

В. П. Гладун

ПАРТНЕРСТВО С КОМПЬЮТЕРОМ

**Человеко-машинные
целеустремленные системы**

«Port-Royal»
Киев – 2000

Гладун В. П.

Партнерство с компьютером. – К.: «Port-Royal», 2000. – 128 с.

В книге раскрывается новая роль компьютеров в осуществлении мыслительных процессов и описываются метапроцедуры, обеспечивающие повышение уровня участия компьютерных систем в организации целеустремленного поведения.

Книга предназначена для научных работников и программистов, интересующихся вопросами искусственного интеллекта.

ISBN 966-7068-16-1

ISBN 966-7068-16-1

© Гладун В. П., 2000
© Издательство «Port-Royal», 2000

ПРЕДИСЛОВИЕ

Создание компьютера стимулировалось необходимостью получить инструмент поддержки мыслительных процессов. Это изначальное понимание назначения компьютера в течение длительного времени оказывало решающее влияние на развитие компьютерных технологий.

По современным представлениям возможности компьютерных систем не ограничиваются функциями инструмента. Расхожее слово «пользователь» не совсем точно отражает складывающиеся отношения между человеком и компьютером. Компьютерная система все в большей степени рассматривается как активная система, способная выполнять все составляющие целеустремленного поведенческого акта: формирование целей, выбор действий и актантов действий. Такого рода компьютерные системы обозначаются термином «агент». Аспекты, связанные с изучением агентов и многоагентных систем, появились во многих традиционных и относительно новых направлениях информатики.

Целеустремленность обычно считалась прерогативой человека. В свете новых представлений можно говорить о появлении систем нового типа – человеко-машинных целеустремленных систем. В этих системах функции человека и компьютера могут быть равносильными, вследствие чего возникает принципиально новый тип взаимодействия между человеком и компьютером. Это взаимодействие наилучшим образом определяется словом «партнерство».

Введение в научную терминологию новых, несколько размытых терминов («агент», «компьютер-партнер» и т.п.) без наполнения их конструктивным

содержанием часто оказывает не только стимулирующее, но и разрушительное воздействие на развитие научных исследований (разрушаются и запутываются классификации, размываются критерии и оценки, появляется база для научных спекуляций). Избежать разрушений можно только на основе системно организованного развития научных направлений. Сейчас имеется много теоретических предложений и практических разработок, касающихся реализаций различных элементов целеустремленного поведения в специфических условиях, при решении частных задач. Объединение частных разработок в систему возможно только на основе представления о целеустремленном поведении как о многослойном иерархическом процессе, в котором общее управление, не зависящее от особенностей предметных областей, компьютеров и языков, а также спецификаций параметров и ограничений, осуществляется метапроцедурами верхнего слоя, построенными на основе некоторых объединяющих принципов и работающими с общей базой данных и знаний. Процедуры более низких слоев адаптируют управление к частным средам и задачам.

Смещение акцентов в определении роли компьютера в целеустремленном поведении существенно влияет на состав аппаратного и математического обеспечения компьютерных комплексов, а также на организацию человеко-машинного интерфейса. Стратегии развития компьютерных систем должны отражать эти изменения.

В книге раскрывается новая роль компьютеров в осуществлении мыслительных процессов и описываются метапроцедуры, обеспечивающие повышение уровня участия компьютерных систем в организации целеустремленного поведения.

Глава 1. **ЦЕЛЕУСТРЕМЛЕННОЕ ПОВЕДЕНИЕ**

1.1. Процессы выбора в поведенческом акте

Поведение можно рассматривать как последовательность поведенческих актов, в основе которых лежит выполнение отдельных действий.

Акофф и Эмери [Акофф, 1974] в системе определений, связанных с целеустремленными системами, используют для описания поведения следующие основные категории: субъект, действие, окружение, результат, инструмент.

Субъект выбирает и осуществляет действие.

Окружение субъекта составляет множество элементов, не являющихся его частями, изменение в которых может вызвать изменение его состояния.

Результатом действия субъекта является изменение в нем или его окружении. Долгосрочный желаемый результат обычно считается целью субъекта, промежуточные результаты в достижении цели рассматриваются как подцели или задачи. Разумеется, понятия цели и задачи относительны и зависят от того, о каком периоде времени идет речь.

Инструмент – объект из окружения субъекта, который используется субъектом для достижения результата действия.

Субъект, элементы окружения, инструмент связаны с действием ролевыми отношениями определенных типов, которые называются глубинными падежами [Штерн, 1998].

К числу глубинных падежей относятся следующие отношения:

- агент – инициатор действия;
- объект – то, что подвергается действию;
- пациент – то, что испытывает эффект действия;
- источник – место, с которого начинается действие;
- инструмент и т.п.

Объекты и свойства, связанные с действием, являются его актантами. Если поведение стимулируется целью, поведенческий акт включает выбор цели, действия, инструмента и других актанта из окружения субъекта действия. Выбор цели, действия и актанта необходим, если эти атрибуты поведенческого акта не определены однозначно в окружении субъекта сложившейся ситуацией или прямыми указаниями.

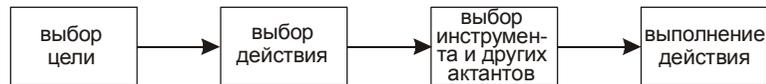


Рис. 1.1. Структура поведенческого акта.

Между целью, действием и актантами действия существует причинно-следственная зависимость, в результате чего возникает временная упорядоченность соответствующих процессов выбора: сначала выбирается цель, затем действие, затем инструмент и другие актанта (рис. 1.1).

Обратим внимание, что в приведенной структуре поведенческого акта помимо действия, служащего его основой, присутствуют обслуживающие действия выбора. Выбор в свою очередь может рассматриваться в качестве базового действия некоторого поведенческого акта. Таким образом, подобно тому, как действие может быть представлено совокупностью составляющих действий, поведенческий акт может расчленяться на более «мелкие» поведенческие акты.

Процессы выбора, входящие в поведенческий акт, осуществляются субъектом действия.

Можно говорить о двух разновидностях процессов выбора целей. Первый состоит в выделении подцелей на основе анализа цели более высокого уровня. Техника сведения (редуцирования) целей к подцелям хорошо разработана. Мы рассмотрим эти вопросы в третьей главе. Намного более сложной является вторая разновидность процессов целеполагания – формирование глобальных целей. Наметим подход к решению этой проблемы в рамках объективной телеологии, т.е. на основе анализа наблюдаемых поведенческих актов. В процессе целеполагания важное значение имеет категория «жизненные потребности субъекта действия». Для человека жизненные потребности – это биологические и духовные параметры его состояния, данные ему в ощущениях, отклонение которых от соответствующих норм является для него опасным или по меньшей мере дискомфортным.

К числу человеческих жизненных потребностей относятся потребности в питании, нормальном осуществлении физиологических процессов, нормальной физико-химической среде обитания, семье, движении, общении, трудовой деятельности, творчестве, впечатлениях, приобретении новых знаний, существовании нравственных идеалов и т.п.

Жизненные потребности, их нормативные значения индивидуальны и переменчивы. Они меняются с возрастом человека и находятся под влиянием его окружения. Жизненные потребности следует отличать от намерений, желаний, целей, формируемых в контексте конкретных окружений. Намерение пойти в библиотеку не является жизненной потребностью, но лишь целью, стимулируемой наличием потребности в приобретении новых знаний. Потребность в пище следует отличать от намерения купить бутерброд.

К числу «жизненных потребностей» компьютерных систем можно отнести потребности в ресурсах (например, данных), необходимых для нормального функционирования системы, в определенной аппаратной и программной среде функционирования, в человеко-машинном интерфейсе, в использовании

результатов работы системы, в приобретении новых знаний и т.п. В компьютерной системе должны быть смоделированы потребности пользователя системы, а также «потребности» объекта, управляемого системой.

Целеполагание состоит в генерации конструктивных целей в контексте существующего окружения субъекта действия, устраняющих отклонения жизненных потребностей от нормативных значений. Жизненные потребности служат критериями оценки выбираемых целей. Моделирование целеполагания на основе жизненных потребностей оказывается простым, если известно отображение жизненных потребностей в элементы окружения субъекта действий. Механическая черепаха формирует цель «вилка питания подключена к розетке», поскольку ее потребность в электрическом токе удовлетворяется при наличии этой связи между ней и ее окружением. В новых, малоизвестных окружениях отображение жизненных потребностей в элементы окружения необходимо формировать, достраивая понятийную сеть субъекта наблюдаемыми объектами, свойствами и связями нового окружения.

Выбор действий и актантов действий осуществляется путем сопоставления цели и окружения субъекта с моделями действий.

Для удовлетворения «жизненных потребностей» компьютерной системы могут быть выбраны, в частности, такие действия, как формирование запроса к пользователю, поиск необходимых ресурсов (например, данных) в коммуникационной сети, поиск в сети агента-партнера и т.п.

Приведенное понимание поведенческого акта требует переосмысление роли компьютера в функционировании человеко-компьютерных систем. Компьютер – не только инструмент действия. Все более широкой становится область участия компьютера в процессах выбора, ранее считавшихся исключительно прерогативой человека.

В классификации функциональных систем Акоф-

фа и Эмери многопрограммные автоматы (компьютеры) отнесены к классу активных многофункциональных, многонаправленных систем. Системы такого типа могут выполнять разные задачи, но при этом сами не определяют, какую именно задачу следует выполнять. Тем не менее, они сами выбирают средства для выполнения своих задач. Таким образом, по представлениям авторов классификации компьютеры способны выбирать действия и актанты действий, но не могут выбирать цели. Функции целеполагания считаются исключительно функцией человека, который отнесен к классу целеустремленных систем, выбирающих как средства выполнения задач, так и сами задачи. Введение категории «жизненные потребности субъекта действия» дает основу для объяснения и моделирования процесса формирования глобальных целей. Современные возможности компьютеров позволяют осуществить такое моделирование. Таким образом, практически все функции субъекта действия становятся доступными для моделирования с помощью компьютера. Роли компьютера и человека в реализации поведенческого акта становятся все более равноценными. Возникает более высокий уровень человеко-компьютерного взаимодействия, который может быть обозначен словом *партнерство*. Компьютер становится сотрудником, мало того, партнером человека в выполнении наиболее сложных процессов поведенческого акта, в том числе процессов целеполагания.

В настоящее время элементы партнерского взаимодействия человека и компьютера разрабатываются в рамках теории агентов и многоагентных систем [Wooldridge, 1994]. Тем не менее современные компьютеры в большей степени ориентированы на поддержку различных видов человеческой деятельности в качестве инструмента. Переориентация компьютеров на выполнение функций субъекта действий требует существенных изменений математического и аппаратного обеспечения. В последующих разделах книги намечены пути продвижения в этом направлении.

1.2. Методологические основы моделирования процессов выбора.

Гипотетическое моделирование

Цели, действия, актанты действий выбираются путем анализа окружения субъекта действия. Анализ осуществляется на основе информационной модели окружения, которая формируется у субъекта действия. При этом исследуются не только наблюдаемые, но и скрытые, имплицитные свойства и отношения, информация о которых возникает не в виде достоверного знания, а в виде гипотез. Гипотеза представляет собой систему суждений, содержащую суждения-предположения, т.е. такие суждения, истинность или ложность которых еще не установлена. Формируемые гипотезы обогащают модель исследуемого или создаваемого объекта. Процесс формирования модели объекта в условиях, когда достоверных знаний о некоторых его существенных характеристиках нет и приобретение таких знаний путем непосредственного наблюдения затруднительно или невозможно, будем называть *гипотетическим моделированием*. Гипотетическое моделирование играет важную роль в творческом мышлении: в науке, проектировании, производственной деятельности, коммерции, политике и т.п. Это – цельный и своеобразный вид творческой деятельности, подчиняющийся особым законам и требованиям. Гипотетические модели часто служат основой при принятии решений, диагностировании, прогнозировании. Изучение и реализация методов гипотетического моделирования на уровне информационных технологий является необходимой ступенькой в развитии интеллектуальных компьютерных систем.

Различные методы гипотетического моделирования в обобщенном рассмотрении сводятся к индуктивным или традуктивным умозаключениям. Индукция – умозаключение, в котором заключение представля-

ет собой знание о классе объектов в целом, полученное в результате исследования отдельных представителей этого класса. В индуктивных умозаключениях заключение всегда является более общим суждением, чем посылки. К формированию гипотез приводит неполная индукция – индуктивное умозаключение, в котором общее заключение о классе объектов выводится на основании изучения лишь части объектов этого класса. В традуктивных умозаключениях посылки и заключение выражают знания одинаковой степени общности. Важнейшим видом традуктивных умозаключений является умозаключение по аналогии. Аналогия, как правило, дает толчок для высказывания предположения. Индукция необходима в тех случаях, когда целью является обобщение данных. В дальнейшем развитии и обосновании гипотезы важную роль играет дедуктивное умозаключение, в котором родовые признаки класса объектов присваиваются отдельным объектам этого класса.

Гипотеза предшествует созданию научной теории. В процессе развития познания гипотеза сменяется теорией либо другой гипотезой. Таким образом, происходит поступательное развитие субъективных представлений к объективному достоверному знанию.

Гипотетическое моделирование является основным методологическим приемом, объединяющим весьма разнородные методы решения задач, описанные в последующих главах.

Сетевые модели

В основе процессов выбора лежат поисковые операции. Объем и время выполнения поисковых операций быстро растут с увеличением объема информации. Основным средством повышения эффективности поисковых операций является использование сетевых структур для моделирования сред, в которых решаются задачи. Ориентация на реальные прикладные среды существенно повышает уровень требова-

ний к их сетевым моделям. Выделим особенности реальных сред, оказывающие сильное влияние на процессы решения задач:

Многосвязность. Реальные среды обычно включают много объектов, связанных большим количеством связей.

Разнородность. Для реальных сред характерно многообразие объектов и отношений.

Иерархичность. В реальных средах приходится оперировать составными объектами, представляющими собой композиции более простых объектов.

Динамичность. Реальные среды, как правило, подвержены частым изменениям.

Учитывая приведенные выше особенности реальных сред, сформулируем требования к сетевой структуре, представляющей среду.

1. Сеть должна обладать развитыми ассоциативными свойствами, т.е. обеспечивать эффективное выполнение разнородных поисковых операций.

2. Сеть должна отражать иерархичность реальных сред и в связи с этим должна быть удобной для представления родо-видовых связей и структур составных объектов

3. В сети должны быть предусмотрены средства, ограничивающие зоны поиска по временным, пространственным или содержательным критериям, т.е. сеть должна обеспечивать избирательность поиска по временному, пространственному или содержательному контексту

4. При построении сети должны формироваться классы объектов и ситуаций; ввод новой информации в сеть должен сопровождаться процессами классификации.

5. Сеть должна допускать параллельное выполнение поисковых операций.

При моделировании сложного поведения сетевая структура должна объединять все виды знаний, используемых в целеустремленном поведении. Различные виды знаний образуют иерархическую систему, отдельные элементы которой связаны структурными



Рис. 1.2. Система знаний

и семантическими связями. Только системное понимание совокупности знаний дает возможность объяснить и смоделировать достаточно сложные процессы мышления. На рис. 1.2 приведена классификация знаний, в основе которой лежат логические формы мышления: понятия, суждения, умозаключения. Понятия и суждения служат «строительным материалом» для формирования агрегированных знаний: умозаключений, моделей действий, планов, сценариев, теорий, моделей процессов и ситуаций.

Система знаний организована в виде сети, вершины которой соответствуют понятиям, а дуги отношениям между понятиями. Понятия наиболее высокого уровня обобщения называют категориями. К числу категорий относятся, например, такие понятия, как «объект», «процесс», «свойство», «действие», «состояние», «отношение», «система», «метод», «вещество», «среда», «территория», «социальная организация», «модель» и т.д. Понятия-категории определяют классы менее общих понятий. В процессах целеполагания, осознания и решения проблем важную роль играет категориальная модель мира, представляющая собой сеть, вершины которой соответствуют категориям, а дуги – обобщенным отношениям, обычно используемым при решении проблем («используется для», «является средством», «преобразуется в», «разрушается в результате», «усиливается при», «происходит в», «является следствием», «агент», «объект» и т.п.). Сеть

категорий образует самый верхний уровень сети понятий. Понятия более низких уровней связаны в сети с соответствующими категориями. Новые понятия классифицируются по категориям путем анализа образующих их признаков.

Сеть категорий включает понятия, определяющие жизненные потребности субъекта действия. С помощью сети категорий осуществляется отображение жизненных потребностей субъекта действия в элементы его окружения, следствием чего является формирование конкретных целей, достижение которых позволяет удовлетворить жизненные потребности.

При решении задач разных типов используются специфические сетевые структуры, возникающие в сети понятий. Мы рассмотрим их при обсуждении методов решения задач.

