДСТВО

---

..Имя кандидата : АРНОЛЬД

..Рекомендуемая специальность : МЕДИЦИНА

..Альтернативная специальность : АДМИНИСТРИРОВАНИЕ

---

..Имя кандидата : ДЖОНСОН

..Рекомендуемая специальность : ИССЛЕДОВАНИЯ

..Альтернативная специальность : СЧЕТОВОДСТВО

---

..Имя кандидата : РИЧАРДС

..Рекомендуемая специальность : СЧЕТОВОДСТВО

..Альтернативная специальность : ИССЛЕДОВАНИЯ

Рис. 9.7. Экспертные системы

	СМИТ	БРАУН	ДЕЙВИС	АРНОЛЬД	ДЖОНСОН	РИЧАРДС
ИССЛЕДОВАНИЯ	81				61	63
ЛИТЕРАТУРА						
АДМИНИСТРИРОВАНИЕ				65		
ПРЕПОДАВАНИЕ		85	65	55		
ПОЛИТИКА				65		
МЕДИЦИНА						
ЮРИСПРУДЕНЦИЯ						
СЧЕТОВОДСТВО			63		57	71
АРХИТЕКТУРА						61

- 1 : ?Принцип отбора специальности
- 2 : ?=====
- 3 :
- 4 : Если матем блестяще и физика блестяще  
то рекомендуются исследования
- 5 :
- 6 : Если матем меньше средне или физика меньше средне  
то исследования не рекомендуются
- 7 :
- 8 :
- 9 : Если искусства больше хорошо и матем больше средне  
то рекомендуется архитектура
- 10 :
- 11 : Если матем больше средне и экономика больше хорошо  
то рекомендуется бухгалтерия
- 12 :
- 13 : Если биология больше хорошо и худш.точн больше средне  
то рекомендуется медицина
- 14 :
- 15 : Если средн.гум меньше хорошо и средн.точн меньше хорошо  
то рекомендуется преподавание
- 16 :
- 17 :
- 18 : Если английский больше хорошо и матем больше средне  
и худш.точн больше слабо то рекомендуется юриспруденция
- 19 :
- 20 : Если английский блестяще и худш.гум больше слабо  
то рекомендуется литература
- 21 :
- 22 :
- 23 : Если английский хорошо и экономика хорошо и матем меньше  
средне то рекомендуется политика
- 24 :
- 25 :
- 26 : Если худш.гум хорошо или худш.точн хорошо и матем средне  
то рекомендуется администрирование

Рис. 9.8. Представление знаний

РЕЖИМ >начертить плохо, слабо, хорошо, блестяще, средне

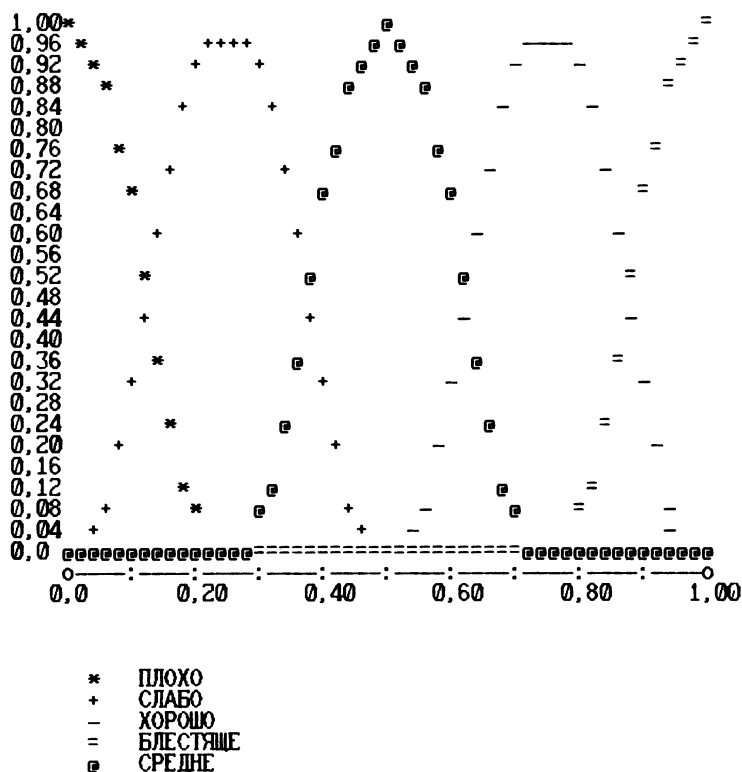


Рис. 9.9. Нечеткие множества

ненты, на которые распадается программа, иллюстрируют то обстоятельство, что как до применения знаний к задаче, так и после их использования может возникнуть необходимость в числовой обработке входных данных (их проверка, преобразование), в пересылке данных в файлы и обратно, выдаче на печать и т. д.

В декларативном языке системы REVEAL предусмотрено несколько других форматов для правил. Можно делать безусловные утверждения. Значение некоторой контекстной переменной может определяться несколькими, возможно, конфликтующими правилами. В этих случаях система выделяет в ходе своей работы наиболее подходящую выходную величину, рассматривая утверждения как нечеткие цели или ограничения в некоторой нечеткой задаче оптимизации. Часто встречаются, конечно, и четкие правила,

```

1 : !Программа: консультант по выбору профессии
2 : !
3 :
4 : !Программа распадается на три части; для каждого кандидата:
5 :
6 : !     1. Подсчитывается наихудшая, наилучшая и средняя оценки
7 : !     по точным и гуманитарным наукам
8 : !     2. Применяется принятый принцип отбора
9 : !     3. Составляется простой отчет
10 :
11 : !Вычисляются средние и другие величины
12 :
13 : лучш.точн=max(max(max(матем,физика),химия),биология)
14 : худш.точн=min(min(min(матем,физика),химия),биология)
15 : средн.точн=среднее(физика,биология)
16 :
17 : лучш.гум=max(max(max(английский,франц),экономика),история)
18 : худш.гум=min(min(min(английский,франц),экономика),история)
19 : средн.гум=среднее(английский,экономика,искусства)
20 :
21 : ! Теперь применим принцип отбора к исходным и вычисленным данным
22 :
23 :     применить('ОТБОР')
24 : ! Дать итоговый отчет по двум лучшим кандидатам
25 :
26 :     печать(30* ' - ')
27 :     печать('')
28 :
29 :     печать('...Имя кандидата': ' ',тимя(шаг))
30 :     печать('')
31 :
32 :     лучш=0; след=0; лучшпр=0; следпр=0
33 :
34 :     повторять индекс=(51,59)
35 :     если серия(индекс,шаг) больше.равно лучш то
36 :         выполнять
37 :             след=лучш
38 :             лучш=серия(индекс,шаг)
39 :             следпр=лучшпр
40 :             лучшпр=индекс
41 :         закончить
42 :     иначе
43 :         выполнять
44 :             если серия(индекс,шаг) больше след то
45 :                 выполнять
46 :                     след=серия(индекс,шаг)
47 :                     следпр=индекс
48 :                 закончить
49 :             закончить
50 :     стопповтор
51 :
52 :     печать('...Рекомендуемая специальность : ',фамил(лучшпр))
53 :     если следпр больше 0 то
54 :         печать('...Альтернативная специальность : ',фамил(следпр))
55 :     печать('')
56 :     печать('')

```

Рис. 9.10. Управляющая программа

которые вполне допустимы. Такие правила, как "Если функция есть "торговля" или если стаж работы продолжительный, то зарплата высокая", содержат как нечеткие, так и жестко заданные элементы.

В приведенном выше примере все правила применялись ровно по одному разу. Это и есть, если хотите, машина вывода — она настолько проста, что не заслуживает такого громкого названия. Обычно знания обладают определенной структурой, и необходимо иметь формальный метод доступа к правилам в последовательности, зависящей от контекста. Конструктор (специалист по использованию знаний) располагает всеми возможностями для проектирования и реализации любой желаемой стратегии вывода в языке системы REVEAL, т. е. в обмен на работу по подготовке кода он получает в свое распоряжение абсолютно гибкую систему.

С другой стороны, в нашем языке имеется набор функций, предназначенных для обработки множества правил в соответствии со многими стандартными подходами к созданию машин логического вывода. Так, например, некоторая совокупность правил (политика) может выполняться как древовидная структура. В результате рабочая система будет занята либо поиском свидетельств в пользу некоторой априорной гипотезы, либо проверкой всех возможных гипотез, обеспечивающих заданное значение какой-то переменной, причем на выход будут поступать все правдоподобные результаты с указанием степени их подтверждения.

#### 9.4. ОБЗОР СИСТЕМЫ REVEAL

Система REVEAL представляет собой объемный программный продукт, доступный для продажи и предназначенный для использования методов инженерии знаний в проблемных средах, в которых необходимо, кроме того, обеспечить поддержку принятия решений. Она нацелена на применение в самых разнообразных приложениях и самыми разнообразными пользователями — от новичка до опытного системотехника. В системе REVEAL нашло воплощение существенное соображение о том, что имеется множество применений, где привнесение в существующую моделирующую программу некоторого небольшого объема знаний и использование их вместе с более привычными средствами может решительно повысить качество моделирования. Совсем не обязательно мыслить исключительно в терминах экспертных систем, работающих лишь на основе знаний, для чего используются сотни или тысячи правил, хотя нет сомнения в важности такого рода приложений, и они включены в круг задач, для которых создана система REVEAL. Дело в том, что область приложения стандартных методов моделирования с помощью вычислительных машин может быть значительно расширена благодаря использованию небольших баз знаний, содержащих около



десятка правил. Такие небольшие объемы позволяют вывести вопрос создания систем из сферы, в которой заняты высокопрофессиональные и весьма искусные специалисты по технологии использования знаний, поскольку ограниченная сложность позволяет сохранять контроль над созданием и использованием системы.

Система REVEAL реализована на широком наборе вычислительных устройств, включая такие ЭВМ, как IBM, ICL, VAX и IBM PC/XT. Предусмотрена связь между микромашиной и главной машиной; поэтому вполне возможно применение системы в приложениях, требующих распределения вычислительных ресурсов, в особенности благодаря тому, что система идентична в каждой своей реализации.

Исследования по искусственному интеллекту развивались со времен создания вычислительных машин на электронных лампах, получивших название "электронного мозга". В течение многих лет плоды таких исследований вкушали лишь ученые академических учреждений, в которых эти работы процветали. Даже сейчас налицо определенная дистанция между технологией использования знаний и основным потоком исследования операций и управления системами. Одно из важнейших предназначений системы REVEAL — уменьшить эту дистанцию, включив науку использования знаний в число рутинных приемов, которыми располагают те люди, которые моделируют на машине реальные задачи.

*Ричард Форсайт*

Грандиозные успехи экспертных систем вызвали новую волну интереса к знаниям – тому “горючему”, на котором такие системы работают. Именно знания специалистов придают силу экспертным системам. Но задача извлечения знаний из людей-экспертов является весьма трудоемкой [11], а поэтому область машинного обучения переживает новый подъем.

#### 10.1. ЧТО ТАКОЕ ОБУЧЕНИЕ МАШИНЫ?

Под обучением машины я понимаю любое улучшение работы вычислительной системы, являющееся результатом накопления опыта. Обучающиеся алгоритмы нацелены на достижение одного или нескольких свойств:

1. Расширить круг решаемых задач.
2. Выдавать более точные решения.
3. Получать ответы с меньшими затратами.
4. Упростить уже имеющиеся знания.

Включение в этот список последнего пункта говорит о том, что упрощение имеющихся знаний имеет самостоятельную ценность, так как выраженные яснее знания становятся более доступными для людей, хотя это и не приводит к улучшению рабочих характеристик компьютерной системы.

Обучение машины может быть использовано почти в любой области, но на практике наиболее успешно работающие системы созданы для классификационных задач, на которых мы и остановимся. Даже в этом подмножестве задач можно найти широкий набор важных применений, например:

1. При заданных сведениях о погоде за последние семь дней построить предсказание того, что в течение 48 ч в Северном ледовитом океане будет шторм.
2. При известных картах игрока в покер и записи о выдвигениях ставок решить, следует ли выйти из игры, или потребовать открыть карты, или поднять ставку.
3. По найденному при раскопках окаменевшему осколку челюсти австралопитека установить, принадлежит он женщине или мужчине.
4. При заданных результатах сейсморазведки выдать рекомендацию нефтедобывающей компании начать бурение или перебазироваться на другое место.

## 10.2. ПАРАДИГМА ОБУЧЕНИЯ

Всякая система, созданная с таким расчетом, чтобы она могла модифицировать и улучшать свою работу, должна содержать следующие четыре главные компоненты:

1. Множество информационных структур, в которых кодируется текущий уровень экспертности системы ("правила").
2. Алгоритм задачи ("исполнитель"), который использует эти правила для управления активностью системы.
3. Модуль обратной связи ("критик"), который сопоставляет достигнутые результаты с желаемыми.
4. Обучающийся механизм ("ученик"), который использует обратную связь, поступающую от критика для усовершенствования правил.

Эти главные компоненты схематически представлены на рис. 10.1. В большинстве экспертных систем используется база знаний, которая допускает лишь чтение. Здесь идет речь о стираемой и программируемой базе знаний.

Чтобы показать наши идеи на конкретном и знакомом примере, в остальной части главы будет рассмотрена простая задача формирования понятий. Мы будем считать, что компьютеру предъявляются сданные на руки игроку совокупности по четыре игральные карты, например Дама Червей (ДЧ), Десятка бубновая (ХБ), Валет Крестей (ВК) и Двойка Пик (2П), вместе с указанием от учителя, относится ли данная комбинация к классу И, т. е. истина, или к классу Л, т. е. ложь. Задачей компьютера является выработка правила (или правил) для различения случаев И и Л.

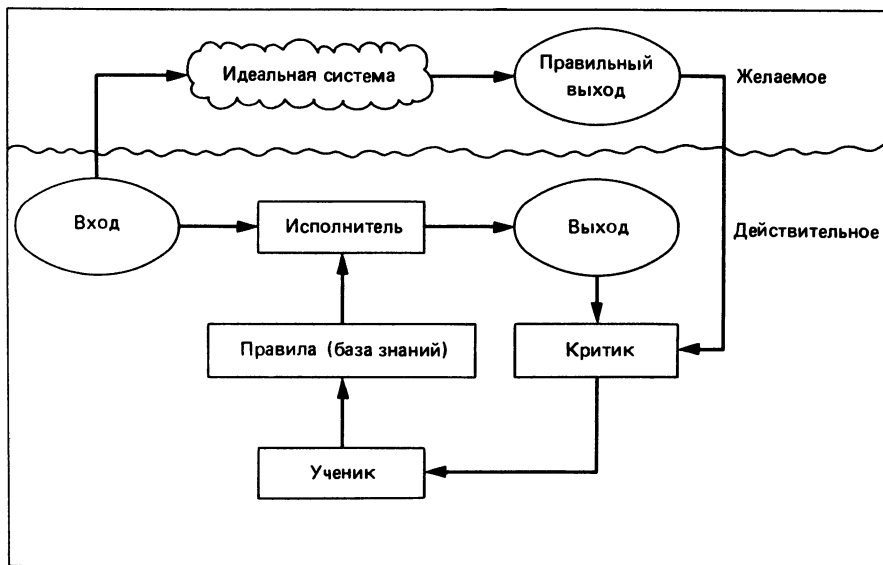


Рис. 10.1. Компоненты системы, улучшающей свое поведение в результате обучения

### 10.3. ЯЗЫК ОПИСАНИЯ

Ключевым моментом для успеха или неудачи в задаче такого рода является выбор языка описания.

Правила различения должны быть выражены в какой-то нотации или структуре данных, пригодной для описания входной информации, определяющей правильный отклик системы. Это положение очевидно, но можно сколько угодно стараться заставить компьютер выучить некоторое понятие, для которого у него не предусмотрено средств представления. Компьютер является последовательным прагматиком: несущественное для него различие вообще не является различием.

Вернемся к нашей игре в карты. Предположим, что компьютеру сообщена следующая последовательность комбинаций карт (Т – туз):

ТЧ, ВБ, ВК, 2П: И

2Б, ВЧ, ТЧ, ДП: Л

8К, ХП, 6Ч, ТК: Л

3Ч, 9Б, 8Б, 7Б: Л

5Ч, 3К, ХЧ, 9Ч: И

ТБ, 3Б, ТЧ, 7Ч: Л

ВБ, ХБ, 7П, 2Б: Л

Какое же правило различения ведет к И или Л? (Попробуйте сами найти это правило, не заглядывая в ответ!)

Если наши правила оформляются подобно поисковым цепочкам символов, используемым в некоторых машинных редакторах текста, типов ?Б, 3?, 3?, ??, где знаком вопроса обозначена некая произвольная карта, стоящая на месте ранга карты или ее масти, тогда приведенное правило представляет собой такое понятие: любая бубновая, какая-то тройка, еще какая-то тройка и любая другая карта. В настоящем текстовом редакторе порядок в этой последовательности был бы существенным, так что карты пришлось бы рассматривать именно в указанном порядке.

Однако от этого ограничения можно избавиться, так что такой "образец" будет сопоставим и с 3К, 3Ч, ВБ, ТП, и с ВБ, 3К, 3Ч, ТП, что несколько расширяет возможности нашего языка.

Но это расширение возможностей не настолько велико, чтобы прийти к правилу, которое я имею на самом деле в виду, а именно:

ЕСЛИ нет ни одной красной карты ТО Л;

ИНАЧЕ ЕСЛИ нет ни одной черной карты ТО Л;

ИНАЧЕ ЕСЛИ самая младшая красная карта выше по рангу самой младшей черной ТО И;

ИНАЧЕ Л,

которое, несомненно, уже пришло вам в голову.

Если вы думаете, что на обычной машине этого легко добиться, то попытайтесь обобщить, скажем, правило относительно двух карт, например

?Ч, ?Ч, ??, ??

так, чтобы охватить понятие пары, т. е. любых двух карт одного достоинства. В таком языке даже столь простое понятие невозможно выразить с помощью одного "образца". Их потребуется несколько, например

2?, 2?, ??, ??

3?, 3?, ??, ??

4?, 4?, ??, ??

и т. д., что чрезвычайно неудобно.

Таким образом, перед началом построения обучающейся системы крайне важно обеспечить такой язык описания правил, который в состоянии выразить необходимые понятия, что совсем не так просто.

#### 10.4. ВЕКТОР ПРИЗНАКОВ

Один из изящных приемов состоит в том, чтобы для входной информации использовать то же представление, что и для правил. Если это удастся, то очень хорошо, так как не возникает необходимости в преобразовании формата данных в формат правил. Один особенно простой формат стал очень популярным в первых подходах к статистическому распознаванию образов.

Если ваши данные представляют собой одномерный массив чисел, каждое из которых является результатом измерения значения какой-то одной характеристики входного объекта  $a[i]$ ,  $i=1, \dots, N$ , например

1, 7, 2, 0, -8, -19, 77, 4,  $\dots$ ,

то этот вектор может быть поэлементно умножен на веса  $w[i]$ ,  $i=1, \dots, N$ , например

0, -1, -2, 0, 0, 1, 1, 0,  $\dots$ ,

а сумма полученных попарных произведений сопоставлена с порогом (как правило, равным нулю). Если эта сумма превышает порог, то машина классифицирует входную величину как положительный пример, в противном случае считает ее отрицательным примером. В этом состоит суть системы PERCEPTON. Алгоритм обучения этой системы действует путем перестройки весов после неверных реакций на вход:

ЕСЛИ сумма произведений больше порога и ответ должен был быть отрицательным,

ТО вычесть значения  $a[i]$  из соответствующих значений  $w[i]$ ;

ЕСЛИ эта сумма меньше порога и ответ должен был быть положительным,

ТО прибавить  $a[i]$  к  $w[i]$ ;

ЕСЛИ ответ был верным, то оставить все как было.

Вначале эти веса могут быть выбраны вполне произвольно, а затем они сойдутся к некоторому почти оптимальному множеству.

На самом деле, весовые коэффициенты  $w[i]$  выражают знания системы: они указывают на важность каждого входного признака для такой задачи различения. Этот алгоритм изящен, прост и является эффективным, если

только два класса "линейно разделимы", а обучающая последовательность не слишком зашумлена. Он допускает также обобщение на случай более двух классов.

К сожалению, многие из реальных задач классификации не являются линейно разделимыми. На практике почти невозможно найти множество признаков, которые между собой были бы абсолютно независимы. Подход с использованием вектора признаков не срабатывает уже при решении очевидно тривиальной задачи (узнавания ситуации) "червовая карта слева от карты черной масти".

#### 10.5. БОЛЕЕ СИЛЬНЫЕ ЯЗЫКИ

При структурном обучении, когда отношения между элементами важны так же, как и признаки этих элементов, исключается использование векторов признаков в их исходной форме.

В большинстве систем структурного обучения используется какая-либо разновидность исчисления предикатов [7]. Язык, предназначенный для нашей задачи с картами, нуждается в таких функциях, как

слева-от (a, b)	справа-от (a, b)
достоинство (a)	масть (b)

а также операторов сравнения и логических операторов

И, ИЛИ, НЕ

$<, =, >, <>, <=, >=$

но даже и этого может не хватить. Например, мы могли бы сказать\*

достоинство (C1) = достоинство (C2) ИЛИ

достоинство (C1) = достоинство (C3) ИЛИ

достоинство (C1) = достоинство (C4) ИЛИ

достоинство (C2) = достоинство (C3) ИЛИ

достоинство (C2) = достоинство (C4) ИЛИ

достоинство (C3) = достоинство (C4),

чтобы выразить понятие пары, но как можно было бы закодировать "имеется бубновая карта меньшего достоинства, чем любая червовая или крести".

Выбор компактного, выразительного обозначения является отнюдь не легким делом. И не забывайте: игра в карты — это всего лишь "игрушечная" проблемная область.

#### 10.6. ОБУЧЕНИЕ ПОСРЕДСТВОМ ПОИСКА

Предположив, что мы в состоянии решить нетривиальную проблему разработки хорошего языка описания, мы сталкиваемся с вопросом о том, каким образом можно на этом языке строить нужные описания.

---

\* C1, C2, C3, C4 — соответственно первая, вторая, третья и четвертая карты, находящиеся у игрока. — *Прим. ред.*

Один из подходов состоит в том, чтобы организовать в пространстве всех возможных описаний поиск таких из них, которые являются и верными, и полезными. Однако число возможных описаний носит астрономический характер, и оно тем больше, чем более выразительным является язык описаний.

Ясно, что должен быть найден какой-то способ управления таким поиском, позволяющий избежать рассмотрения громадного числа потенциально возможных описаний (т. е. понятий), которые не имеют отношения к изучаемой в данный момент задаче. Было показано, что этого можно достигнуть многими методами, если использовать незашумленные обучающие последовательности (см. например, [5, 8]). Мы же ограничимся рассмотрением здесь четырех конкретных примеров систем, допускающих использование данных с шумами.

Во всех этих системах обучение правилам классификации происходит в результате рассмотрения цепочки правильно классифицированных обучающих примеров — обучающего множества. При этом предполагается, что такие правила впоследствии могут быть применены к новым данным такого же типа, которых не было в обучающем множестве. Правильная классификация лежит в основе многих систем искусственного интеллекта, таких как интеллектуальные управляющие устройства, системы диагностики, советчики, чувствительные устройства роботов и тому подобные системы.

### 10.7. СИСТЕМА ID3

Хотя система ID3\* [9] не является слишком устойчивой, если допустить использование зашумленных данных, но в принципе это ограничение перестает быть существенным, если отказаться в ней от обязательного поиска "совершенного" правила. Программа работает следующим образом:

1. Выбрать наугад некоторое подмножество (окно)  $W$  из множества всех обучающих примеров.
2. Применив алгоритм CLS, построить для текущего окна правило классификации.
3. Просмотреть полностью всю базу данных, а не только это окно, чтобы найти исключения для последнего из выработанных правил.
4. Если исключения обнаружены, то включить некоторые из таких примеров в окно и повторить шаги, начиная со второго; в противном случае прекратить работу и выдать полученное правило.

Алгоритм CLS [4] циклически разбивает обучающиеся примеры на классы в соответствии с переменной, имеющей наибольшую классифицирующую силу. Для этой цели Куинлан воспользовался теоретико-информационной мерой — энтропией. Каждое подмножество, выделяемое такой переменной с наибольшей классифицирующей силой, вновь разбивается на классы (пока в нем не будут содержаться данные лишь одного класса) с использованием следующей переменной с наибольшей классифицирующей способностью, и т. д. Разбиения заканчиваются, когда в подмножестве оказываются лишь однотипные данные. В ходе этого процесса образуется дерево решений.

---

\* ID — Interactive Dichotomizer, т. е. интерактивный дихотомайзер. — *Прим. ред.*

Деревья решений, полученные в алгоритме ID3, хорошо себя проявили при решении шахматных задач типа "королевская ладья против королевского коня". Система ID3 способна работать в условиях лишь небольшого шума, поэтому шахматы явились для нее столь удобной областью применения. При сильно зашумленных данных деревья решений начинают сильно ветвиться, чтобы хорошо описать обучающие последовательности, но при этом затрудняется их модификация при поступлении новых примеров. Другое ограничение, присущее системе ID3, состоит в том, что в ней используются векторы признаков для описания данных (хотя и не для описания правил).

Более серьезный недостаток системы обусловлен бедностью используемого языка описаний. Правила в ней представляют собой деревья решений конкретного вида, в которых каждая вершина является тестом, дающим, как правило, две ветви, в зависимости от результата. А чтобы общий объем поиска оставался в разумных пределах, необходимо чтобы каждый тест представлял собой просто сравнение переменной с константой, например "осадки > 10". Столь простые тесты, однако, исключают применение составных проверок любого типа, в которых использовались бы логические или арифметические операции. Невозможно даже производить сравнение одной переменной с другой, что является очень серьезным ограничением. Перечисленные ограничения системы взваливают на плечи пользователя задачу выработки очень эффективного множества переменных, описывающих рассматриваемую ситуацию, в которые уже включены все необходимые предварительные вычисления.

Система ID3 лежит в основе продукта, продаваемого в Великобритании под названием "Expert-Ease".

#### 10.8. СИСТЕМА INDUCE

Диттерих и Михальски разработали структурно обучающуюся программу, названную INDUCE, в которой используется метод, известный как лучевой поиск [1].

- (1) Пусть множество  $H$  содержит наугад выбранное подмножество размером  $W$  множества обучающих примеров.
- (2) Обобщить каждый пример/понятие в  $H$  в минимально возможной степени (т. е. проделать минимальные обобщения).
- (3) Отбросить неоправданные гипотезы в соответствии с их размером и работоспособностью, т. е. сохранить те гипотезы, которые просты и охватывают много примеров, отбросив сложные и охватывающие лишь несколько случаев. В результате остаются только лучшие правила для  $W$ .
- (4) Если какое-то описание в  $H$  охватывает все (или достаточное число) примеров, то выдать его на печать.