1.3. Метапроцедуры поведенческого акта

Моделирование процессов поведения, в связи с их многообразием, должно быть организовано как многослойный иерархический процесс, отображающий естественную иерархию процедур управления поведением. На нижнем уровне иерархии находятся процедуры, учитывающие специфические детали конкретных задач, особенности субъектов действий и их окружений. На самом верхнем уровне находятся метапроцедуры управления поведением, не зависящие от особенностей объектных областей, спецификаций параметров и ограничений, особенностей компьютеров и языков представления информации.

Исследование метапроцедур поведения позволяет выявить принципы управления поведением, знание которых упрощает формирование конкретных алгоритмов, учитывающих особенности объектных областей, компьютеров и языков программирования. Метапроцедуры представляют собой стандартную часть матобеспечения компьютерных систем.

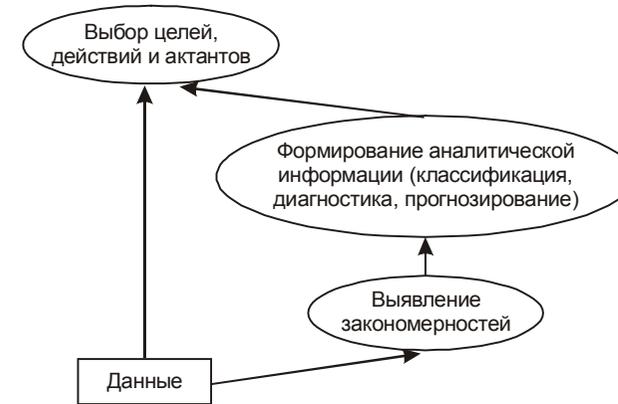


Рис. 1.3. Метапроцедуры управления целеустремленным поведением

Основные метапроцедуры управления целеустремленным поведением показаны на схеме (рис. 1.3).

При выборе целей, действий и актантов используются не только заданные или наблюдаемые характеристики окружения субъекта действия, но и производная аналитическая информация, которая формируется метапроцедурами классификации, диагностики и прогнозирования. Для выполнения этих метапроцедур должны быть известны зависимости определяемой величины от других характеристик окружения субъекта действия. Обычно это имплицитная информация, которая не задана непосредственно и поэтому должна быть сформирована путем анализа известных данных. Таким образом, возникает проблема выявления закономерностей, которая занимает важное место в формировании поведенческого акта [Бонгард, 1967, Аркадьев, 1971, Ивахненко, 1975, Загоруйко, 1988, Закревский, 1988].

В соответствии с приведенной схемой последующий текст книги разделен на две главы. Во второй главе рассматриваются аналитические метапроцедуры (выделение закономерностей, классификация, диагностика, прогнозирование), в третьей – метапроцедуры выбора целей, действий и актантов.

Глава 2

АНАЛИТИЧЕСКИЕ ПРОЦЕССЫ

2.1. Сетевые структуры для аналитических задач

Атрибутивный анализ

Можно выделить четыре наиболее общих класса аналитических задач: обнаружение закономерностей, классификация, диагностика, прогнозирование. Обычно нет расхождений в понимании задач обнаружения закономерностей и классификации. Под диагностикой понимается деятельность, цель которой состоит в определении существующих в данный момент характеристик объекта исследования или факторов, являющихся причинами некоторых его свойств. При прогнозировании целью является определение таких неизвестных характеристик объекта исследования, которые могут возникнуть в результате его дальнейшего развития или при наличии определенных условий.

Названные задачи иногда возникают и решаются отдельно, но часто объединяются в одно комплексное исследование. Как уже было сказано, знание закономерностей необходимо при диагностировании и прогнозировании. Классификация является одним из приемов формирования диагнозов и прогнозов.

Многообразие сред, в которых возникают задачи, сводится к двум нечетко разграниченным типам. Это – среды, образуемые непрерывными объектами, и среды, в которых преобладает антипод непрерывности – дискретность. Для сред первого типа характерны количественные отношения и вычислительные опе-

рации. В средах второго типа обрабатывается качественная, смысловая информация. В непрерывных средах для решения задач используются вычислительные модели, в дискретных – логико-лингвистические [Валькман, 1998], в которых объекты и ситуации представляются атрибутивными (признаковыми) описаниями, а процессы решения задач являются процессами обработки атрибутивных описаний. Логико-лингвистические модели в большей степени пригодны для реализации адаптивных процессов, а также для решения задач в условиях неопределенности и неполноты информации. Они служат основой моделирования процессов мышления и в связи с этим удобны для организации человеко-машинного взаимодействия при автоматизации сложных поведенческих актов.

В этой главе описываются метапроцедуры атрибутивного анализа. В них используются методы гипотетического моделирования, в частности индуктивный вывод и вывод по аналогии. В результате атрибутивного анализа формируется аналитическая имплицитная информация, необходимая для выбора основных элементов поведенческого акта: целей, действий и актантов действий.

Пирамидальные сети

При решении задач разных типов в сети понятий могут формироваться сетевые структуры, наиболее адекватно отображающие особенности решаемых задач. Требованиям атрибутивного анализа в полной мере отвечают пирамидальные сети [Gladun, 1980, 1995, Гладун, 1987, 1995].

Пирамидальной сетью называется ациклический ориентированный граф, в котором нет вершин, имеющих одну заходящую дугу.

Примеры пирамидальных сетей можно видеть на рис. 2.1., 2.2.

Вершины, не имеющие заходящих дуг, называются *рецепторами*, остальные вершины – *концепторами*.

ми. Подграф пирамидальной сети, включающий вершину a и все вершины, от которых имеются пути к вершине a , называются *пирамидой* вершины a . Вершины, входящие в пирамиду вершины a , образуют ее *субмножество*. Множество вершин, к которым имеются пути от вершины a , называется ее *супермножеством*. В субмножестве и супермножестве вершины выделяются θ -субмножество и θ -супермножество, состоящие из тех вершин, которые связаны с ней непосредственно.

Рисунок 2.1. иллюстрирует представление в виде пирамидальной сети фрагмента семантической сети, соответствующего ситуациям:

- 1– Молекула 1 имеет атом А;
- 2– Молекула 2 имеет атом В.

Сеть, показанная на рисунке, является двухъярусной. Рецепторы соответствуют именам отношений, свойств, состояний, действий, объектов и классов

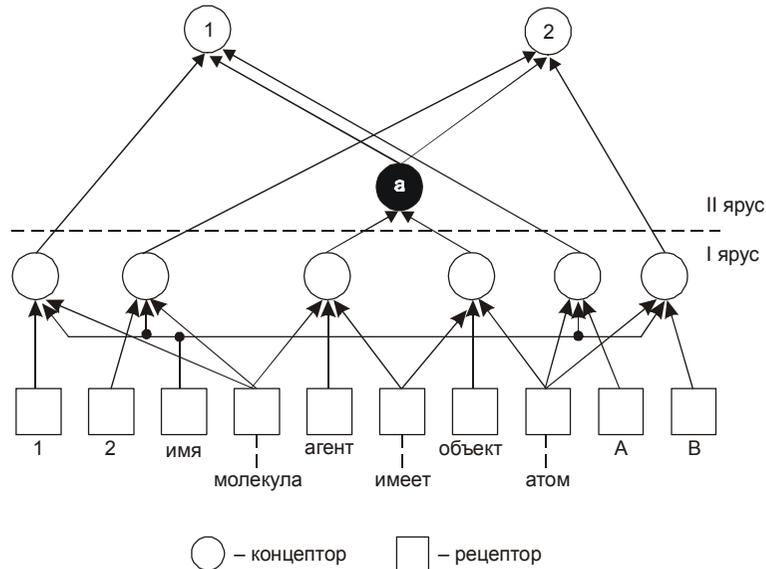


Рис. 2.1. Пирамидальная семантическая сеть.

объектов. Пирамиды первого яруса представляют свойства, состояния объектов, связи между объектами, связи между объектами и действиями, модификации действий, отношений, свойств. «Основание» каждой такой пирамиды образуют рецепторы, один из которых указывает имя отношения, а остальные соответствуют именам объектов, классов объектов, свойств, состояний, действий. Второй ярус образуют концепторы, которые представляют ситуации и повторяющиеся фрагменты ситуаций. Просмотрев пирамиду концептора сверху вниз до рецепторов, можно восстановить описание соответствующего фрагмента ситуации. Например, концептору a соответствует ситуация «Молекула имеет атом», являющаяся общей частью ситуаций 1 и 2. Концепторы сети могут представлять неупорядоченную информацию (например, описания наборов признаков, симметричные отношения) или описания, для которых определено отношение порядка (например, слова, описания несимметричных отношений). Во втором случае упорядоченность элементов, которым в совокупности соответствует концептор, отображается в сети путем нумерации его входящих дуг.

Выделение повторяющихся фрагментов описаний и формирование соответствующих концепторов производится в результате работы алгоритмов построения пирамидальных сетей. Рассмотрим один из наиболее распространенных алгоритмов построения сети для неупорядоченной информации. Новые вершины и дуги вводятся в сеть при переводе какой-либо группы рецепторов в состояние возбуждения. Процесс возбуждения распространяется по сети. Концептор переводится в состояние возбуждения, если возбуждены все вершины его θ -субмножества. Рецепторы и концепторы сохраняют состояние возбуждения в течение выполнения всех операций достройки сети.

Пусть Fa – подмножество возбужденных вершин θ -субмножества вершины a ; G – множество возбужденных вершин сети, не имеющих других возбужденных вершин в своих супермножествах.

Ввод новых вершин производится по следующим двум правилам.

Правило A1

Если вершина a не возбуждена и множество Fa содержит более одного элемента, то дуги, соединяющие вершины из множества Fa с вершиной a , ликвидируются, и в сеть вводится новый концептор, который соединяется заходящими дугами с вершинами множества Fa и исходящей дугой с вершиной a . Новая вершина находится в состоянии возбуждения.

Выполнение правила A1 иллюстрируется рис. 2.2 (I, II). Сеть II возникает после возбуждения в сети I рецепторов 2, 3, 4, 5.

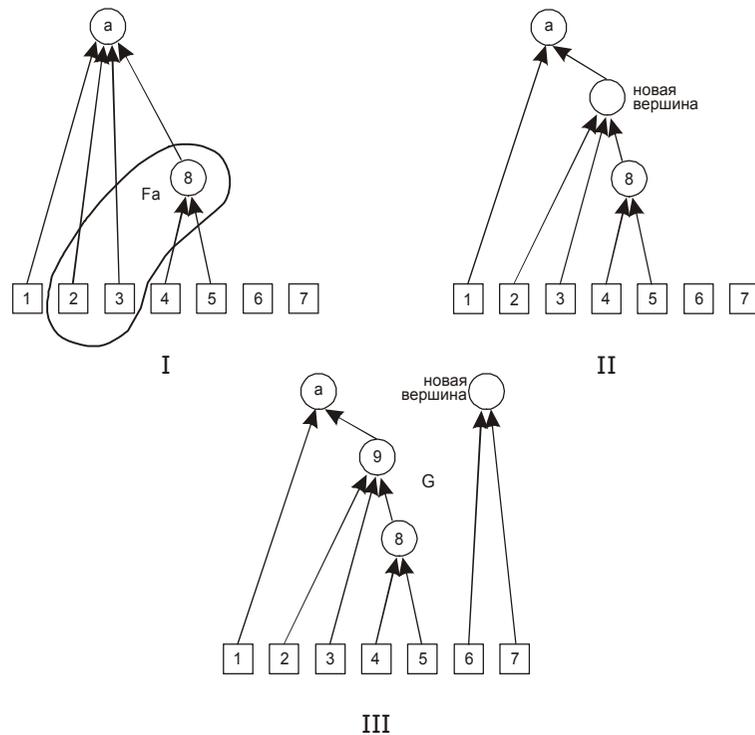


Рис. 2.2. Построение пирамидальной сети.

Как следует из правила A1, условием ввода в сеть новой вершины является ситуация, когда некоторая вершина сети оказывается неполностью возбужденной (возбуждаются не все, но не менее двух вершин ее 0-субмножества). Новые вершины вводятся в субмножества неполностью возбужденных вершин.

После введения новых вершин во все участки сети, где удовлетворяется условие правила A1, выполняется правило A2.

Правило A2

Если множество G содержит более одного элемента, к сети присоединяется новый концептор, который соединяется заходящими дугами со всеми вершинами множества G . Новая вершина находится в возбужденном состоянии.

Выполнение правила A2 иллюстрируется рис. 2.2(II, III). Сеть III возникает после возбуждения в сети II рецепторов 2, 3, 4, 5, 6, 7.

Построение многоярусных пирамидальных сетей осуществляется последовательно по ярусам. Для всех ярусов, кроме первого, роль рецепторов выполняют вершины более низкого яруса, не получившие при его формировании исходящих дуг. Приведенный выше алгоритм может быть применен, например, для формирования второго яруса сети, показанной на рис. 2.1. Различные варианты теоретико-множественного описания процессов построения пирамидальных сетей содержатся в [Гладун, 19987, 1994]

Отметим некоторые свойства пирамидальных сетей.

В зависимости от прикладной области, в которой используются сети, рецептор может представлять значение признака, измеряемый или вычисляемый параметр, элементарный факт из описания ситуации, значение экономического показателя, симптом болезни, букву, слово и т.п. Концепторы соответствуют описаниям объектов, ситуаций, реализаций процессов или явлений, словам, фразам, планам, а также пересечениям описаний.

Пирамидальная сеть является сетевой памятью, автоматически настраиваемой на структуру входной

информации. В результате достигается оптимизация представления информации за счет адаптации структуры сети к структурным особенностям входных данных. Причем в отличие от сетей нейроразподобных элементов эффект адаптации достигается без введения априорной избыточности сети. В некоторых случаях новые возможности обработки информации могут быть получены за счет ввода порогов, дополнительно ограничивающих процессы распространения возбуждения в сети [Морозов, 1997].

Пирамидальные сети удобны для выполнения различных операций ассоциативного поиска. Например, можно выбрать все объекты, включающие заданное сочетание значений признаков, прослеживая пути, исходящие из вершины сети, которая соответствует этому сочетанию. Для выборки всех объектов, описания которых пересекаются с описанием заданного объекта, достаточно проследить пути, исходящие из вершин, образующих его пирамиду. Алгоритм построения сети обеспечивает автоматическое установление ассоциативной близости между объектами по общим элементам их описаний. Все процессы, связанные с построением сети, при обработке одного описания локализуются в относительно небольшой части сети – пирамиде, соответствующей этому описанию.

Важным свойством семантических сетей пирамидальной структуры является их иерархичность, позволяющая естественным образом отображать структуру составных объектов и родо-видовые связи.

Концепторы сети соответствуют сочетаниям значений признаков, определяющих конъюнктивные классы объектов. При включении возбужденных вершин в пирамиду объекта осуществляется привязка объекта к классам, определения которых представлены этими вершинами. Таким образом при построении сети формируются конъюнктивные классы объектов, т.е. осуществляется классификация без учителя. Классифицирующие свойства пирамидальной сети имеют большое значение для автоматизации процессов моделирования сред и ситуаций.

Достоинством пирамидальных сетей является также экономичность за счет того, что одинаковые сочетания значений признаков нескольких объектов представляются в сети одной общей пирамидой.

В пирамидальной сети информация хранится путем ее отображения в структуре сети. Информация об объектах и классах объектов представлена ансамблями вершин (пирамидами), распределенными по всей сети. Внесение новой информации вызывает перераспределение связей между вершинами сети, т.е. изменение ее структуры.

Конечно, в полной мере достоинства пирамидальных сетей проявляются при их физической реализации, допускающей параллельное распространение сигналов по сети. Важным свойством сети как средства хранения информации является то, что возможность параллельного распространения сигналов сочетается в ней с возможностью параллельного приема сигналов на рецепторы от воспринимающих органов системы. Это свойство оказывается полезным при применениях пирамидальных сетей в робототехнических системах, автоматизированных системах научных исследований, системах автоматизированного проектирования.

2.2. Методы, включающие индуктивный вывод

Обнаружение знаний

Для того, чтобы выполнять классификацию, диагностирование или прогнозирование, нужно знать зависимости неизвестных величин, определяемых при решении этих задач, от известных. Возникает необходимость в знании закономерностей, характеризующих объект исследования. Иногда закономерности можно выделить с помощью экспертов. Но в этом случае надежность аналитического процесса уменьшается из-за элементов субъективизма, кото-

рые присущи всем методам экспертного оценивания. В некоторых ситуациях человеческие возможности оказываются недостаточными для выделения сложных закономерностей. Поэтому большое значение имеет проблема автоматизации этого процесса, получившая название «обнаружение знаний» (knowledge discovery). Какой тип знаний является более предпочтительным при диагностировании и прогнозировании? Можно использовать одно-, двух- или трехмерные зависимости, но для получения более надежного диагноза или прогноза лучше иметь модель, отображающую все существенные связи, свойственные объектам, относительно которых решаются эти задачи. Модель классов объектов, используемая для решения задач классификации, диагностики и прогнозирования, должна включать все наиболее важные признаки, характеризующие класс, и отображать характерные для этого класса логические связи между существенными признаками. Поэтому мы, прежде всего, сосредоточим внимание на формировании обобщенных логических многомерных моделей классов объектов. Такого рода модели, по сути, являются понятиями, соответствующими классам объектов [Войшвилло, 1967, Горский, 1985].

В логике понятие обычно определяется как мысль, отражающая сущность объектов. Большинство употребляемых понятий является результатом обобщения представлений об объектах некоторого класса по существенным, специфическим для этого класса признакам. К признакам относится все, что характеризует объекты и может быть использовано в таких логических операциях, как выделение, распознавание, отождествление и т.п. Следует отметить, что разделение признаков на существенные и несущественные в значительной степени условно и зависит от задач, для решения которых они используются.

Понятие по отношению к слову – его смысл, информация, которую содержит слово о возможных денотатах – обозначаемых словом реалиях. Например, денотатом слова «книга» может быть некоторая

конкретная книга, о которой идет речь в тексте. Понятие, связанное со словом «книга», характеризует множество всех книг – возможных денотатов этого слова. Множество обобщенных в понятии объектов составляет его *объем*. Понятие является лексическим правилом употребления слова. В системе знаний понятия играют роль базовых элементов, из которых складываются суждения и другие логические формы мышления. Переход от чувственной ступени познания к абстрактному мышлению, по сути, означает переход от отражения мира в форме ощущений и представлений к отражению его в понятиях. Являясь концентратом знания, итогом некоторого этапа познания, понятие служит важнейшим инструментом формирования новых знаний и решения задач.

На основе имеющегося запаса понятий осуществляются классификация, обобщение, структурирование воспринимаемой информации, включение ее в систему знаний. В этих процессах реализуются две основные функции понятий – распознавание и генерация моделей элементов мира, в котором оперирует носитель знаний. Процессы распознавания давно стали объектом исследования и автоматизации, в то время как генерация моделей пока еще является малоисследованной проблемой. Генерация моделей представляет собой процесс формирования моделей конкретных объектов путем введения в понятия других понятий и констант. Генерация моделей играет важную роль в творческой деятельности. Только путем конкретизации понятий мы можем создавать (например, рисовать) образы конкретных домов, деревьев, автомобилей и т.п. Генерация моделей элементов мира лежит в основе проектирования инженерных объектов.

Входящие в понятие признаки по их роли в реализации основных функций понятия делятся на два типа – *разделительные* и *объединительные*. К разделительным признакам относятся признаки, которые за пределами объема понятия не встречаются вообще или встречаются редко. Эти признаки наи-

более эффективны при реализации функции распознавания. Объединительными признаками называются те признаки, которые присущи всем или многим элементам объема понятия, но могут быть распространены и за пределами объема понятия. Без этих признаков невозможна генерация моделей элементов мира. Например, для всех берез характерны такие признаки, как наличие ствола, корней, кроны. Это их объединительные признаки, которые присущи всем деревьям. Хорошо известным разделительным признаком берез является белый цвет коры.

Степень детализации создаваемой на основе понятия модели зависит от цели задачи, при решении которой осуществляется ее генерация. Модель моста, создаваемая при решении задачи «нарисовать мост», существенно отличается по степени детализации от модели моста, создаваемой при решении задачи «сконструировать мост». Успех решения задач, включающих генерацию моделей, зависит от того, насколько используемые понятия правильно и полно характеризуют соответствующие классы элементов мира.

Теперь можно дать более конструктивное определение понятия, более пригодное при рассмотрении информационно-технических аспектов проблем формирования и обработки понятий.

Понятие – элемент системы знаний, представляющий собой обобщенную модель некоторого класса объектов, с помощью которой реализуются процессы распознавания и генерации моделей конкретных объектов.

В процессах распознавания и генерации моделей понятие используется как логическая функция признаков, имеющая значение «истина» для объектов из объема понятия и значение «ложь» в остальных случаях.

Совокупность понятий, входящих в систему знаний, в свою очередь обладает рядом системных характеристик, в связи с чем целесообразно говорить о системе понятий носителя знаний.

Рассмотрим некоторые свойства систем понятий, соответствующих предметным областям.

Системы понятий, как правило, иерархичны. Объемы понятий всех уровней иерархии, кроме нижнего, образуются объединением объемов ряда понятий более низких уровней. Например, объем понятия «фрукт» объединяет объемы понятий «яблоко», «груша» и т.п.

Системы понятий динамичны. Состав понятий меняется в результате взаимодействия их носителей с окружающей средой, а также в процессе решения задач.

В каждый момент времени состояние системы понятий отражает индивидуальный опыт ее носителя. Поэтому отдельные понятия и системы понятий в целом субъективны.

Ни одна система понятий в силу своей дискретности, ограниченности состава понятий, несовершенства отдельных понятий не может отразить многообразие и непрерывность реального мира. Объемы понятий, не введенных «по определению», как правило, не имеют четких разделяющих границ. Существует много переходных форм, затрудняющих проведение условных границ между объемами понятий. Из-за неполноты отображения мира в системе понятий, а также субъективности понятий однозначная идентификация элементов мира на основе системы понятий часто оказывается затруднительной или даже невозможной. В связи с этим объемы многих понятий можно рассматривать как нечеткие множества, т.е. такие множества, на элементах которых задана функция принадлежности, указывающая степень уверенности в принадлежности элемента множеству. Каждый носитель системы понятий обладает своей функцией принадлежности, которая, таким образом, имеет субъективный характер. Понятия, которым соответствуют нечеткие множества, получили название *нечетких понятий*.

По логической структуре различают конъюнктивные понятия, т. е. понятия, которые можно описать

конъюнкцией признаков, дизъюнктивные понятия, описываемые дизъюнкциями конъюнкций, а также понятия с исключаящими признаками, отражающие отсутствие у объектов из объема понятия некоторых признаков.

Логическая структура понятия зависит от пространства признаков, в котором оно сформировано. Правильный выбор пространства признаков позволяет сформировать понятия простейшей логической структуры. Понятия, вошедшие в широкое употребление, конъюнктивны. Более сложная логическая структура характерна для понятий, формируемых в процессе научно-исследовательской деятельности. В этом случае причиной логической сложности понятий обычно являются следующие обстоятельства: неправильно выбрано пространство признаков; обучающая выборка, на основе которой формируется понятие, неполно отражает специфику объема понятия; объем формируемого понятия в соответствии с целями исследования искусственно образован из объектов, значительно отличающихся друг от друга.

Эти причины обычно сопутствуют начальным, недостаточно высоким уровням познания объекта исследования.

Рассмотрим задачу индуктивного формирования понятий для непересекающихся множеств объектов V_1, V_2, \dots, V_n . Пусть L – множество объектов, используемое в качестве обучающей выборки. Имеют место соотношения $L \cap V_i \neq \emptyset$ и $V_i \not\subset L$ ($i=1, 2, \dots, n$). Заданы признаковые описания всех элементов L . Каждый признак $l \in L$ снабжен указанием типа $l \in V_i$. Требуется путем анализа L сформировать n понятий с объемами V_1, V_2, \dots, V_n , достаточных для правильного распознавания всех объектов $l \in L$.

Понятие, сформированное на основе обучающей выборки, в общем случае является приближением к действительному понятию, причем степень близости этих понятий зависит от представительности обучающей выборки, т.е. от того, насколько полно в ней отражены особенности объема понятия. Задача ин-

дуктивного формирования понятий близка к задаче обучения распознаванию образов. И в том и в другом случае в результате обучения строится некоторая модель класса объектов. При формировании по-

№	OBJECT	CLASS	U	GAP	A	C	CA
1	ZnSnAs2	1	3	3	10	12	15
2	CuInSe2	1	2	5	8	12	15
3	CuTlSe2	1	3	6	8	12	15
4	CuInS2	1	1	8	4	7	15
5	ZnSnP2	1	4	9	4	10	15
6	CuGaSe2	1	5	9	4	6	13
7	AndInS2	1	5	11	8	8	7
8	CuAlTe2	1	5	11	13	12	13
9	AgAlTe2	1	7	12	16	12	5
10	ZnGeP2	1	6	12	3	4	11
11	AgGaS2	1	11	13	8	1	2
12	CuAlSe2	1	7	13	4	6	9
13	AgAlSe2	1	9	13	12	4	2
14	AgAlS2	1	14	14	6	1	2
15	CuAlS2	1	9	15	2	2	11
16	CdSnAs2	2	8	1	15	13	11
17	CuFeS2	2	9	2	1	3	12
18	CdGeAs2	2	10	2	12	9	6
19	ZnGeAs2	2	5	4	5	7	12
20	AgInTe2	2	5	5	18	14	10
21	AgGaTe2	2	7	7	17	13	6
22	CdSnP2	2	8	7	11	11	8
23	CuGaTe2	2	5	7	14	13	14
24	CdSiAs2	2	13	8	9	5	3
25	CdGeP2	2	12	10	7	5	4
26	AgGaSe	2	9	10	7	1	1

Таблица 2.1. Пример обучающей выборки.

нятий к этой модели (понятию) предъявляются более сильные требования. Она должна обеспечивать не только распознавание, но и возможность генерации моделей конкретных объектов. В связи с этим в ней должны быть отражены признаковые, структурные, логические характеристики объектов.

При формировании понятия, соответствующего множеству V_p , объекты обучающей выборки, входящие в V_p , рассматриваются как примеры множества V_i (положительные объекты), а объекты, не входящие в V_p – как контрпримеры множества V_i (отрицательные объекты).

Обучающая выборка обычно имеет форму таблицы (табл.2.1). Строки соответствуют признаковым описаниям объектов, столбцы – признакам. В приведенном примере объектами являются химические соединения, принадлежащие двум классам. Имена классов указаны в специальном столбце. Понятие, которое формируется в результате анализа обучающей выборки, обычно описывается логическим выражением, в котором переменными служат имена значений признаков.

Известные методы формирования понятий [Бонгард, 1967, Поспелов, 1986, Гладун, 1987, Glagun, 1980, 1995, Вагин, 1988, Michalski, 1986, Piatetsky-Shapiro, 1991] по сути являются методами управляемого выбора значений признаков, характеризующих классы объектов. Выбор может быть упрощен за счет использования адекватного представления анализируемой информации. Во многих алгоритмах с этой целью используется специальная структура, называемая «деревом решений». Дерево решений разделяет объект на конечное число классов. В дереве решений [Quinlan, 1986], показанном на рис. 2.3, объектами являются состояния погоды. Вершины дерева отмечены именами признаков, ребра соответствуют возможным значениям признаков. Листья (концевые вершины) помечены именами классов. Объект классифицируется путем прослеживания дерева сверху вниз по ребрам, соответствующим значениям его при-

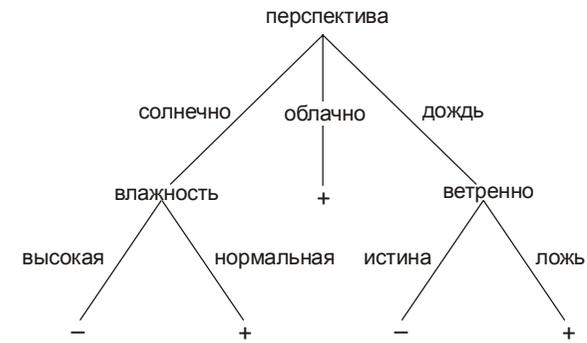


Рис. 2.3. Дерево решений

знаков. Недостатком алгоритмов, использующих деревья решений, является быстрый рост временных затрат при увеличении объема обучающей выборки. Это серьезно препятствует их применению для решения практических задач. Еще в большей степени этот недостаток присущ другим методам, (например, эволюционным, использующим технику «грубых» множеств или нейрноподобные структуры [Michalski, 1986, Piatetsky-Shapiro, 1991]).

Сформулируем требования к методам формирования понятий.

1. Для повышения надежности диагноза или прогноза необходимо учитывать зависимость определяемой величины от сочетаний известных признаков, т.е. принимать во внимание совместное одновременное влияние признаков. Формируемое понятие должно отражать зависимости такого рода.

2. В зависимости от выбора метода обучения понятиям для одного и того же класса объектов могут быть получены различные логические описания. Естественно, возникает вопрос о качестве логических моделей. Наилучшие результаты применения понятий для классификации, диагностики и прогнозирования, как правило, соответствуют более обобщенным понятиям, т.е. понятиям, которые описываются более простыми логическими выражениями. Степень сложности логического выражения может быть оценена числом его переменных. Метод обучения поня-

пирамиды вершины 2 в качестве контрольной вершины выбирается вершина 6, так как она имеет наибольшее k из всех вершин, имеющих наибольшее m_i (6, 12, 13).

Правило В2

Если в пирамиде объекта из объема понятия A_i есть контрольные вершины других понятий, не содержащие в своих супермножествах возбужденных контрольных вершин понятия A_i , в каждом из этих супермножеств вершина, имеющая наибольшее k из всех возбужденных вершин с наибольшим m_i , отмечается как контрольная вершина понятия A_i .

В соответствии с правилом В2 возбуждение пирамиды вершины 2 (рис. 2.5.а) при условии, что она представляет объект из объема понятия A_i , приводит к выделению в качестве контрольной вершины понятия A_i вершины 5 (рис. 2.5б).

С помощью контрольных вершин осуществляется выделение наиболее характерных (имеющих наибольшее m_i) сочетаний значений признаков, принадлежащих объектам из объема понятия. Например, выделение вершины 8 (рис. 2.5) в качестве контрольной вершины означает выделение сочетания значений признаков, соответствующих рецепторам 16, 17, 18.

Если при просмотре всех объектов обучающей выборки появилась хотя бы одна новая контрольная вершина, т.е. хотя бы один раз выполнялись условия, содержащиеся в правилах В1 и В2, осуществляется новый просмотр всех объектов обучающей выборки. Работа алгоритма оканчивается, если при очередном просмотре обучающей выборки не возникает ни одной новой контрольной вершины.

После этого может быть применено следующее правило распознавания.

Объект входит в объем понятия A_i , если в его пирамиде имеются контрольные вершины понятия A_i и нет ни одной контрольной вершины какого-либо другого понятия, не содержащей возбужденных кон-

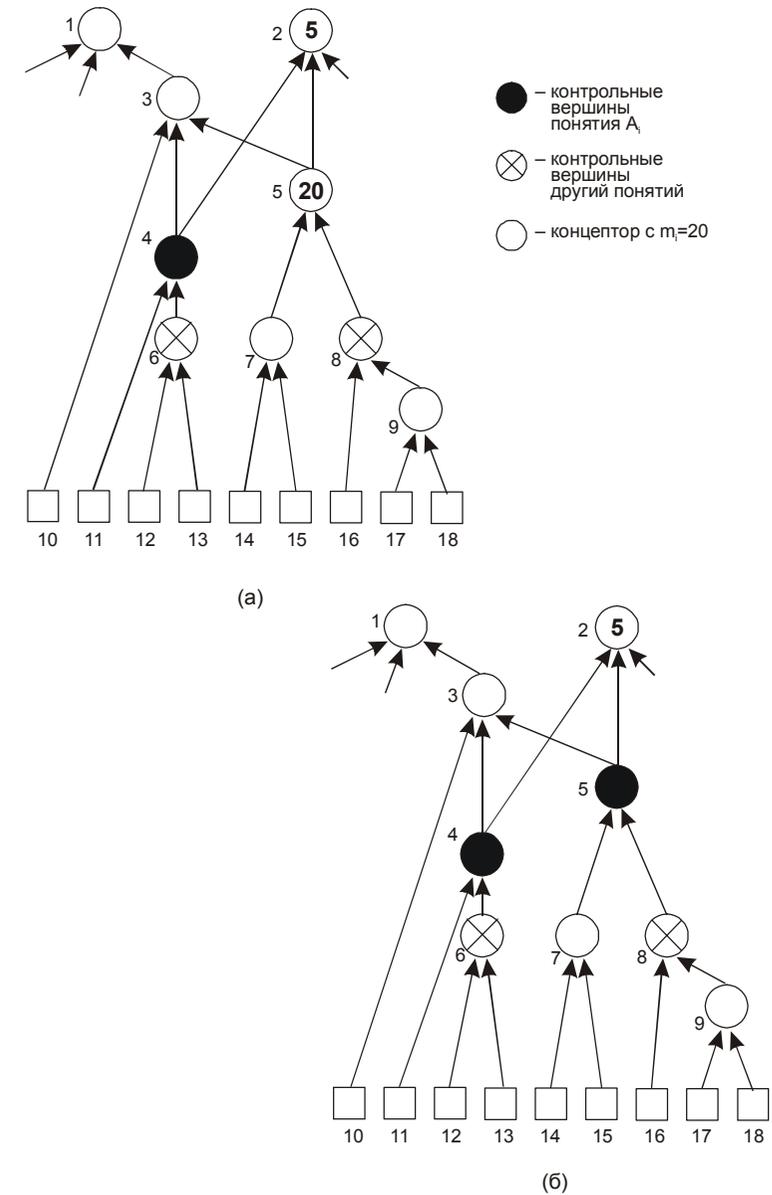


Рис. 2.5. Правило В2 алгоритма формирования понятий.