(Шаги со второго по четвертый повторяются до тех пор, пока либо  $H$  не станет пустым, либо достаточное число понятий не поступит на печать.) Этот метод оказывается достаточно помехоустойчивым и может быть использован в целом ряде задач. Он был создан как развитие оказавшейся весьма успешной программы, использованной при обучении диагностике заболеваний



бобовых культур, и представляет собой результат продолжающихся передовых исследований в области обучения машин, проводимых Михальски и его коллегами.

Эффективность системы INDUCE целиком определяется адекватностью языка описания ("аннотированное исчисление предикатов") и критерия качества работы, на основании которого принимается решение, какое правило следует сохранить, а какое отбросить. В общем случае этот критерий нельзя указать заранее. Пользователь должен его подбирать, поскольку в различных ситуациях по-разному складывается соотношение между сложностью и эффективностью правил.

Другим существенным для метода фактором является процедура обобщения. Вопрос этот непростой. Правило (или пример), такое как

масть (C1) = бубны И достоинство (C1) = 8 И

масть (C2) = крести И достоинство (C2) = дама

может быть обобщено несколькими способами:

- (1) И становится ИЛИ, например,  
масть (C1) = бубны И достоинство (C1) = 8 ИЛИ  
масть (C2) = крести и достоинство (C2) = дама.
- (2) Отбрасываются условия, например,  
масть (C1) = бубны И достоинство (C1) = 8 И  
масть (C2) = крести.
- (3) Расширяется область, например,  
масть (C1) = бубны И достоинство (C1) > 7 И  
масть (C2) = крести И достоинство (C2) = дама.
- (4) Ослабляется условие, например,  
цвет (C1) = красный И достоинство (C1) = 8 И  
цвет (C2) = черный И достоинство (C2) = дама

Каждый метод обобщения может быть применен несколькими способами. Например, И может быть заменено на ИЛИ в целом семью способами (каждое, каждая пара и все три). Михальски уменьшает число возможностей, допуская, что за один раз можно делать лишь одно изменение. И даже при этом число возможных одиночных изменений, которые нужно рассмотреть на каждом шаге в реальных задачах, очень велико, а если отбрасывается слишком много вариантов, то может случиться так, что оптимальное правило так никогда и не будет найдено.

Чтобы оставаться в рамках разумного, на практике необходимо привлекать и другие ограничения, например рассматривать только правила определенной синтаксической формы.

#### 10.9. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ХОЛЛАНДА

Поначалу теоретическое исследование проблемы адаптации, проведенное Холландом [3], специалистами по искусственному интеллекту никак не отмечалось, но в последнее время некоторые ученые (например,

Смит) воспользовались его идеями как основой для создания эффективных обучающихся систем [10]. Основным алгоритм характеризуют некоторые биологические аналогии, копирующие идеи Дарвина, Менделя и других исследователей, которые учили нас представлению об эволюции, проверявшей в течение 3600 млн. лет "полевых испытаний"!

1. Случайным образом создать начальную популяцию из  $M$  правил-структур.
2. Вычислить и занести в память для каждого правила показатель его работы. Если их среднее значение достаточно высокое, то остановить вычисления и выдать эти правила.
3. Для каждого правила подсчитать вероятность его выбора  $p=e/E$ , где  $e$  — индивидуальный показатель, а  $E$  — суммарный показатель для всех  $M$  правил.
4. Создать следующую популяцию правил в соответствии с вычисленной вероятностью выбора и применяя операторы из генетики.
5. Повторить начиная со второго шага.

Таким образом, среднее число "потомков" каждого правила в следующем поколении пропорционально измеренному успеху при решении стоящей перед системой задачи: выживают наиболее приспособленные.

Генетическими операторами, используемыми на шаге 4, являются кроссинговер, мутация и инверсия. Кроссинговер (или перекрест) — это операция, при которой, например, правила  $(A, B, C, D)$  и  $(a, b, c, d)$  могут в качестве потомков в следующем поколении дать  $(A, B, C, d)$  и  $(a, b, c, D)$ . Холланд настаивал на том, что его алгоритм и вытекающие из него другие алгоритмы чрезвычайно экономным образом просматривают пространство правил, и результаты Смита подтверждают на практике его предсказание. Основная трудность состоит в том, что для оператора кроссинговера требуется, чтобы правила были цепочками с равной длиной, содержащими не-зависящие от положения компоненты. Это затрудняет создание работоспособного языка описания, и, кроме того, чрезвычайно трудно создать такой язык, который был бы к тому же понятен и людям.

Мутации, которые были включены в алгоритм, чтобы обеспечить возможность создания произвольных описаний, состоят в совершении случайных ошибок при записи правила. (Эти методы более подробно описаны в гл. 11.)

Теперь обратимся к методу обучения, истоки которого также уходят к Дарвину, но в котором используется несколько менее ограниченный язык описания, основанный на булевой алгебре.

#### 10.10. СИСТЕМА BEAGLE

Система BEAGLE (Biological Evolutionary Algorithm Generating Logical Expressions — биологический эволюционный алгоритм генерации логических выражений) представляет собой пакет машинных программ для построения правил решения путем индукции на базе данных [2]. Таким об-

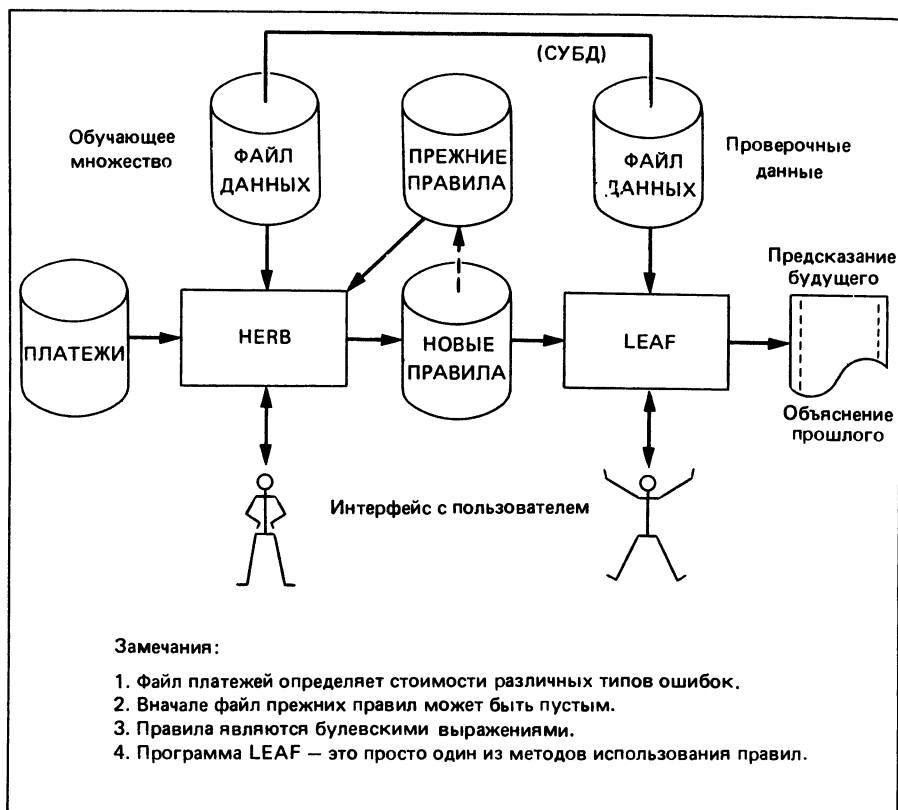


Рис. 10.2. Краткое описание системы BEAGLE

разом он нацелен на решение проблемы (которую часто забывают) — откуда появляются правила в системе базы правил.

Система BEAGLE работает по принципу "естественного отбора", когда правила, которые плохо соответствуют данным, отбрасываются с заменой мутациями более подходящих правил или с заменой новыми правилами, возникающими в результате объединения двух более подходящих правил. Сами правила являются булевыми выражениями, представляемыми посредством деревьев.

Система содержит две программы на языке Паскаль, а именно HERB (Heuristic Evolutionary Rule Breeder) — программу эвристического выращивания правил и LEAF (Logical Evaluation and Forecast) — программу логической оценки и предсказания, которые вместе выполняют задачу классификации выборок по двум и более классам на основе значений некоторого числа переменных или признаков. Программа HERB создает и модифицирует правила, которые в дальнейшем используются программой LEAF (рис. 10.2).

Программе HERB нужны три входных файла: файл данных, файл штрафов

Машинное решение		Действительный класс		Образец правила (61 МИНУС £14) МЕНЬШЕ – РАВНО £6) \$
		Выжили	Умерли	
- 1	Нет	-1	+ 1	Значение переменной £14 (мочевыделен. ...), вычитенное из 61, меньше или равно значе- нию переменной £6 (среднее артериальное давление) .
0	Возможно	0	0	
+ 1	Да	+1	-1	

Рис. 10.3. Пример платежной матрицы системы BI AGL1

и файл прежних правил (возможно, пустой). Она порождает файл новых правил, которые по крайней мере не хуже старых. Файл данных содержит обучающее множество, для которого известно распределение по классам. Пользователь должен также сообщить матрицу штрафов, которая определяет значение или стоимость каждой удачной и каждой неверной классификации (рис. 10.3).

Правило представляется в виде "скобочного" выражения, заканчивающегося знаком доллара, например

((£4 GE 20) OL (£7 LT £8)) \$

где утверждается, что переменная 4 (£4) должна быть больше или равна (GE) 20 или что переменная 7 должна быть меньше (LT) переменной 8, чтобы результат был истинен. ("Странные" имена, вроде OL, вместо OR, т. е. ИЛИ, были выбраны, чтобы избежать путаницы со стандартными символами языка Паскаль.)

Программа LEAF проще. Она воспринимает файл данных, записанных в том же формате, что и обучающее множество, и применяет к нему файл правил. Кроме всего прочего, эта программа способна расположить различные примеры в соответствии с "мнением" правила в порядке от наиболее вероятного "да" до наиболее вероятного "нет". Заметим, что использование правил, создаваемых программой HERB, не связано со значительными вычислениями, и их можно было бы проделать вручную. Программа LEAF просто удобнее.

Алгоритм обучения системы BEAGLE состоит из повторения следующего процесса на протяжении ряда поколений, где каждое поколение – это один прогон на обучающей выборке:

1. В соответствии с матрицей штрафов построить оценку для каждого правила на каждой выборке, давая дополнительные очки более коротким правилам.

2. Упорядочить правила по убыванию их "качества", а затем отбросить худшую половину правил.
3. Заменить "убитые" правила, применяя процедуру MATE к паре случайных выбранных правил из числа оставленных, комбинируя таким образом части хороших правил.
4. Подвергнуть мутации несколько наугад выбранных правил из числа созданных и применить процедуру TIDY ко всем новым правилам — новое поколение сформировано.

Процедура TIDY состоит в устранении синтаксической избыточности, такой как двойные отрицания, константные выражения и т. д., в результате чего остающееся дерево правил имеет то же значение, но выражается более аккуратно.

Система была опробована в медицинском приложении (классификация исходов сердечных заболеваний) и в спорте (классификация бегунов на короткие и длинные дистанции). При этом выяснилось, что данный метод работает лучше, чем стандартная методика дискриминантного анализа на основе линейных функций, и оказывается довольно устойчивым в отношении зашумленных данных. Кроме того, эта наивная неodarвинистская теория дала программу, которая оказалась жизнеспособной.

Регулярно наблюдались появление и последующее исчезновение доминантных "особей" правил. Каждая из них процветала какое-то время, а затем довольно внезапно вытеснялась новой, имеющей преимущество наследственной линией, являющейся обычно результатом мутации одного из потомков этой особи. Когда такое происходило, то исчезновение более примитивных форм было быстрым и полным. На самом деле ценность систем, подобных BEAGLE, может состоять в том, что они будут помогать биологам в отработке теорий и проверке их следствий.

В настоящее время я работаю над более удобной в пользовании коммерческой версией системы BEAGLE, которую можно будет "пристроить" к базе данных. Цель состоит в создании базы данных, способной обучаться.

#### 10.11. СИСТЕМА EURISCO

В нашем кратком обзоре не упомянута наиболее важная из всех программ, способных к "открытиям", — программа Лената EURISCO [6]. Однако по многим причинам здесь было бы невозможно описать ее подробно.

Во-первых, программа EURISCO слишком искусно сделана, чтобы можно было ограничиться кратким резюме. Невозможно указать на какой-то один алгоритм, сказав: "Вот сущность всего метода". Это не годилось бы даже в качестве краткого реферата.

Во-вторых, система EURISCO постоянно меняется — это движущаяся цель. Она скорее является исследовательской разработкой, чем законченным продуктом. Она меняется почти каждый день (в этом вся ее суть), причем Ленат сам участвует в процессе обучения, добавляя, вычеркивая и модифицируя эвристические правила, т. е. обучение здесь полуавтоматическое. Благодаря

этому программа работает более эффективно, но не становится легче для описания.

Главная же причина состоит в том, что своим успехом система EURISCO обязана не столько совершенным алгоритмам, которые используются при обучении, сколько богатству ее языка представлений. Любое понятие в системе, включая правила, представлено в виде "блока". Каждый блок имеет ряд ячеек-слотов.

Для тех из нас, кто вырос на традиционных языках программирования, блок представляет собой структуру данных, подобную записи, в которой слоты соответствуют полям. Но блоки исключительно гибки. Блок может иметь любое число слотов, а новые слоты могут добавляться или уничтожаться в ходе работы. Слоты могут содержать данные всех типов (строки, данные, числа, функции и т. д.), поскольку структуры языка Лисп, на котором написана система EURISCO, автоматически обеспечивают контроль типов. Кроме того, по ходу процесса обеспечения могут создаваться и слоты новых типов. Несколько правил в системе связано с решением вопроса, когда создавать слот нового типа.

Кроме всего прочего, вся система устроена рекурсивным образом, так что не должно вызывать удивление, что каждый слот описывается своим собственным блоком. В этом состоит одно из самых замечательных свойств системы EURISCO. Поскольку все объекты данных, включая процедурные блоки (т. е. эвристические правила), описаны в терминах блоков и слотов, то легко строятся как рассуждения, так и метарассуждения. Ничего не препятствует тому, чтобы одно из правил применялось и даже изменяло другое и, в частности, само себя. Так, система EURISCO может начать с правила вида

Правило 10

ЕСЛИ            некоторое правило сработало лишь однажды,

ТО              обобщить его

(где ЕСЛИ и ТО могут быть слотами).

Когда система уделит внимание этому правилу (т. е. Правило 10 сработает в первый раз), то система начнет рассматривать правила, которые сработали лишь однажды. Правило 10 будет одним из тех, которые могут привести к созданию другого правила:

Правило 20

ЕСЛИ            некоторое правило сработало менее 5 раз,

ТО              обобщить его.

Это правило носит несколько более общий характер и может, в свою очередь, применяться к самому себе или к Правилу 10, его прародителю.

Этот процесс мог бы идти до бесконечности, и поэтому я характеризую систему EURISCO как интроспективную, но у каждого правила имеется слот "ценность", который указывает на то, как много процессорного времени может быть отведено этому правилу. Лишь правила, которые прямо или косвенно полезны, могут быть обобщены или специализированы.

В заключение отметим, что система EURISCO привнесла в область обучения машин среди прочего два важных момента. Она достигла вполне заметных успехов, среди которых – создание совершенно нового логического элемента для СБИС – трехмерной схемы типа И/ИЛИ. Кроме того, в ней был разработан самый изощренный из всех существующих сегодня языков представления. Но если весьма несложно достать программы, основанные на идеях систем ID3 или INDUCE, и запустить их на вашем персональном компьютере, то следует сразу же отсоветовать вам брать программу EURISCO в качестве модели для собственной системы, если только ваша комната не забита специализированными LISP-машинами и у вас масса свободного времени.

## Глава 11

### АДАПТИВНЫЕ ОБУЧАЮЩИЕСЯ СИСТЕМЫ

*Стефен Смит*

#### 11.1. ВВЕДЕНИЕ

Успехи, достигнутые за последние несколько лет в области экспертных систем, с очевидностью показали, что основной компонентой, обеспечивающей экспертные способности, являются специфические для данного приложения знания. Было показано, что снабженные соответствующими знаниями системы могут действовать на уровне эксперта в таких разнообразных областях, как медицинская диагностика [23], выбор конфигурации ЭВМ [20], химический анализ [5] и поиск полезных ископаемых [9]. Несмотря на эти успехи, сегодняшнее состояние техники экспертных систем страдает серьезными недостатками:

конструирование экспертной системы по-прежнему остается искусством. Хотя появилось множество инструментальных средств и методологических рекомендаций (например, [13]), процесс выявления, представления и очищения знаний, используемых специалистами в каждой прикладной области, все еще остается плохо определенной и трудоемкой задачей;

применение методики экспертных систем по-прежнему ограничивается достаточно узкими, замкнутыми областями, а работа системы, как правило, ухудшается довольно быстро при приближении к границам того, что ей известно. Экспертные системы в самой малой степени способны адаптироваться или реорганизовывать знания по мере изменения предъявляемых к ним требований.

По-видимому, наибольший потенциал для решения этих проблем заключен в области обучающихся машин. Современные исследования по методам преобразования советов в операции [22], обучения по аналогии [6], обобщения [21] и открытия [18] предлагают широкий спектр перспективных подходов к облегчению наблюдающихся затруднений с усвоением знаний и к созданию гибких экспертных систем будущего. В настоящей главе описывается один

конкретный подход к этим вопросам и рассматривается адаптивная стратегия обучения, открывающая несколько путей усвоения, уточнения и реорганизации системных знаний в соответствии с наблюдаемым поведением системы.

Следуя Митчеллу, процесс обучения можно формулировать как поиск [21]. При заданном конкретном языке выражения знаний механизм обучения состоит в обследовании результирующего пространства представлений в поиске точек (представлений), ведущих к хорошей работе, при их использовании механизмом рассуждений системы. Определяющим для эффективности алгоритма обучения является то, как проводится такой поиск. Можно выделить несколько критических факторов:

Размер соответствующего пространства поиска для когнитивных задач практической значимости исключает систематическое рассмотрение подряд всех альтернатив. Механизм обучения должен располагать эвристическими методами для уменьшения диапазона поиска.

Такое сужение внимания не должно повлиять на достижимость хороших точек в этом пространстве. В частности поэтому, процесс поиска должен обладать достаточной свободой, чтобы выйти за пределы любых заранее очерченных понятий для данной проблемы.

Последнее замечание направлено против чрезмерного учета в механизме обучения критериев и предположений, специфических для данной предметной области. Хотя на пути приспособления методов обучения к решаемой в данный момент задаче и были достигнуты определенные успехи (например, [5]), похоже, что для этого необходимо иметь способность какого-то *туннельного видения* пространства представлений. Иными словами, те усилия, которые тратятся на разработку таких методов, мало что дают при решении крупных проблем, которые были упомянуты выше, ибо методы, как правило, оказываются непереносимыми на другие предметные области\*.

Признавая внутреннюю ограниченность специфических для данной области стратегий обследования пространства представлений, можно сформулировать нашу задачу так: требуется найти проблемно-независимую методику поиска, которую можно было бы эффективно использовать в подобном пространстве. Для выполнения такого серьезного заказа требуется механизм, который обладает согласованными возможностями по использованию структур, приводящих к хорошему функционированию в прошлом, и по исследованию новых структур. Пример такого рода механизма дает нам природа, где, несмотря на необозримое число возможностей разного рода, процесс эволюции быстро создает структуры (организмы), которые прекрасно адаптированы к специфическим условиям той среды, в которой они существуют. Поэтому при некоторых попытках в области обучения машин (например, [10, 11]) строились модели подобного процесса на компьютере. Однако в результате

---

\* Следует подчеркнуть, что мы не выступаем здесь против использования специальных для данной области знаний, а интересуемся тем, каким образом их следует применять. Идея здесь состоит в том, что такие знания должны служить в качестве сырья для "мельницы" обучающегося механизма, а не в качестве основы ее построения.



этих усилий не удалось создать одижаемого поведения при обучении, и от парадигмы эволюционного поиска пришлось в значительной степени отказаться.

Главным просчетом в этих первых работах было то, что используемые вычислительные процессы недостаточно точно отражали само моделируемое явление. Предположение, что процесс эволюции движим исключительно случайными мутациями (точка зрения, которая, кстати, все еще бытует сегодня в некоторых биологических кругах), просто неверно. Это существенно более сложный процесс, в ходе которого эффективно выделяются и применяются регулярности в структурах (организма), которые порождаются со временем. В работе [14] изложена теория, в которой явно выделяются некоторые из таких базовых операций и в которой результирующий процесс излагается с вычислительных позиций. Воспользовавшись этой более разумной моделью адаптации, мы в данной главе пересмотрели вопрос применимости эволюционного подхода к проектированию и разработке универсальных обучающихся систем.

Остальная часть главы построена следующим образом. В разд. 11.2 вводит-ся общий класс адаптивных стратегий поиска и исследуются его основные свойства. Одно приложение описанной методологии изучается в разд. 11.3 в контексте системы, которая обучается эвристиками управления процессом применения некоторого множества операторов решения проблем. Для демонстрации жизнеспособности такого подхода приведены результаты, полученные системой при попытке обучиться принимать правильные решения о ставках в игре в покер. В разд. 11.4 выделяются и обсуждаются некоторые альтернативные подходы к обучению с применением идей генетики. В разд. 11.5 высказываются некоторые общие соображения.

## 11.2. ГЕНЕТИЧЕСКИЕ АДАПТИВНЫЕ АЛГОРИТМЫ

В этом разделе находится некоторый класс адаптивных стратегий поиска, которые появились как центральная компонента недавно созданной теории адаптации, принадлежащей Холланду [14]. Эти так называемые *генетические алгоритмы* основаны на стандартных моделях наследственности и эволюции из области популяционной генетики, являющихся модельным воплощением тех механизмов адаптации, которые имеются в живых системах. Среди таких механизмов главными являются следующие:

Концентрация внимания на наблюдаемом поведении построенных структур.

Совокупность операторов поиска, которые обследуют коадаптирующиеся множества структурных компонентов (т. е. структурных конфигураций, которые делают совместный вклад в хорошую работу системы) при порождении новых структур для их последующей проверки.

Благодаря выделению этих процессов из специфического контекста генетики получаемые алгоритмы оказываются применимыми к широкому спектру проблем, встречающихся при конструировании адаптивных систем.

### 11.2.1. Обзор

В генетическом алгоритме динамика популяционной генетики имитируется путем организации базы знаний (популяции) структур (сущностей), которая эволюционирует со временем в соответствии с наблюдаемым поведением (пригодностью) ее структур в их рабочей среде. Каждая структура предъявляется алгоритму как последовательность составляющих ее частей (генотипов), с которой имеют дело операторы поиска. Конкретная интерпретация этой структуры (например, как совокупности значений параметров, правила вида "условие—действие" и т. д.) дает единственную точку в пространстве альтернативных решений для рассматриваемой проблемы (фенотипа), которая затем может быть включена в эволюционный процесс и которой можно приписать меру полезности. Поиск происходит посредством постоянного выбора структур из текущей базы знаний на основе ассоциированных мер полезности, получаемых благодаря интерпретации, и применения к этим структурам идеализированных *генетических операций*, создающих в эволюции новые структуры (потомки). Основная парадигма более точно выражена на рис. 11.1.

Чтобы как-то прочувствовать поведение такой поисковой стратегии, рассмотрим вероятности отбора, определенные на шаге 3 рис. 11.1. Мы видим, что если  $m$  структур выбраны для участия в генерации новых структур в

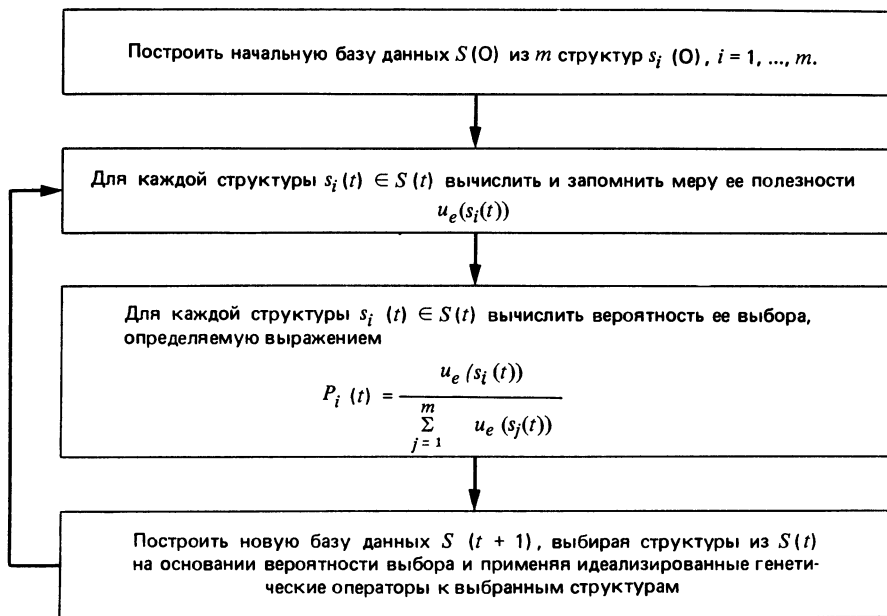


Рис. 11.1. Основной метод обучения

момент  $t$ , то среднее число новых структур, получаемых в текущей базе знаний, составляет

$$m \times \frac{u_e(s_i(t))}{\sum_{j=1}^m u_e(s_j(t))} = \frac{1}{m} \frac{\sum_{j=1}^m u_e(s_j(t))}{\sum_{j=1}^m u_e(s_j(t))} = \frac{u_e(s_i(t))}{\bar{u}_e(S(t))},$$

где  $\bar{u}_e$  — средняя полезность аргумента. Таким образом, эти вероятности обеспечивают селективное смещение в пользу структур, ведущих себя лучше среднего по сравнению с остальными структурами в базе знаний. В отсутствие других механизмов такое избирательное давление приведет к тому, что хорошо ведущие себя структуры в начальной базе знаний со временем будут занимать в ней все больше и больше места.

Однако простое распространение наилучших структур в базе знаний не дает никакого вклада в дальнейший поиск еще более совершенных структур. Этой цели служат генетические операторы поиска, трансформирующие структуры, отобранные из текущей базы знаний, в новые, еще не проверенные структуры. Хотя эти операторы выполняют над отобранными структурами простейшие синтаксические преобразования, в подразд. 11.2.3 будет показано, что они оказывают весьма изощренное и высокопараллельное воздействие, когда применяются совместно с вышеописанной процедурой отбора.