трольных вершин понятия A_i в своем супермножестве. Если это условие не выполняется ни для одного из понятий, объект считается неопределенным.

Утверждение 2.1. Время выполнения алгоритма всегда конечно. Если объемы формируемых понятий V_1, V_2, \dots, V_i не пересекаются, то после выполнения алгоритма приведенное правило распознавания полностью разделяет обучающую выборку на подмножества $L_i = V_i \cap L (i=1, n)$.

Утверждение 2.1 обосновывается следующими соображениями. В процессе работы алгоритма при неправильном распознавании объекта обучающей выборки осуществляется корректировка формируемого понятия путем выделения внутри возбужденной пирамиды контрольных вершин таким образом, чтобы получить правильное распознавание. Если правильное распознавание объекта, которому соответствует пирамида, не будет достигнуто раньше, то процесс корректировки внутри пирамиды приведет к выделению в качестве контрольной самой верхней вершины пирамиды. После этого для распознавания объекта, которому соответствует пирамида, по приведенному правилу распознавания будет использоваться только эта контрольная вершина и, следовательно, никакие последующие изменения внутри пирамиды (выделение новых контрольных вершин в связи с возбуждением других пирамид) не смогут повлиять на процесс распознавания и сделать распознавание неправильным. Таким образом, процесс выделения контрольных вершин в пирамиде всегда приводит к правильному распознаванию объекта, которому она соответствует. Этот процесс конечен, поскольку конечно число вершин пирамиды. Из сходимости процесса выделения контрольных вершин внутри каждой пирамиды следует сходимость алгоритма в целом. Поскольку выделение контрольных вершин внутри каждой пирамиды приводит к правильному распознаванию объекта, которому соответствует пирамида, алгоритм в целом приводит к правильному разделению объектов обучающей выборки на подмножества L_i .

Заметим, что доказательство сходимости никак не обусловлено сложностью формируемых понятий, т.е. алгоритм является сходящимся для понятий любой сложности.

Понятия, формируемые в результате работы алгоритма, представлены ансамблями контрольных вершин пирамидальной сети. Существует алгоритм с помощью которого формируется описание понятия, представленного в пирамидальной сети, в виде логического выражения [Гладун, 1994]. Например, понятие, представленное на рис. 2.5б контрольными вершинами с «крестами», описывается следующим выражением

$$(12 \wedge 13) \wedge \neg 11 \vee (16 \wedge 17 \wedge 18) \wedge \neg (14 \wedge 15).$$

Распознавание объектов по записи понятия в виде логического выражения можно выполнять, вычисляя значение логического выражения. Переменным, которые соответствуют значениям признаков, принадлежащим распознаваемому объекту, присваивается значение 1, остальным переменным – значение 0. Единичное значение всего выражения означает, что объект входит в объем понятия. В понятии, которое формируется алгоритмом, отражены общие существенные признаки объектов из объема понятия и логические связи между признаками, характерные для объема понятия. Объединительные признаки выделяются в результате выполнения правила $B1$. При выполнении правила $B2$ выделяются разделительные признаки.

В зависимости от характера задачи при распознавании новых объектов выполняется диагностирование или прогнозирование.

Метод позволяет осуществлять диагностирование или прогнозирование на основе многомерных зависимостей исследуемых величин от различных сочетаний признаков, т.е. дает возможность учесть эффект совместного влияния нескольких признаков.

Важной особенностью метода формирования понятий в пирамидальных сетях является возможность

включения в понятия признаков, не принадлежащих объектам соответствующего класса, т.е. признаков таких объектов других классов, которые подобны объектам исследуемого класса и поэтому могут рассматриваться как «исключения» из него. В результате формируемые понятия имеют более компактную логическую структуру, что в принципе дает возможность повышать точность диагноза или прогноза. В логическом выражении признаки «исключений» представлены переменными с отрицаниями.

Реализация процесса формирования понятий в пирамидальной сети позволяет избежать больших переборов информации, в результате чего появляется принципиальная возможность решать практические аналитические проблемы на основе больших объемов данных.

Для обобщенной характеристики алгоритма формирования понятий в пирамидальной сети используем геометрическую интерпретацию. Каждой вершине сети, имеющей k рецепторов в своем субмножестве, соответствует конъюнкция имен значений признаков k -го ранга (рецептору соответствует одно имя, т.е. конъюнкция первого ранга). Конъюнкции имен значений признаков описывают многомерные плоскости признакового пространства. Следовательно, каждой вершине сети, имеющей k рецепторов в своем субмножестве, в s -мерном признаковом пространстве соответствует $(s-k)$ -мерная плоскость, в которую входят все точки, представляющие объекты, при восприятии которых возбуждается эта вершина. $(s-k)$ -мерные плоскости, соответствующие контрольным вершинам понятия A_r , будем называть *зонами понятия* A_r . Для пирамидальных сетей справедливы следующие утверждения.

Утверждение 2.2. Зона любой вершины сети целиком входит в зоны вершин ее субмножества и полностью включает зоны вершин ее супермножества.

Утверждение 2.3. Точка, представляющая объект в признаковом пространстве, находится внутри области, образованной пересечением зон контрольных

вершин, которые возбуждаются при восприятии этого объекта.

Будем говорить, что точка a , представляющая объект в пространстве объектов, непосредственно входит в зону Z понятия A_r , если не имеется других зон того же понятия, содержащих точку a и целиком входящих в зону Z .

Дадим геометрическую интерпретацию приведенных выше правил.

Правило В1

Для каждого объекта из объема понятия A_i $(s-k)$ -мерная плоскость одной из возбужденных вершин, имеющих наибольшее значение m_r , преобразуется в зону понятия A_i .

Правило В2

Если точка, представляющая объект из объема понятия A_i в признаковом пространстве, оказывается непосредственно внутри зон других понятий, то внутри каждой из этих зон создается зона понятия A_i .

Работа алгоритма формирования понятий оканчивается, когда при очередном просмотре обучающей выборки точки, представляющие объекты из объема любого из формируемых понятий, ни разу не попадают непосредственно внутрь зон других понятий. После обучения объект считается принадлежащим объему понятия A_r , если точка, представляющая его в признаковом пространстве, входит непосредственно хотя бы в одну зону этого понятия и не входит непосредственно ни в одну из зон других понятий.

Таким образом, в результате работы алгоритма для каждого из формируемых понятий в признаковом пространстве из зон строится область, содержащая все точки, представляющие те объекты обучающей выборки, которые входят в объем понятия, и не содержащая ни одной из точек, представляющих другие объекты обучающей выборки. Эта область аппроксимирует область распределения объектов из объема понятия. Поскольку аппроксимирующая об-

ласть состоит из линейных элементарных областей (гиперплоскостей), ограничивающая ее поверхность является кусочно-линейной. Следовательно, алгоритм осуществляет кусочно-линейной разделением объектов, входящих в объемы различных понятий.

Назовем зоны понятия A_i , непосредственно содержащие как точки, соответствующие объектам из его объема, так и точки, соответствующие объектам, не входящим в его объем, *границными зонами* понятия A_i .

Утверждение 2.4. По правилу *B2* новые зоны могут создаваться только непосредственно внутри граничных зон.

Формирование новых зон внутри граничных зон приводит к дроблению граничных зон.

Построение аппроксимирующей области понятия A_i состоит из двух процессов – грубого покрытия зонами понятия A_i области распределения объектов обучающей выборки, входящих в объем понятия A_i (правило *B1*), и дробления возникающих граничных зон (правило *B2*).

Объясним сходимость алгоритма, используя приведенную геометрическую интерпретацию.

Для каждого понятия полное покрытие зонами области распределения объектов обучающей выборки, входящих в его объем, достигается в течение одного просмотра обучающей выборки. Из определения граничных зон следует, что они содержат точки объектов обучающей выборки, для которых выполняются условия правила *B2*. Поэтому при каждом просмотре обучающей выборки осуществляется деление всех граничных зон, возникших на предыдущем просмотре.

Процесс деления граничных зон продолжается, пока существуют граничные зоны, и может привести к выделению в качестве зон отдельных точек признакового пространства. Так как число точек, представляющих объекты обучающей выборки, в каждой граничной зоне конечно, процесс деления граничных зон конечен. Отсутствие граничных зон после окончания процесса деления означает, что для каждого

из понятий в признаковом пространстве построена область, содержащая все точки, представляющие объекты обучающей выборки, входящие в его объем, и не включающая ни одной из точек, представляющих другие объекты обучающей выборки. Таким образом, после окончания деления граничных зон достигается полное разделение обучающей выборки на подмножества $L_i = V_i \cap L$ ($i=1, 2, \dots, n$). Возможны различные модификации алгоритма формирования понятий на основе пирамидальной сети, отличающиеся формой зон контрольных вершин.

Недетерминистский процесс формирования понятий.

Рассмотрим модификацию алгоритма формирования понятий в пирамидальной сети для недетерминистского процесса обучения, т.е. для случая, когда возможно пересечение объемов разных понятий. При недетерминистском процессе обучения в обучающей выборке имеются одинаковые объекты с противоречивыми указаниями $L \in V_i$ и $L \in V_j$ при $i \neq j$. Очевидно, в этом случае правило распознавания объектов после обучения должно быть вероятностным и должно отображать статистические свойства обучающей выборки. Недетерминистское формирование понятий широко распространено в практике, поэтому изучение и автоматизация этого процесса имеют важное значение. Мы рассмотрим здесь лишь отличия модифицированного алгоритма от алгоритма, описанного в предыдущем разделе, поскольку во многом эти алгоритмы совпадают. Пусть обучающая выборка включает объекты, входящих в объемы двух понятий A_1 и A_2 ($i=1, 2$). Вершины сети отмечаются одним из трех признаков – П1, П2 и П3, указывающих тип объекта, которому соответствует пирамида. Признак П1 соответствует объектам из объема понятия A_1 , признак П2 – объектам из объема понятия A_2 . Признак П3 вносится в двух случаях – при вос-

приятии объекта из объема понятия a_1 , если его пирамида уже отмечена признаком П2, или при восприятии объекта из объема понятия a_2 , если его пирамида уже отмечена признаком П1. Таким образом, признаком П3 отмечаются пирамиды объектов, которые встречаются в обучающей выборке несколько раз, причем с противоречивыми указаниями.

Помимо контрольных вершин понятия A_i , введем новый тип контрольных вершин – *двузначные контрольные вершины*.

Правило В2 алгоритма, описанного в предыдущем параграфе, несколько изменяется.

Правило В2

Если пирамида объекта из объема понятия A_i не помечена признаком П3 и содержит возбужденные двузначные контрольные вершины или контрольные вершины другого понятия, не содержащие в своих супермножествах возбужденных контрольных вершин понятия A_i , в каждом из этих супермножеств вершина, имеющая наименьшее k из всех возбужденных вершин с наибольшим m_k , отмечается как контрольная вершина понятия A_i .

Правило В3

Если пирамида объекта помечена признаком П3 и в ней содержатся контрольные вершины каких-либо понятий, не содержащих в своих супермножествах других возбужденных контрольных вершин, одна из вершин, имеющих наименьшее k среди всех вершин пересечения этих супермножеств, отмечается как двузначная контрольная вершина.

Дадим геометрическую интерпретацию приведенных правил.

Зоны двузначных контрольных вершин будем называть *двузначными зонами*. В соответствии с правилом В3 двузначная зона создается при восприятии объекта, имеющего противоречивые указания, внутри области пересечения зон, в которые непосредственно входит точка, представляющая воспринимаемый объект в признаковом пространстве. Изменения,

внесенные в правило В2, дают возможность формировать зоны понятий внутри двухзначных зон. Как и прежде, в результате работы алгоритма в пространстве признаков строится область, аппроксимирующая область распределения объектов из объема понятия. Двузначными зонами покрывается область пересечения областей распределения объектов из объемов различных понятий. Необходимо предусмотреть возможность преобразования зон, состоящих из отдельных точек, в двузначные зоны в тех случаях, когда оказывается, что объект, которому соответствует зона, встречается в обучающей выборке повторно с противоположным указанием. Это преобразование выполняется в соответствии со следующим правилом.

Правило В4

Если в пирамиду вносится признак П3, причем во главе пирамиды стоит контрольная вершина понятий A_1 или A_2 , эта контрольная вершина преобразуется в двузначную контрольную вершину.

Формирование понятия оканчивается, если при очередном просмотре обучающей выборки не возникает ни одной новой контрольной вершины.

После обучения применяется следующее правило распознавания.

Объект входит в объем понятия A_i ($i=1, 2$), если в его пирамиде: 1) имеются контрольные вершины понятия A_i и нет контрольных вершин других понятий, не содержащих возбужденных контрольных вершин понятия A_i в своих супермножествах; 2) для всех двузначных контрольных вершин, не содержащих других возбужденных контрольных вершин в своих супермножествах, выполняется условие

$$\frac{c^+ \sum_S m_i}{c^- \sum_S m_j} > 1, \quad (2.1)$$

где S – множество возбужденных двузначных контрольных вершин, не содержащих других возбужден-

ных контрольных вершин в своих супермножествах; c^+ – вес ошибки, состоящей в нераспознавании объекта из объема понятия A_i ; c^- – вес ошибки, состоящей в нераспознавании объекта из объема другого понятия; $j \neq i$. Если условие правила не выполняется ни для одного из понятий, объект считается неопределенным.

Как следует из правила распознавания, объект не причисляется к объему понятия A_i , если точка, представляющая его в признаковом пространстве, входит непосредственно хотя бы в одну зону другого понятия. В случае если точка, соответствующая объекту, входит в двузначные зоны, решение принимается путем проверки неравенства (2.1).

С ростом объема обучающей выборки правило распознавания на основе неравенства (2.1) в пределе дает известное правило Байеса, минимизирующее риск ошибки распознавания.

Применения методов анализа на основе пирамидальных сетей.

Метапроцедуры анализа на основе пирамидальных сетей реализованы в системе вывода и анализа закономерностей CONFOR [Гладун, 1995, Киселева, 1998]. Система CONFOR предназначена для решения задач обнаружения знаний, классификации, диагностики, прогнозирования.

Основные функции системы:

- Обнаружение знаний (формирование многомерных моделей классов объектов) на основе анализа данных, хранящихся в базах данных;
- Использование сформированных моделей в целях классификации новых объектов, диагностики и прогнозирования.

Исходными данными для системы CONFOR служат описания примеров и контрпримеров исследуемого класса объектов (процессов, ситуаций, явлений и т.п.), заданных наборами значений признаков.

Закономерность представляется в виде логического выражения в терминах исходных значений признаков, благодаря чему результаты работы системы CONFOR являются наглядными и легко интерпретируемыми. Помимо решения указанных выше аналитических задач CONFOR предоставляет пользователю широкий спектр возможностей логического анализа данных, позволяющего, например:

- Установить сходство и различие между заданными объектами;
- Получить перечень объектов, включающих заданное сочетание значений признаков;
- Посмотреть, на какие объекты похож заданный объект, и в чем это сходство проявляется;
- Определить сочетания значений признаков, которые являются наиболее характерными для исследуемого класса объектов.

Характерными задачами для системы CONFOR являются прогнозирование существования химических соединений с заданными свойствами, прогнозирование получения необходимых материалов (сегнетоэлектриков, ферромагнетиков, полупроводников, сверхпроводников, с оптическими свойствами и т.п.), анализ и прогнозирование объемов сбыта и ценовых характеристик товаров, вывод закономерностей, характеризующих предприятия и регионы, классификация экономических ситуаций, диагностика заболеваний, проектирование технологий, прогнозирование солнечной активности, диагностика неисправностей в технических устройствах и т.п.

Приведем краткие описания некоторых из применений системы.

Прогнозирование существования химических соединений. По ряду обстоятельств экспериментальная проверка существования соединений некоторых химических элементов затруднена. В связи с этим важно иметь описания типов соединений, с помощью которых можно было бы прогнозировать возможность образования соединений для любых заданных химических элементов. В этом случае объектами

служат пары химических элементов. Положительными объектами обучающей выборки считаются те пары элементов, относительно которых известно, что они образуют соединение исследуемого типа, отрицательными – пары элементов, не образующие соединения. В качестве признаков используются физические характеристики элементов, в частности атомарные.

Диагностика заболеваний. CONFOR применяется для составления описаний болезней, диагностика которых затруднительна. Объектами служат наборы симптомов, характеризующие течение болезни у отдельных больных.

Прогнозирование сильных вспышек на Солнце. В этой задаче объектами для обучения служили наборы значений физических параметров, измеряемых при наблюдении вспышек на Солнце. В результате обучения сформировалось обобщенное описание класса ситуаций, предшествующих появлению сильных вспышек. Построенное описание использовалось для прогнозирования вспышек.

Качество понятий, его уровень обобщения зависят от:

1. Эффективности используемого алгоритма формирования понятий
2. Представительности обучающей выборки, в частности от того, насколько полно в ней представлены граничные объекты, т.е. объекты, точки которых в пространстве признаков находятся вблизи поверхности, разделяющей положительные и отрицательные области;
3. Выбора исходного набора признаков, которыми определяются объекты.

Выбор пространства признаков. Формирование знаний на основе изучения объектов является сложным исследовательским процессом, в котором можно выделить два основных этапа:

1. Выбор признаков, используемых для характеристики исследуемых объектов, и формирование описаний объектов.
2. Выделение закономерностей, свойственных исследуемым объектам, путем анализа их описаний.

Начальный отбор признаков, характеризующих, по мнению исследователя, объекты и свойственные им закономерности, выполняется на основе уже имеющихся опытных знаний. К пространству признаков предъявляются следующие требования.

1. *Разделимость классов.* Области распределения в пространстве признаков объектов разных классов должны быть разделены. Это означает, что обучающая выборка не должна включать объекты разных классов, имеющих одинаковые признаковые описания.

2. *Компактность распределений.* Следует стремиться к компактности распределений объектов разных классов в пространстве признаков. Компактность распределений двух классов [Аркадьев, 1971] измеряется числом так называемых граничных точек. Точка, представляющая объект некоторого класса, считается граничной точкой этого класса, если среди ее соседних точек в пространстве признаков имеются точки, представляющие объекты других классов. Другими словами, точка, представляющая объект некоторого класса, является граничной точкой этого класса, если признаковое описание соответствующего объекта может быть трансформировано в описание объекта другого класса путем замены значения хотя бы одного признака соседним значением по шкале значений этого признака.

Более компактные распределения содержат меньше граничных точек. Компактные распределения разделяются более простыми разделяющими поверхностями. Компактным распределениям соответствуют более простые логические модели.

3. *Простота.* Для построения более простых логических моделей классов объектов необходимо стремиться по возможности уменьшать число признаков и число их значений, конечно, не нарушая разделимости классов. Пространство признаков должно включать лишь существенные признаки, характеризующие классы объектов в целом.

При сопоставлении наборов признаков следует отдавать предпочтение тем из них, которые, по мнению

исследователя, в большей степени полезны для характеристики исследуемого множества объектов. Например, для прогнозирования тенденций в экономике более полезны показатели прироста, а не объема производства.

Принимая во внимание двойственное назначение понятий, определяющих классы объектов – для классификации и для генерации моделей конкретных объектов, необходимо стремиться включить в множества признаков как разделительные так и объединительные признаки.

В результате многолетнего применения метода формирования понятий на основе пирамидальных сетей возникла методика человеко-компьютерного формирования множества признаков. В основе методики лежит понимание процесса обнаружения знаний как итерационного процесса, при каждом повторении которого совершенствуется как выделенное знание, так и пространство признаков, характеризующих объекты. Если выделенное знание (в данном случае – понятие) по какой-либо причине оказывается некачественным (например, неэффективным при использовании в целях классификации, диагностики или прогнозирования) исследователь повторяет процесс, вводя некоторые изменения в множество признаков, определяющих объекты. При этом важно, чтобы форма представления знаний на выходе системы была бы удобной для формирования решений относительно необходимых корректировок множества признаков. Таким образом, в данной методике реализуется обратная связь с выхода системы к обучающей выборке, подаваемой на ее вход.

В системах, реализующих методы формирования понятий в пирамидальных сетях, сформированное понятие представлено логическим выражением, в котором переменными служат обозначения признаков, входящих в описания объектов обучающей выборки. Для каждой конъюнкции признаков, входящей в логическое выражение, указывается число ее включений в конъюнкции признаков, описывающие объек-

ты обучающей выборки. Однотипность представления информации на входе и выходе системы упрощает сопоставление выделенного знания с исходными описаниями, в результате чего возникают идеи по совершенствованию используемого множества признаков. Обычно изменение множества признаков сводится к следующим операциям: исключение признаков, встречающихся в понятиях как положительных, так и отрицательных объектов (при этом нельзя исключать существенные объединительные признаки, свойственные многим объектам обучающей выборки), объединение сильно коррелирующих признаков, добавление новых признаков.

Правильный выбор пространства признаков позволяет сформировать понятия простейшей логической структуры. Понятия, вошедшие в широкое употребление, как правило, конъюнктивны. Более сложная логическая структура характерна для понятий, формируемых в процессе научно-исследовательской деятельности. В этом случае причиной логической сложности обычно являются следующие обстоятельства:

- неправильно выбрано пространство признаков;
- обучающая выборка неполно отражает специфику объема понятия;
- объем формируемого понятия в соответствии с целями исследования искусственно образован из объектов, значительно отличающихся друг от друга.

Дискретизация непрерывных величин. В методах обнаружения знаний обычно обрабатываются признаковые описания объектов, в которых каждый признак имеет конечное множество значений. В связи с этим на этапе формирования признаков описаний диапазоны непрерывного изменения численных значений признаков разделяются на непересекающиеся интервалы, каждому из которых сопоставляется одно дискретное значение признака. Операцию дискретизации целесообразно выполнять на основе сопоставления распределений объектов обучающей выборки, принадлежащих разным классам, по шкале численных значений признаков. В качестве от-

дельных значений признаков выделяют интервалы с наибольшей плотностью распределения объектов, а также интервалы, содержащие разделительные значения признаков объектов одного класса. Более подробно этот метод описан в разделе 2.4.

Обработка неопределенностей обучающей выборки. В описаниях объектов обучающей выборки встречаются неопределенности трех видов:

1) отсутствие информации о значении признака объекта;

2) недостоверность информации о признаке объекта;

3) совпадение описаний объектов, принадлежащих разным классам.

Один из подходов к обработке неопределенностей обучающей выборки состоит в использовании априорных вероятностных оценок степени достоверности информации о принадлежности признака объекту. Главная трудность при реализации подхода связана с определением исходных вероятностей. На практике исходные вероятности часто могут быть заданы только на основе волюнтаристских субъективных представлений. Поэтому в применениях системы CONFOR лучше опускать малодостоверные значения признаков, полагая, что неполнота описаний отдельных объектов в достаточно представительной обучающей выборке не должна оказать существенного влияния на качество выделяемого знания.

Причиной совпадения описаний объектов разных классов может быть несовершенство множества признаков или принципиальная неразделимость формируемых понятий, если они являются нечеткими. В частности, совпадение описаний объектов разных классов может быть результатом чрезмерного обобщения признаков, характеризующих объекты. Если совпадение описаний не удастся устранить путем изменения множества признаков, следует исключить из обучающей выборки совпадающие описания или использовать недетерминистский метод формирования понятий в пирамидальных сетях.

Неоднозначность представления выделяемых

закономерностей. Метод формирования понятий в пирамидальных сетях относится к классу методов, которые позволяют «дотраивать» выделенные закономерности по мере ввода новых объектов. При использовании таких методов формируемое логическое выражение зависит от порядка просмотра обучающей выборки. В методах этого типа важной целью является сохранение качества выделяемого знания при изменении порядка просмотра обучающей выборки. Отличие логических выражений не имеет значения, если они одинаково эффективны при их дальнейшем использовании для решения задач классификации, диагностики и прогнозирования. Метод формирования понятий в пирамидальных сетях обеспечивает высокую степень сохранности качества выделяемого знания при изменении порядка просмотра обучающей выборки. На каждом шаге формирования понятий осуществляется минимальная коррекция уже сформированного понятия, обеспечивающая правильность классификации объекта.

Эффект сохранения качества формируемого понятия при изменении порядка просмотра обучающей выборки является следствием адаптации коррекционных воздействий к ситуациям, возникающим на промежуточных этапах обработки обучающей выборки.

Оценка качества выделенного знания. При решении прикладных задач возникает естественный вопрос: до какой степени обучена система, в какой мере можно доверять ее ответам при распознавании после обучения. Оценить степень обученности системы можно путем экзамена на контрольной группе объектов.

Каждый раз при обучении системы в построении понятия участвует лишь часть объектов, введенных в систему. Некоторые из введенных объектов оказываются «лишними», так как при их восприятии в сети не возникают новые контрольные элементы, т.е. не осуществляется корректировка формируемого понятия. Это свойство системы используется для оценки степени ее обученности в случаях, когда выделение контрольной группы объектов для экза-

мена невозможно. Обучение системы выполняется несколько раз на различных по объему частях обучающей выборки, находящейся в распоряжении исследователя. По результатам счетов строится график зависимости числа объектов, вызвавших корректировку формируемого понятия, от размера обучающей выборки. При введении в обучающую выборку новых объектов приращение числа объектов, вызывающих корректировку понятия, происходит за счет тех объектов из числа вновь введенных, которые неправильно распознаются с помощью понятия, сформированного до их ввода. Тогда для расчета вероятности правильного ответа может быть использована формула:

$$P_a = \frac{N_a - \Delta_a}{N_a},$$

где P_a – вероятность правильного распознавания в точке a графика на рис. 2.6.; Δ_a – прирост числа объектов, вызывающих корректировку понятия, при введении в обучающую выборку N_a новых объектов. Замедление роста числа объектов, участвующих в формировании понятия, при увеличении обучающей выборки свидетельствует об уменьшении числа неправильно распознаваемых объектов, т.е. о росте степени обученности системы.

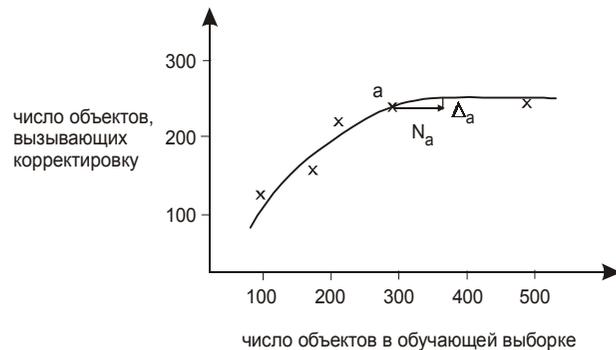


Рис. 2.6. Расчет вероятности правильной классификации объектов.

2.3. Методы, содержащие вывод по аналогии

Сеть «объект-свойство»

Принцип вывода по аналогии [Hall, 1989, Гладун, 1997] можно описать следующим образом. Пусть Q_A и Q_B – множество фактов, характеризующих соответственно объекты A и B , причем некоторые из фактов множества Q_B неизвестны.

Пусть известно о существовании соответствия (анalogии) $\varphi: Q'_A \rightarrow Q'_B$, где $Q'_A \subset Q_A$ и $Q'_B \subset Q_B$. Тогда для некоторого $a \in Q'_A \setminus Q_A$ может быть выдвинута гипотеза о существовании факта $b \in Q_B \setminus Q'_B$ такого, что $a \varphi b$.

Объект B обычно называется целью, объект A – базисом или аналогом. При выводе по аналогии выдвигается гипотеза о существовании нового, ранее неизвестного факта, характеризующего цель, на основании существования аналогичного факта среди фактов, характеризующих базис. Иногда выдвигенные гипотезы по аналогии называют переносом фактов (свойств или отношений) от базиса к цели.

При использовании аналогии естественно возникает вопрос о степени сходства. Для оценки степени аналогии нужна классификация видов аналогии и формализованные правила установления аналогии.

В основу классификации видов аналогии положим представление о мире в структурном плане как о системе объектов, свойств и отношений. Слово «объект» в философской литературе является синонимом слов «предмет», «вещь» и всегда обозначает нечто качественно обособленное. Объект тоже можно рассматривать как систему, элементами которой являются свойства и другие объекты. Элементы объекта связываются между собой отношениями. Объекты в результате взаимных воздействий подвергаются преобразованиям и превращениям. При изучении аналогии важное значение имеет также понятие «ситуация», обозначающее композицию взаимосвязанных объектов, свойств и отношений.

Среди свойств объекта будем выделять собственные свойства, т.е. те свойства, которые всегда присущи объекту независимо от ситуаций или процессов, в которых он участвует. Это, прежде всего, физические, химические и биологические свойства. К числу собственных свойств не относятся ситуативные связи объектов с другими объектами, а также роли (функции) объектов в различных процессах. Объект будем считать элементарным, если он рассматривается как единое целое, без учета его внутренней структуры. Введем несколько обозначений:

$a(b)$ – объект (свойство) a входит в состав (является собственным свойством) объекта b ;

$a \equiv b-a$ тождественно b ;

$a \approx b-a$ аналогично b ;

$a r b$ – объекты a и b связаны отношением r ;

$S_i(a)$ – внутренняя связь i объекта a (связь i между объектами, входящими в состав объекта a);

$S'_i(a)$ – внешняя связь i объекта a (связь i объекта a с другим объектом);

$a \rightarrow b$ – процесс (действие), преобразующий a в b ,

$F_i(a)$ – функция i объекта a .

Правила установления аналогии сформулируем следующим образом.

1. Элементарные объекты (свойства) a и b аналогичны, если $a \equiv b$ или существует класс объектов (свойств) D такой, что $a \in D, b \in D$.

2. Отношения r_i, r_j аналогичны, если $r_i \equiv r_j$ или существует класс отношений R такой, что $r_i \in R, r_j \in R$.

3. Связи $a r_i b$ и $c r_j d$ аналогичны, если $r_i \equiv r_j$ и $a \approx c$ или $b \approx d$.

4. Процессы $a \rightarrow b$ и $c \rightarrow d$ аналогичны, если $a \approx c$ или $b \approx d$.

Используя введенные правила, определим основные виды аналогии.

1. Между объектами a и b существует *атрибутивная* аналогия, если $\exists c(a)\exists d(b) (c(a) \approx d(b))$.

2. Между объектами a и b существует *структурная* (или релятивная) аналогия, если $\exists S_i(a)\exists S_j(b) (S_i(a) \approx S_j(b))$.

Из приведенных определений и правила 3 следует, что при наличии структурной аналогии всегда имеет место атрибутивная аналогия.

3. Между объектами a и b существует *контекстная* (или внешняя) аналогия, если $\exists S'_i(a)\exists S'_j(b) (S'_i(a) \approx S'_j(b))$.

4. Между объектами a и b существует *функциональная* аналогия, если: $\exists F_i(a)\exists F_j(b) (F_i(a) \approx F_j(b))$.

Между ситуациями может существовать атрибутивная, структурная или контекстная аналогии. Эти виды аналогии для ситуаций определяются так же, как и для объектов. В конкретных приложениях может возникнуть необходимость усилить ограничения, используемые в приведенных выше определениях, и таким образом детализировать приведенную классификацию.

Степень сходства объектов и ситуаций можно оценивать на основе представления их в виде множеств свойств, связей и объектов. Оценка степени сходства зависит от числа и значимости аналогичных элементов множеств, представляющих сравниваемые объекты или ситуации. Введем следующие обозначения:

η – степень аналогии сравниваемых объектов (ситуаций);

U_{ab} – множество аналогичных элементов в каждом из сравниваемых объектов (ситуаций);

U_a – множество элементов объекта (ситуации) a , не имеющих аналогов в представлении объекта (ситуации), который сравнивается с a ;

p_i^a – оценка значимости элемента i для объекта (ситуации) a .

Степень аналогии объектов (ситуаций) a и b может быть определена с помощью следующей формулы

$$\eta = \frac{\sum_{u \in U_{ab}} (p_u^a + p_u^b)}{\sum_{u \in U_a} p_u^a + \sum_{u \in U_b} p_u^b}.$$

Если значимости всех элементов сравниваемых объектов (ситуаций) равны 1,

$$\eta = \frac{2 \times |U_{ab}|}{|\bar{U}_a| + |\bar{U}_b|}.$$

Достоверность гипотезы, выведенной по аналогии, тем больше, чем выше степень аналогии между базисом и целью. В случае отсутствия аналогичных элементов у сравниваемых объектов (ситуаций) $\eta=0$. Если все элементы сравниваемых объектов (ситуаций) имеют аналоги, $\eta=\infty$.

В качестве критериев классификации выводов по аналогии используют характер факта, переносимого от базиса к цели. В зависимости от того, является ли переносимый факт свойством или отношением, различают вывод свойств по аналогии и вывод отношений по аналогии. Переносимым свойством может быть свойство существования. Вывод по аналогии, в результате которого устанавливается аналогия причин базиса и цели, называется выводом причин по аналогии, или каузальной аналогией.