### 11.2.2. Операторы поиска

Главным средством создания новых структур является оператор *кроссинговера*. Он берет две структуры, случайным образом выбирает точку разрыва (т. е. место разделения компонент) на этих структурах и меняет местами последовательности компонент, находящиеся справа от точки разрыва. Например, если эти две структуры представляют собой последовательности компонент  $c_1, c_2, c_3, c_4, c_5$  и  $c_1', c_2', c_3', c_4', c_5'$  и перекрещиваются между второй и третьей позициями, то новыми построенными структурами будут следующие:  $c_1, c_2, c_3', c_4', c_5'$  и  $c_1', c_2', c_3, c_4, c_5$ . Набор различных возможных реализаций определяется конкретными решениями относительно того, следует ли обе полученные структуры включать в базу знаний, следует ли сохранить их родителей и какие из других структур, если это вообще необходимо, должны быть удалены.

При построении новых структур для последующего их опробывания оператор кроссинговера использует лишь информацию, присутствующую в структурах текущей базы знаний. Если конкретная информация отсутствует либо в силу ограниченности памяти, либо из-за потери в процессе отбора на предыдущем этапе итерации, то этот оператор не в состоянии создать содержащую ее новую структуру. Для привнесения новой информации в базу знаний предусмотрен оператор *мутации*, который произвольным образом изменяет одну или несколько компонент выбранной структуры. Однако в противоположность старым вычислительным моделям эволюционных процессов оператор мутации действует исключительно как фоновый оператор внутри

генетического алгоритма (т. е. вероятность его применения очень мала). Его наличие обеспечивает достижимость всех точек в пространстве поиска.

Третий оператор, *инверсия*, изменяет характер связей между компонентами структуры. Он берет одну структуру, случайным образом выбирает на ней две точки разрыва и располагает в обратном порядке компоненты, попавшие между этими двумя точками разрыва. Например, инверсией структуры  $c_1, c_2, c_3, c_4, c_5$  с точками разрыва, расположенными между первой и второй, между четвертой и пятой позициями, будет структура  $c_1, c_4, c_3, c_2, c_5$ . Заметим, что оператор инверсии не создает совсем другую структуру, а лишь альтернативное представление той же самой структуры (переупорядочивая составляющие ее компоненты). Важность таких изменений представления станет очевидной из дальнейшего.

### 11.2.3. Источник эффективности

Сила адаптивных стратегий поиска, описанных выше, лежит не в опробывании отдельных структур, а в эффективном использовании той богатой информации, которую дает такое опробывание в отношении взаимодействия между компонентами этих структур. Специфические конфигурации значений компонент, которые дали заметный вклад в хорошее поведение (например, конкретные пары значений параметров, конкретная группа условий правила и т. д.), сохраняются и распространяются по структурам в базе знаний высокопараллельным образом. А это, в свою очередь, создает предпосылки для последующего возникновения подобных конфигураций все большего и большего размера. Интуитивно мы можем рассматривать эти структурные конфигурации как регулярности в пространстве, которые проявляются по мере порождения и испытания отдельных структур. Встретившись однажды, они начинают служить в качестве *строительных блоков* при создании новых структур. Попробуем точнее охарактеризовать этот источник силы стратегий. Для упрощения анализа ограничимся рассмотрением проблемных пространств, в которых структуры состоят из фиксированного числа компонент ( $l$ ), каждая из которых определяется по ее положению в структуре, отражая присутствие или отсутствие определенного атрибута. Так, последовательность  $1, 0, 1, \dots, 1$  описывает структуру, которая обладает первым атрибутом, не обладает вторым, обладает третьим и т. д.

Договорившись об этом, мы можем формально охарактеризовать поиск, показав, что в таком случае происходит эффективная концентрация просмотра в тех разбиениях соответствующего  $l$ -мерного пространства, которые оказались связанными с хорошей работой системы. Разбиение (или гиперплоскость) такого пространства определяется связыванием значений  $k$  из  $l$  структурных компонент ( $1 \leq k \leq l$ ). Например, разбиение, обозначаемое  $1, \square, 0, \square, \dots, \square$  ( $\square$  читается как "не связанный"), выделяет все структуры, которые *обладают* указанным здесь значением первого атрибута и *отсутствием* значения для третьего. При оценивании структуры, которая лежит в  $j$ -м разбиении этого пространства, выявляется также информация и о полезности всей области пространства поиска, очерчиваемой этим разбиением.

Иначе говоря, оценивание структуры является также оцениванием конкретной конфигурации значений компонентов, которая служит для *определения* разбиения  $j$  (например, как показано выше, первый атрибут конфигурации присутствует, третий отсутствует). Заметим, что одна структура принадлежит  $2^l - 1$  разбиениям в рассматриваемом пространстве (т. е.  $\sum_{k=1}^l C_l^k$ ). В со-

ответствии с этим мера полезности, выявленная при оценивании в базе знаний данной структуры, на самом деле дает информацию в отношении полезности этой конкретной конфигурации величин компонентов для каждого из  $2^l - 1$  разбиений, которому она принадлежит.

Имея это в виду, интересно рассмотреть те структуры в базе знаний, которые принадлежат одним и тем же разбиениям пространства. Для этого обозначим через  $S_j(t)$  подмножество структур в базе знаний в момент  $t$ , которые принадлежат разбиению  $j$ , а через  $m_j(t)$  — число структур в  $S_j(t)$ . Обращаясь еще раз к вероятностям отбора, определенным на рис. 11.1, можно показать, что среднее число новых структур, выводимых из структур, принадлежащих  $S_j(t)$ ,

$$\sum_{s \in S_j(t)} \frac{u_e(s)}{\bar{u}_e(S(t))} = m_j(t) \frac{\bar{u}_e(S_j(t))}{\bar{u}_e(S(t))}.$$

Если результирующие структуры также принадлежат разбиению  $j$ , то

$$m_j(t+1) = m_j(t) \frac{\bar{u}_e(S_j(t))}{\bar{u}_e(S(t))}.$$

Это означает, что число структур\*, принадлежащих данному разбиению пространства, будет возрастать или убывать со временем прямо пропорционально наблюдаемому качеству работы структур, принадлежащих этому разбиению. Таким образом, мы видим, что процесс порождения новых структур постоянно ориентируется на наиболее перспективные области пространства поиска и, кроме того, такое обследование пространства неявно происходит параллельным образом. Заметим, что для базы знаний, содержащей  $m$  структур, вышеуказанный темп создания структур достигается одновременно для разбиений пространства, число которых лежит между  $2^l - 1$  и  $m$  ( $2^l - 1$ ).

Достижение указанного темпа создания в отношении данного разбиения  $j$ , конечно, зависит от конкретной конфигурации значений компонент, определяющих разбиение, которая передается неизменной от родителей в новую структуру. Отсюда возникает необходимость в анализе разрушительных эффектов, приносимых порождающими операторами. Оператор кроссинговера разорвет данную структурную конфигурацию, если выбранные точки разрыва попадут между двумя значениями компонент этой конфигурации. Вероятность такого события прямо пропорциональна длине наименьшей последовательности компонент, содержащей саму конфигурацию, которую

\* Речь все время идет о среднем числе структур. — Прим. ред.

мы будем называть *определяющим фрагментом* соответствующего разбиения. Например, вероятность разрушения оператором кроссинговера разбиения  $\square, 0, \square, 1, 1, \square, \dots, \square$  (имеющего определяющий фрагмент  $0, \square, 1, 1$ ) равна  $3/(l - 1)$ . Таким образом, у оператора кроссинговера имеется тенденция к сохранению указанного темпа выборки в отношении тех разбиений, определяющие фрагменты которых малы по сравнению с  $l$ , имеется также тенденция к разрушению в отношении разбиений с большими определяющими фрагментами. Но поскольку структуры, принадлежащие специфическим разбиениям, обеспечивающим высокое качество функционирования, с малыми определяющими фрагментами со временем начинают преобладать в базе знаний, то произойдет эффективное уменьшение числа определяющих фрагментов других разбиений, ослабляющее разрушительное влияние оператора кроссинговера на темп, с которым последние выбираются. Оператор мутации оказывает незначительное влияние на фактор выбора, поскольку ему отводится в поиске фоновая роль.

В таком поиске могут возникнуть все же трудности, если существенные разбиения (т. е. примитивные строительные блоки) обладают длинными определяющими фрагментами. Эта проблема связана с представлением и возникает при неудачном выборе порядка структурных компонент. В этом случае продуктивность поиска повышается путем использования оператора инверсии. Благодаря изменению порядка компонент в структуре, его применение будет создавать тенденцию к уменьшению длины длинных определяющих фрагментов, благодаря чему открываются большие возможности применения кроссинговера\*.

Таким образом, возможности генетического алгоритма связаны с его способностью обследовать почти оптимальным образом информацию о полезности экспоненциально большого числа структурных конфигураций без необходимости явного вычисления и запоминания. Благодаря этому ведется концентрированное изучение пространства поиска, когда внимание сосредоточивается на областях, содержащих структуры с полезностью выше средней. Тем не менее база знаний остается широко распределенной по пространству, не давая поиску остановиться в каком-то локальном минимуме. Читателю рекомендуется ознакомиться с работой [14], где дается более глубокое описание алгоритма и его свойств.

---

\* Заметим, что принятые нами условия в отношении представления в данном анализе накладывают некоторые ограничения на одновременное применение в процессе поиска операторов кроссинговера и инверсии. В частности, необходимо сохранять знания о местоположении каждой компоненты в данной структуре, чтобы гарантировать, что оператор кроссинговера всегда применяется к аналогично представленным структурам. Другие предположения о представлении могут привести к более естественному режиму сосуществования этих операторов (см., например, [24]).

### 11.3. ПРИМЕНЕНИЕ ОПИСАННОЙ МЕТОДИКИ К ОБУЧЕНИЮ ЭВРИСТИКАМ РЕШЕНИЯ ЗАДАЧ

В предыдущем разделе содержится достаточно соображений в пользу генетического поиска как универсального механизма обучения. В настоящем разделе показана жизненность этих методов на примере использования их в сложной задаче обучения. Конкретнее, мы опишем структуру и работу прототипа обучающейся системы, названной LS-1 [24, 25]. Система LS-1 обучается множеству эвристик, представленных в виде правил продукций, которые направляют примечание некоторой совокупности операторов в определенной задаче. Это достигается накоплением опыта в соответствующей проблемной области, причем генетический алгоритм является единственным средством улучшения способности системы к решению задач. Система LS-1 является хорошим примером возможностей генетического поиска, обеспечивая весьма эффективное манипулирование сложными символьными структурами при обучении эвристикам решения задач.

#### 11.3.1. Подход к обучению в системе LS-1

Функционально система LS-1 состоит из трех взаимодействующих компонент:

Компонента решения задачи – машина логического вывода для применения альтернативных множеств управляющих эвристик к примерам рассматриваемой задачи. Критик – механизм оценки работы данного множества эвристик при решении примеров рассматриваемой задачи.

Обучающаяся компонента – генетическая стратегия поиска, ответственная за порождение новых множеств эвристик в ответ на оценки работы системы, поставляемые критиком.

Взаимодействие этих компонент схематически показано на рис. 11.2.

Работа системы осуществляется на основе базы знаний из  $m$  структур, каждая из которых является множеством управляющих эвристик (или множеством правил) – кандидатов на решение задачи. На данном цикле этапа обучения каждое множество правил применяется компонентой решения к  $k$  примерам задачи. Критик анализирует  $k$ -операторную последовательность, порождаемую множеством правил в ходе такой проверки "профессиональной пригодности", а также характеристики оцениваемого множества правил и выдает меру работоспособности, указывающую на относительные достоинства этого множества правил в качестве потенциального решения исходной задачи. Как только все структуры из базы знаний оказываются оцененными подобным образом, вызывается генетический алгоритм, который генерирует новую базу знаний для последующей ее проверки, и цикл повторяется. База знаний из структур и ассоциированных с ними мер работоспособности рассматривается как внутренняя память системы LS-1, представляющая в каждый момент сумму того опыта, который имеется у системы в отношении проблемной области. Внешне текущей гипотезой системы в отношении решения рассматриваемой задачи является множество правил, получивших  $k$

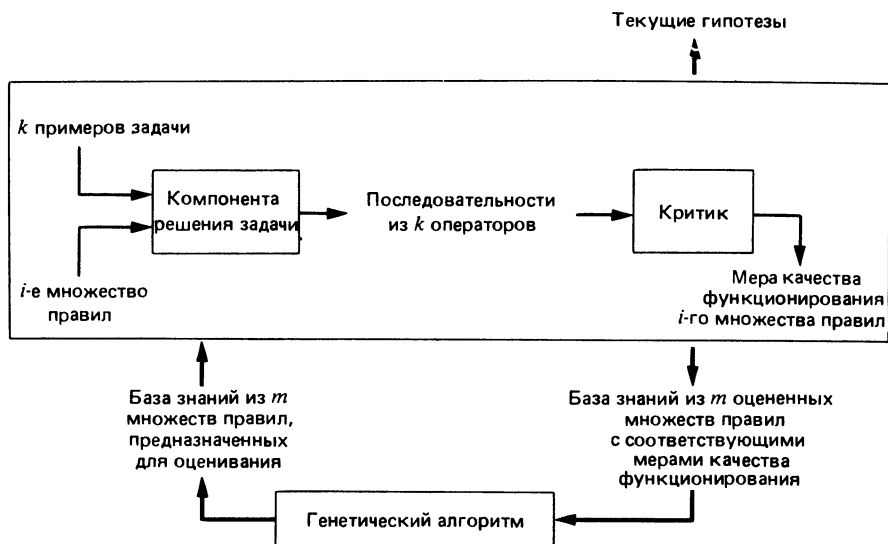


Рис. 11.2. Организация обучения в системе LS-1

этому моменту в ходе поиска наивысшую оценку. Прогресс системы LS-1 в проблемной области является результатом учета последовательности (возможно, разрывной) гипотез, построенных в ходе существования системы.

Как говорилось выше, область структур, среди которых предстоит осуществлять поиск обучающемуся алгоритму системы LS-1, довольно серьезно отличается от простого представления, использованного для иллюстрации свойств генетического алгоритма в разд. 11.2. Хотя на нижнем уровне данную структуру (множество правил) в базе знаний системы LS-1, безусловно, можно рассматривать как последовательность символов из некоторого небольшого алфавита (например, как двоичную последовательность); на более высоких уровнях эти символы комбинируются между собой, образуя осмысленные символичные компоненты (например, образцы, действия, продукции и т. д.). Кроме того, эти компоненты верхнего уровня никак не связаны с определенной позицией в структуре в целом. Например, на уровне правил компоненты имеют одну и ту же интерпретацию независимо от их местоположения в структуре. Чтобы эффективно использовать регулярности пространства на всех уровнях абстракции, стратегия обучения должна учитывать все такие характеристики представления. В соответствии с этим в алгоритме системы LS-1 используется другое множество операторов, которое позволяет манипулировать структурами на различных уровнях их деятельности. Теоретические последствия таких изменений, которые выходят за рамки настоящей книги, были исследованы Смитом [24], и было показано, что результирующие свойства оказываются аналогичными свойствам исходных алгоритмов.



### 11.3.2. Представление эвристик решения проблем

Использование генетического алгоритма для манипулирования структурами знаний в соответствии с наблюдаемым функционированием обуславливает определенные требования к форме языка представлений. Принимая во внимание, что генетические операторы поиска преобразуют структуры, не учитывая никакой их специфической интерпретации, важно добиваться, чтобы продукции имели достаточно простую синтаксическую форму, чтобы повторные применения операторов порождали допустимые точки в пространстве поиска. Это требование должно быть согласовано с желанием иметь достаточно выразительное представление, позволяющее эффективно описывать все виды решения задач в данной предметной области. Представление знаний в системе LS-1 — это результат попытки совместить эти внешне противоречивые цели.

В самом общем смысле компонента решения задач системы LS-1 организована как проблемно-независимый интерпретатор для правил продукций, в который необходимо передать конкретные для данной задачи элементы, если речь идет о применении в определенной области. Конкретизация интерпретатора для данной области решения проблем достигается путем указания соответствующего множества переменных для состояний и множества операторов решения проблем. Эти переменные задают системе необходимую специфику области, а операторы — репертуар тех действий, которые возможны в данном состоянии. Эвристики решения задач даются в виде простых продукций, которые при их активации запускают операторы и направляют в рабочую память сообщения в соответствии с полученной переменной состояния и конфигурациями в рабочей памяти.

В отдельных продукциях используется простой и однородный синтаксис. Каждый антецедент состоит из фиксированного числа элементарных образцов, которые реагируют на каждую переменную состояния, поставляемую в систему, и заранее указанного числа компонент, предназначенных для обслуживания сообщений, появляющихся в рабочей памяти. Каждый консеквент представляет собой сообщение, которое предстоит отправить в рабочую память, если эта продукция сработает, и указание на тот оператор, который должен быть применен к текущему состоянию. Продукция может быть предназначена и исключительно для целей внутренней коммуникации (см. ниже); в таком случае предусматривается специальный оператор *безоп*. На рис. 11.3 продукция, используемая в системе LS-1, показана в стилизованном виде.

Информация, воспринимаемая продукцией (как значения, характеризующие внешнее состояние обработки, так и сообщения в рабочей памяти), представляется в двоичной форме. Это позволяет воспользоваться синтаксически простым, но достаточно выразительным языком для сопоставления с образцом. Конкретнее, каждый образец, появляющийся в продукциях, имеет вид строки в алфавите  $\{0, 1, \# \}$ , где нули и единицы играют роль констант образца, а  $\#$  сопоставим с любым символом. Например, образец  $\#1\#0$  будет сопоставим с любым входом 0100, 0110, 1100 и 1110. Дополнительные выразительные возможности обеспечиваются использованием образца *префикс*, 168

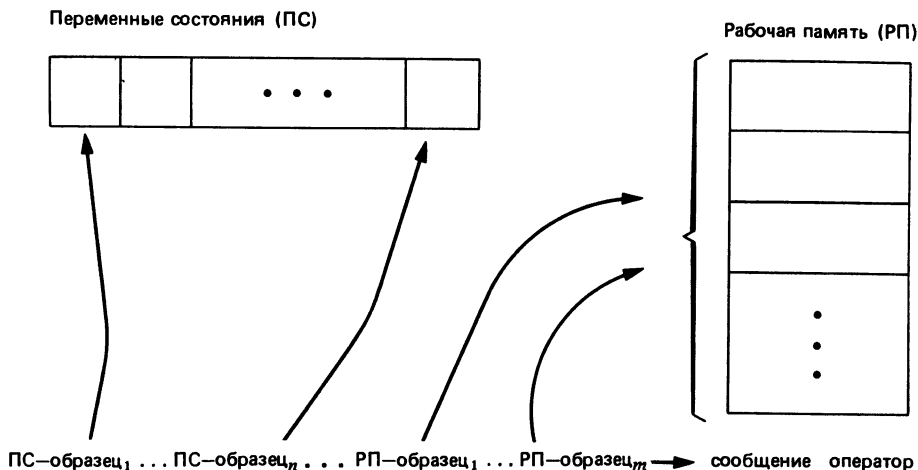


Рис. 11.3. Стилизованный вид продукции системы LS-1

который по желанию может быть использован либо для образования из образца его дополнения (т. е. множество входных величин, которые раньше сопоставлялись с образцом, становится множеством, которое теперь не сопоставимо с ним), либо просто как команда для системы сопоставления не принимать соответствующий образец во внимание. Последнее является удобным средством для ограничения контекста некоторой продукции определенной частью обрабатываемого состояния.

Прямым следствием таких соглашений относительно представления является кодирование эвристик в виде последовательностей с фиксированной длиной, которые легко преобразуются в новые эвристики на чисто синтаксическом уровне. (Напомним, что данное множество продукций рассматривается в базе знаний системы LS-1 как одна структура.) Однако этого удалось достичь не без дополнительных ограничений на опознающие способности образцов продукций. Хотя образцы, выражимые на описанном языке, характеризуют большой диапазон потенциально полезных подмножеств величин, существуют некоторые ситуации (подмножества), которые таким образом опознать нельзя. Эти недочеты уравниваются благодаря использованию некоторого нестандартного режима управления, который обеспечивает кооперативное опознание более сложных аспектов текущей ситуации отдельными продуктами. Конкретнее, этап разрешения конфликта изъят из основного цикла "распознавание, выбор, действие", характерного для типичных интерпретаторов правил: активизируются все виды означивания переменных, найденные в фазе распознавания в данном цикле. Глобальная синхронизация откладывается до конца цикла, когда в рабочую память направляются все сообщения, порожденные активными продуктами, а предложенные операторы решения задач, если они есть, применяются к текущему внешнему

состоянию. Сообщения, направленные в рабочую память, существуют только в течение одного следующего цикла. Поэтому некоторое описание состояния может запустить несколько продукций, чья активность выразится лишь в размещении в рабочей памяти сообщений. Эти сообщения на следующем цикле могут быть обнаружены другой продукцией, что приведет к применению некоторого конкретного оператора. Благодаря такой параллельной архитектуре система становится вычислительно полной по отношению к тем элементам проблемной области, которые на задаются\*.

Если операторы, выдвигаемые для применения в некотором цикле, по своему действию являются взаимоисключающими, то возникает необходимость в координации внешней деятельности по решению задач. В подобных случаях используется *механизм разрешения по умолчанию*, который вероятностным образом выбирает один из этих конфликтующих операторов, учитывая при этом число активных продукций, рекомендующих каждую из альтернатив. Такое недерминированное поведение рассматривается как отражение текущей неопределенности системы в отношении текущего состояния решения задачи. Чем больше перевес в пользу применения какого-то одного оператора (т. е. чем больше процент активных продукций, рекомендующих именно этот оператор), тем меньше неопределенность системы в отношении того, как ей реагировать.

### 11.3.3. Вопрос назначения поощрения

Успех любой обучающейся системы зависит, с одной стороны, от способности эффективно поощрять те единицы знания, которые привели к хорошему функционированию в данной рабочей среде, и, с другой стороны, штрафовать те единицы, которые оказались неподходящими. Без глобального внимания к этому вопросу всякие попытки улучшить работу системы будут весьма неуклюжими. В контексте системы LS-1 вопрос о правильном поощрении превращается в вопрос построения адекватного механизма для оценки относительного достоинства каждого множества правил, созданного как потенциальное решение задач, используемой при обучении. Как мы видели при анализе стратегии генетического алгоритма обучения в подразд. 11.2.3, этот алгоритм прекрасно выделяет те характеристики структуры, которые в действительности обеспечивают хорошее поведение системы, используя их при построении новых структур. Однако степень, в которой такие поисковые свойства могут быть выявлены в пределах конкретной области решения проблем, в конечном счете зависит от различительной способности функционирования. Таким образом, уровень интеллектуальности, достижимый системой

---

\* Обычно разрешение конфликтов рассматривается как необходимое условие обучения, поскольку оно координирует взаимодействие продукций и дает самим продукциям значительную степень автономии [19]. Заметим, что такое мнение оправдано в свете обучения, рассчитанного на постепенное добавление новых правил. В системе LS-1 множества правил со временем видоизменяются, образуя целостные структуры.

LS-1 в некоторой конкретной области, определяется качеством имеющейся оценочной функции.

Оценочная функция, поставляемая критиком системы LS-1, при оценке качества множеств правил, порождаемых генетическим алгоритмом, учитывается как источники знаний, специфические для данной задачи, так и проблемно-независимые источники. Источники знаний первой группы дают измерения, непосредственно связанные с внешним поведением, полученным при профессиональном испытании множества правил, и им уделяется основное внимание при построении меры полезности. Вторая группа источников дает измерения, отражающие общие характеристики множества правил, которые существенны для хорошей работы независимо от конкретной предметной области. Эти источники в меньшей мере влияют на общую меру полезности, обеспечивая более тонкий уровень анализа.

Для оценки качества решения задач в данной области критику системы LS-1 необходимы средства анализа правильности (или частичной правильности) последовательностей из  $k$  операторов, создаваемых данным множеством правил, в ходе профессионального испытания. Природа и область таких знаний, очевидно, будут изменяться от одной области к другой и в какой-то степени являться функцией сообразительности конструктора. В самом деле, критик – это та компонента системы, в которой с пользой могут быть применены любые объемы знаний о конкретной области. Единственным ограничением является то, что эти знания относятся к функциональным (наблюдаемым) аспектам работы системы в намеченной области, а не к структурным характеристикам знаний по решению задач, выявление которых является задачей системы. При использовании специальных для данной области знаний последнего вида на поиск, производимый обучающимися механизмам, потенциально были бы наложены слишком жесткие границы, особенно в случае плохо понятных областей, где применение обучающихся систем и было бы наиболее выигрышным.

Проблемно-независимые меры, используемые критиком системы LS-1, могут быть разнесены по трем довольно широким классам:

Структурные свойства правил: по-видимому, имеются отчетливые характеристики правил, которые *способствуют* хорошему функционированию. В компоненте критика системы LS-1 оценивание таких характеристик производится путем использования мер, отражающих перспективы для взаимодействия между сообщениями правил и образцами в рабочей памяти, уровень специфики задачи в образцах правил и наличие каких-то конкретных недостатков (например, продукции, которые невозможно означить, бесконечные циклы и т. д.).

Динамические свойства правил: другие общие указания на полезность некоторого множества правил как потенциального решения могут быть получены при изучении истории их выполнения на этапе профессиональной проверки. Аналогичным образом получают измерения, указывающие степень доверия механизму решения по умолчанию при выборе операторов решения задач, а также процент правил, на самом деле участвующих в процессе решения.

Соображения эффективности: здесь ряд продукций в множестве правил используется для обеспечения порождения компактных множеств правил.

Использование таких мер увеличивает дискриминационные возможности компоненты критика системы LS-1, делая возможным различие даже в ситуациях, когда либо совсем отсутствуют результаты наблюдения внешнего функционирования, либо их недостаточно. (Это — важная черта, обеспечивающая проблемно-независимый характер стратегии обучения.)