Обязательной частью всех процессов, связанных с аналогией, является поиск аналогичных объектов, причем от этой операции в наибольшей степени зависит время реализации всего процесса. Поэтому одним из главных показателей качества поиска по аналогии является время его выполнения, которое в основном зависит от объемов информации, просматриваемой во время поиска. Принципиальное решение проблемы сокращения времени поиска состоит в перенесении действий по установлению подобия объектов в операции ввода информации в память. При этом общий выигрыш в эффективности достигается за счет более редкого применения операций ввода по сравнению с поисковыми операциями. Таким образом, проблемы поиска по аналогии необходимо решать в комплексе с вопросами организации памяти, обеспечивающей установление подобия объектов и выделения классов подобия на этапе занесения информации. В свете современных представлений это должна быть память коннекционистского типа, до-

пускающая параллельное выполнение поисковых операций и отражающая в своей структуре семантические пересечения блоков информации. Такого рода требованиям удовлетворяют пирамидальные сети. Алгоритм построения сети, работающий при вводе описаний новых объектов, является одновременно алгоритмом поиска в сети аналогов нового объекта, имеющих с ним общие фрагменты описаний.

Приведем еще один тип сетевой структуры, специально ориентированной на вывод по аналогии. Сеть «объект-свойство» представляет собой ориентированный ациклический граф, в котором вершины соответствуют «объектам» и «свойствам». Имеется два типа объектов: первичные и составные – соединения.

Дуги соответствуют отношениям: «включает» и «имеет свойство». Дуги «включает» соединяют вершины, соответствующие объектам, и всегда направлены от вершин, соответствующих соединениям, к вершинам, соответствующих первичным объектам. Дуги «имеет свойство» соединяют вершины, представляющие объекты, с вершинами, представляющими свойства, и направлены от первичных объектов к свойствам и от свойств к соединениям.

Сеть «объект-свойство» удобно изображать четырехслойным графом (рис. 2.7), в котором отдельные слои образуют вершины, представляющие соответственно свойства первичных объектов, первичные объекты, соединения, свойства составных объектов. Эта структура может использоваться отдельно, или как локальный фрагмент более общей сети понятий. Она является удобным инструментом вывода гипотез о свойствах и составе объектов на основе аналогии.

Объекты, задающие базис и цель для вывода по аналогии, могут быть заданы непосредственно именами соответствующих вершин сети или путем указания характеризующих их свойств. Отображение, связывающее базис с целью, устанавливается в пространстве свойств первичных объектов. Составные или первичные объекты, группы объектов считаются

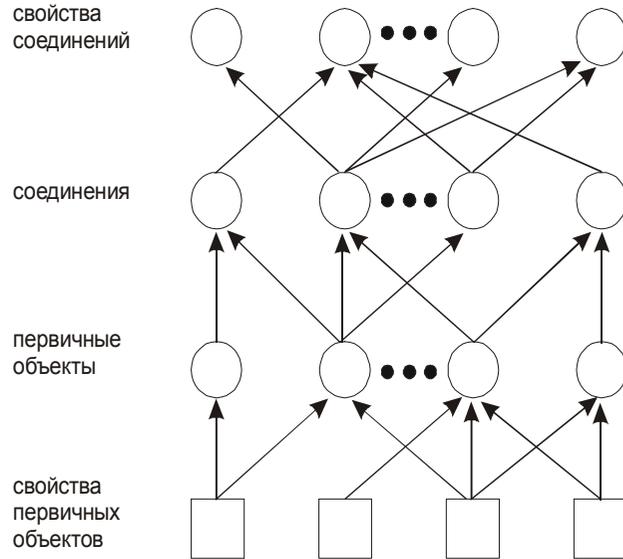


Рис. 2.7. Сеть «объект-свойство».

аналогичными, если они характеризуются общим набором свойств первичных объектов. На основе аналогии объектам-целям присваиваются свойства объектов, служащих базисом.

Все поисковые процессы, связанные с выводом по аналогии, реализуются в сети с помощью двух поисковых операций – просмотров путей от заданной вершины в направлении или против стрелок.

При решении различных задач часто приходится вводить дополнительные ограничения на выбор базиса и цели, обусловленные, например, спецификой типов или классов объектов, структурой соединений, возможной несовместимостью первичных объектов при формировании соединений и т.п.

Степень аналогии зависит от числа и значимости общих свойств, связывающих базис и цель. Значимость свойства оценивается весовыми коэффициентами, которые формируются при выполнении поисковых операций, и присваиваются вершинам сети.

Проектирование состава составных объектов.

К числу наиболее характерных задач, решаемых на сети «объект-свойство», относится задача проектирования состава составных объектов – задача комплектации [Гладун, 2000].

Введем обозначения:

A – множество первичных объектов;

$B = \{b \mid b \subset A, |b| > 1\}$ – семейство множеств, представляющих составные объекты (соединения);

C – множество свойств первичных объектов;

D – множество свойств соединений.

Известны отображения множеств A и B в множества C и D .

$$\forall a \in A (a \rightarrow C_a \mid C_a \subset C)$$

$$\forall b \in B (b \rightarrow D_b \mid D_b \subset D)$$

Пусть заданы $D^+ \subset D$ и $D^- \subset D$, $D^+ \cap D^- = \emptyset$, а также ограничение R , определяющее допустимую структуру соединений.

Необходимо сформировать новые допустимые соединения, имеющие все свойства из множества D^+ и ни одного свойства из множества D^- .

При решении задачи комплектации подбор первичных объектов, входящих в состав соединения, осуществляется на основе знания зависимостей между заданными свойствами соединения и свойствами первичных объектов. Эти зависимости могут быть простыми или обусловленными многими факторами.

Например, при проектировании состава вычислительных комплексов, свойства соединения (вычислительного комплекса), такие как частота, разрядность, объем памяти либо совпадают со свойствами составляющих (процессоров, блоков памяти, мониторов и т.п.), либо являются их функцией (объем внешней памяти является суммой емкостей дисков). В этом случае задача комплектации является вполне детерминированной.

При проектировании состава химических соединений зависимости между заданными свойствами соединения и свойствами химических элементов обусловлены большим количеством факторов. Результат решения представляет собой гипотезу о составе соединения, который мог бы обеспечить заданные свойства соединения. В этом случае задача недетерминирована и по сути является задачей прогнозирования состава соединения.

Проектирование состава составных объектов с заданными свойствами состоит из следующих этапов:

1. Выделение в сети вершин верхнего слоя, соответствующих требуемым свойствам соединения (множество D^+), а также тем свойствам, которые не должно иметь проектируемые соединения (множество D^-).
2. Выделение в сети вершин, соответствующих известным соединениям, которые удовлетворяют заданным требованиям.
3. Выделение релевантных свойств, т.е. свойств таких первичных объектов, которые входят в состав известных соединений, удовлетворяющих заданным требованиям.
4. Выделение первичных объектов, аналогичным тем первичным объектам, которые входят в состав соединений, выделенных на этапе 2.
5. Формирование новых соединений из первичных объектов, выделенных на 4-ом этапе, в соответствии с заданными ограничениями на структуру соединений.

Рассмотрим более детально отдельные этапы метапроцедуры комплектации составных объектов.

Вершины, представляющие те известные соединения, которые удовлетворяют заданным требованиям, отыскиваются в третьем снизу слое путем прослеживания заходящих дуг вершин верхнего слоя, представляющих свойства из множеств D^+ и D^- .

При этом в конкретных задачах можно допускать неполное соответствие соединений заданным требованиям. Например, можно выделять лишь соединения, имеющие свойства множества D^+ , не обращая внимания на отсутствие свойств множества D^- .

На третьем этапе просматриваются сверху вниз (против стрелок) пути, заходящие в вершины, выделенные на первом этапе, и проходящие через вершины, выделенные на втором этапе. Для каждого найденного свойства, представленного вершиной c самого нижнего слоя, записывается строка:

$$(n_1^c, n_2^c, \dots, n_{\overline{I}}^c, \dots, n_J^c); (m_1^c, m_2^c, \dots, m_j^c, \dots, m_J^c),$$

где n_i^c ($i=\overline{1, I}$) – количество прослеженных путей, соединяющих вершину c с вершиной, представляющей i -ое свойство из множества D^+ (вес вершины c по i -ому заданному свойству);

n_j^c ($j=\overline{1, J}$) – количество прослеженных путей, соединяющих вершину c с вершиной, представляющей j -ое свойство из множества D^- (вес вершины c по j -ому запрещенному свойству);

I – мощность множества D^+ ; J – мощность множества D^- .

Могут быть учтены различного рода ограничения, уменьшающие число релевантных свойств. Например, могут быть установлены пороговые значения для n_i^c ($i=\overline{1, I}$) и n_j^c ($j=\overline{1, J}$). В число релевантных свойств вносятся лишь те найденные свойства первичных объектов, у которых вес хотя бы по одному заданному свойству соединений превышает соответствующий порог.

На четвертом этапе выделяются первичные объекты, представленные теми вершинами второго слоя, которые имеют связи с вершинами, представляющими релевантные свойства. При просмотре сети снизу вверх от вершин, соответствующих релевантным свойствам, для каждого найденного первичного объекта записывается строка:

$$(N_1^b, N_2^b, \dots, N_i^b, \dots, N_I^b); (M_1^b, M_2^b, \dots, M_j^b, \dots, M_J^b),$$

где b – имя вершины, соответствующей объекту;

$$N_i^b = \sum_{c \in G_b} n_i^c, (i = \overline{1, I}); M_j^b = \sum_{c \in G_b} m_j^c, (j = \overline{1, J});$$

G_b – множество релевантных свойств, присутствующих

первичному объекту b , т.е. множество заходящих дуг вершины b , которые исходят от вершин, соответствующих релевантным свойствам.

Список первичных объектов-кандидатов, найденных на этапе 4, упорядочивается. Наиболее приоритетными для формирования новых соединений считаются объекты, имеющие

$$\min \sum_{j=1}^J M_j^b \text{ и } \max \sum_{i=1}^I N_i^b.$$

Таким образом, для формирования новых соединений отбираются первичные объекты, аналогичные тем объектам, которые входят в состав известных соединений заданного типа. Критерием существования аналогии является наличие общих релевантных свойств; степень аналогии оценивается их числом.

По упорядоченному списку объектов-кандидатов на пятом этапе формируются с учетом заданного структурного ограничения R различные варианты соединений, которые предположительно могут отвечать заданным исходным требованиям, т.е. генерируются гипотезы о составе соединений, которые бы могли иметь свойства множества D^+ и не иметь свойств множества D^- .

Модификации описанной метапроцедуры отличаются характером ограничений, с помощью которых отбираются известные соединения (этап 2), релевантные свойства и объекты-кандидаты, а также средствами учета совместимости первичных объектов при формировании соединений.

Методы проектирования состава составных объектов на основе сетей «объект-свойство» реализованы в программной системе ANALOG. Система включает набор программ, выполняющих различные поисковые операции, операции вывода по аналогии и оценивания степени аналогии.

Система предназначена для решения научно-технических, экономических и социальных задач, содержащих вывод по аналогии. Области применения

комплекса являются химия, материаловедение, рыночная экономика, медицина, экология, геология и т.п.

Результаты применения системы ANALOG в целях проектирования состава химических соединений сравнивались с результатами такого же применения системы CONFOR, в которой используются методы индуктивного вывода.

При использовании индуктивных методов гипотеза о составе соединения формируется путем обобщения информации об известных соединениях, имеющих полный набор заданных свойств. Наличие такой информации (обучающей выборки) является принципиальным фактором во всех процессах индуктивного вывода. Достоверность выведенной гипотезы зависит от качества обучения, т.е. от того, насколько полно объекты обучающей выборки отражают существенные свойства и закономерности исследуемого класса соединений. Необходимость иметь представительную обучающую выборку является серьезной трудностью в практическом применении индуктивных методов в целях проектирования состава соединений. Использование при проектировании состава вывода по аналогии позволяет значительно смягчить требования к исходной информации. В качестве базы для вывода по аналогии могут привлекаться известные соединения, лишь частично удовлетворяющие заданным требованиям. В результате расширяется круг генерируемых гипотез, возникает принципиальная возможность генерации гипотез о составе даже при отсутствии известных соединений, полностью удовлетворяющих заданным требованиям.

2.4. Дискретный анализ распределений объектов по шкалам признаков.

В исследовательских процессах часто возникает необходимость выделять на шкалах признаках интервалы, которые являются наиболее характерными

для отдельных классов объектов или общими для нескольких классов. Задача решается путем сравнительного анализа распределений объектов разных классов на шкалах признаков. Таким образом осуществляется, например, сравнение технических и рыночных характеристик изделий, выпускаемых разными фирмами; сопоставление физико-химических характеристик различных материалов; сравнение числовых показателей, характеризующих предприятия или регионы в разные периоды времени и т.п.

Дискретный анализ распределений объектов на шкалах признаков, помимо самостоятельного значения, играет важную роль как процесс преобразования информации в различных задачах атрибутивного анализа. Дело в том, что для повышения эффективности рассмотренных выше логических метапроцедур атрибутивного анализа следует упрощать пространство признаков, используемых для описания объектов. Диапазоны изменения численных значений признаков должны быть разделены на непересекающиеся интервалы, которым присваиваются индивидуальные имена. Таким образом непрерывные шкалы числовых признаков трансформируются в дискретные, а сами числовые признаки становятся номинальными. В соответствии с распространенной практикой дискретизация непрерывных шкал часто выполняется путем деления их на равные интервалы. «Слепая» дискретизация значительно снижает результаты классификации, диагностики или прогнозирования. Интервалы, выделенные в результате дискретизации, должны в наибольшей степени характеризовать исследуемые классы объектов.

Сформируем требования к методу дискретизации.

1. Номинальные признаки, сформированные в результате дискретизации шкал, должны обеспечивать разделимость классов в пространстве признаков.

2. В пространстве признаков, сформированном в результате дискретизации, области распределений объектов разных классов должны быть максимально компактными.

3. Разделимость и компактность распределений классов достигаются, если метод дискретизации выделяет наиболее характерные интервалы в распределениях объектов сравниваемых классов на шкалах числовых признаков. Наиболее характерными являются интервалы, в которых преобладают значения признаков некоторого класса. Метод дискретизации должен обеспечивать выделение именно таких интервалов.

4. При соблюдении требований разделимости и компактности, а также требования 3 метод дискретизации должен выделять как можно более крупные интервалы. В результате уменьшается число значений номинальных признаков и упрощаются последующие процессы обработки.

5. Метод дискретизации должен обеспечивать заданную точность определения границ выделяемых интервалов.

6. Пользователь должен иметь возможность расширять выделенные интервалы за счет соседних пустых интервалов. Величина сдвига границы выделенного интервала задается пользователем на основе его понимания проблемы, для решения которой выполняется дискретизация. Обобщение выделенных интервалов иногда приводит к улучшению результатов последующих процессов обнаружения закономерностей, классификации, диагностики или прогнозирования.

В процессах дискретного анализа используются интервалы следующих типов:

1. Пустые, не содержащие значений признаков объектов из исследуемых классов;

2. Однородные, содержащие значения признаков объектов только одного класса;

3. Равномерные, в которых имеются значения признаков разных классов и разность между количествами значений признаков, принадлежащих объектам разных классов, не превышает заданный порог;

4. Неравномерные, в которых имеются значения признаков объектов разных классов, но превагируют

ют значения признаков объектов некоторого одного класса.

Алгоритм дискретного анализа классов объектов по признаку D_i состоит в последовательном выполнении следующих операций:

1. Формирование шкалы значений признака D_i .

1.1. Определение границ шкалы. Операция состоит в нахождении наибольшего (d_i^{max}) и наименьшего (d_i^{min}) значений признака D_i на множестве всех анализируемых объектов.

1.2. Определение размера δ начального интервала

$$\delta = \frac{(d_i^{max} - d_i^{min})}{Z},$$

где Z – количество начальных интервалов. От выбора Z зависит точность определения границ интервалов, время работы алгоритма, удобство визуализации шкалы. С учетом этих факторов обычно $Z=100$.

1.3. Определение границ начальных интервалов.

Левая и правая граница начальных интервалов вычисляются следующим образом:

$$b_1^l = d_i^{min}; \quad b_1^r = d_i^{min} + \delta;$$

$$b_m^l = b_{m-1}^r; \quad b_m^r = b_m^l + \delta; \quad m=2, \dots, Z$$

2. Формирование распределений объектов исследуемых классов на шкале признака D_i .

Для каждого из начальных интервалов подсчитывается число объектов каждого из классов, значения признаков которых попадают в этот интервал.

Значение d_i признака D_i принадлежит интервалу b_m ($m=1, Z$), если ($b_m^l \leq d_i < b_m^r$). Для последнего интервала ($b_Z^l \leq d_i \leq b_Z^r$).