#### 11.3.4. Некоторые экспериментальные результаты

Рабочие характеристики системы LS-1 изучались в двух не связанных между собой областях принятия решений, которые были взяты с учетом других работ по созданию обучающихся систем:

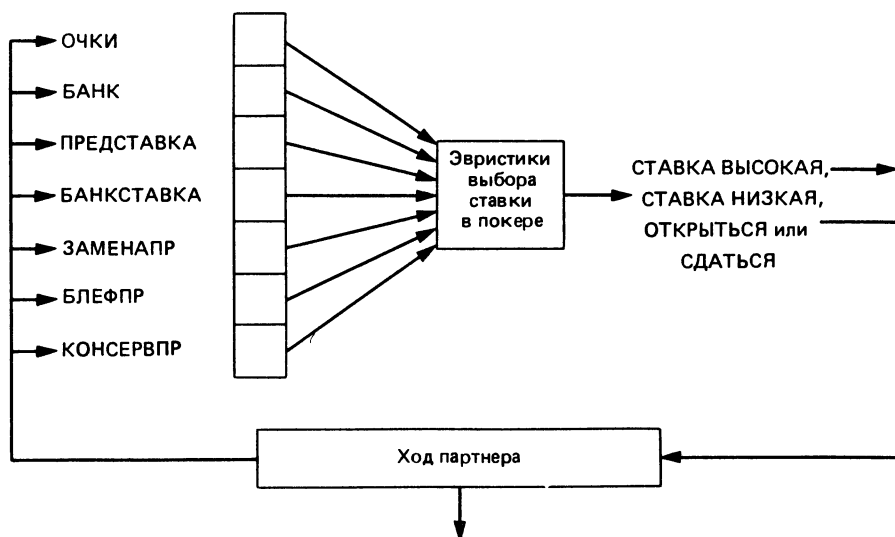
1. Обучение решению простой задачи прохождения лабиринта. Это проблемная область была использована в предыдущей генетической обучающейся системе (см. разд. 11.4) и была нами взята, чтобы провести сравнение двух разных подходов.
2. Обучение решению о ставках в игре в покер. Эта проблемная область была использована в классическом труде Уотермана по машинному обучению [26].

В обеих областях система LS-1 проявила способность значительно улучшать свои действия с течением времени. Мы остановимся здесь на экспериментах, проведенных в более интригующей области карточных ставок, поскольку результаты, которые были получены, оказываются отличными от результатов Уотермана, который пользовался стратегией обучения, в большей степени зависящей от предметной области. Детальное обсуждение обеих групп экспериментов может быть найдено в работе [24].

Кратко проблема ставок в покере, рассмотренная Уотерманом, состоит в следующем. Считается, что система выступает в качестве одного игрока в стандартной игре в покер\* с целью правильным выбором решения о ставках добиться максимального выигрыша. Каждый раз, когда наступает очередь системы делать ставку в игре, она должна выбрать наиболее подходящее решение, учитывая текущее состояние игры. Система воспринимает текущее состояние игры посредством некоторой совокупности значений переменных состояния, кроме того, ей сообщается фиксированное множество альтернатив, допустимых при выборе ставки. Общий сценарий и специфический набор элементов, поступающих в систему, изображены на рис. 11.4. Чтобы иметь возможность оценить эксперименты, которые будут описаны нами ниже, перед системой была поставлена цель — построить правильные решения о ставках в каждом из десяти последовательных раундов игры.

---

\* В этой игре игрокам раздается "втемную" по пять карт, после чего один из игроков делает денежную ставку на то, что его комбинация карт в определенном смысле лучше, чем у других. Следующий игрок либо сдается, либо удваивает ставку, либо поднимает ее еще выше. В рассматриваемом варианте азартной игры допускается после объявления ставки заменять карты на новые, взятые из колоды. — *Прим. ред.*



ОЧКИ — достоинство сочетания карт системы

БАНК — количество денег в банке

ПРЕДСТАВКА — предыдущая ставка

БАНКСТАВКА — отношение суммы денег в банке к размеру предыдущей ставки

ЗАМЕНАПР — число карт, которые были заменены противником

БЛЕФПР — вероятность того, что противник может блефовать

КОНСЕРВПР — мера консервативности стиля игры противника

Рис. 11.4. Область карточной игры в покер

Для определения правильности или ошибочности данного решения о ставках в системе Уотермана использовалась дедуктивная процедура, основанная на некоторой аксиоматизации игры в покер. Та же самая процедура была применена и для критика в системе LS-1, т. е. была основной для оценки качества функционирования в этой области. Конкретнее, в оценочной функции была определена и использована мера, указывающая *процент согласия построенных решений о ставках с аксиомами*. Вторая, связанная с качеством функционирования мера — *число успешно законченных раундов игры*, была использована, чтобы преждевременные решения сдать (т. е. сдать до замены карт у игрока) не оценивались критиком как положительные.

Система LS-1 играла против системы P [built-in], программы выдвижения ставок, написанной Уотерманом, про которую было известно, что она играет примерно на уровне хорошего игрока в покер [26]. Имелась внутренняя база знаний из 50 правил, которые вначале были созданы путем случайного выбора условий и операторов для отдельных продукций в каждой структу-

ре\*. В течение двух часов процессорного времени на ЭВМ DEC-10 удалось наблюдать действие системы LS-1 против P [build-in] на интервале на 4200 оценок множества правил (примерно 85 циклов обучения). Как показывает качество ее функционирования, описанное в табл. 11.1, цели, поставленные перед системой, за этот интервал времени были достигнуты. Было построено множество продукций, которые успешно себя проявили в десяти последовательных раундах игры и которые были в полном согласии с аксиомами покера в отношении решения о ставках\*\*.

Т а б л и ц а 11.1

Результаты работы систем в эксперименте 1

Число выполненных оцениваний	500	1000	2000	3000	4200
Оценка качества функционирования текущей гипотезы по шкале 0 – 1	0,461	0,490	0,562	0,748	0,893
Процент согласия с аксиомами покера	71,4	80	76,1	85,7	100
Число успешно законченных раундов	3	4	7	8	10

Исследование этих результатов показало, что эвристики, используемые системой P [build-in], которые вполне соответствовали подходу Уотермана, оказались неприспособленными для столь продолжительной игры в покер (скажем, 42 000 раундов игры). В общем, система P [build-in] неправильно решила, что система LS-1 является исключительно консервативным игроком; поэтому, опираясь на это неверное представление, можно было бы при правильном "блефовании" легко заставить ее сдаться. Система LS-1 реагировала на оппонента в точности так, как нужно, создавая множества правил, имеющих повышенную тенденцию настойчиво "торговаться".

Чтобы проверить систему LS-1 в игре с более достойным игроком, в системе P [build-in] были сделаны необходимые изменения, исключающие формирование такого ошибочного представления, и проведен второй эксперимент. Результат эксперимента отображен в табл. 11.2. Из табл. 11.2 мы видим устойчивое и существенное улучшение гипотез, порождаемых системой LS-1

\* Решение начать работу системы LS-1 с нулевой информации в этих экспериментах мотивировалось желанием получить непосредственное сравнение с результатами Уотермана. Безусловно, можно было бы начать работу системы с несколько более высокого уровня начального опыта.

\*\* Представленные здесь результаты являются на самом деле усредненными по нескольким запускам системы LS-1 в работу, ибо результат одного запуска мог бы быть объяснен просто некоторыми вероятностными факторами (например, начальной установкой генератора случайных чисел ЭВМ).

## Результаты работы системы в эксперименте 2

Число выполненных оцениваний	500	1000	2000	3000	4000
Оценка качества функционирования текущей гипотезы по шкале 0 – 1	0,414	0,483	0,619	0,689	0,74
Процент согласия с аксиомами покера	58,75	73,03	89,9	75,1	81,93
Число успешно законченных раундов	3	4	6	8	9

в процессе игры в покер. После 4000 оцениваний было построено множество правил, которое привело к успешному завершению 9 из 10 раундов игры, причем решения относительно ставок согласовались с теми, которые предлагал критик, в 82 % всего времени игры.

Этот относительный успех системы LS-1 в области выдвижения ставок можно сравнить также с результатами, полученными системой Уотермана. Уотерман подверг свою *обученную* систему профессиональной проверке, состоявшей из 50 раундов игры в покер с человеком, также оценив работоспособность в процентах соответствия построенных решений с аксиомами. При этом его системой был получен результат в 86 % [26]. Таким образом, хотя множества правил системы LS-1 были подвергнуты более короткому испытанию (10 раундов), уровень функционирования, которого достигли эти системы, оказался примерно таким же, как и в работе [26]. Это приобретает особое значение, если учесть те специфические для проблемной области зависимости, которые присущи методу, использованному в системе Уотермана при манипулировании ее множеством продукций. Создание новых продукций в системе направлялось *априорно* имевшейся *матрицей решений*, определяющей для каждого возможного решения о ставке причину, почему каждая переменная состояния имеет отношение (если она имеет такое отношение) к этому решению. Такое знание было непосредственно использовано при формулировании предикатов antecedenta каждой новой продукции. Однако такое знание не требовалось и не давалось системе LS-1.

#### 11.4. ДРУГИЕ ПОДХОДЫ К ОБУЧЕНИЮ С ИСПОЛЬЗОВАНИЕМ ИДЕЙ ГЕНЕТИКИ

Результаты работы, достигнутые системой LS-1, ясно показывают большие возможности адаптивной стратегии обучения. Тем не менее подход, принятый в системе LS-1 и состоящий в манипулировании базой знаний альтернативных множеств продукций, иллюстрирует лишь одну точку во всем спектре схем обучения с привлечением идей из генетики. Ограничения на форму и сферу приложения знаний, с которыми работает система,



порождают целый набор полезных механизмов обучения. В настоящем разделе дается обзор ряда работ, которые были выполнены с использованием генетических стратегий поиска, и выделяются некоторые другие подходы к обучению.

Первоначально областью изучения свойств генетических алгоритмов поиска была область оптимизации. Первое ясное экспериментальное исследование было проделано Деджонгом, сравнившим достоинства различных вариантов некоторого класса генетических алгоритмов поиска и сопоставившим их с более традиционными методами оптимизации [7]. Было показано, что у генетических алгоритмов поиска имеется определенное преимущество перед другими алгоритмами при решении нелинейных и многомерных задач оптимизации. Анализ рабочих характеристик алгоритмов был продолжен в работах [3, 8]. В недавнем исследовании Грэфенстетта был сделан следующий шаг в экспериментальном направлении: сам генетический алгоритм был использован для исследования пространства возможных реализаций [12]. Интересно, что полученные результаты подтвердили многие заключения, полученные в прежних работах. Некоторые начальные шаги по формальной характеристике классов функций, наиболее пригодных для решения с помощью генетических поисковых стратегий, были проделаны Бетке [1].

Оптимизация функций является простейшей разновидностью обучения с использованием генетических алгоритмов. Однако способность эффективно оптимизировать значения большой совокупности параметров находит множество полезных применений, имеющих непосредственное отношение к современной технике экспертных систем. Одно такое применение возможно в системах, в которых в правилах вводятся определенные степени доверия (недоверия), чтобы обеспечить правдоподобное рассуждение при наличии неопределенностей. Такие меры обычно представляют собой субъективные оценки, поставляемые специалистами из данной области, а поэтому по мере роста базы правил задача поддержания согласованных доверий становится все более и более трудной. Проблема еще более усложняется, когда учитываются оценки, поступающие от нескольких экспертов. Генетическая адаптивная стратегия является прямым средством для тонкой подстройки этих мер доверия по мере накопления системой опыта работы.

Альтернативный подход к обучению правилам разрабатывался Холландом и Букером [2, 14 – 17]. Они пришли к построению класса систем, улучшающих свою работу, посредством создания и усовершенствования конструкций типа правил, названных *классификаторами*, во многих отношениях подобных представлению, описанному в подразд. 11.3.2. Однако этот подход отличается от принципов системы LS-1 в способе применения генетической стратегии поиска. Эти системы работают с базой знаний из отдельных классификаторов, и генетическому алгоритму отводится несколько иная роль, состоящая в обеспечении достаточного разнообразия классификаторов, создаваемых с ходом времени. Поэтому классификаторы, обеспечивающие различные типы поведения (например, назначающие применение взаимоисключающих операторов решения задач), физически разделены в базе знаний и

каждое множество классификаторов преобразуется генетическим алгоритмом независимым образом. Чтобы придать определенное направление отбору, с каждым классификатором в базе знаний ассоциируется мера его силы, отражающая степень, в которой его использование в прошлом вело к достижению целей системы. Эти меры силы используются в сложной стратегии распределения поощрений (названной *алгоритмом пожарников*), когда система работает над конкретными когнитивными задачами. По мере того как генетической обучающейся стратегией создаются новые классификаторы, они замещают те классификаторы в базе знаний, которые имеют наименьшую силу. Экспериментальные исследования этих систем классификаторов в различных когнитивных областях (например, в упомянутой в подразд. 11.3.4 проблеме прохождения лабиринта) показали устойчивые способности к обучению.

## 11.5. ЗАКЛЮЧЕНИЕ

Настоящая глава была посвящена применимости конкретного класса адаптивных стратегий поиска, названных генетическими алгоритмами, к проблемам, связанным с усвоением знаний в современных экспертных системах. Перечислим кратко еще раз основные моменты. Мы начали с того, что для универсальных обучающихся способностей требуются методы поиска, способные работать без использования заранее установленных понятий из проблемной области, а потому они не должны основываться на предположениях, обусловленных особенностями рассматриваемой области. С учетом эффективности адаптивного поиска в естественных системах на эту роль были предложены генетические алгоритмы, которые являлись абстракцией существенных механизмов, используемых в эволюции. Формальное исследование их свойств показало определенное достоинство такого подхода, выявив высокопараллельный и широкомасштабный характер поиска в пространстве представлений системы. Затем мы уделили внимание конкретному применению в случае обучения эвристикам решения задач, остановившись на рабочих характеристиках такой методики в достаточно сложной задаче обучения. Была также описана система LS-1, являющаяся реализацией обучающейся системы на генетическом принципе, и рассмотрены результаты, полученные при попытке обучиться эвристикам выдвижения ставок в игре в покер. В этой проблемной области система LS-1 показала свою способность улучшать функционирование с течением времени и достигла уровня компетентности, сравнимого с уровнем компетентности предшествующей обучающейся системы, в которой использовались более специальные для данной предметной области методы. Наконец, мы рассмотрели и альтернативные подходы к обучению, в которых используются генетические адаптивные стратегии, что показало широкий спектр возможностей, возникающих в этом случае.

Эта работа отчасти финансировалась факультетом информатики и вычислительной техники Питтсбургского университета (США) и Институтом роботики при Университете Карнеги-Меллона.

*Рой Рада*

## 12.1. ВВЕДЕНИЕ

Все люди по-разному понимают содержание искусственного интеллекта, но при стандартном подходе к этой науке в качестве основных вопросов выделяются *поиск* и *представление знаний*. Ни один алгоритм поиска или решения задач не может быть равно эффективным во всех пространствах поиска или во всех задачах. Важной задачей искусственного интеллекта является классификация алгоритмов поиска и пространств поиска с указанием того, в каких пространствах поиска хорошо работают определенные алгоритмы. В большинстве задач, представляющих интерес для людей, решающие алгоритмы, по-видимому, опираются на большие запасы знаний, специфических для рассматриваемых задач. При этом успех программ искусственного интеллекта обусловлен адекватным представлением этих знаний. Применения искусственного интеллекта, в частности на примере *экспертных систем*, предназначенных для медицины, точных наук и образования, помогают проиллюстрировать то громадное воздействие, которое искусственный интеллект оказывает на эти высокопрофессиональные области.

Основное затруднение при создании экспертных систем возникает на этапе усвоения знаний. Опыт, накопленный в таких областях как медицина или юриспруденция, сегодня не слишком легко ввести в компьютер. В настоящей главе для иллюстрации методов автоматического уточнения знаний используется одна медицинская экспертная система.

Студенты-медики на первых двух курсах обучения обычно получают громадный объем специальной информации. В последующие годы ожидается, что студенты научатся структурировать и использовать эту информацию, наблюдая за действиями более опытных практиков [6]. Формальное обучение методам решения проблем не получило распространения отчасти из-за ограниченной возможности описать такие методы. Обычно при создании компьютерных медицинских экспертных систем требуется, чтобы врач ввел в машину определенные им правила, позволяющие решать проблемы. Однако наука о том, как это делать, развита довольно слабо. Обычно то, как эти правила создаются одним человеком, радикально отличается от того, как они создаются другим, поэтому мало случаев, когда знания компьютерной экспертной системы могут пополняться кем-то, кроме ее разработчика. Если взять в качестве тезиса то, как студент становится экспертом, а в качестве антитезиса то, как экспертом становится компьютер, то в качестве синтеза можно вывести новую цель — попытаться создать вычислительные системы, способные усваивать большие объемы фактической информации, а затем структурировать ее, автоматически создавая эффективные стратегии решения задач.

Вообще, что позволяет решить некоторую задачу? На метауровне: что позволяет системе решения задач улучшать свою работу с накоплением опыта? Ответ на оба вопроса состоит в том, что пространство альтернатив, среди которых необходимо осуществлять поиск, должно обладать тем свойством, что изучение малой части этого пространства дает полезную информацию о неисследованных его частях. Нужно, чтобы для системы решения задач пространство альтернатив, определяющих задачу, выглядело как пространство, имеющее свойства непрерывности или предсказуемости. Темой настоящей главы является демонстрация того, что, опираясь на представление о непрерывности, можно разработать более совершенные методы усвоения знаний. Более конкретно рассматривается проблема наращивания объема знаний в области медицинской диагностики. Из литературы по нейрорадиологии было взято некоторое множество правил интерпретации томограммы мозга. Затем были исследованы методы уточнения этих знаний, чтобы они вернее интерпретировали изображения, применяя представление о непрерывности пространства возможных правил.

Настоящая глава разбита на четыре основные части: "Обзор области искусственного интеллекта", "Непрерывность", "Обзор обучающихся систем" и "Другой подход к уточнению знаний". В первой части, которая служит введением к определению непрерывности, содержащемуся во второй части, подчеркивается роль поиска и пространств поиска. В третьей части дается обзор ключевых моментов во многих проектах обучающихся устройств. В четвертой части работы других авторов и представление о роли непрерывности сведены вместе, позволяя создать новые методы усвоения знаний. Четвертая часть сама по себе распадается на четыре главных раздела. Сначала в ней описывается область интерпретации радиологических изображений, затем представлена простая схема улучшения весов и, наконец, предлагается более совершенная схема уточнения правил.

## 12.2. ОБЗОР ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

### 12.2.1. Поиск

Решение задач можно рассматривать как поиск. В распространенном подходе к поиску используются правила, данные и управление [19]. Правила оперируют данными, а управление работает с правилами. Например, в задаче коммивояжера данные представляют собой маршруты и их стоимости в некотором графе с весами, правила — это способы перехода от маршрута к маршруту, а управление связано с решением вопроса о том, какое правило применять и в какой момент.

Поскольку главной целью искусственного интеллекта является установление связи между методами решения задач и самими задачами, то необходимо охарактеризовать и то и другое. Обычно классы методов решения задач могут различаться по степени использования: 1) цепочки рассуждений в системах порождающих правил; 2) сопоставления и 3) процедуры порождения и проверки. В очень простой системе порождающих правил может быть два

правила: "ЕСЛИ  $a$  ТО  $b$ " и "ЕСЛИ  $b$  ТО  $c$ ". Если дан факт  $a$ , то с помощью некоторого алгоритма можно строить прямую *цепочку* рассуждений, ведущую сначала к  $b$ , а затем к  $c$ . Если  $c$  является решением, то алгоритм прекращает работу. При построении обратной цепочки к заданной цели, например  $c$ , поиск идет сначала к  $b$ , а затем к  $a$ . Если известно, что  $a$  — истинно, то обратный поиск прекращается. Степень пригодности прямой или обратной цепочки рассуждений, разумеется, зависит от решаемой задачи. Методы *сопоставления* часто являются важной составляющей частью стратегии решения задач. Например, в случае прямой цепочки рассуждений правило "ЕСЛИ  $a$ , ТО  $b$ " срабатывает, только если в числе данных имеется  $a$ . Причем  $a$ , которое содержится в данных, и то, которое имеется в этом правиле, не обязательно должны точно совпадать. Часто для того, чтобы установить наличие адекватного соответствия, необходимо прибегать к различным индуктивным и дедуктивным методам. При использовании *методов порождения и проверки* решаемая задача рассматривается как некоторое пространство поиска: порождается элемент этого пространства, который затем подвергается проверке. В зависимости от тех ограничений, которые используются в алгоритме порождения, возникают различные алгоритмы порождения и проверки. В тех случаях когда элементы пространства поиска связаны между собой структурой типа дерева, то, скажем, при использовании механизма "сначала в глубину" осуществляется возможно быстрое движение от корня дерева к его листьям.

Задача коммивояжера в принципе может быть решена с применением любой процедуры исчерпывающего поиска (полного перебора), однако на практике для ускорения процесса поиска необходимы эвристики и другие соображения, использующие специфику этой задачи. На самом деле, иногда дают такое определение дисциплине искусственного интеллекта: "Изучение методов решения экспоненциально сложных задач за полиномиальное число шагов, используя знания о предметной области" [23]. Такое использование знания возможно лишь при наличии общих сведений о задаче или пространстве поиска. В общем случае задача разрешима в той степени, в которой исследование части пространства поиска дает существенную информацию о характере оставшейся части этого пространства. При попытке охарактеризовать пространство поиска может быть задано множество вопросов. Разбивается ли задача достаточно хорошо на совокупность более мелких подзадач? Существует ли точная, непротиворечивая информация о задаче? Ожидается ли, что в процессе решения задачи человек будет взаимодействовать с вычислительной машиной?

### 12.2.2. Представление знаний

Знания могут быть как декларативными, так и процедурными. При декларативном представлении большая часть знаний представлена в виде статической совокупности фактов, сопровождаемой небольшим набором универсальных процедур для манипулирования этими фактами. При про-

цедурном представлении знания воплощены в виде выполнимых программ, которые в процессе работы выявляют смысл этого знания. В большинстве областей возникает необходимость в обоих таких типах информации. Декларативные представления могут быть подвергнуты дальнейшей классификации в соответствии со степенью их структурированности. Считается, что исчисление предикатов первого порядка (ИППП) менее структурировано, чем семантические сети.

Исчисление предикатов первого порядка является адекватной системой для представления любого знания, которое люди в состоянии формализовать. Кроме того, ИППП — стандартный язык для специалистов по математической логике. Существуют алгоритмы для преобразования любого выражения ИППП к каноническому виду, получившему название конъюнктивной нормальной формы. Методы выполнения дедукции в ИППП были всесторонне изучены. Для облегчения хранения информации ИППП было соответствующим образом расширено. В некоторых из таких обобщений ИППП допускают использование противоречивых дедукций или вероятностных дедукций.

Первоначально семантические сети были разработаны как способ представления слов английского языка. В семантической сети информация представлена в виде множества вершин, соединенных друг с другом множеством помеченных дуг, представляющих отношения между вершинами. Например, можно взять три вершины "мальчик", "человек", и "рука", соединив их дугами "является" и "есть-часть". Легко представить себе возможность создания произвольно сложных семантических сетей, но, по практическим соображениям, обычно стараются как-то ограничить число возможных вершин и соединяющих их дуг. Информация извлекается из семантической сети путем прохождения по графу. Диаграммы концептуальной зависимости являются вариантом семантической сети с небольшим числом типов дуг, например "поглощать" и "схватывать". Они используются главным образом в программах работы с естественным языком.

### 12.2.3. Экспертные системы

Двумя самыми старыми и самыми известными экспертными программами в искусственном интеллекте являются DENDRAL и MACSYMA [7]. Система DENDRAL получает масс-спектрограмму некоторого вещества и определяет его молекулярную структуру. Система MACSYMA помогает ученым при решении математических задач. Обе системы представляют собой крупные программы, создававшиеся на протяжении ряда лет. Обе программы часто используются профессионалами, для которых они и были разработаны. Принципы построения обеих систем довольно сложные, поскольку на этапе формирования этих проектов необходимые элементы искусственного интеллекта только начинали формироваться.

За последнее десятилетие было написано множество программ, позволяющих хранить в форме математического обеспечения профессиональный опыт специалистов. Одна из программ обеспечивает диагностику заболеваний

инфекционного характера на том же, если не на наиболее высоком уровне, чем это делают медики. Другая программа отыскивает признаки полезных ископаемых в данных сейсморазведки, третья — обеспечивает интеллектуальное машинное преподавание ряда учебных курсов. Был сделан вывод, что обычно недостаточно всего лишь несколько сотен правил, чтобы представить знания, необходимые для экспертного поведения в соответствующей предметной области. Удивительно, но области деятельности человека, опирающиеся существенно на здравый смысл, оказались теми областями искусственного интеллекта, в которых достигнуты наименьшие успехи.

#### 12.2.4. Обучение и творчество

Сразу же после появления электронных вычислительных машин начались эксперименты по их обучению. В настоящее время основное внимание здесь уделяется передаче машине опыта специалиста. Во многих экспериментах по обучению вычислительной машине предоставляется некоторое множество примеров, на основе которых машина должна выделить характеризующие их понятия. Обучающиеся программы должны при этом вести поиск среди различных концептуализаций, сохраняя в базе знаний лишь наиболее полезные варианты.

Были написаны программы, задачей которых было открытие интересных гипотез [13]. По-видимому, неизбежно создание машин, которые будут пытаться найти интересные задачи и решить их. Были созданы роботы, снабженные развитыми сенсорами и двигателями, обеспечивающими контакт с внешним миром. В конечном счете, центром интересов искусственного интеллекта может стать создание машин, которые будут играть активную, созидательную роль в нашем обществе.

### 12.3. НЕПРЕРЫВНОСТЬ

#### 12.3.1. Непрерывность в обучении

Когда обучение рассматривается в широком контексте как часть интеллектуального поведения, характеризующая многие живые системы, то роль *влияния и непрерывности* при обучении становится очевидной. В настоящем подразделе основные методы искусственного интеллекта, связанные с обучением, рассмотрены именно с точки зрения присущих им влияния и непрерывности. Является ли механизм обучения столь узким, как запоминание, или столь широким, как индукция, всегда можно увидеть, что важную роль в распространении влияния играет избирательный рост. Опирается ли метод обучения на числа и алгоритм системы PERCEPTRON, на символы плюс одиночные понятия в пространстве вариантов или символы плюс множество понятий в системе творческого характера, всегда в ходе обучения приходится наращивать некоторые структуры обучения, а другие при этом убирать. Для того чтобы процесс наращивания и удаления протекал достаточно гладко, система должна обладать свойством непрерывности. В системах

искусственного интеллекта часто используется единая схема представления для изучаемых примеров и изучаемых понятий. Тогда переход от примеров к понятиям может быть осуществлен добавлением или вычеркиванием нескольких символов таким образом, чтобы обеспечить гладкий или пошаговый переход от примеров к понятиям и от одного понятия к другому.

### 12.3.2. Влияние

Ретроспективный взгляд на мир наводит на мысль, что функцией живых существ является *влияние*. Организм может осуществлять свое влияние следующим образом:

1. Оставаясь живым, сохраняя себя.
2. Создавая семью.
3. Делясь информацией с другими или вступая с ними в контакт.
4. Создавая систему или "машину", которая имеет влияние.

Четвертый метод влияния одновременно и самый сильный, и самый загадочный, и наиболее труднодостижимый. Ученые часто пользуются третьей формой влияния, когда они публикуют статьи и одновременно читают лекции, участвуют в конференциях. Самосохранение и обзаведение семьей — это те цели, которые у человека являются общими со всеми развитыми живыми организмами.

Чтобы система оказывала влияние, она должна придать неожиданный поворот передаваемой ею информации. Простая передача чьей-то информации еще не обеспечивает ее сохранения. Однако заранее невозможно определить; какой именно "поворот" будет иметь в нашем мире успех. Поэтому нужно произвести какое-то число проб. Интеллектуальные (разумные) организмы стараются минимизировать ошибки, делая попытки, имеющие наибольшую вероятность успеха, а самым известным критерием успеха в сильно связанном мире является соображение о том, что хорошо сейчас, вероятно, подобно тому, что будет хорошо и в будущем. Этот вероятностный аргумент разрабатывается более формально и более глубоко Конрадом [3].

### 12.3.3. Число инверсий

Организм в длительном процессе проб и ошибок сталкивается с вопросом, какую попытку предпринять в следующую очередь. Проблемы всегда можно рассматривать как пространства поиска, а одной удобной формализацией такого пространства ( $SS$ ) является четверка величин  $\{S, O, I, G\}$  [1, 7]. Величина  $S$  — множество состояний,  $O$  — множество операторов, которые переводят из одного состояния в другое,  $I$  — начальное множество состояний, а  $G$  — множество целевых состояний. Задача состоит в выборе операторов, которые ведут из  $I$  в  $G$ . Обычно этот поиск опирается на некоторую эвристическую оценочную функцию  $f$ .

Непрерывность в пространстве поиска определяется степенью, в которой подобные состояния в  $S$  имеют подобные величины  $f$ . Подобие двух состоя-



ний в  $S$  обратно пропорционально числу операторов в  $O$ , требуемых для перехода между этими состояниями. Подобие в значениях  $f$  для двух состояний, если  $f$  принимает численные значения, может определяться просто близостью двух чисел.

Чтобы охарактеризовать предсказуемость  $f$  на  $SS$ , была предложена мера NIV (Number of Inversions — число инверсий) [20]. Предсказуемость часто связывают с вычислительной сложностью, но мера NIV непосредственно характеризует отношения в пространстве поиска. Мера NIV для  $(SS, f)$  указывает на степень, в которой подобные состояния имеют подобные значения  $f$ . Необходимо ввести следующие понятия: классы эквивалентности состояний, эвристические предсказания и инверсии. Класс эквивалентности  $S_i$  определяется следующим образом:

$$\{s | s \in S \text{ и } s \text{ может быть достигнуто из } I \text{ применением последовательности из } i \text{ операторов}\}.$$

Функция эвристического предсказания  $h$  на  $s \in S_i$  определяется так:

$$h(s) = \max \{f(w) | w \in S_{i+1} \text{ и один шаг связывает } s \text{ с } w\}.$$

Порядок, вызываемый  $h$  на  $S_i$ , сравнивается с порядком, накладываемым  $f$  на  $S_i$ , а чтобы придать этому сравнению количественный характер, вводят инверсии. Пусть  $a_1, \dots, a_n$  будет некоторой перестановкой множества  $\{1, \dots, n\}$ . Если  $i < j$ , а  $a_i > a_j$ , то пара  $(a_i, a_j)$  называется инверсией этой перестановки [11]. Например, в перестановке 2, 4, 1, 3 имеется три инверсии: (2, 1), (4, 1), (4, 3).

Для определения  $NIV_k(S_i, f)$  необходимо выполнить следующие три шага:

1. Упорядочить все  $s \in S_i$  так, чтобы  $f(s_1) \leq f(s_2) \leq \dots \leq f(s_m)$ .
  2. Вычислить  $h$  для каждого  $s$  и построить последовательность  $h(s_1), h(s_2), \dots, h(s_m)$ , где  $f(s_j) \leq f(s_{j+1})$ . Если  $f(s_j) = f(s_{j+1})$ , то  $h(s_j) \leq h(s_{j+1})$ .
  3.  $NIV_k(S_i, f)$  = числу инверсий в последовательности  $h(s_1), \dots, h(s_k)$ .
- $NIV_k(S_i, f)$  зависит от  $k, S_i, S_{i+1}$  и  $f$ .

Системы, основанные на использовании правил, такие как STRIPS и DCOMP [19], укладываются в представления настоящей главы. В качестве простого примера обозначим регистры компьютера через  $r_i$ , а их содержимое — через  $c_i$ . Множество  $S$  пространства  $SS$  состоит из  $\{(r_1, c_1), (r_2, c_2), (r_3, c_3), (r_4, c_4)\}$ , где  $c_i \in \{0, a, b\}$ . Начальным состоянием является  $\{(r_1, a), (r_2, b), (r_3, 0), (r_4, 0)\}$ . Для одной задачи обмена  $E_1$  целью является  $\{(r_1, 0), (r_2, 0), (r_3, a), (r_4, b)\}$ , а для другой,  $E_2$  целью является  $\{(r_1, b), (r_2, a), (r_3, 0), (r_4, 0)\}$ . В качестве оператора выступает правило присваивания, описанное Нильсоном [19]:

правило: assign  $(r_i, a, r_j, b)$

предусловие:  $(r_i, a) (r_j, b)$

вычеркнуть:  $(r_i, a)$

внести:  $(r_i, b)$

В качестве эвристики  $f(s)$  выступает число регистров в  $s$ , содержимое которых совпадает с содержимым соответствующих регистров в целевой задаче. Задачи  $E_1$  и  $E_2$  могут быть решены за четыре шага, однако  $E_2$  кажется более трудной, поскольку для нее характерен шаг перехода из состояния с высоким значением эвристики в состояние с низким значением. Вычисления NIV для  $E_1$  и  $E_2$  показывают, что NIV для  $E_2$  намного больше. Это подтверждает гипотезу, что NIV различает задачи по их трудности.

## 12.4. ОБЗОР ПРОЦЕДУР МАШИННОГО ОБУЧЕНИЯ

### 12.4.1. Среда обучения

В большинстве экспериментов по обучению среда изолирована от машины. Она предоставляет примеры или правила обучающейся машине, но машина не соревнуется с другими машинами в этой среде. Всякое влияние должно относиться к ограниченному контексту самой машины. Другими словами, именно на микроуровне внутренних частей обучающейся машины должна осуществляться эволюционная борьба, в ходе которой фрагменты программы сталкиваются друг с другом. На микроуровне взаимодействие обучающейся машины с окружающей средой настолько ограничено, что только по косвенным признакам можно говорить о наличии влияния.

### 12.4.2. Оптимальная стратегия

Холланд изучал стратегии обучения в разнообразных контекстах. Для многих случаев обучение можно рассматривать как поиск оптимума некоторой функции. Если эта функция связывает стратегии игровых автоматов с потерями, которые возникают, если эта стратегия используется на некотором конкретном множестве игровых автоматов, то естественно попытаться узнать, при какой стратегии эти потери минимальны.

Если считать, что игральные автоматы ведут себя как случайные переменные, то минимизирующей потери стратегией является такая, при которой все большее число испытаний приходится на автоматы, получающие большее вознаграждение. В своей основе эта стратегия сводится к воспроизведению с изменением плюс отбор наиболее подходящего. Экспоненциально большее число испытаний выделяется на те игральные автоматы, от которых ожидается наибольший доход.

Эта наилучшая стратегия имеет вид эволюционного процесса, при котором создается некоторая популяция из испытаний, получается обратная связь и испытания, оказавшиеся удачными, повторяются, а малоудачные имеют малую вероятность повториться в будущем. Все время должна сохраняться некоторая выборка игровых автоматов, дающих пока что плохой доход, но которые могут тем не менее на больших сериях испытаний оказаться выгодными.

Анализ этой ситуации с привлечением обсуждавшейся ранее меры влияния показывает, что некоторая часть популяции испытаний, сначала представ-

ляющая менее половины всей популяции, со временем начинает доминировать во всей популяции. После какого-то промежутка времени все испытания приходится на долю автоматов с самым лучшим поведением, и с этого времени и далее влияние перестает быть исключительным свойством этих игровых автоматов. Время от времени порождаются необычные испытания, которые, как правило, должны выделяться, очевидно, лучшим автоматам. Если эти "непослушные" испытания достаются автомату, который начинает приносить доход, больший ожидаемого, то, возможно, такие испытания начнут доминировать во всей популяции испытаний. Эти новые испытания можно рассматривать как потомки испытаний, которые были доминирующими в предыдущей игре, т. е. доминирующая подпопуляция может способствовать возникновению новой доминирующей популяции.

#### 12.4.3. Запоминание

Запоминание можно считать разновидностью обучения. Однако одного запоминания было бы совершенно недостаточно для большинства интересных задач обучения. При запоминании можно использовать разнообразные приемы для повышения его эффективности. Один из таких приемов — селективное забывание. Сэмюэл попытался воспользоваться запоминанием для оценки позиции в шахматах и в борьбе с переполнением памяти использовал одну из схем селективного забывания [25].

При селективном забывании стираются те элементы из памяти, значение которых по оценкам машины находится ниже некоторого порога. В память добавляются новые элементы, которым дается шанс доказать свою полезность. Иными словами, те элементы, которые годятся, остаются, и в некотором смысле они сохраняют сами себя. Такое сохранение или влияние ограничено, поскольку не используется процесс копирования элементов памяти с некоторым их изменением и последующим добавлением в память. Новые элементы поступают извне. Таким образом, схема запоминания позволяет окружающей среде эффективно влиять на обучающуюся машину, но не дает возможности обучающейся машине контролировать влияние внутри самой себя.

#### 12.4.4. Отладка знаний

Промежуточное по сложности положение между обучением путем запоминания и обучением на примерах занимает обучение путем отладки. Здесь обучающейся машине сообщается то, что она должна знать, но она должна преобразовать это знание из его входной формы в форму, которая позволит машине эффективно функционировать в ее окружении на основе полученных знаний. Иными словами, входным знаниям необходимо придать рабочий вид.

Мосту создал программу для вычислительной машины, которая получает советы и подвергает их переработке [18]. В этой программе рассматривается целый набор подпрограмм, предназначенных для обработки входной инфор-

мации, пока не будут найдены программы для соответствующего преобразования. Отобранные подпрограммы на самом деле получают возможность распространить свое влияние на остальную систему, поскольку теперь часть подпрограммы копируется и размещается вместо входного совета. Однако возникающее влияние носит лишь ограниченный характер. В более устойчивой (робастной) системе хорошо было бы иметь обучение, связанное также и с выбором обрабатывающих подпрограмм. Для этого можно было бы воспользоваться результатами работы существующих преобразующих подпрограмм, предоставив тем подпрограммам, которые оказывают хорошее влияние, возможность порождать новые подпрограммы. В этом случае то влияние, о котором шла речь ранее, также носило бы более робастный характер.

Система TEIRESIAS [4] позволяет человеку взаимодействовать с системой MYCIN и управлять улучшением ее базы знаний. Система TEIRESIAS способна просматривать базу правил системы MYCIN, чтобы ответить на вопросы "как?" и "почему?", которые помогают специалисту отладить базу знаний. Кроме того, система TEIRESIAS создает метазнания, касающиеся этих правил, так что эта система в состоянии непосредственно воздействовать на процесс уточнения знаний.

#### 12.4.5. Некоторые понятия из области обучения

*a) Система PERCEPTRON.* Система PERCEPTRON распознает классы образов. Системой PERCEPTRON можно считать функцию  $\Psi(x)$ , которая рассматривается как пороговая:

$$\Psi(x) = \text{да, если } a_1 \omega_1(x) + \dots + a_j \omega_j(x) \geq \Delta$$

$$\Psi(x) = \text{нет в противном случае.}$$

Величины  $a_i$  — целые числа, играют роль коэффициентов при  $\omega_i$ . Величины  $\omega_i$  — бинарные предикаты, которые некоторым простым образом "срабатывают" при предъявлении определенных изображений.

Интерес к системам PERCEPTRON обусловлен не столько их классифицирующими возможностями, сколько свойством обучения. Предположим, что у нас имеется два множества образов  $F_+$  и  $F_-$ . Предположим также, что существуют  $a_j$  для  $1 \leq j \leq n$ , такие, что линейная пороговая функция  $a_1 \omega_1(x) + \dots + a_j \omega_j(x) \geq \Delta$  истина, когда  $x \in F_+$ , и ложна, когда  $x \in F_-$ . Теорема сходимости для системы PERCEPTRON гласит, что PERCEPTRON, исходя из произвольного множества величин  $a$  и следуя простому алгоритму обучения через конечное число шагов, сходится к требуемым  $a$  [16].

Система PERCEPTRON работает, незначительно изменяя свои коэффициенты. Рассмотрим вектор  $A = (a_1, \dots, a_j)$ , который будем рассматривать как точку в  $j$ -мерном пространстве. Небольшое изменение в  $A$  соответствует небольшому изменению в  $a_1 \omega_1(x) + \dots + a_j \omega_j(x)$ . Для системы PERCEPTRON поиск можно считать идущим по поверхности функции  $G$ , где  $G(A) = A \times A^* / |A|$ , а  $A^*$  — вектор коэффициентов, который верно идентифицирует образы из множества  $F_+$ . Система PERCEPTRON обеспечивает перемещение

вдоль  $G(A)$  в направлении больших величин  $G(A)$  [16]. Поверхность  $G$  является гладкой.

Если под непрерывностью в случае дискретной функции понимать то, что небольшие изменения в области определения функции ведут к небольшим изменениям в области ее значений, то алгоритм обучения системы PERCEPTRON можно рассматривать как поиск по "непрерывной" дискретной функции. Такой непрерывности  $G(A)$ , однако, недостаточно для сходимости обучающегося алгоритма. Достаточными являются условие унимодальности  $G(A)$  и движение вверх по градиенту обучающегося алгоритма. Те же самые характеристики могут оказаться важными и для алгоритма, который обучается самим  $\omega$ .

*б) Обучение, направляемое данными.* Методы обучения одиночным понятиям, отталкивающиеся от данных, представлены пространствами вариантов Митчелла [17] и системой BACON Ленгли [12]. При использовании пространства вариантов наиболее конкретные и наиболее общие понятия определяют верхнюю и нижнюю границы частично упорядоченного пространства поиска. Обучающие примеры заставляют обобщать "конкретные" описатели и делать более специальными "общие" описатели. В этом процессе полезные описатели сохраняют в основном свой первоначальный характер. Конечно, наиболее исчерпывающее описание с большой вероятностью содержит части описаний, которые существовали с самого начала, а указанные влиятельные описатели обеспечивают такому подходу наличие характеристик влияния. В системе BACON признаки с некоторыми изменениями распространяются по пространству признаков. Когда некоторое множество признаков сходится к константному значению, то система заключает, что была открыта некоторая концепция. Манипулирующие признаками операторы оказывают влияние на систему и в конечном счете приводят к тому, что одна из характеристик пространства признаков становится доминирующей.

*в) Обучение, направляемое моделями.* Михальски описывает метод обучения одиночным понятиям, направляемый моделями [15]. Используемая модель содержит схожий с эволюцией механизм порождения и проверки гипотез. Влияние как понятие основывается на эволюционной идее, так что естественно, что в основанных на этой идее методах обучения влияние становится заметным фактором.

Обучающийся алгоритм Михальского [15] имеет четыре основных этапа: 1) выбрать начальную популяцию гипотез; 2) создать новые гипотезы путем обобщения старых; 3) выбрать лучшие гипотезы и использовать снова этап 2; 4) просмотреть результаты каждого этапа 3, чтобы решить, следует ли прекратить этот итерационный процесс.

Цикл из этапов 2 и 3 является циклом воспроизведения с изменением плюс отбор, который приводит к тому, что некоторая популяция гипотез распространяет свое влияние более успешно, чем другие.

Система XPLAIN Швартаута [27] первоначально создавалась для объяснения действий системы Digitalis Therapy Adviser (советчик по лечению дигиталисом). При построении адекватной модели терапии, Швартаут понял, что

он мог бы также строить конкретные правила, необходимые для лечебных рекомендаций. Его модель имеет набор компонент, обеспечивающих гладкие переходы в ходе построения правил терапии.

г) *Обучение, направляемое эвристиками.* В системе АМ Лената понятия генерируются и совершенствуются способом, который можно считать эволюционным [14]. Существующие структуры шлифуются с помощью эвристик, которые наращивают базу структур и добавляют к ней информацию, необходимую для лучшего соответствия ситуации. В зависимости от этого соответствия некоторая структура может стать или нет основой для построения других структур.

Структуры или понятия, которые в какой-то момент сохраняют себя, получают возможность сохраняться и дальше. Такие концепции включают в себя как математически специальные объекты типа "числа", так и эвристические понятия, например "построй крайние примеры". Когда создается что-то новое, то вместе с ним сохраняются и понятия обоих типов. Некоторые понятия неизбежно оказываются более полезными, чем другие, и в большей степени распространяют свое влияние. Когда происходит построение метаэвристики, то возможности для дифференцированного влияния возрастают в еще большей степени

## 12.5. ДРУГОЙ ПОДХОД К УТОЧНЕНИЮ ЗНАНИЯ

### 12.5.1. Эксперт по радиологии

Несколько статей было написано по использованию компьютеров для диагностики по рентгеновским снимкам. В Университете шт. Огайо (США) работа по применению искусственного интеллекта в медицине была распространена на задачу интерпретации радиологического обследования печени [2]. В Великобритании [10] были разработаны правила для автоматической интерпретации компьютерной томограммы (КТ) мозга. В целом медики и специалисты по вычислительной технике стали вкладывать больше усилий в развитие систем, способных облегчить интерпретацию радиологических данных. Кроме того, эта тенденция усиливается благодаря растущему применению цифровых машин для радиографического анализа.

Изучив несколько учебников по нейрорадиологии, таких как [22, 28], мы сконструировали около 200 правил интерпретации компьютерной томограммы мозга человека. Правила, выведенные из текста, варьировали в зависимости от характера этого текста. Обзор результатов проведенного исследования можно найти в работе [21]. В некоторых книгах выделялись данные, касающиеся истории болезни пациента, некоторые фокусировали внимание на локализации нарушения. В любом случае ширина дерева правил была больше его глубины. Эти данные по КТ подтвердили заключение Лената [14] о том, что, по-видимому, экспертные качества определяются не глубиной дерева правил, а тем, насколько оно ветвится.

Для работы с правилами для интерпретации КТ была создана система ЕТ

(Expert Tomographer – опытный томографист), которая имела структуру, во многом аналогичную структуре системы MYCIN [26]. В ней используются вероятностные знания, основанные на использовании правил. Коэффициенты уверенности обрабатываются так же, как в системе EMYCIN [29]. Отличия состояли в том, что использовались обратные цепочки рассуждений, а не прямые и применялись переменные пороги. С каждым суждением, появляющимся в антецеденте правила, связывается некоторый порог. Для того чтобы правило сработало, необходимо, чтобы факт или суждение имели бы коэффициент уверенности, превосходящий порог. Вес, связанный с суждением в консеквентной части правила, используется для ослабления заключения (см. [30], где можно найти пояснения с похожей терминологией).

### 12.5.2. Подбор весов

*а) Метод.* Взаимодействие правил, первоначально предназначавшихся для системы ЕТ, не носило осмысленного характера. Из нескольких проблем одной из самых простых была проблема, связанная с распространением уверенностей. То, что в середине дедукции выглядело как разумный порог и ослабление для правила, при реализации оказывалось узким местом на пути потока информации. Было неясно, что же именно надо сделать. Манипулирование весами с целью улучшения работы системы выглядело полезным, но тяжелым делом.

Началась разработка алгоритма для систематической корректировки весов (порогов и ослаблений). В одной очевидной стратегии предлагалось повысить ослабление (число, на которое умножается уверенность в заключении правила), когда некоторое суждение было получено при недостаточной уверенности, и понизить усиление, когда это суждение было получено при слишком большой степени уверенности. Эта логика может быть обобщена и на случай, когда правило не сработало или сработало, но не в свое время. Величина, на которую следует изменить вес правила, должна зависеть от прошлого опыта использования правила.

Алгоритм регулировки весов, или обучающийся алгоритм (УЧЕНИК) работает с использованием обучающего множества (ОМ). Каждый элемент ОМ представляет собой упорядоченную пару  $(I, D)$ , где  $I$  – множество начальных суждений и ассоциированных с ним уверенностей, а  $D$  – множество заключений и их уверенностей. Множества  $I$  или  $D$  представляют собой сочетания  $\{(\text{суждение}_1, \text{уверенность}_1), \dots, (\text{суждение}_m, \text{уверенность}_m)\}$ .

Кроме обучающего множества УЧЕНИК использует множество правил (ПРАВИЛА). ПРАВИЛА должны быть такими, чтобы ОМ допускало правильную обработку, если веса соответствующим образом подогнаны. Такого же рода предположение необходимо и для системы PERCEPTRON, а именно детекторы признаков должны быть адекватными [16].

Цель алгоритма УЧЕНИК – заставить ПРАВИЛА создать то же множество  $D$  для данного  $I$ , которое рекомендуется обучающим множеством. Каждое суждение в множестве ПРАВИЛА помечено специальным первым словом,

обозначающим некоторую категорию. В случае системы ЕТ такими словами являются ИСТОРИЯ, КТ и ДИАГНОЗ. В обучающем множестве нет необходимости задавать промежуточные заключения, которые могут делаться множеством ПРАВИЛА, поэтому с  $D$  в ОМ сравниваются только те заключения из правил, которые начинаются со слова ДИАГНОЗ.

У алгоритма УЧЕНИК имеется две фазы — раздача поощрений (РП) и регулировка весов (РВ). В каждой фазе из ОМ считывается некоторый элемент  $S_i$ , затем к  $I_i$  из  $S_i$  применяются ПРАВИЛА. Созданные при этом суждения, которые начинаются с предлога ДИАГНОЗ, помещаются в группу, названную  $\text{CONC}_i$ . Для последующего сравнения  $\text{CONC}_i$  и  $D_i^*$  суждения с приставкой ДИАГНОЗ сначала разделяются на три класса:

1. Те, которые встречаются и в  $\text{CONC}_i$  и в  $D_i$ .
2. Те, которые встречаются в  $D_i$ , но не в  $\text{CONC}_i$ .
3. Те, которые встречаются в  $\text{CONC}_i$ , но не в  $D_i$ .

Затем класс 1 разделяется еще на три категории:

- а) "Совершенными" являются такие, у которых коэффициент уверенности одинаков и в  $\text{CONC}_i$  и в  $D_i$ .
- б) "Слишком-много" — это те, у которых в  $\text{CONC}_i$  больше коэффициент уверенности, чем в  $D_i$ .
- в) "Слишком-мало" — это те, у которых в  $\text{CONC}_i$  меньше коэффициент уверенности, чем в  $D_i$ .

У каждого правила есть список со следующими идентификаторами: Совершенное, Слишком-много, Слишком-мало, Нужен-путь, Плохой-путь. Для данного  $S_i$  а фазе РП определяются последовательности правил, ведущие к каждому суждению в  $\text{CONC}_i$ . Свойства каждого правила для случаев а) — в), т. е. Совершенное, Слишком-много и Слишком-мало, пополняются  $S_i$ . Для суждения, которое в  $D_i$ , но не в  $\text{CONC}_i$ , в фазе РП определяется, какой путь вел бы из  $I_i$  в  $D_i$ . В свойство Нужен-путь тех правил, которые не сработали, а должны были бы сработать, добавляется атом  $S_i$ . Когда некоторое суждение из  $\text{CONC}_i$  не находится в  $D_i$ , то в фазе РП отмечается этот факт в свойствах Плохой-путь у тех правил, которые оказались на пути в  $\text{CONC}_i$ .

В фазе РВ изменяются веса с целью улучшения работы множества ПРАВИЛА на ОМ. В случае 1, а) нет необходимости в изменении. В случае 1, б) должны быть понижены ослабления тех правил, которые вели от  $I_i$  в  $D_i$  (и, возможно, понижен соответственно и порог). Степень, в которой уменьшается ослабление в каждом правиле для суждения  $p$ , бывшего на пути из  $I_i$  в  $D_i$ , пропорциональна одновременно (а) разности в коэффициентах уверенности в множестве  $\text{CONC}_i$  и в множестве  $D_i$  и (б) алгебраической оценке истории, хранящейся в  $p$ . Возможной оценкой для  $p$  является величина  $|\text{Слишком-много}| + |\text{Плохой-путь}| - |\text{Слишком-мало}| - |\text{Совершенное}|$ . Случай 1, в) обрабатывается аналогично случаю 1, б).

---

\*  $D_i$  — это то подмножество из  $D$ , которое согласно ОМ соответствует  $I_i$ . — *Прим. ред.*



В случае класса 2 алгоритм РВ стремится увеличить вероятность того, что сработает тот путь, который не сработал. В фазе РВ сначала определяются пути, который должен сработать, и что произойдет, если некоторая начальная последовательность правил из этого пути сработала бы. В этом случае порог следующего правила, которое должно было бы сработать, понижается, так что повышается вероятность запуска этого правила. В случае класса 3 было выведено суждение, которого не должно быть. В соответствии с этим порог правила, создавшего это суждение, повышается, так что в следующий раз это правило не сработает.

б) *Результаты.* Первая проверка алгоритма УЧЕНИК показала, что легко сконструировать такие обучающие множества, которые совершенно невозможно правильно идентифицировать с использованием имеющихся правил. В качестве простого примера отметим, что  $S_1 = \{(a; 0,7), (b; 0,6)\}$  и  $S_2 = \{(a; 0,4), (b; 0,8)\}$  не могут быть одновременно обработаны одним и тем же множеством правил. Более тонкие трудности могут встречаться всякий раз, когда два различных множества начальных суждений в ОМ имеют общие элементы.

Другое соображение, которое прояснилось после некоторого экспериментирования с весами, состоит в том, что все существенные веса могут быть локализованы в правилах, которые находятся на первом уровне. Правила первого уровня — это такие правила, антецедентами которых являются суждения, содержащиеся в  $I$  из некоторого  $S$ , принадлежащего обучающему множеству. Существенный вес — это порог, отличный от  $-1$ , или ослабление, отличное от  $1$ . Алгоритму УЧЕНИК приходится манипулировать весами во всем множестве ПРАВИЛА потому, что людям удобнее кодировать свои значения в весах правил, относящихся ко многим уровням.