3. Обобщение интервалов.

Непустые интервалы шкалы, граничащие с пустыми интервалами, просматриваются, начиная от левой границы шкалы, и расширяются в обе стороны путем присоединения к ним нескольких соседних

пустых начальных интервалов. Максимальное число пустых начальных интервалов, которое может быть присоединено к непустому интервалу с каждой стороны, является параметром алгоритма и определяется пользователем.

4. Объединение однородных интервалов.

Соседние однородные интервалы, содержащие значения признаков объектов одного класса, объединяются.

5. Объединение соседних неравномерных интервалов.

6. Объединение соседних равномерных интервалов.

Приведенный алгоритм дискретного анализа нацелен на обеспечение делимости и компактности распределений классов объектов в пространстве признаков. Это достигается путем выделения наиболее характерных интервалов, а именно, интервалов, содержащих значения признаков только или преимущественно одного класса (однородные и неравномерные интервалы).

В результате дискретного анализа выделяются интервалы наибольшего размера, в результате чего уменьшается общее число сформированных номинальных признаков. Пользователь имеет возможность обобщать сформированные интервалы на основе его понимания проблемы, для решения которой производится дискретный анализ.

В отличие от других методов дискретизации шкал [Маркин, 1988] описанный метод формирует интервалы с учетом соотношения распределений исследуемых классов на шкалах признаков.

Метод реализован в программной системе DISCRET, которая применяется в целях предобработки данных для систем CONFOR и ANALOG, а также самостоятельно, для решения различных задач дискретного анализа.

Глава 3. ПРОЦЕССЫ ВЫБОРА

1.1. Формирование системы знаний целеустремленной системы

Формирование сети понятий

Выбор целей, действий и актантов действий производится на основе системы знаний целеустремленной системы. Знания разных типов (рис. 1.2) объединены в цельную иерархическую неоднородную структуру – сеть понятий.

Сеть понятий – саморазвивающаяся структура. Рассмотрим процессы формирования иерархии связанных понятий [Гладун, 1994].

Понятие a называется видовым по отношению к понятию b , если объемы этих понятий связаны соотношением $V_a \subset V_b$ и не существует понятия c такого, что $V_a \subset V_c \subset V_b$. Понятие, по отношению к которому имеются видовые понятия, называется *родовым*. Множество непосредственных связей понятия с другими понятиями назовем его *видовым контекстом*.

Введем следующие обозначения:

K_a – видовой контекст понятия a ; L_a – дочернее множество понятия a , т.е. множество видовых понятий понятия a . Сеть понятий развивается по следующим правилам.

Правило 1

Если для понятий a и b выполняются условия $K_a \cap K_b \neq \emptyset$, $K_a \not\subset K_b$, $K_b \not\subset K_a$, то формируется новое понятие d с $K_d = K_a \cap K_b$ и $L_d = \{a, b\}$. Для понятий a и b формируются новые видовые контексты $K_a = K_a \setminus K_d$ и $K_b = K_b \setminus K_d$.

Правило 1 является основной операцией формирования родовых понятий при возникновении пересечений видовых контекстов.

Правило 2

Если для понятий a и b выполняется условие $K_a \subset K_b$, то формируется новое дочернее множество понятия a $L_a = L_a \cup b$ и новый видовой контекст понятия b $K_b = K_b \setminus K_a$.

Правило 2 вводит новый вид в дочернее множество ранее сформированного родового понятия.

Правило 3

Если для родовых понятий a и b выполняется условие $L_a \subset L_b$, то для понятия b формируется новое дочернее множество $L_b = (L_b \setminus L_a) \cup a$.

При выполнении правила 3 родовые понятия включаются в дочерние множества других родовых понятий, вследствие чего образуется иерархия родовых понятий.

Правило 4

Если для родовых понятий a и b $L_a = L_b$, понятия a и b ликвидируются и формируется новое родовое понятие d с $K_d = K_a \cup K_b$ и $L_d = L_a = L_b$.

Правило 4 объединяет родовые понятия, имеющие общее дочернее множество.

Правило 5

Если для родовых понятий a и b $|L_a \cap L_b| > 1$, $L_a \not\subset L_b$, $L_b \not\subset L_a$, то формируется новое родовое понятие d с $K_d = \emptyset$ и $L_d = L_a \cap L_b$. Для понятий a и b формируются новые дочерние множества $L_a = (L_a \setminus L_d) \cup d$ и $L_b = (L_b \setminus L_d) \cup d$.

Правило 5 формирует родовые понятия, которые соответствуют пересечениям дочерних множеств родовых понятий, сформированных ранее.

На рис. 3.1 показаны преобразования в сети понятий, соответствующие приведенным выше правилам. Пунктирными линиями показаны связи родо-

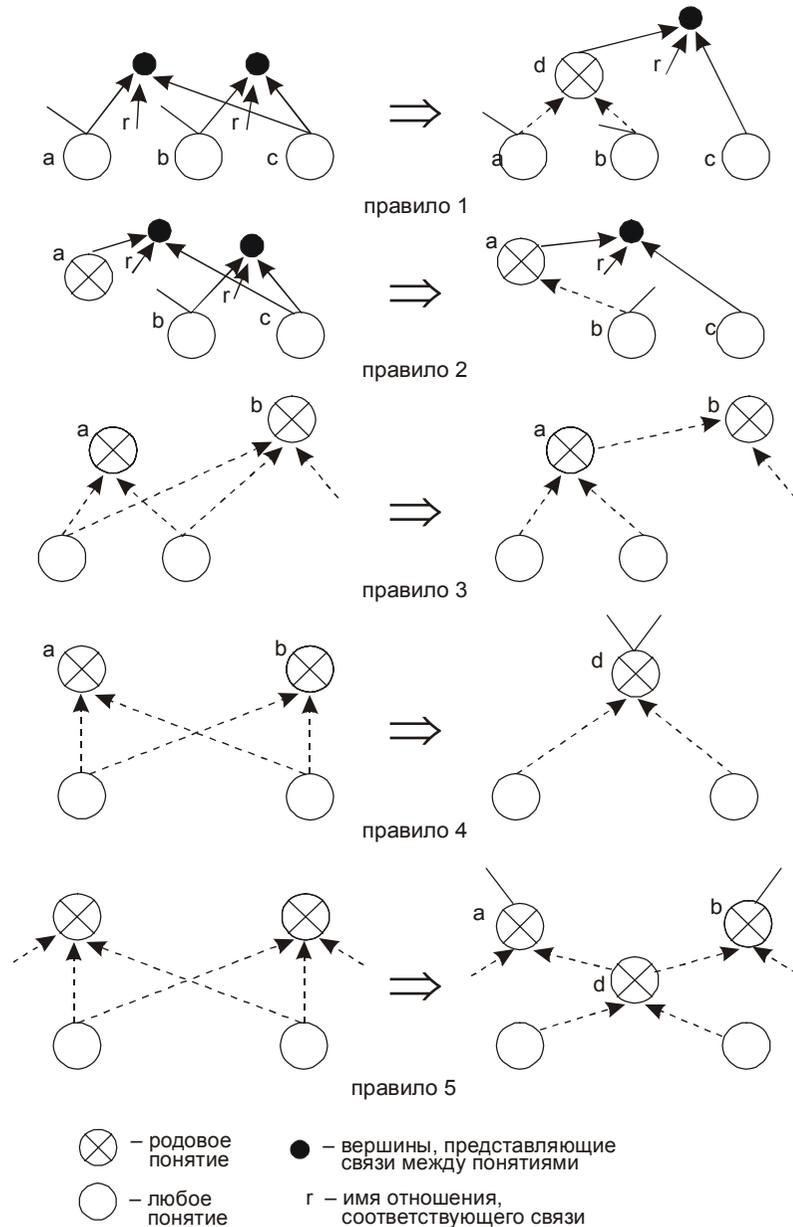


Рис. 3.1. Преобразования в сети понятий

вых понятий с их дочерними множествами. Этим связям соответствует отношение «элемент класса».

Информация о новых понятиях и связях может поступать по интерфейсным каналам системы. Приведенные правила обеспечивают интеграцию новых фрагментов в основную сеть понятий.

При формировании родовых понятий первоначальные видовые контексты понятий распределяются между родовыми понятиями, с которыми они связаны.

Объединение видового контекста понятия и видовых контекстов всех его родовых понятий называется полным контекстом этого понятия. Полный контекст отражает содержание понятия.

Структура, возникающая в результате применения приведенных правил, включает сетевые структуры, рассмотренные ранее, – пирамидальные сети и сети «объект-свойство». Эти структуры в ситуациях индуктивного вывода или вывода по аналогии могут быть созданы другими методами, например, теми, которые были описаны в предыдущих разделах. В случае необходимости построенные отдельно пирамидальные сети или сети «объект-свойство» интегрируются в общую сеть понятий.

В сети понятий выделим три вида моделей: понятийную модель проблемы, понятийную модель объектной области и категориальную модель мира.

Понятийной моделью проблемы называется граф $M_p = \langle W_p, L_p \rangle$, где W_p – множество вершин, соответствующих понятиям, которые имеют отношения к проблеме, и L_p – множество дуг, представляющих связи между понятиями.

Понятийной моделью предметной области называется граф $M_d = \langle W_d, L_d \rangle$, где W_d – множество вершин, соответствующих понятиям объектной области, к которой относится проблема, и L_d – множество дуг, представляющих связи между понятиями. В общем случае из-за неполноты или неточности моделирования предметной области граф M_p может не быть частью графа M_d .

Категориальной моделью мира называется граф $M_k = \langle W_k, L_k \rangle$, где W_k – множество вершин, соответствующих наиболее общим понятиям – категориям, и L_k – множество дуг, представляющих связи между категориями.

Во всех трех моделях используется одно и то же множество обобщенных отношений, содержащие отношения, которые обычно используются в рассуждениях при решении проблем. Примеры категорий и отношений приведены в разделе 1.2.

В модели проблемы значительную часть понятий составляют имена конкретных объектов. В моделях более высоких уровней используются более общие понятия. Понятия, образующие модели более низких уровней, соединены связями «элемент класса» с понятиями моделей более высоких уровней. По этим связям при решении проблем осуществляются переходы от одной модели к другой.

Понятийные модели проблем могут быть кратковременными. Часто они формируются и используются лишь в период решения проблемы. Модели объектных областей и категориальная модель мира развиваются по мере накопления опыта в решении проблем разных типов.

В развитии сети понятий важную роль играют операции переноса категориальных знаний. Перенос категориальных знаний используется для построения моделей более низкого уровня, например, модели проблемы, на основе информации, содержащейся в моделях более высоких уровней.

Каждой категории c соответствует некоторый стандартный, свойственный только ей, набор отношений с другими категориями – R_c . Отношения из этого набора будем называть *валентностями категории*, опираясь на некоторую аналогию в использовании термина «валентность» в лингвистике.

В сети понятий возникновение проблемы проявляется в отклонении от нормативных значений величин, представляющих жизненные потребности системы, или в появлении сетевых фрагментов, пред-

ставляющих, возможно неполно, исходную и целевую ситуации проблемы. Понятия и отношения, отражающие начальные представления о проблеме, вводятся в сеть понятий по интерфейсным каналам системы. Таким образом возникает начальная понятийная модель проблемы.

Построение M_p состоит в формировании гипотез о связях вершин множества W_p . Для каждой вершины должны быть построены связи, соответствующие валентностям родовой категории (т.е. категории, представленной вершиной, к которой ведут связи типа «элемент-класс»). Таким образом осуществляется перенос категориальных знаний о связях в понятийные модели более низких уровней. В результате переноса на вершину $w \in W_p$ валентностей категории c вершина w получает недостроенные связи типа wrx , где $r \in R_c$ и x – неизвестное понятие.

Валентности характеризуются категориями, которые они связывают. Таким образом, перенесенная на вершину w валентность r определяет родовую категорию неизвестного понятия x . Понятием x может быть любое видовое понятие этой категории, входящее в уже построенную часть модели проблемы или в соответствующую понятийную модель предметной области. Предпочтение дается тем понятиям, которые входят в модель проблемы. Так осуществляется перенос категориальных знаний о понятиях в понятийные модели более низких уровней. При наличии нескольких вариантов подстановки понятий на место неизвестного понятия x возникает связи, соединенные логическим отношением «или». Процесс развития модели проблемы продолжается, пока в ней существуют недостроенные связи. В результате формируется понятийная модель проблемы, включающая представления целей, действий, необходимых для достижения целей, а также актантов этих действий. На основе этой модели формируется план решения проблемы (раздел 3.2.)

Мы рассмотрели две группы процессов построения сети понятий: процессы построения верхних уровней

сети по правилам обобщения 1÷5 и процессы построения нижних уровней путем переноса категориальных знаний. Категориальные знания формируются по правилам обобщения, понятийная модель проблемы – путем переноса категориальных знаний. В построении понятийной модели предметной области участвуют обе группы процессов. Понятия и связи любых уровней могут быть введены человеком – партнером компьютерной системы.

Для фрагментов сети понятий, представляющих агрегированные знания (рис. 1.2), строятся объединяющие вершины, которые соединяются специальными структурными связями со всеми вершинами, образующими соответствующие фрагменты.

Концептуальные умозаключения

Сеть понятий обогащается также новыми понятиями и связями, возникающими в результате процессов правдоподобного логического вывода. Рассмотрим процесс рассуждения, приводящий к построению гипотетических моделей. Приведем несколько примеров «человеческих» умозаключений из работ Р. Шенка [Schank, 1980].

- А. Джон держал камень.
Он ранил Билла.
⇒ Джон ранил Билла (этим) камнем.
- В. Джон ранил Мери камнем.
⇒ Джон, вероятно, был зол на Мэри.
- С. Джону нужна книга.
⇒ Джон, вероятно, хочет прочитать ее.

Исследование подобных умозаключений вместе с ситуациями, в которых они реализуются, позволяет выделить следующие особенности.

1. Чаще всего процесс рассуждения возникает в ситуациях, модели которых не содержат в явном

виде полных наборов посылок, достаточных для каких-либо заключений. Тем не менее процесс рассуждения происходит, новые суждения создаются. Ни в одном из приведенных выше умозаключений набор посылок не является достаточным для вывода заключений. Дополнительные посылки, необходимые для вывода заключения, выбираются из базы понятийных знаний, отражающей опыт субъекта процесса рассуждений.

2. В качестве дополнительных знаний, используемых для вывода заключений, иногда привлекаются признаки, входящие в содержание понятий. Рассмотрим, например, умозаключение А. Для вывода заключения привлекаются знания, отраженные в понятиях «ранить» и «камень». В содержание понятия «ранить» входит признак «твердый». Ранить можно чем-то твердым. Этот же признак входит в содержание понятия «камень». На основании пересечения содержаний понятий «ранить» и «камень» по признаку «твердый» с учетом явно заданных посылок выводится заключение «Джон ранил Билла камнем».

Использование знаний, содержащихся в понятиях, является спецификой всех приведенных выше умозаключений. Поэтому предложение Шенка называть такие умозаключения «концептуальными» (concept – понятие) представляется удачным.

3. Концептуальные умозаключения вероятностны, т.е. относятся к классу правдоподобных умозаключений. Можно выделить три причины недостоверности концептуальных умозаключений: 1) игнорирование фактов, не отраженных непосредственно в посылках умозаключения; 2) многовариантность неявных посылок, содержащихся в базе знаний; 3) многовариантность следствий, соответствующих посылкам.

Например, умозаключение А вероятно, поскольку в его посылках игнорируется такой возможный факт, как наличие у Джона помимо камня еще каких-нибудь предметов, например палки. В основе умозаключения В лежит знание, что причиной ране-

ния может быть злость. Но возможны и другие причины (презрение, обида, зависть, ревность, несчастный случай и т.д.) Поэтому заключение «Джон был зол на Мэри» является лишь одним из нескольких возможных вариантов заключений и в силу этого вероятно. В умозаключении С используется знание о том, что книги предназначены для чтения. Но опять-таки это лишь одно из возможных использований книги. Книга может использоваться как подставка, как украшение интерьера, как пресс и т.д. В данном случае причиной ненадежности умозаключения является многовариантность неявной посылки.

Концептуальные умозаключения представляют собой эффективный, отшлифованный природой инструмент правдоподобных рассуждений. Эффективность его состоит в возможности генерировать гипотезы, планы поведения, цепочки следствий в информационно бедных средах и ситуациях. Сильной стороной концептуальных умозаключений является отсутствие жестких требований к полноте и достоверности посылок, многовариантность заключений. Концептуальные умозаключения максимально эвристичны. Эта особенность позволяет отыскивать решения в условиях неопределенности и формировать планы успешного поведения в, казалось бы, безвыходных ситуациях.

Каким требованиям должны удовлетворять компьютерная система и правила, реализующие концептуальные умозаключения?