Для эксперимента были отобраны 40 правил из множества 200 правил, которое мы упоминали ранее. Все эти правила относились к церебральным инфарктам. Было создано ОМ, проверяющее все главные случаи работы алгоритма УЧЕНИК, который представляет собой программу, содержащую примерно 500 строк команд на языке Лисп. После трех циклов применения РП и РВ множество ПРАВИЛА прекрасно работало на обучающем множестве.

Создание ОМ ручным подбором примеров является трудной работой. Планируется, однако, что обучающее множество будет строиться автоматически при рассмотрении клинических ситуаций в госпитале. В вычислительной машине хранится много изображений, полученных при компьютерной томографии, и алгоритм УЧЕНИК будет пополнен так, чтобы он был в состоянии читать такие файлы и автоматически создавать обучающее множество.

В настоящее время автор исследует роль непрерывности в пространствах поиска, возникающих при извлечении знаний для экспертных систем. На пути к уточнению алгоритма обучения сделана формализация пространства поиска  $SS$  и применена мера  $NIV$ . Выбрано некоторое множество  $R$  правил с порогами и ослаблениями (весами) в высказываниях каждого правила. Эта взвешенная база правил является начальным состоянием  $SS$ . Указанные

веса принадлежат некоторому небольшому множеству, и множество  $S$  состояний включает все возможные присваивания весов правилами из  $R$ . Обучающее множество из ассоциативно связанных "фактических" высказываний и "заключающих" высказываний является косвенным заданием целевого состояния. Каждое применение некоторого оператора  $O$  из  $SS$  изменяет ровно один весовой коэффициент.

Используя представления о пространстве поиска  $SS$  и мере  $NIV$ , можно количественно оценить качество работы алгоритма обучения и соответствующие характеристики пространства взвешенных правил. Исследования подтверждают, что требуемое представление пространства поиска обеспечивает низкое значение меры  $NIV$ . В контексте точно определенной меры  $NIV$  удается достаточно глубоко рассмотреть и вопросы упорядочения примеров в обучающем множестве и выбор шага алгоритма обучения.

### 12.5.3. Генетический алгоритм, эвристики и фреймы

Алгоритм, подобный алгоритму системы PERCEPTRON, может хорошо работать лишь при условии, что суждения или детекторы признаков используемых правил с уверенностью могут различать образы в обучающем множестве. Часто, однако, подбор такого множества суждений оказывается очень трудной задачей. Среди попыток автоматизировать процесс усвоения этих суждений следует упомянуть старую работу Сэмюэля [25]. Пространство поиска при обучении такого рода суждениям, по-видимому, не является унимодальным, по-выглядит вполне гладким. Генетический алгоритм Холланда [8, 9] рассчитан на работу с гладкими и многомодальными пространствами поиска. Главным препятствием на пути реализации обучающих схем с применением этого алгоритма является проблема представления. Генетический алгоритм только тогда работает хорошо, когда удается найти представление, в котором смысл имеют многие подстроки символов.

Эвристики и фреймы являются основными компонентами системы Лената, рассчитанной на поиск нового [13]. Ленат считает, что ключом к успеху использованного им представления является то обстоятельство, что многие комбинации элементов оказываются полезными. Это требование вполне аналогично требованию, высказанному в случае системы Холланда. Система Лената открыла множество интересных понятий и эффективных планов. Система Холланда имеет весьма устойчивую математическую базу. И Ленат, и Холланд чувствуют, что настало время объединить то лучшее, что было ими достигнуто порознь.

Хотя генетический алгоритм предназначен для работы со строками символов, мы применяем его к представлению, ориентированному на фреймы. Каждый фрейм соответствует правилу системы ЕТ, но теперь каждое суждение заменено иерархией слотов. Имеются слоты для данных о компьютерной томограмме (КТ), об истории болезни пациента (ИБП) и для пациента (ДП). В слоте имеются подслоты, например КТ включает ПОЛОЖЕНИЕ, ПЛОТНОСТЬ и ФОРМУ. Слот ИБП включает НАЧАЛО и ПРОДОЛЖИТЕЛЬНОСТЬ,

а слот ДП – Васкулярный, Инфекционный и Неопластический. В подслотах могут быть свои подслоты. Например, подслот ПОЛОЖЕНИЕ включает подслоты артериальный, лобный и вентрикулярный, а ПЛОТНОСТЬ – гипоплотный, равноплотный и гиперплотный.

Автором был накоплен богатый словарь терминов для организации иерархий слотов. Кроме того, правила ЕТ переводятся во фреймы. Осуществляется реализация генетико-эвристического алгоритма для полуавтоматического улучшения фреймов. При применении генетического алгоритма необходимо, чтобы фреймам выдавался кредит, пропорциональный качеству их функционирования. Для этого используется метод "пожарной команды" Холланда [9]. Генетический алгоритм работает с большими популяциями, в которых множество отдельных членов (фреймов в нашем случае) не сильно отличаются друг от друга. Соответственно каждое из прежних 200 правил распадается на несколько фреймов, являющихся не сильно отличающимися друг от друга вариантами.

Для иллюстрации работы обучения суждениям рассмотрим примерный набросок двух фреймов на рис. 12.1. Предположим, что генетический алгоритм должен скопировать и пересечь их друг с другом с использованием операции кроссинговера. Одно возможное пересечение могло быть между компонентами истории болезни, связанными с продолжительностью наблюдения. В результате пересечения должен быть порожден потомок фрейма 1 с продолжительностью "1 неделя" и потомок фрейма 2 с продолжительностью "6 недель".

Полезные эффекты, связанные с таким пересечением, могут быть достигнуты и другими способами. Из медицины известно, что на рентгеновском снимке скопление крови вначале является гиперплотным, а по прошествии примерно двух недель становится изоплотным. Программа, располагающая такими знаниями, могла бы так изменить фрейм 1, чтобы гиперплотность ожидалась не в течение 6 недель, а лишь в первую неделю. Если это знание

Фрейм 1	Фрейм 2
ЕСЛИ	ЕСЛИ
КТ нарушение	КТ нарушение
гиперплотный	гиперплотный
ИБП	ИБП
нейросимптомы	нейросимптомы
< 6 недель	< 1 недели
ТО	ТО
ДП	ДП
инфаркт	гематома

Рис. 12.1. Два фрейма, предназначенные для оператора кроссинговера

известно, то компьютер должен им воспользоваться. Генетический алгоритм предназначен для случаев, когда подобное знание недоступно. С другой стороны, можно попытаться в системе воспользоваться обоими методами — генетическим и с применением знаний, если ситуация это позволяет.

Другой альтернативой генетическому алгоритму является рассуждение по аналогии. При таком рассуждении изучаются два события с тем, чтобы извлечь из одного события информацию, которая могла бы быть полезной для другого. Для нашей иллюстрации можно считать, что генетический алгоритм извлекает информацию из фрейма 2 и передает ее во фрейм 1. В некотором смысле генетический алгоритм ведет рассуждение по аналогии. Этот алгоритм не располагает конкретными знаниями, позволяющими ему решить, что данный перенос информации лучше, чем какой-либо другой, поэтому он больше подходит для ситуаций, когда мало известно о том, какие именно изменения могут быть полезны.

В радиологическом эксперименте автор предлагает использовать комбинацию рассуждения по аналогии и генетический алгоритм для уточнения слотов. Несколько сотен фреймов, совокупность иерархически организованных термов и некоторые специальные для радиологии эвристики будут включены в программу, использующую генетический алгоритм и эвристики для выработки улучшенного множества фреймов. Первоначально фреймы не будут содержать весов (порогов, ослаблений и коэффициентов уверенности), обсуждавшихся в предыдущем разделе. Система ЕТ сначала будет обучаться выдавать правильные диагностические классы, а затем определять их внутреннюю структуру.

## 12.6. ОБСУЖДЕНИЕ

В обучающихся алгоритмах искусственного интеллекта, рассмотренных в настоящей главе, часто прибегают к тому, что Диттерих и другие называют "трюк использования единого представления" [5]. Это означает, что вход от окружающей среды, представление этого входа в машине и окончательное описание, даваемое машиной, — все основано на одних и тех же символах и том же синтаксисе. Например, в пространстве вариантов Митчелла и системе INDUCE Диттериха используется исчисление предикатов первого порядка (ИППП), где несколько предиктов и атомов применяются для представления входа, ассоциаций и выхода. Этот трюк использования единого представления облегчает достижение непрерывности.

В работе обучающихся программ искусственного интеллекта часто происходят обобщение или конкретизация некоторого понятия. Это может быть достигнуто добавлением или стиранием символов из выражения ИППП. Добавления и стирания ограничены механизмами обучающейся программы, поэтому, как правило, результирующие понятия оказываются очень близкими к родительским. Выражение ИППП само по себе является структурой, а интерпретация этого выражения — в некотором смысле функцией. В системах искусственного интеллекта, рассмотренных в настоящей главе, такая структурно-функциональная связь способствует непрерывности.

Многие области науки и техники сотнями лет отшлифовывали сжатые способы, позволяющие отмечать, связывать и проверять то, что сегодня мы называем идентификаторами этих областей. Возьмите например, математику: аксиома, правило вывода, лемма, теорема, утверждение. Или медицину: заболевание, лечение, симптомы. Искусственный интеллект намного моложе математики или медицины, и здесь подходы менее четко очерчены и определены. Из-за этого затрудняется сравнение результатов экспериментов в области искусственного интеллекта.

В искусственном интеллекте обычно выбирают какую-то четко очерченную область, а затем применяют некоторые принципы построения программы, которая в этой области ведет себя разумно. Обычно эти принципы являются интересными и интуитивно оправданными. Как правило, программа имеет определенный успех, подтверждая целесообразность используемых принципов. Необходима дополнительная работа, чтобы дать строгое определение интеллекту и найти необходимые и достаточные условия интеллектуального поведения. В настоящей главе интеллект связывается с влиянием. Под влиянием понимается разновидность избирательного роста. Необходимым, хотя и труднодостижимым условием для влияния является непрерывность. Располагая точным определением непрерывности, исследователь в состоянии изучать принципы интеллекта и добиваться количественных результатов, позволяющих провести сравнение различных экспериментов между собой. Такого рода сравнения в науке совершенно необходимы, поскольку результаты одного множества экспериментов должны использоваться в качестве ограничений для пространства параметров другого множества измерения.

Одним из барьеров на пути широкого использования экспертных систем является малочисленность технических приемов уточнения, совершенствования знаний. В настоящей главе разработан метод УЧЕНИК, позволяющий подбирать веса в вероятностной экспертной системе, основанной на использовании правил. Корни этого метода лежат в области системы PERCEPTRON, но работа нашей системы отличается от действия системы PERCEPTRON в двух отношениях. Во-первых, используется несколько уровней правил. Во-вторых, используется несколько форм эвристик. Веса, которые в начале характеризуют правила, выражают то мнение об этих весах, которое имеет у эксперта. Программа УЧЕНИК очень осторожно изменяет эти веса. Эвристики также применены в этом процессе, позволяя учесть историю изменения каждого правила сразу с нескольких точек зрения.

Соображения непрерывности могут обеспечить автоматическое усвоение знаний, направляя выбор представления для пространств поиска и выбор алгоритмов поиска. База правил, созданная для интерпретации компьютеризированных аксиальных томограмм, нуждается в уточнении. Такое уточнение может производиться на нескольких уровнях, включая веса правил и высказываний. Пространство правил, в котором необходимо вести поиск с целью улучшения правил, выглядит достаточно гладким. Показано, что классические алгоритмы типа алгоритма системы PERCEPTRON или генетического

алгоритма хорошо работают в гладких пространствах. Сложность состоит в том, что в них обычно не предусмотрено использование знаний из данной предметной области. В настоящей главе автор предложил два метода усвоения и уточнения знаний для медицинской экспертной системы, в которых используются достоинства классических алгоритмов. Кроме того, вводится некоторая зависимость от самих знаний, что необходимо для практического успеха подобных систем.

## Глава 13

### ИНДУСТРИЯ ЗНАНИЙ

*Том Стониер*

#### 13.1. ЭКОНОМИКА ИНФОРМАЦИИ

##### 13.1.1. Роль информации как компоненты экономики

Без притока информации невозможна экономическая деятельность. Когда в каменном веке охотник шел по следу зверя, он пользовался информацией. Средневековый крестьянин пашет поле или его жена прядет шерсть — это использование информации. Часто информация в накопленном виде оказывается воплощенной в историю изобретений, таких как плуг или прядильное колесо. Даже животные, перенимая опыт друг у друга, можно сказать, используют информацию. Например, голубая синица, не равнодушная к блестящим на солнце предметам, научилась вскрывать клювом молочные бутылки, стоящие у порога дома, и снимать с молока сливки.

В этой главе понятие информации употребляется в самом широком смысле. С одной стороны, имеется изолированный факт. В простейшем случае это двоичная информация типа все-ничего, нуль-один, да-нет. На более высоком уровне информационные образы составляют знание и образы знаний, т. е. мудрость. Информация может существовать вне мозга человека, например, в компьютере или в головке голубой синицы. На самом деле, информация может заключаться просто даже в организованных структурах окружающего мира, например в молекуле ДНК. Однако этот аспект информации, раскрытый в другой нашей работе [Стониер, 1984], не является содержанием настоящей главы.

Полезно провести некоторую аналогию между информацией и текстильной отраслью. Необработанная шерсть равносильна данным, она может быть spun в нить, которая представляет собой информацию. Из нити, в свою очередь, может быть соткано платье, так же как из информационных образов может ткаться знание. На ткацком станке можно из шерстяных нитей выткать ткань, точно так же, как компьютер может из информационных нитей создать знание. Из ткани, в свою очередь, может быть сшит костюм, а знание, должным образом проанализированное и систематизированное,

может вылиться в мудрость. В настоящей главе термин "информация" охватывает все — от простейших данных до мудрости. Понятие "информация" не следует путать с понятием "интеллект", который анализирует информацию, чтобы в дальнейшем ее использовать. Не следует также путать получение информации или само ее существование с передачей по разнообразным каналам связи.

### 13.1.2. Знания — важная входная компонента современного производства

Информация оттеснила землю, труд и капитал, заняв место наиболее важного входного ресурса современного производства. Она снижает потребность в земле, труде и капитале, уменьшает расход сырья и расход энергии. Она вызывает к жизни новые виды производства. Информация сама является товаром и служит исходным сырьем для бурно растущего сектора экономики — индустрии знаний.

Каждая машина заключает в себе целую историю нововведений и изобретений. Такое накопление информации так же важно, как и накопление капитала. Добавив информацию, можно сделать более ценным любой предмет или материал. Пустующие, заброшенные земли плюс информация превращаются в высокоурожайные земли. Неподготовленные рабочие при наличии образования превращаются в искусных производителей. При наличии информации мертвый капитал становится выгодным капиталовложением. Бесплезная энергия солнечных лучей или океанских волн может быть превращена в полезную работу, если знать, как это сделать. Информация в состоянии не только придать дополнительную ценность другим ресурсам, таким как земля или труд, — ее собственная ценность также может быть при этом увеличена. Анализируя, делая перекрестные ссылки, отбирая, сортируя, резюмируя или каким-либо иным способом организуя данные, их можно превращать в информацию. Информация имеет большую ценность, чем данные, ставшие руководством для достижения определенных целей. Дополнительная информация, получаемая людьми в процессе образования, повышает их ценность. Именно по этой причине образование является непрерывно расширяющейся индустрией, на которую в конце концов будет расходоваться значительно больше национального продукта, чем на любую другую экономическую деятельность, и которая поглотит большую часть рабочей силы, чем любой другой наниматель.

Экономист Ф. Мэхлап помог определить понятие индустрии знаний, указав, что образование — это самая крупная ее отрасль. Мэхлап имел в виду не только официальные институты образования, но также и образование, получаемое дома, в церкви и на военной службе. Американский социолог Д. Белл расширил наблюдения, сделанные Мэхлапом, и дополнил высказанные им концепции. Белл также высказался за "центральное" место знаний в современной экономической системе. Уже позже М. Порат модернизировал эти пионерские исследования Мэхлапа и Белла. Порат разделил информационный сектор

экономики на две части: "первичный информационный сектор", включающий фирмы, поставляющие информацию на рынок (т. е. другим фирмам и потребителям) в виде товара или услуг и "вторичный информационный сектор", который включает все информационные услуги, предназначенные для внутреннего потребления правительственными учреждениями и частными неинформационными фирмами. Например, промышленной фирме приходится еще держать крупные административные подразделения, и на самом деле, как результат достижений в области технологии производства, административные подразделения часто потребляют гораздо большую часть трудовых ресурсов, чем само предприятие.

Согласно данным Пората [5] первичный информационный сектор США в 1967 г. потреблял 25 % валового национального продукта, а вторичный — 21 %. Следовательно, на общую информационно-экономическую деятельность приходилось 46 % валового национального продукта, производимого в США. Прошло полтора десятка лет с того времени, и информационная технология (ИТ) получила формальное признание. Чтобы составить представление о развитии ИТ, отметим, что в 1967 г. в США в пользовании находилось примерно 15 тыс. компьютерных терминалов, а к 1980 г. их число, видимо, превысило 2 млн. Компьютерные базы данных были настолько редки в 60-х годах, что прямой доступ к ним для поиска данных едва ли в то время существовал. В 1970–1971 гг. таких систем уже насчитывалось 100–200 тыс., а к 1980 г. их стало по меньшей мере 2 млн.

### 13.1.3. Информация как ресурс

Информацию можно продавать непосредственно в виде электронных баз данных или в форме газет, но она может продаваться и как основа для создания благ. Так обстоит дело, когда изобретатель или автор продает соответственно патент или авторские права или же предоставляет лицензию другим лицам на право использовать изобретение. Хороший патент может быть гораздо ценнее, чем обладанием целым предприятием. Итак, информация сама по себе может создавать богатства.

Когда информация поступает к людям, то мы видим, что образованный и квалифицированный трудящийся обычно работает с большей производительностью. Этим объясняется то, что общество отдает предпочтение образованным квалифицированным людям, тогда как необразованные и неквалифицированные оказываются в худших условиях. Рабочего с лопатой заменяет бульдозерист, а конторского служащего — программист ЭВМ.

Университеты обеспечивают занятость и способствуют росту возможностей человека. В них также вырабатывают новые идеи и новые промышленные направления. В Кремниевой долине, Оксфорде и Кембридже много промышленных предприятий, которые непосредственно черпали и идеи, и людей из университетов. Большой интерес для информационной технологии представляют фундаментальные исследования в области физики твердых тел, которые непосредственно привели к созданию транзистора. Еще до феномена



Кремниевой долины в начале 50-х годов полупроводниковая промышленность концентрировалась в окрестностях дороги № 128 в Бостоне (США). Многие основатели новых фирм были выпускниками Массачусеттского технологического института в Гарвардского университета и часто набирали штат в местных университетах или приглашали специалистов из университетов в качестве консультантов.

Сказанное должно прояснить некоторые аспекты, свидетельствующие о важности информации и знаний для развития современной, постиндустриальной экономики\*. Что нужно сегодня — так это новый раздел в экономической науке — информационная экономика. Она должна заниматься поиском ответов на целый ряд вопросов. Как измерять информацию в экономических терминах? Как измерять производительность труда людей, работающих с информацией? Можно довольно легко установить, какой процент валового национального продукта расходуется на информацию, но как измерить долгосрочное влияние информации на экономику? Как следует определять экономические последствия изобретения или поступления информации? Какая часть годового роста национального продукта или повышения уровня жизни (что не всегда одно и то же) обусловлена успехами в области информации или образования?

### 13.2. ПОЯВЛЕНИЕ ИНФОРМАЦИОННЫХ РАБОТНИКОВ

Возможно, самым первыми людьми, которые стали частично или полностью зарабатывать на жизнь как информационные работники, были врачи и жрецы. Постепенно они превратились в профессионально мудрых людей. В Древней Греции такие люди могли зарабатывать очень много денег. Адам Смит подсчитал, сколько могли получать наиболее известные из древних информационных работников — греческие философы. Он пришел к заключению, что Сократ, например, получал десять мин от каждого ученика, а имел он сто учеников. Таким образом, он зарабатывал тысячу мин, или свы-

---

\* Постиндустриальная экономика или новое индустриальное общество — это термины, используемые некоторыми буржуазными экономистами [БСЭ, 1975. — Т. 20. — С. 1241] для обозначения того нового этапа развития капитализма, который начался в США в 60-е годы, годы очередного циклического подъема экономики этой страны. Этот этап характеризует дальнейшее укрепление монополий и их фактическое сращивание с государством. При этом государственное регулирование капиталистического воспроизводства осуществляется в интересах правящей монополистической верхушки и направлено на обеспечение наиболее благоприятных условий получения прибыли, расширение и захват новых рынков, повышение конкурентоспособности товаров. Внешне это выражается в возрастании темпов научно-технического прогресса, дальнейшем повышении роли техноструктуры. Крупные корпорации все шире используют электронно-вычислительную технику, в большем масштабе развертывают научные исследования и опытно-конструкторские разработки. Следует иметь в виду, что практически все дальнейшие рассуждения автора данной главы относятся именно к государственно-монополистическому капитализму. — *Прим. ред.*

ше трех тысяч фунтов стерлингов (по курсу XVIII века) за каждый цикл лекций. Другой пример — Горгий, который преподнес в дар Дельфийскому храму свою собственную статую, выполненную в натуральную величину из чистого золота. Судя по различным источникам, Плутарх, Платон, Аристотель — все были состоятельными людьми.

Сегодня информационные работники составляют большинство трудового населения, т. е. намного больше людей зарабатывают себе на жизнь, имея дело с информацией, по сравнению с числом людей, занятых каким-то другим делом. Даже те, кто работает на фермах, заводах или имеет дело с машинами, — все в большей степени нуждаются в пополнении своих знаний, чтобы не отставать от новой техники. Имеется шесть категорий информационных работников:

1. Относительно небольшая группа людей, связанных с созданием новой информации. В нее входят ученые, художники, архитекторы, конструкторы и статистики, т. е. те, кто создает либо новую информацию, либо новые знания на основе существующей информации.

2. Обширная группа людей, вовлеченных в передачу информации, включает в себя, во-первых, работников справочных телефонных служб, почтовых работников, машинисток и т. д., а во-вторых, поставщиков техники связи, журналистов и других работников, взаимодействующих с широкими массами, и, наконец, гигантскую индустрию разнообразных воспитателей: преподавателей, священнослужителей, руководителей производственного обучения и т. д.

3. Третья группа связана с хранением и извлечением информации и включает в себя работников, занимающихся учетом и регистрацией, библиотекарей и, конечно, программистов ЭВМ.

4. В четвертую группу входят профессионалы: юристы, врачи, бухгалтерские работники, т. е. люди, применяющие информацию, которая накопилась у них в голове, и предлагающие другим людям свой опыт.

5. Существует большая группа людей, работающих с информацией, которые когда-то были частью трудового населения, а теперь уже к нему не относятся. Это школьники и студенты. Следует напомнить, что в Великобритании вплоть до первой мировой войны в среднем каждый двенадцатилетний подросток и большая часть десятилетних проводили по крайней мере половину своего времени на шахтах и фабриках, причем девочки работали на фабриках, служили продавцами и прислугой. Они были частью рабочей силы, а теперь заняты в сфере восприятия (поглощения) информации.

6. Наконец, мы должны обратиться к организационным работникам. Современные производства представляют собой комплексное взаимодействие труда, капитала, земли, энергии, материалов и самой разнообразной техники со столь же сложными системами транспорта, связи и распределения. В современном производстве с момента посадки зерна пшеницы в землю до того момента, когда ломоть хлеба можно будет обжарить в домашнем тостере, продукт участвует в сложной цепочке взаимодействующих процессов. Чтобы учесть все эти различные аспекты производства, требуются не только производители и потребители, но и множество посредников и организационных работников — менеджеров.

Менеджеры обладают организационным опытом. Они создают блага, составляя систему работать. Они участвуют в создании *новых* благ, поставляя информацию существующим организациям или производствам и тем самым

способствуя снижению производственных затрат либо созданию новых продуктов или новых услуг. Они должны вовлекать в единую организацию производителей, работают ли те на заводах, фермах, шахтах или, если хотите, в художественных студиях или школьных классах, и целый коллектив других информационных работников с тем, чтобы производимый продукт достиг своего потребителя. Их средствами являются телефон и карандаш или пишущая машинка, процессор для обработки текстов, телекс и т. д. (за работу которых часто отвечают другие информационные работники). Организационные работники принадлежат к наиболее высокооплачиваемым членам постиндустриального общества, поскольку они выполняют очень важную функцию не просто передавать информацию от отдельного лица или организации другому лицу или другой организации, а принимать важнейшие решения в отношении того, кому, куда и когда должна направляться информация. В больших организациях менеджеры руководят работой множества других информационных работников, которые сами являются экспертами в области финансов, учета, юриспруденции, промышленных связей, рекламы, контакта с населением, планирования и прогнозирования, работой ученых и инженеров, связанных с исследованиями и разработкой, создателей новой продукции, руководителей профессионального образования и подготовки, агентов по снабжению и специалистов по сбыту, руководителей контор и персонала, связанного с обработкой данных. Ясно, что объединение такой разнообразной смеси специальных знаний в единое целое уже само по себе требует большого опыта. Поэтому не удивительно, что навыки организатора в нашем обществе признаются наиболее важными и наиболее ценными информационными навыками.

Не может быть сомнений, что существующая система образования часто оказывается непригодной для подготовки специалистов, способных руководить постиндустриальной экономикой. Во-первых, обучение оказывается неадекватным или, говоря точнее, устаревшим, не способным совладать с быстро возрастающими потоками информации во всех сферах деятельности. Человека с лопатой в руках уже заменили те, кто знает, как управлять работой механического канавокопателя. Женщину, работающую в отделе регистрации, заменяет программист ЭВМ. Тенденция такова, что опытных работников готовят очень мало и слишком поздно, причем их продолжают выпускать уже и после того, как в них отпадает необходимость. Возьмите, например, программистов. По мере развития языков высокого уровня и создания "дружественных" вычислительных систем будет отпадать необходимость в профессиональных программистах, работающих с языками самого низкого уровня. Кажется сомнительным, чтобы система образования или общество в целом хорошо понимали, когда же следует прекратить выпуск таких специалистов по программированию, потребность в которых уже упала.

Второй недостаток системы образования носит более серьезный характер. Он связан с неспособностью давать людям достаточно широкую подготовку, которая позволяла бы им понимать проблемы, стоящие в других областях, и эффективно взаимодействовать с соответствующими специалистами.

По-видимому, правильным будет сказать, что в Великобритании и, в несколько меньшей степени, в других странах Запада большинство решений принимаются людьми, не разбирающимися в технике. В то же время большинство инженеров, специалистов по обработке данных, ученых, т. е. людей технически образованных, характеризует общая безграмотность в гуманитарном и социальном отношениях. Результатом является то, что торговые работники не могут контактировать с техниками, те не могут контактировать с финансистами и т. д., а руководство обычно пропускает удачные возможности или на самом деле оказывается полностью не у дел, когда появляются новые системы, воспринять которые они просто не в состоянии.

### 13.3. БУДУЩИЕ ИНДУСТРИИ ЗНАНИЙ

#### 13.3.1. От энергетических машин к машинам информационным

Промышленная революция привела к созданию дополняющих мускулатуру человека механизмов, которые поднимают, переносят, вращают предметы. Все эти машины и механизмы заменяют ручные операции. Напротив, революция в электронике привела к созданию машин, как бы дополняющих нервную систему человека. Радио и телефон являются продолжением органов слуха. Кино, первоначально немое, а затем использующее звук, и, разумеется, телевидение расширяют возможности и зрения, и слуха. И наконец, компьютер дополняет возможности мозга человека.

Компьютер представляет собой определенную разновидность машинного мозга — это устройство, которое позволяет совершать вне мозга человека те действия, которые прежде можно было сделать только внутри него. Мы все еще находимся на самых первых этапах создания искусственного интеллекта. Однако в настоящее время предпринимаются серьезные шаги в направлении объединения энергетических и информационных машин.

#### 13.3.2. Объединение энергетических и информационных устройств

Появились роботы! Вначале это были примитивные устройства, предназначенные для выполнения некоторых шаблонных задач. Действие этих первых электромеханических устройств можно было изменить, только изменяя монтаж проводов или какую-то другую часть их конструкции.

Эти первые электромеханические устройства в настоящее время быстро вытесняются электромеханическими устройствами, которые управляются компьютером. Появилась возможность изменять поведение или характер работы устройства простым изменением программы ЭВМ. Именно эта возможность перепрограммирования таких роботов второго поколения придает им значительно большую универсальность по сравнению с устройствами первого поколения. Однако и они еще слишком примитивны. Если робот второго поколения запрограммирован так, чтобы взять некоторую деталь, а ее на месте не оказывается, то он по-прежнему продлевает всю последовательность

необходимых движений, хотя на самом деле он при этом будет "хватать" просто воздух.

У роботов третьего поколения имеются встроенные сенсоры и системы обратной связи, позволяющие им контролировать состояние среды, в которой они действуют. Они в состоянии "разумно" реагировать на окружающие условия, что выражается в способности анализировать информацию об окружающей среде и соответствующим образом реагировать. В настоящее время мы входим в этот третий период эволюции роботов, когда и сенсоры, и обратные связи все более и более усложняются.

Этот процесс приведет к созданию роботов четвертого поколения — сетей роботизированных устройств, связанных как бы искусственной нервной системой сенсорных механизмов и механизмов обратной связи, так что все они взаимодействуют друг с другом и используют экспертные системы. Теперь отдельные роботы будут кооперироваться друг с другом, выполняя операции на линии сборки, в частности корректируя любые ошибки, которые могут возникнуть в ходе работы. В конечном счете это означает, что каждая линия сборки, а затем и каждый завод сами становятся сложными машинами. Сегодня уже нет необходимости держать оператора для каждого станка. В будущем отпадет нужда в управлении работой целого завода. Нужны будут лишь операторы, *поддерживающие* фабрику в рабочем состоянии и при необходимости *перестраивающие* ее работу.

### 13.3.3. Становление информационной технологии

Информационная технология — это некоторая смесь из вычислительных машин и систем связи, т. е. то, что французы называют "телематикой". Она содержит широкий спектр устройств и методов для манипулирования информацией вне мозга человека. В нее входят компьютеры и программное обеспечение, периферийные устройства и широкий спектр средств связи — от оптических волокон до спутниковых систем связи.

Сфера человеческой деятельности, получающая выгоды от применения информационной технологии, растет не по дням, а по часам. Не вызывает удивления то, что впервые информационная технология заявила о себе в отраслях, связанных с переработкой информации. Банки теперь уже не занимаются пересылкой больших денежных сумм, заменив этот процесс обменом информацией о кредитах. Точно так же финансовые и страховые компании, многие правительственные учреждения заняты сбором, выверкой, анализом, выдачей и обменом информацией.

Среди правительственных учреждений Министерство обороны является, без сомнения, главным потребителем информационной технологии, и не только потому, что у него имеются те же проблемы, что и у любого другого государственного учреждения, но и потому, что отныне вся военная структура все в большей и большей степени будет опираться на информационную технологию.

Другой сферой интересов государственных учреждений является здравоохранение. Здесь снова возникает проблема получения и переработки информации

мадных объемов информации. Но теперь ситуация значительно улучшилась, так как компьютеры и экспертные системы стали применяться для диагностики и лечения и оказывают большую помощь врачам и фармакологам.

#### 13.3.4. Приход экспертных систем

Врачи — одни из многих потребителей информационной технологии, которая быстро завоевывает свое место в разных профессиях. Беседа с пациентом с помощью вычислительной машины, в ходе которой компьютер использует экспертные системы как для расспроса больного, так и для анализа полученных данных, начинает выходить за рамки научных экспериментов и получает широкое применение. То же самое начинает происходить в юриспруденции. Вскоре это станет обычным делом для большинства профессий. В будущем всем информационным работникам высокого уровня станут доступны в том или ином виде интеллектуальные базы данных.

По мере полной автоматизации конторской работы у менеджеров появляется что-то вроде "неограниченного стола". Последнее означает, что, используя компьютеры, факсимильные устройства, видеотелефоны, а в дальнейшем и голографические передающие системы, менеджеры будут в состоянии определить и обсудить вместе со своими партнерами, находящимися в любом месте, всевозможные аспекты рассматриваемого вопроса или деловой операции.

Столь же яркими будут последствия перестройки быта в результате внедрения в наш дом информационной технологии, и не только в виде систем контроля температуры, освещенности и уровня шума и тому подобного, но и для организации нашего отдыха, развлечений, для целей образования, а во многих случаях и для работы. Возможность кардинального изменения системы образования в последующие два десятилетия, состоящего в переходе от школьной системы к домашнему обучению, обсуждалась нами в другой работе (Стониер, 1979).

Мы уже касались вопроса создания заводов, которые больше будут похожи на гигантские сложные машины. На самом деле, полностью роботизированные заводские системы будут подсоединены к полностью автоматизированным конторским системам. Это означает, что вы сможете иметь продукцию или какие-то услуги без вмешательства людей. Ситуация аналогична прямому набору номера телефона по междугородной автоматической линии связи, когда отпадает необходимость в телефонистках, помогающих вам соединиться с вашим абонентом, расположенным даже на другой половине земного шара. Аналогично в будущем, пользуясь домашним терминалом или терминалом вашего учреждения, можно будет напрямую заказывать на заводе автомобиль, удовлетворяющий вашим требованиям. Заказ может принять при этом компьютер, установленный в конторе завода. Проверив состояние вашего банковского счета, он отдаст команду на изготовление автомобиля по вашим спецификациям и организует его доставку. По ходу дела он будет информировать вас о выполнении заказа. Закончив работы и отправив автомобиль к вам домой, компьютер свяжется с компьютером вашего

отделения банка о переводе необходимой суммы. В этом процессе не возникает необходимости вмешательства человека, не считая, может быть, транспортировки автомобиля заказчику.

Подобные фирмы, производящие автомобили, будут нуждаться в людях по общему надзору и контролю и для контакта с покупателем, если что-то было не так. Не более того. Принципиальная необходимость в неквалифицированной рабочей силе сохранится в отношении персонала, обеспечивающего охрану учреждений, и в отношении водителей грузовиков. Менеджеры должны обеспечить включение такого заводского комплекса и в общую международную сеть. Они будут вести переговоры о приобретении необходимых компонентов (часто через свой компьютер и компьютеры поставщиков), заниматься вопросами рынка, определения цен, отвечать за престиж фирмы и производимого продукта и т. п. Как и в прошлом, останется необходимость в контактах с общественностью, правительством, трудящимися, финансовыми кругами и т. д.

### 13.3.5. Распространение интеллектуальной технологии

Одним из наиболее ярких свойств компьютера является возможность моделировать на нем другие системы. По мере углубления понимания и накопления опыта работы как с самими системами, так и с их моделями мы будем переходить к моделированию все более сложных систем. Это — экономические системы с использованием анализа вход-выход, политические системы (базирующиеся на избирательной системе) и, в конечном счете, сама история человечества.

Среди наиболее изощренных способов применения информационной технологии найдут свое место разработки интеллектуальных методов достижения надежного прогноза. При предсказании погоды информационная технология, включая современные ЭВМ и спутники, уже интенсивно применяется. Факторы, определяющие погоду, образуют сложную систему, но, возможно, не такую сложную, как другие системы, например системы метаболического обмена, экологические системы, экономические или политические системы, характеризующие человеческое общество.

Причина, по которой пока что предсказание не является точным, состоит в отсутствии как последовательного теоретического подхода к анализу будущего, так и фундамента для такого рода теории. Основным в развитии индустрии знаний будет создание возможности предсказывать экономические циклы, обеспечивать техническое прогнозирование и решать такие интеллектуальные задачи, которые кажутся столь же недостижимыми сегодня, как казалось невозможным предсказать затмение в средние века. Обратимся к некоторым аспектам этой области индустрии знаний.

*а) Системный анализ человеческих общественных формаций.* При этом человеческое общество воспринимается как эволюционирующая система, идущая от сообществ гоминидов и приматов, которые ему предшествовали. Доминирующим фактором, определяющим общественную эволюцию приматов, служит среда обитания. Среди сообществ приматов и сегодня такое

экологическое давление является очень важным. С появлением техники у древних людей изменились отношения между сообществами и окружающей средой. Оружие, огонь, речь, одомашнивание растений и животных означали, что выслеживание дикого зверя или неравномерность поступления пищи уже перестали быть такими важными, как раньше. К концу каменного века культурная эволюция стала вытеснять эволюцию биологическую, поскольку в основном была решена главная проблема, с которой сталкиваются почти все животные, — добыть достаточно пищи и не быть съеденным самому. На ее место, как более важная, встала проблема отношений с соседями и врагами.

С самого зарождения человечества общественная эволюция определялась эволюцией техники. С теоретической точки зрения информационная технология особенно интересна тем, что она является самой мощной технологией, которая начала создаваться со времен появления речи (также являющейся ИТ). Информационная технология — это метатехнология в том смысле, что это не только разновидность технологии сама по себе, но и технология, оказывающая влияние на обширные области имеющейся технологии. Фундаментальной предпосылкой системного подхода к анализу воздействия информационной технологии на общество является мысль, что *впредь ИТ будет двигателем технической эволюции, так же как техника явилась двигателем общественной эволюции.*

*б) Прослеживание тенденций.* Хотя у этого метода имеются серьезные ограничения, у него находятся и свои области применения. Он может быть полезным при выяснении характера и степени изменений, которых естественно ожидать. Одним из простых практических правил могло бы быть следующее: если нужно посмотреть на 25 лет вперед, то следует заглянуть на 50 лет назад в прошлое. Если взять 1985 г., то, чтобы увидеть, что будет достигнуто к 2010 г., необходимо вернуться назад — к 1935 г.

Некоторые типы технологического развития характеризует волнообразная кривая роста. Такого рода кривые часто встречаются в биосистемах, например, в случае роста бактерий в ограниченной среде. Рост мировой телефонной сети (как и рост плесени в чашечке Петри), по-видимому, может быть проанализирован таким образом. Кривая, построенная на материале 1935–1985 гг., может дать разумные основания для предсказания роста в 2010 г.

Однако *прослеживание тенденций не позволяет предвидеть скачки.* Если бы в 1935 г. кто-то, заглядывая в будущее, применял этот метод для прогноза на 25 лет, то вряд ли он смог бы предсказать, что кроме радио в каждом доме будет стоять еще и телевизор. Прослеживание тенденций не предсказало бы тогда появление реактивных самолетов и тем более космических кораблей. И, разумеется, в 1935 г. целый мир электронных вычислительных машин, транзисторов, интегральных схем и тому подобного просто не существовал.

*в) Анализ истории техники.* Компьютеры представляют идеальную иллюстрацию скачка в техническом развитии. В 1935 г. счетные машины — это все, что было известно в мире коммерции. К 1960 г. уже ни одна крупная уважающая себя организация не обходилась без компьютера при подготовке пла-



тежной ведомости. В 1960 г., однако, бытовые компьютеры были не известны, а о цифровых электронных часах и карманных калькуляторах никто не мечтал. К 1985 г. Такие устройства стали вполне обыденными, и шли разговоры о криогенных компьютерах, биокристаллах, оптических волокнах, сетях спутниковой связи, домашних роботах и множестве других технических чудес.

Можно ли было предвидеть появление компьютеров и транзисторов 50 лет назад? Можно дать положительный ответ, если знать, куда посмотреть. К середине 30-х годов К. Зус четко определил понятие калькулятора общего назначения и начал строить работающую модель у себя дома. Р. Пол уже предвидел в 1933 г., что электронные лампы в радиоприемниках будут заменены на небольшие твердотельные устройства [4]. Если возникает необходимость в предвидении технологических скачков, то нужно проанализировать фундаментальные научные исследования и подумать, во-первых, о технических возможностях, вытекающих из состояния науки или фундаментальных успехов технологии, а во-вторых, об экономических или других факторах рыночного характера, которые преобразуют открытие или изобретение в реальное нововведение или продукт.

И если прослеживание тенденций полезно при прогнозе роста глобальной телефонной системы, эта методика бессильна при предсказании появления новой технологии. Лучший способ прогнозирования таких скачков — новых видов техники будущего — попытаться найти коллективы или отдельных людей, занятых фундаментальными исследованиями и их развитием, которые с большой вероятностью приведут к коммерчески полезным нововведениям в ближайшие 25 лет, а затем проанализировать потенциальный рынок.

г) *Экспертная рекомендация.* Некоторые специалисты высказываются в пользу применения методов Дельфи\* при построении оценок будущего развития. Однако даже самые лучшие эксперты часто не хотят или неспособны кооперироваться друг с другом. Кроме того, в этой процедуре не удастся надлежащим образом взвесить мнения и установить различия между проницательной экспертизой, разумной догадкой, народной мудростью и фантастическими спекуляциями. Напротив, глубокий анализ, проводимый отдельными людьми, выбранными из-за их опыта и творческого воображения, может оказаться чрезвычайно полезным в определении характера будущего развития событий. Затем, когда концепции созреют, всегда полезно провести встречи, начиная от совещаний малых коллективов избранных специалистов и до открытых международных симпозиумов и конференций. Такого рода процедура должна быть одним из необходимых условий создания экспертных систем.

д) *Анализ экономических последствий.* Чтобы точно оценить все положительное воздействие информационной технологии на экономику, недостаточно

---

\* Метод Дельфи — процедура получения экспертных оценок посредством неоднократного опроса группы специалистов. — *Прим. ред.*

сведений о прямых затратах на информационную технологию: объеме продаж или ее доле в валовом национальном продукте страны. Причина, по которой ИТ и индустрия знаний в целом так сильно выросли за последние десятилетия, состоит в том, что они оказываются очень выгодными. Те фирмы и страны, которые соответствующим образом пользуются информационной технологией, процветают. Но как измерить роль информационной технологии как входной компоненты экономики.

Одни из возможных подходов иллюстрируется исследованием роста производительности на американских железных дорогах между 1870 и 1910 гг., проведенным А. Фишлоу. Он пришел к выводу, что использование менее эффективной технологии 1870 г. стоило бы потребителю дополнительно 1,3 млрд. дол. при сохранении грузовых потоков на уровне 1910 г. Таким образом, ценность усовершенствований в технике (воздушные тормоза, автоматическое сцепление, замена железных рельсов стальными, более мощные локомотивы, более эффективные подвижные составы) состоит в том, что поезда смогли перевозить большой груз и быстрее, что позволило снизить затраты на 1,3 млрд. дол.

*е) Анализ моделей затраты-выпуск.* К наиболее полезным методам анализа в экономике относится использование моделей затраты-выпуск Леонтьева [3]. Если имеется надежная информация для построения таких моделей, то становится возможным предсказать, как изменения в одной части экономики повлияют на ее другие части. Основным недостатком матрицы затраты-выпуск является относительная статичность. Ее можно сделать динамической, приняв два предположения: 1) технология является двигателем постиндустриальной экономики; 2) темпы развития технологии определяются качеством и объемом поступающей информации, включая образование и информационную технологию.

В основе первого предположения лежит тот факт, что постиндустриальные страны не растут ни по площади, ни по числу населения. Следовательно, долговременный рост валового национального продукта не может происходить за счет роста рабочей силы или расширения территории. Не может он быть объяснен и притоком капитала из других стран, по крайней мере на длительном отрезке времени. Скорее, этот рост должен быть отнесен за счет успехов в технологиях, объясняющих рост производительности труда и общественно-го благосостояния\*.

Экономист Е. Денисон (1966 г.) провел анализ, используя подобные соображения. Он рассмотрел все факторы, которые, по его мнению, могли бы привести к росту экономики. Затем он дал им оценку. Сумма таких оценок оказалась не равной действительному росту. Эту разницу, или остаток, он приписал росту знаний. Денисон заключил, что примерно половина среднегодового роста экономики в США между 1929 и 1957 гг. непосредственно обеспечивалась распространением и созданием информации.

---

\* И разумеется, за счет более интенсивной эксплуатации трудящихся. — *Прим. ред.*

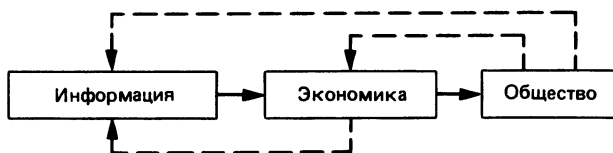


Рис. 13.1

Если Денисон прав, то наполовину темп роста обусловлен информацией. Имеются веские основания считать, что в подходе Денисона серьезно *недооценивается* истинная важность информации. Отсюда можно надеяться на создание динамической модели экономики, содержащей традиционную матрицу затраты-выпуск, дополненную новой внешней входной величиной, в качестве которой выступает новая информация. Эта внешняя сила движет системой.

*ж) Международный аспект.* Постиндустриальная экономика носит транснациональный характер. Никакой утонченный анализ экономики, проведенный в пределах одной страны, не может дать верную оценку будущего. Одной из главных задач для экономистов могло бы быть создание некоторой глобальной таблицы затраты-выпуск, которая при использовании компьютеров могла бы стать мировой моделью существующей экономики. Некоторые экономисты на самом деле заняты глобальным моделированием, и, когда их работа будет закончена, полученные результаты следует подвергнуть всестороннему анализу.

Технология также носит транснациональный характер. Особенно это верно в отношении информационной технологии. Большая часть оборудования, связанного с информационной технологией и используемого в Великобритании, поступила из-за рубежа. То же самое можно было бы сказать и о большей части фундаментальных научно-исследовательских работ, хотя Великобритания может претендовать на роль источника многих идей. Это фактически одно из самых больших ее достоинств.

Никакой анализ будущего не будет иметь смысла, если в нем не будет использована как историческая, так и глобальная матрица. Чтобы завершить анализ, необходимо дополнить анализ затраты-выпуск внешней выходной величиной, называемой обществом. Таким образом, необходимо изучать динамическую систему, показанную на рис. 13.1.

На рис. 13.1 изображена теоретическая конструкция, которая является попыткой вывести анализ затраты-выпуск за пределы чистой экономики, добавив на ее вход информационную технологию, с одной стороны, и рассматривая ее влияние на общество — с другой.

Штриховыми линиями показаны цепи обратной связи. Экономика может регулировать рост информации, например, финансируя исследования и разработки. Аналогично общество способно выработать законодательства, регулирующие расходы на образование, что приведет к большему распространению информации. Однако основной процесс идет слева направо, поскольку инфор-

мационная технология обладает следующими тремя уникальными свойствами:

1. Рост информации и технологии, по существу, необратим. Можно о чем-то забыть и что-то и проигнорировать, но как только что-то становится известным, то почти невозможно это утаить. Аналогично, если что-то изобретено, то почти невозможно отменить это изобретение.
2. Рост информации и технологии носит экспоненциальный характер: чем больше известно и изобретено, тем легче узнать и изобрести что-то еще.
3. Невозможно представить себе предел роста знаний и изобретений.

Этими тремя свойствами не обладает никакая другая компонента эволюционирующей человеческой общественной системы — ни экологическая, ни экономическая, ни политическая. По этой причине рост информации и технологии является основной движущей силой в общественной эволюции человека. Новая информация — топливо, технология — это пар, а экономика — локомотив, влекущий поезд человеческого прогресса (иногда, возможно, толкая его и назад).

#### 13.4. ВЫВОДЫ И ЗАМЕЧАНИЯ

Средства информационной технологии получают быстрое развитие. Интенсивно развиваются не только оборудование, но и необходимое программное обеспечение. Наша способность к логической организации данных, изобретению все усложняющихся алгоритмов и все более детальному моделированию систем соответствует нашему владению математикой, программированием и логикой. Скоро ограничивающим фактором в создании простых баз данных, интеллектуальных баз данных и экспертных систем станет ограниченность человеческого знания и понимания.

Как бы полезны и необходимы ни были такого рода системы, они могут привести к серьезным деформациям в нашей культуре. Промышленная революция на ее начальном этапе привела к величайшим страданиям. К концу индустриальной эры, хотя многие прежние проблемы и удалось разрешить, образовалось нечто вроде социального чудовища с мощной мускулатурой и крошечным мозгом — глобальное общество с военной технологией, способной разрушить нашу цивилизацию, не способное создать политическую технологию, гарантирующую, что такого не произойдет, глобальное общество, в котором имущий сектор не знает, куда деть излишки пищи, а беднейший сектор находится на грани голодной смерти.

Экспертные системы могут помочь и, вполне вероятно, что они помогут. Однако, возможно, что мы теперь создаем новую разновидность социального чудовища, воображающего, что все проблемы могут быть разрешены применением логики и алгоритмов умного игрока в шахматы. Шахматы — это игра с нулевой суммой, чего нельзя сказать о самом существовании человека. Создание экспертной системы, дающей советы, как играть в шахматы, не приведет к фатальным ошибкам. А создание системы, дающей советы врачу, — может. Очевидно, что нам необходимо создание интеллектуальной технологии, помогающей решить человеческие проблемы, включая проблемы

в области экономики, политики, системы убеждений, образования и нашей культуры в целом. Однако мы должны проявлять предельную осторожность и консервативность в отношении возникающих комбинаций сложностей.

Применяя к чрезвычайно сложным системам, которые пока еще плохо поняты (например, экономическим системам), дополнительно новую группу сложностей, которые сами по себе тоже плохо поняты, мы создаем новые уровни неопределенности, порождая в то же время иллюзию возрастающей точности. Всегда существует опасность, что люди, склонные обычно искать простые решения, примут результаты работы машинного интеллекта с большим доверием, чем они того заслуживают.

Поэтому индустрия знаний должна со всей серьезностью поставить перед собой задачу вскрытия законов экономики, понимания человеческого общества и человеческой психики, и не только там, где возникают конфликты, но и в сфере кооперации. В частности, нам необходима предельная ясность, какие области знаний являются сравнительно надежными, а какие совсем неизвестны и какие части человеческого знания относятся к туманным областям, расположенным между ними.

Создание ценных, служащих человеку экспертных систем является одной из величайших задач, стоящих перед мыслящими людьми и индустрией знаний в ближайшие десятилетия. Сможем ли мы понять ситуацию настолько, чтобы остановиться, подумать, а потом воспользоваться этими инструментами?

Автор выражает признательность М. Эллисон за терпеливую помощь при подготовке этой рукописи.

Части этой главы были заимствованы из следующих публикаций автора:

- Stonier, T. (1979) Changes in western society: educational implications. In *World Yearbook of Education 1979: Recurrent Education and Lifelong Learning* (eds T. Schuller and J. Megarry) Kogan Page, London, 31–44.
- Stonier, T. (1983) *The Wealth of Information: A Profile of the Post-Industrial Economy*, Thames/Methuen, London.
- Stonier, T. (1984) Information and the deep structure of the universe – the need for a new physics. In *Infostorms*, Council for Educational Technology (In press).

К главе 1

1. Aldiss, B. (1975) *Billion Year Spree*, Corgi, London.
2. Ernst, G. and Newell, A. (1969) *GPS: a Case Study in Generality and Problem Solving*, Academic Press, New York.
3. Feigenbaum, E. (1971) On generality and problem solving. *Machine Intelligence*, 6.
4. Lenat, D. (1982) Eurisko: a program that learns new heuristics and domain concepts. *Artificial Intelligence*, 21.
5. McCorduck, P. (1979) *Machines Who Think*, Freeman, San Francisco.
6. McCulloch, W. and Pitts, W. (1943) A logical calculus of ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5.
7. Minsky, M. and Papert, S. (1969) *PERCEPTRON; an Introduction to Computational Geometry*, MIT Press, Massachussetts.
8. Rosenblatt, F. (1957) *The PERCEPTRON: a Perceiving and Recognizing Automaton*, Cornell Aeronautical Lab, New York.
9. Shortliffe, E. (1976) *Computer Based Medical Consultations: MYCIN*, American Elsevier, New York.

К главе 2

1. Feigenbaum, E. and McCorduck, P. (1983) *The Fifth Generation*, Michael Joseph, London.
2. Forsyth, R. (1981) BEAGLE: a Darwinian approach to pattern recognition. *Kybernetes*, 10.
3. Lenat, D. (1982) The nature of heuristics. *Artificial Intelligence*, 19, 189-249.
4. Michalski, R.S. and Chilausky, R.L. (1980) Learning by being told and learning from examples. *International Journal of Policy Analysis and Information Systems*, 4.
5. Michie, D. (ed.) (1982) *Introductory Readings in Expert Systems*, Gordon and Breach, New York.
6. Naylor, C. (1983) *Build Your Own Expert System*, Sigma Technical Press/John Wiley, Chichester.
7. Quinlan, J.R. (1979) *Induction over Large Databases*, Report HPP-79-14, Stanford University, Palo Alto, California.

К главе 3

1. The Alvey Directorate (1984) *Alvey MMI Strategy*, Department of Industry, London.
2. The Alvey Directorate (1982) *The Alvey Report*, Department of Industry, London.
3. The Alvey Directorate (1983) *Alvey VLSI and CAD strategy*. *ALVEY News*.

Department of Industry, London.

4. Commission of the European Communities (1982) *On Laying the Foundations for a European Strategic Program of Research and Development in Information Technology: The Pilot Phase*. COM (82) 486 final /2, Commission of the European Communities, Brussels.
5. d'Agapeyeff, A. (1982) *The fifth generation*. In *The Fifth Generation – Dawn of the Second Computer Age*, SPL International, Abingdon, Oxfordshire.
6. Fuch, F. (1983) Final aiming for knowledge information processing systems. In *Adopting the First European Strategic Programme for Research and Development in Information Technologies*, (ESPRIT)COM (83) 258, Commission of the European Communities, Brussels.
7. Gevarter, W.B. (1982) *An Overview of Expert Systems*. NBSIR 1982 – 2505, US Department of Commerce, National Bureau of Standards, Washington, D.C.
8. JIPDEC (1981) *Preliminary Report on Study and Research on Fifth Generation Computers 1979–1980*. Japan Information Processing Development Centre, Keidan-Ren-Kaikan, Tokyo.
9. Muller, B. (ed.) (1983) *SPL Insight Program*. The Fifth Generation World Conference 1983, SPL International, Abingdon, Oxfordshire.
10. Point Paper (1983) *DARPA Strategic Computing Program*, US Department of Defence, Washington, D.C.
11. SERC–DOI (1982) *IKBS Architecture Study*, HMSO, London.
12. Talbot, D. and Witty, R.W. (1983) *Alvey Program – Software Engineering Strategy*. Department of Industry, London.
13. Yokoi, T. (1983) *A Perspective of the Japanese Fifth Generation Computer Systems Project*. ICOT TM – 0026, Minato-Ku, Tokyo.

#### К главе 4

1. Adams, D. (1979) *The Hitchhiker's Guide to the Galaxy*, Pan, London.
2. Jaynes, J. (1976) *The Origin of Consciousness in the Breakdown of the Bicameral Mind*, Houghton Mifflin, New York.
3. Kubrick, S. (1968) *2001: A Space Odyssey* (from *The Sentinel* by Arthur C. Clarke) MGM.
4. Lighthill, J. (1973) *Report to the Science Research Council on Funding for Artificial Intelligence Research*, HMSO, London.

#### К главе 5

1. Duda, R., Hart, P., Barrett, p. et al. (1978) *Development of the PROSPECTOR System for Mineral Exploration*, SRI Report Projects 5822 and 6415, Stanford Research Institute, Palo Alto, Ca.
2. Efstathiou, J. (1984) Non-classical logics and the handling of uncertainty. *British Computer Society Specialist Group on Expert Systems Newsletter*, 10.

3. Forsyth, R. (1984) *HULK User Guide*, Brainstorm Computer Publications, London.
4. Mamdani, A. and Gaines, B.R. (1981) *Fuzzy Reasoning and its Applications*, Academic Press, New York.
5. Shortliffe, E. (1976) *Computer Based Medical Consultations: MYCIN*, American Elsevier, New York.
6. Weiss, S. and Kulikowski, C. (1983) *A Practical Guide to Designing Expert Systems*, Methuen, New York.
7. Zadeh, L. (1965) Fuzzy sets. *Information and Control*, **8**, 338–353.

## К главе 7

1. Alvey, P. (1982) *A Programme for Advanced Information Technology, The Report of the Alvey Committee*, HMSO.
2. Attisha, M. and Yazdani, M. (1983) A microcomputer-based tutor for teaching arithmetic skills. *Instructional Science*, **12**, 333–342.
3. Attisha, M. and Yazdani, M. (1984) An expert system for diagnosing children's multiplication errors. *Instructional Science*, **13**.
4. Barber, E.O. (1984) Expert systems survey. *Working Paper W.122*, Department of Computer Science, University of Exeter.
5. Barr, A. and Feigenbaum, E. (1982) *Handbook of Artificial Intelligence*, Pitman Press, London.
6. Clark, K.L. and McCabe, F.G. (1982) PROLOG: a language for implementing expert systems. *Machine Intelligence*, **10**, 455–475.
7. Clocksin, W. and Mellish, C. (1982) *Programming in PROLOG*, Springer Verlag, Berlin.
8. Duda, R.O. and Gaschnig, J.G. (1981) Knowledge-based expert systems come of age. *Byte*, **6:9**, 238–278.
9. Hardy, S. (1983) *PROLOG for Knowledge Engineers*, Tecknowledge Internal Memo.
10. Lighthill, J. (1972) *Artificial Intelligence: Report to the Science Research Council*, Science Research Council.
11. Michie, D. (1980) Expert systems. *Computer Journal*, **23:4**, 369–377.
12. Minsky, M. (ed.) (1968) *Semantics Information Processing*, MIT Press.
13. O'Shea, T. and Self, J. (1983) *Learning and Teaching with Computers*, The Harvester Press, Brighton.
14. Shortliffe, E.H. (1976) *Computer-based Medical Consultations: MYCIN*, American Elsevier, New York.
15. Sleeman, D. and Brown, J.S. (eds.) (1982) *Intelligent Tutoring Systems*, Academic Press, London.
16. Sloman, A., Hardy, S. and Gibson, J. (1983) POPLOG: a multi-language program development environment. *Information Technology: Research and Development*, **2**, 109–122.
17. Spacek, L. (1981) *The Production Systems and PROLOG*, Department of Computer



Science, University of Essex.

18. Wellbank, M. (1983) *A Review of Knowledge Acquisition Techniques for Expert Systems*, Issued by Martlesham Consultancy Services, British Telecom Research Laboratories, Ipswich, England.
19. Winston, P. (1977) *Artificial Intelligence*, Addison-Wesley, Reading, Mass.
20. Winston, P. and Horn, B. (1980) *LISP*, Addison-Wesley, Reading, Mass.

#### К главе 8

1. Duda, R., Hart, P., Barrett, P. et al. (1978) *Development of the PROSPECTOR Consultation System for Mineral Exploration: Final Report*, SRI International, Stanford Research Institute, Palo Alto, Ca.

#### К главе 9

1. Sprague, R.H. (1980) Framework for the development of decision support systems. *M.I.S. Quarterly*, 4:4.
2. Zadeh, L.H. (1965) Fuzzy sets. *Information and Control*, 8, 338-353.

#### К главе 10

1. Dietterich, T. and Michalski, R. (1981) 'Inductive learning of structural descriptions'. *Artificial Intelligence*, 16, 257-294.
2. Forsyth, R. (1981) BEAGLE: a Darwinian approach to pattern recognition. *Kybernetes*, 10.
3. Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
4. Hunt, E., Marin, J. and Stone, P.T. (1966) *Experiments in Induction*, Academic Press, New York.
5. Langley, P. (1977) Rediscovering physics with BACON-3. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*.
6. Lenat, D. (1983) The nature of heuristics III. *Artificial Intelligence*, 19, 189-249.
7. Michalski, R., Stepp, R. and Diday, E. (1981) A recent advance in data analysis. *Progress in Pattern Recognition*, North-Holland, Amsterdam.
8. Mitchell, T. (1979) An analysis of generalization as a search problem. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*.
9. Quinlan, J. (1979) *Induction over Large Databases: Report HPP-7914*, Stanford University, Palo Alto, Ca.
10. Smith, S. (1980) *A Learning System Based on Genetic Adaptive Algorithms*, Ph.D. Thesis, University of Pittsburgh, Pa.
11. Wellbank, M. (1983) *A Review of Knowledge Acquisition Techniques for Expert Systems*, British Telecom Research, Ipswich, England.

1. Bethke, A.D. (1981) *Genetic Algorithms as Function Optimizers*. PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan.
2. Booker, L.B. (1982) *Intelligent Behavior as an Adaptation to the Task Environment*. PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan.
3. Brindle, A. (1981) *Genetic Algorithms for Function Optimization*. PhD thesis, Dept. of Computing Science, University of Alberta.
4. Buchanan, B.G., Sutherland, G. and Feigenbaum, E.A. (1969) Heuristic DENDRAL: a program for generating explanatory hypotheses in organic chemistry. *Machine Intelligence*, 4, 209–254.
5. Buchanan, G.B., and Mitchell, T.M. (1978) Model-directed learning of production rules. In *Pattern-Directed Inference Systems* (eds. D. A. Waterman and F. Hayes-Roth), Academic Press, New York.
6. Carbonell, J.G. (1983) Learning by analogy: formulating and generalizing plans from past experience. In *Machine Learning* (eds. R. S. Michalski, J. G. Carbonell and T.M. Mitchell) Tioga Publishing Co., Palo Alto, California.
7. DeJong, K.A. (1975) *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan.
8. DeJong, K.A. (1980) Adaptive system design: a genetic approach. *IEEE Transactions on Systems, Man and Cybernetics*, 10(9), September, 566–574.
9. Duda, R.O., Hart, P.E., Barrett, J.G. et al (1978) *Development of the PROSPECTOR Consultation System for Mineral Exploration: Final Report*. Technical Report, SRI International.
10. Fogel, L.J., Owens, A.J., and Walsh, M.J. (1966) *Artificial Intelligence Through Simulated Evolution*, Wiley, New York.
11. Friedberg, R.M. (1958) A learning machine, Part 1. *IBM Journal of Research and Development*, 2, 2–13.
12. Grefenstette, J.G. (1983) *Optimization of Genetic Search Algorithms*, Technical Report CS-83–14, Vanderbilt University.
13. Hayes-Roth, F., Waterman, D.A., and Lenat, D.B. (eds.) (1983) *Building Expert Systems*, Academic Press, Reading, Mass.
14. Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
15. Holland, J.H., (1978) Cognitive systems based on adaptive algorithms. In *Pattern-Directed Inference Systems* (eds. D. A. Waterman and F. Hayes-Roth) Academic Press, New York.
16. Holland, J.H. (1980) Adaptive algorithms for discovering and using general patterns in growing knowledge bases. *International Journal on Policy Analysis and Information Systems*, 4(2), 217–240.

17. Holland, J.H. (1983) Escaping brittleness. In *Proceedings 2nd International Workshop on Machine Learning*, July, University of Illinois.
18. Lenat, D.B. (1983) The role of heuristics in learning by discovery: three case studies. In *Machine Learning, An Artificial Intelligence Approach* (eds. R. S. Michalski, J.G. Carbonell and T.M. Mitchell) Tioga Publishing Co., Palo Alto, California.
19. McDermott, J., and Forgy, C.L. (1978) Production system conflict resolution strategies. In *Pattern-Directed Inference Systems* (eds. D. A. Waterman and F. Hayes-Roth), Academic Press, New York.
20. McDermott, J. (1980) *RI: A Rule-Based Configurer of Computer Systems*. Technical Report CMU-CS-80-119, Carnegie-Mellon University, Pa..
21. Mitchell, T.M. (1982) Generalization as search, *Artificial Intelligence*, **18**, 203–226.
22. Mostow, D.J. (1981) *Mechanical Transformation of Task Heuristics into Operational Procedures*. PhD thesis, Computer Science Dept., Carnegie-Mellon University, Pa.
23. Shortliffe, E.H. (1976) *MYCIN: Computer-Based Medical Consultations*, American Elsevier, New York.
24. Smith, S.F. (1980) *A Learning System based on Genetic Adaptive Algorithms*. PhD thesis, Dept. of Computer Science, University of Pittsburgh.
25. Smith, S.F. (1983) Flexible learning of problem solving heuristics through adaptive search. In *Proceedings 8th International Joint Conference on Artificial Intelligence*. August, William Kaufman Inc., Los Altos, California.
26. Waterman, D.A. (1970) Generalization learning techniques for automating the learning of heuristics. *Artificial Intelligence*, **1**, 121–170.

## К главе 12

1. Banerji, R. (1982) Theory of problem solving: a branch of artificial intelligence. *Proceedings of the IEEE*, **70:12**, 1428–1448.
2. Chandrasekar, B., Mittal, S. and Smith, J. (1980) RADEX – towards a computer-based radiology consultant. In *Pattern Recognition in Practice* (eds Gelsema and Kanel) North Holland, Amsterdam.
3. Conrad, M. (1979) Bootstrapping on the adaptive landscape. *BioSystems*, **11**, 167–182.
4. Davis, R. and Lenat, D. (1982) *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill, New York.
5. Dietterich, T., London, B., Clarkson, K. and Dromey, G. (1982) Learning and inductive inference. In *Handbook of Artificial Intelligence*, Vol. 3 (eds P. Cohen and E. Feigenbaum) William Kaufman, Los Altos, Ca., 323–512.
6. Elstein, A., Shulman, L. and Sprafka, S. (1978) *Medical Problem Solving: an Analysis of Clinical Reasoning*, Harvard University Press, Cambridge, Mass.
7. *Handbook of Artificial Intelligence*, Vol 1 (1981) (eds A. Barr and E. Feigenbaum) Vol 2 (1982) (eds A. Barr and E. Feigenbaum) Vol 3 (eds P. Cohen and

- E. Feigenbaum) William Kaufmann, Inc, Los Altos, Ca.
8. Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
  9. Holland, J. (1983) Escaping brittleness. *Proceedings of the International Machine Learning Workshop*, 92-96.
  10. Innocent, P., Teather, D., Wills, K. and Du Boulay, G. (1983) Operational system for computer assisted diagnosis of cerebral disease. *Proc. MEDINFO '83*, 467.
  11. Knuth, D. (1973) *The Art of Computer Programming, Vol 3, Sorting and Searching*, Addison-Wesley, Reading, Mass.
  12. Langley, P. (1977) Rediscovering physics with BACON.3. *Proceedings of the International Joint Conference on Artificial Intelligence*, 505-507.
  13. Lenat, D. (1982) The nature of heuristics. *Artificial Intelligence*, **19**, 189-249.
  14. Lenat, D. (1983) The role of heuristics in learning by discovery. In *Machine Learning* (eds R. Michalski, J. Carbonell and T. Mitchell) Tioga Publishing, Palo Alto, Ca., 243-306.
  15. Michalski, R. (1983) Theory and methodology of inductive learning. In *Machine Learning* (eds R. Michalski, J. Carbonell and T. Mitchell) Tioga Publishing, Palo Alto, Ca., 83-134.
  16. Minsky, M. and Papert, S. (1972) *PERCEPTRONS*, MIT Press, Massachusetts.
  17. Mitchell, T. (1977) Version spaces: a candidate elimination approach to rule learning. *Proceedings of the International Joint Conference on Artificial Intelligence*, 305-310.
  18. Mostow, D. (1982) Learning by being told. In *Machine Learning* (eds R. Michalski, J. Carbonell and T. Mitchell) Tioga Publishing, Palo Alto, Ca., 367-404.
  19. Nilsson, N. (1980) *Principles of Artificial Intelligence*, Tioga Publishing, Palo Alto, Ca.
  20. Rada, R. (1983) Characterizing search spaces. *Proceedings of the International Joint Conference on Artificial Intelligence*, 780-782.
  21. Rada, R. and Ackerman, L. (1983) Computerized tomography expert. *Proceedings of the Conference on Artificial Intelligence*, 233-241.
  22. Ramsey, R. (1981) *Neuroradiology with Computed Tomography*, W. B. Saunders, Philadelphia.
  23. Rich, E. (1983) *Artificial Intelligence*, McGraw-Hill, New York.
  24. Samuel, A. (1959) Some studies in machine learning using the game of checkers. *IBM J. Res. Devel.*, **3**, 210-229.
  25. Samuel, A. (1967) Some studies in machine learning using the game of checkers, Part II. *IBM J. Res Devel.*, **11**, 601-618.
  26. Shortliffe, E. (1976) *Computer Based Medical Consultations: MYCIN*, American Elsevier, New York.
  27. Swartout, W. (1983) XPLAIN: a system for creating and explaining expert consulting systems. *Artificial Intelligence*, **21:3**, 285-325.

28. Valk, J. (1980) *Computed Tomography and Cerebral Infarctions*, Raven Press, New York.
29. van Melle, W. (1979) Domain-independent production-rule system for consultation programs. *Proceedings of the International Joint Conference on Artificial Intelligence*, 923–925.
30. Winston, P. (1984) *Artificial Intelligence*, Addison-Wesley, Reading, Mass.

### К главе 13

1. Denison, E.F. (1966) Measuring the contribution of education (and the residual) to economic growth. In *The Residual Factor and Economic Growth*, OECD, 13–55.
2. Fishlow, A. as cited in N. Rosenberg (1983) *Inside The Black Box*, Cambridge Univ Press.
3. Leontief, W. (1966) *Input/Output Economics*, Oxford University Press, New York.
4. Pohl, R.W. (1934) *Mitteilungen der Universitäts bundes Göttingen*, 15 (as cited by E. Braun and S. MacDonald (1978) in *Revolution in Miniature*, Cambridge University Press.)
5. Porat, M. (1976) *The Information Economy*, Center for Interdisciplinary Research, Stanford University.
6. Smith, A. (1776) *Inquiry into the Nature and Causes of the Wealth of Nations*, Book 1.
7. Zuse, K. cited in Evans, C. (1981) *The Making of the Micro*, Victor Gollancz, London.

### ДОПОЛНИТЕЛЬНЫЙ СПИСОК ЛИТЕРАТУРЫ

Приводится рекомендуемая литература общего характера, предназначенная для тех, кто хочет глубже познакомиться с экспертными системами. В перечисленных далее книгах и статьях содержится материал, с которым необходимо познакомиться всем серьезно интересующимся инженерией знаний и ее приложениями. Предлагаемая литература охватывает область искусственного интеллекта со специальным уклоном в сторону усвоения, представления и использования знаний в вычислительных системах.

- BARR, A. and FEIGENBAUM, E.A. (eds.) (1981) *The Handbook of Artificial Intelligence, Vol. 1*, Pitman Books, London.
- BARR, A. and FEIGENBAUM, E.A. (eds.) (1982) *The Handbook of Artificial Intelligence, Vol. 2*, Pitman Books, London.
- COHEN, D. and FEIGENBAUM, E.A. (eds.) (1983) *The Handbook of Artificial Intelligence, Vol. 3*, Pitman Books, London.
- DAVIS, R. and LENAT, D. (1981) *Knowledge Based Systems in Artificial Intelligence*, McGraw-Hill, New York.
- FEIGENBAUM, E.A. and MCCORDUCK, P. (1984) *The Fifth Generation*, Michael Joseph, London.
- FORSYTH, R. (1981) BEAGLE: A Darwinian approach to pattern recognition. *Kybernetes*, 10.

- GRAHAM, N. (1979) *Artificial Intelligence*, Tab Books, Blue Ridge, Pa.
- HAYES-ROTH, F., WATERMAN, D. and LENAT, D. (eds.) (1983) *Building Expert Systems*, Addison-Wesley, New York.
- HOFSTADTER, D. (1981) *Goedel, Escher, Bach . . .*, Harvester Press, Brighton.
- HOLLAND, J.H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
- JAMES, M. (1984) *Artificial Intelligence in Basic*, Newnes Technical Books, Sevenoaks, Kent.
- LANGLEY, P. (1981) Data-driven discovery of physical laws. *Cognitive Science*, 5.
- LENAT, D. (1983) EURISKO: A program that learns new heuristics and domain concepts. *Artificial Intelligence*, 21.
- MICHALSKI, R.S., CARBONELL, J.G. and MITCHELL, T.M. (eds.) (1983) *Machine Learning*, Tioga Press, Palo Alto, Ca.
- MICHALSKI, R.S. and CHILAUSSKY, R.L. (1980) Learning by being told and learning from examples. *International J. Policy Analysis and Information Systems*, 43, 125-161.
- MICHIE, D. (ed.) (1982) *Introductory Readings in Expert Systems*, Gordon and Breach, New York.
- MITCHELL, T. (1982) Generalization as search. *Artificial Intelligence*, 18, 203-226.
- NAYLOR, C. (1983) *Build Your Own Expert System*, Sigma Technical Press.
- QUINLAN, R. (1982) Semi-autonomous acquisition of pattern-based knowledge. *Machine Intelligence*, 10.
- SAMUEL, A. (1967) Some studies in machine learning using the game of checkers, part II. *IBM Journal of Research and Development*.
- SHORTLIFFE, E. (1976) *Computer-Based Medical Consultations: MYCIN*, Elsevier, New York.
- WATERMAN, D. and HAYES-ROTH, F. (eds.) (1978) *Pattern-Directed Inference Systems*, Academic Press, New York.
- WEISS, S. and KULIKOWSKI, C. (1984) *A Practical Guide to Designing Expert Systems*, Chapman and Hall, London.
- WINSTON, P.H. (1977) *Artificial Intelligence*, Addison-Wesley, Reading, Mass.
- WINSTON, P.H. (1982) Learning new principles from precedents and examples. *Artificial Intelligence*, 19.
- ZADEH, L. and FUKANAKA, K. (eds.) (1975) *Fuzzy Sets and their Applications to Cognitive Decision Processes*, Academic Press, New York.

#### СПИСОК РАБОТ, ПЕРЕВЕДЕННЫХ НА РУССКИЙ ЯЗЫК

1. Банерджи Р. Б. Теория решения задач как раздел искусственного интеллекта. — ТИИЭР, т. 70, № 12, 1982. С. 64–90.
2. Кнут Д. Е. Искусство программирования для ЭВМ. Т.3. Сортировка и поиск: Пер. с англ. — М.: Мир, 1978. — 848 с.
3. Маккалок У., Питтс У. Логическое исчисление идей, относящихся к нервной активности// Автоматы: Пер. с англ. — М.: ИЛ, 1956. — С. 362–384.
4. Минский М., Пейперт С. Перцептроны: Пер. с англ. — М.: Мир, 1971. — с. 261.
5. Нильсон Н. Принципы искусственного интеллекта: Пер. с англ. — М.: Радио и связь, 1985. — 376 с.
6. Семюзль А. Некоторые исследования возможности обучения машин на примере игры в шашки// Вычислительные машины и мышление: Пер. с англ. под ред. Э. Фейгенбаума и Дж. Фельдмана. — М.: Мир, 1967. — С. 71–110.
7. Хейес-Рот Ф., Уотерман Д., Ленат Д. Построение экспертных систем: Пер. с англ. — М.: Мир, 1987. — 430 с.
8. Уинстон П. Искусственный интеллект: Пер. с англ. — М.: Мир, 1980. — 519 с.

#### СПИСОК ЛИТЕРАТУРЫ К ПРЕДИСЛОВИЮ РЕДАКТОРА ПЕРЕВОДА

1. Нильсон Н. Искусственный интеллект. Методы поиска решений: Пер. с англ. — М.: Мир, 1973. — 265 с.
2. Нильсон Н. Принципы искусственного интеллекта: Пер. с англ., — М.: Радио и связь, 1985. — 376 с.
3. Уинстон П. Искусственный интеллект: Пер. с англ. — М.: Мир, 1980. — 519 с.
4. Поспелов Д. А., Стефаниук В. Л. Искусственный интеллект в зарубежных исследованиях//Информационные материалы № 3 Научного совета по комплексной проблеме "Кибернетика", 1986. — С. 3–33.
5. Реальности и прогнозы искусственного интеллекта: Пер. с англ./Под ред. В. Л. Стефаниука. — М.: Мир, 1987. — 245 с.
6. Алексеева Е. Ф., Стефаниук В. Л. Экспертные системы — состояние и перспективы//Изв. АН СССР. Сер. Техническая кибернетика. — 1984. — № 5. — С. 153–167.
7. Представление знаний в человеко-машинных и робототехнических системах. — Т. С.: Прикладные человеко-машинные системы, ориентированные на знания/Под. ред. Г. С. Поспелова. — М.: ВЦ АН СССР, ВИНТИ АН СССР, 1984. — 380 с.
8. Хейес-Рот Ф., Уотерман Д., Ленат Д. Построение экспертных систем: Пер. с англ. — М.: Мир, 1987. — 430 с.

## ОГЛАВЛЕНИЕ

Предисловие редактора перевода . . . . .	5
Предисловие . . . . .	7
<b>Часть I. ОСНОВНЫЕ ИДЕИ . . . . .</b>	<b>9</b>
Глава 1. Феномен экспертных систем . . . . . Ричард Форсайт	9
Глава 2. Архитектура экспертных систем. . . . . Ричард Форсайт	14
Глава 3. Проект машин пятого поколения . . . . . Анни Брукинг	23
Глава 4. Как мы собираемся оценивать работу экспертной системы?. . . . . Антони Стевенс	40
<b>Часть II. ЛОГИЧЕСКИЙ ВЫВОД . . . . .</b>	<b>51</b>
Глава 5. Системы нечеткого рассуждения. . . . . Ричард Форсайт	51
Глава 6. Как построить машину вывода. . . . . Крис Нейлор	62
<b>Часть III. ИНЖЕНЕРИЯ ЗНАНИЙ . . . . .</b>	<b>84</b>
Глава 7. Техника использования знаний на языке Пролог. . . . . Масауд Яздани	84
Глава 8. Как мы строим систему "Микроэксперт" . . . . . Фил Кокс	103
Глава 9. REVEAL – вычислительная среда для создания экспертных систем . . . . . Питер Джонс	124
<b>Часть IV. ОБУЧЕНИЕ МАШИН . . . . .</b>	<b>145</b>
Глава 10. Стратегии обучения машин . . . . . Ричард Форсайт	
Глава 11. Адаптивные обучающиеся системы . . . . . Стефен Смит	158
Глава 12. Автоматизация усвоения знаний . . . . . Рой Рада	178
Глава 13. Индустрия знаний . . . . . Том Стониер	197
Список литературы . . . . .	213
Дополнительный список литературы . . . . .	220
Список работ, переведенных на русский язык . . . . .	222
Список литературы к предисловию редактора перевода . . . . .	222



Производственное издание

Анни Брукинг, Питер Джонс, Фил Кокс и др.

## ЭКСПЕРТНАЯ СИСТЕМА

### ПРИНЦИПЫ РАБОТЫ И ПРИМЕРЫ

Заведующая редакцией *О. В. Толкачева*

Редактор *Е. А. Засядько*

Художественный редактор *Т. В. Бусарова*

Обложка художника *К. М. Прасолова*

Технический редактор *Т. Н. Зыкина*

Корректор *Л. А. Буданцева*

ИБ № 1549

---

Подписано в печать 16.03.87	Формат 60x84/16	Бумага офс. № 2	Гарнитура
"Пресс-роман"	Печать офсетная	Усл. печ. л. 13,02	Усл. кр.-отт. 13,485
Тираж 30 000 экз.	Изд. № 21881	Зак. № 2285	Уч.-изд. л. 16,59
		Цена 1 р. 10 к.	

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

---

Московская типография №4 Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.  
129041, Москва, Б. Переяславская ул., д. 46