1. В базе знаний системы должны быть представлены знания о связях между понятиями и знания о признаках, составляющих содержание понятий. Этим требованиям отвечает рассмотренная выше сеть понятий.

2. Умозаключений, основанных на специфике понятий и связей между понятиями, можно привести очень много. Совершенно нереально иметь правила, моделирующие все такие умозаключения. Выход из положения состоит в типизации концептуальных умозаключений, что позволило бы использовать от-

носителю небольшое число обобщенных правил (метаправил), соответствующих типам концептуальных умозаключений. В метаправилах обобщение достигается за счет использования переменных, а также родовых понятий и обобщенных отношений вместо конкретных понятий и отношений.

3. Метаправила должны содержать посылки двух типов: те, которые должны быть заданы явно, и те, которые задаются в знаниях о понятиях.

4. В системе должны быть реализованы процессы распознавания применимости метаправил на основе сопоставления обобщенных посылок метаправил с конкретными фактами.

Прежде чем рассматривать примеры метаправил, приведем некоторые элементы синтаксиса их описаний.

В записи метаправил суждения представлены выражениями типа $(a)r(b)$, где r – имя отношения, a и b – имена понятий. Имя понятия может быть простым или составным. Простое имя – строка любых символов, кроме скобок и тире, или переменная X с индексом. Составное имя состоит из двух простых имен, разделенных тире. Первым стоит имя понятия, которое является родовым по отношению к понятию, имя которого указывается на втором месте. Использование переменной X в качестве имени понятия означает, что имеется в виду любое понятие. Переменная в составном имени означает любое видовое понятие по отношению к другому понятию, входящему в составное имя. В пределах описания одного метаправила каждая переменная всегда обозначает один и тот же денотат.

Левая и правая части правил разделяются стрелкой. Посылки, истинность которых проверяется в сети понятий, обозначаются звездочкой. Символ \cap , связывающий имена понятий, указывает на существование пересечения множеств признаков, составляющих содержание этих понятий.

Приведем примеры метаправил, в первую очередь правила, используемые для моделирования умозаключений, приведенных выше.

1. (X_1) иметь (X_2) ;
 (X_1) агент (действие – X_3);
 *(действие – X_3) инструмент (X_4) ;
 * $(X_4) \cap (X_2)$
 \Rightarrow (действие – X_3) инструмент (X_2) .
2. (X_1) агент (действие – X_2);
 (действие – X_2) адресат (X_3) ;
 * $((X_1)$ отношение – X_4 (X_3)) является причиной
 (действие – X_2)
 $\Rightarrow(X_1)$ отношение – $X_4(X_3)$
3. (X_1) нуждаться в (X_2) ;
 *(действие – X_3) объект (X_2) ;
 $\Rightarrow(X_1)$ иметь целью (действие – X_3);
 (действие X_3) объект (X_2) .
4. (процесс – X_1) реализуется в (система – X_2);
 * (X_3) средство (процесс – X_1)
 \Rightarrow (система – X_2) использует (X_3) .

Родовые понятия «действие – X », «отношение – X » означают соответственно любое действие, любое отношение. Отношение «иметь» является обобщением отношения «держат». Используются ролевые отношения: «агент» – инициатор действия, «адресат» – лицо, в пользу или во вред которого совершается действие, «объект» – то, что подвергается действию.

Если в понятийной модели предметной области имеются связи, соответствующие суждению: (метод резолюции) средство (логический вывод), правило 4 позволяет получить следующее умозаключение (в естественном языке выразении):

Логический вывод реализован в системе G .
 \Rightarrow Система G использует метод резолюций.

Можно выделить несколько характерных классов метаправил, использующих знания о понятиях. В результате выполнения метаправил формируются гипотезы: об актантах действий на основе знаний о действиях; об условиях, причинах, целях или ре-

зультатах действий на основе знаний о действиях; об условиях, причинах или целях действий на основе знаний о их актантах; об условиях, причинах возникновения свойств или отношений на основе знаний об этих свойствах или отношениях.

Если в каждом из классов можно сформировать до пяти метаправил (этого обычно достаточно для решения практических задач), общее число метаправил такого рода оказывается не больше 20.

3.2. Планирование решений

Цели целеустремленной системы рождаются в сети понятий при отклонении от норм жизненных потребностей системы или в качестве подцелей других, уже существующих целей. Вместе с целью возникает проблема. Если проблема достаточно определена совокупностью связанных понятий, которые представляют исходную ситуацию и цель, уместнее говорить о задаче, т.к. в этом случае может быть начат процесс составления плана действий, направленный на решение проблемы [Гладун, 1987, Коваль, 1999, Рабинович, 1995].

Решение задачи представляет собой процесс преобразования исходной ситуации в целевую. Последовательность действий, преобразующая исходную ситуацию в целевую, есть путь решения задачи. Путь решения – деятельность, обеспечивающая решение конкретной задачи. Процессы решения множества близких по целям и исходным ситуациям задач управляются структурой, называемой обычно планом решения. План решения может определять альтернативные последовательности действий. При решении конкретной задачи по плану решения выбор последовательности действий из числа нескольких альтернативных зависит от промежуточных результатов, полученных в процессе решения. Строгое представление плана решения в виде точно определенных правил называется алгоритмом.

Приведем простую схему целеполагания на основе применения показателей типа «жизненные потребности». Рассмотрим систему, формирующую для клиентов планы действий сообразно их текущим ситуациям и целям. Например, это может быть система, представляющая клиентам юридические консультации. Пусть A – жизненная потребность системы, состоящая в максимальном удовлетворении требований клиентов; Q_A – степень удовлетворения жизненной потребности A ; $E = \{e\}$ – множество целевых критериев, используемых при построении планов (максимальный доход, минимальные финансовые затраты, минимальная трудоемкость, минимальное время выполнения и т.п.). Если система строит для клиента план, соответствующий предпочтительному для него целевому критерию, Q_A увеличивается на некоторую величину. Пусть N_e – число планов, построенных в соответствии с целевым критерием e , давших прирост Q_A . Если предпочтительный целевой критерий клиента неизвестен, система стремится построить для него план, соответствующий тому целевому критерию, у которого N_e максимально, т.е. стремится построить план, который с наибольшей вероятностью вызовет увеличение Q_A .

Методы формирования планов решения включают анализ цели задачи, выделение подцелей, выбор

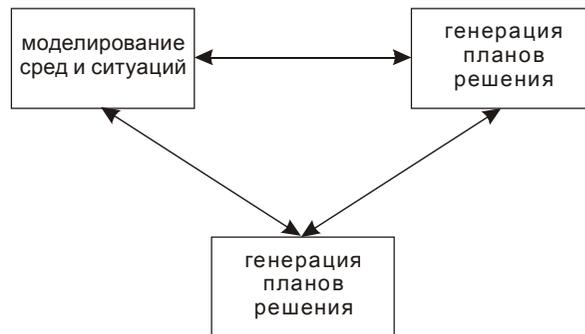


Рис. 3.2. Основные процессы планирования решений

действий, обеспечивающих достижение цели или подцелей. В результате рождаются взаимозависимые последовательности подцелей и действий. Процесс целеполагания (генерирование подцелей) при решении задачи неразрывно связан с процессом выбора действий. Сформированный план решения может включать в качестве структурных единиц модели действий (операторы), подцели или подзадачи. От характера задачи зависит вид действий, образующих план решения. В частности, действиями могут быть преобразования в среде, в которой задана задача, или вычисления.

Процессы формирования понятийной модели проблемы, генерации и выполнения плана решения взаимосвязаны (рис. 3.2). При генерации и выполнении плана обнаруживается недостаток информации, содержащейся в модели проблемы, в результате чего возникает необходимость ее доработки.

Сложность процесса формирования плана зависит от свойств предметной области, в которой решаются задачи. В некоторых случаях для каждой ситуации, получаемой в процессе решения, априори могут быть указаны применимые действия. Это может быть сделано путем задания графа, представляющего все возможные преобразования, которые выполняются в заданной предметной области. При решении задач на графе выделяются вершины, соответствующие исходной и целевой ситуациям, а сам процесс формирования плана состоит в выделении цепочки преобразований, соединяющей эти вершины. Для каждой из ситуаций, возникающих при решении задачи, граф задает все применимые действия. Если же задать граф, моделирующий область поиска планов решения, невозможно, формирование планов решения существенно усложняется. Задание такого графа оказывается нецелесообразным или невозможным в двух случаях: 1) если предметная область характеризуется слишком большим числом различных действий; 2) если в заданной предметной области планы решения различаются в основном не составом действий, а ва-

риантами применения одного или нескольких действий и число вариантов применения действий велико. Однородные предметные области, соответствующие второму случаю, характерны для инженерной деятельности. Например, планы решения задач сборки конструкций, в основном, состоят из различных вариантов применения одного действия – присоединения детали к конструкции. Планы решения задач пространственного размещения объектов состояются из различных вариантов перемещений объектов.

Формирование оптимального плана решения.

При генерации плана решения по графу ситуаций естественным является стремление сформировать оптимальный план. Параметр, по которому производится оптимизация, назовем стоимостью плана. Оптимальным будем считать план с наименьшей стоимостью.

Пусть $g^*(x)$ – стоимость оптимального пути от вершины s графа ситуаций, соответствующей исходной ситуации, к вершине x ; $h^*(x)$ – стоимость оптимального пути из вершины x к вершине, представляющей целевую ситуацию. Значением функции $f^*(x) = g^*(x) + h^*(x)$ является стоимость оптимального пути из s к целевой вершине через вершину x . Функция $f^*(x)$ дает полную информацию об оптимальном пути, проходящем через вершину x . При решении прикладных задач задание такой функции в большинстве случаев оказывается невозможным. Поэтому на практике чаще приходится пользоваться оценочной функцией

$$f(x) = g(x) + h(x), \quad (3.1)$$

в которой $f(x)$ и $h(x)$ аппроксимируют $g^*(x)$ и $h^*(x)$. Предложение применять для поиска планов решений оценочную функцию (3.1) сформировалось в результате развития теории динамического программирования и процедур поиска в графах. Общая теория

использования оценочных функций для управления поиском планов решения изложена в [Нильсон, 1985].

С помощью функции $h(x)$ в процесс поиска плана решения вовлекается эвристическая информация о целях и проблемной области, в которой решается задача. $h(x)$ получила название *эвристической функции*. В [Нильсон, 1985] показано, что при выполнении условия

$$h(x) \leq h^*(x) \quad (3.2)$$

алгоритм, выбирающий на каждом шаге прослеживания графа ситуаций вершину с минимальным значением $f(x)$, всегда обеспечивает построение оптимального плана решения. Условию (3.2) удовлетворяет и случай отсутствия информации о путях, ведущих к целям, т. е. случай, когда $h(x) = 0$. При $h(x) = 0$ осуществляется перебор путей в графе ситуаций, на каждом шаге которого выбирается вершина с наименьшим значением $g(x)$. Это неэффективный способ поиска плана решения, хотя и обеспечивающий с гарантией построение оптимального плана. Приближение $h(x)$ к $h^*(x)$ при соблюдении условия (3.2) влечет увеличение эффективности поиска (уменьшение объема просматриваемой части графа ситуации) с сохранением гарантии построения оптимального плана.

При решении трудных задач для повышения эффективности поиска приходится нарушать условия (3.2), идя на риск потерять оптимальный план. Следует также иметь в виду, что выполнение условия (3.2) часто невозможно обеспечить из-за недостатка информации для построения функции $h^*(x)$.

При оценке эффективности поиска плана решения следует учитывать не только число вершин прослеженной части графа ситуаций, но и объем работы, связанной с определением значений функции $f(x)$. Поэтому иногда приходится игнорировать функцию $g(x)$, тем более, что влияние на процесс поиска этой составляющей $f(x)$ с ростом $h(x)$ уменьшается. Сохранение $g(x)$ в составе $f(x)$ обеспечивает при недостижении цели продолжение поиска планов решения

вплоть до полного перебора путей в графе ситуаций, что гарантирует нахождение какого-нибудь, пусть не оптимального, плана решения.

Относительными весами $g(x)$ и $h(x)$ в оценочной функции можно управлять, представив ее в виде $f(x)=g(x)+wh(x)$, где w – положительное число. Большое значение w усиливает эвристический компонент, в результате чего поиск становится более целеустремленным, при малых значениях w поиск планов решения приобретает характер перебора путей в графе. Компромиссное решение состоит в изменении w в зависимости от длины пройденного пути в графе ситуаций. В начале поиска для уменьшения просматриваемой части графа ситуаций следует усилить влияние эвристического компонента. По мере удаления от исходной ситуации целесообразно путем уменьшения w увеличивать перебор с тем, чтобы не пройти мимо целевых ситуаций.

Направление поиска плана решения

Планы решения можно строить, формируя цепочки операторов, применимые к исходной ситуации,

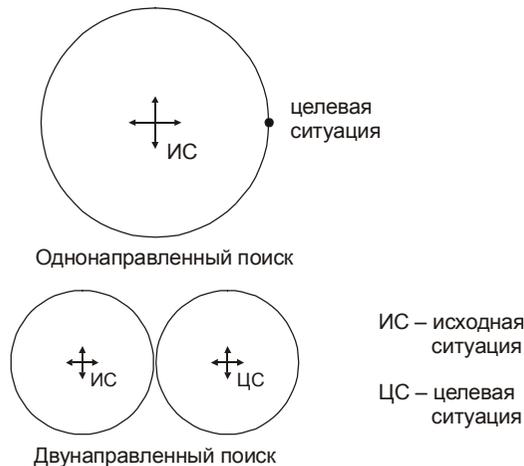


Рис. 3.3. Построение плана с помощью функции $g(x)$

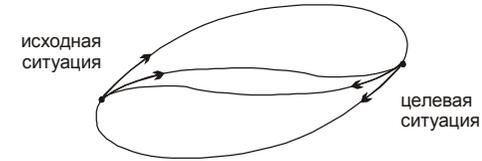


Рис. 3.4. Двунаправленный эвристический поиск плана решения (неудачный вариант)

или цепочки подцелей главной цели задачи. В графе ситуаций этим двум способам формирования плана решения соответствуют два направления прослеживания графа: от вершины, представляющей исходную ситуацию, к целевым вершинам и от целевых вершин к вершине исходной ситуации. Сокращения числа операций выбора оператора при поиске плана решения можно добиться, если организовать встречный двунаправленный поиск от исходной и целевой ситуаций.

Уменьшение объема просматриваемой части графа ситуаций при двунаправленном поиске по сравнению с однонаправленным в случае, когда поиск ведется только с помощью функции $g(x)$, иллюстрируется рис. 3.3.

При использовании эвристической функции двунаправленный поиск не всегда дает выигрыш и может привести к удвоению числа просматриваемых вершин графа, если пути в графе, прослеживаемые от исходной ситуации и цель, не пересекаются (рис. 3.4).

Для решения сложных задач в случаях, когда нет возможности обеспечить строгое выполнение условия (3.2), целесообразно организовать несимметричный

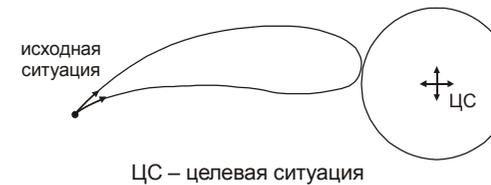


Рис. 3.5. Несимметричный двунаправленный поиск плана решений

двунаправленный поиск плана решения так, как показано на рис. 3.5 перебором со стороны цели и узконаправленно со стороны исходной ситуации. В результате удастся сочетать достаточно высокую эффективность поиска с высокой вероятностью достижения цели даже при невыполнении условия 3.2 [Гладун, 1974].

Граф подцелей

При построении планов решения возможны два случая выбора оператора: 1) имеется априорная информация о приоритетах операторов или вариантов их применения (например, статистические оценки операторов, план решения аналогичной задачи и т. п.); 2) отсутствует априорная информация, определяющая приоритеты операторов.

Если априорной информации о приоритетах операторов нет, построение плана решения осуществляется на основе анализа формулировки задачи (моделей исходной и целевой ситуаций). В этом случае могут быть использованы методы построения планов решения путем трансформации задач в задачи доказательства теорем в исчислении предикатов первого порядка. Таким образом могут быть построены планы решения в *коммутативных системах продукции* [Нильсон, 1985] (правил «если ..., то»), т.е. при использовании множеств продукции, обладающих следующим свойством: каждая продукция после того, как удовлетворяется ее условие применимости, остается применимой, какие бы последовательности продукции не применялись к данным. Такие процессы планирования решений называются монотонными. При решении прикладных задач чаще приходится использовать некоммутирующие системы продукции. В связи с этим мы рассмотрим более общие методы построения планов решения, действующие как в коммутативных, так и в некоммутирующих системах. Техника выбора оператора на основе использования

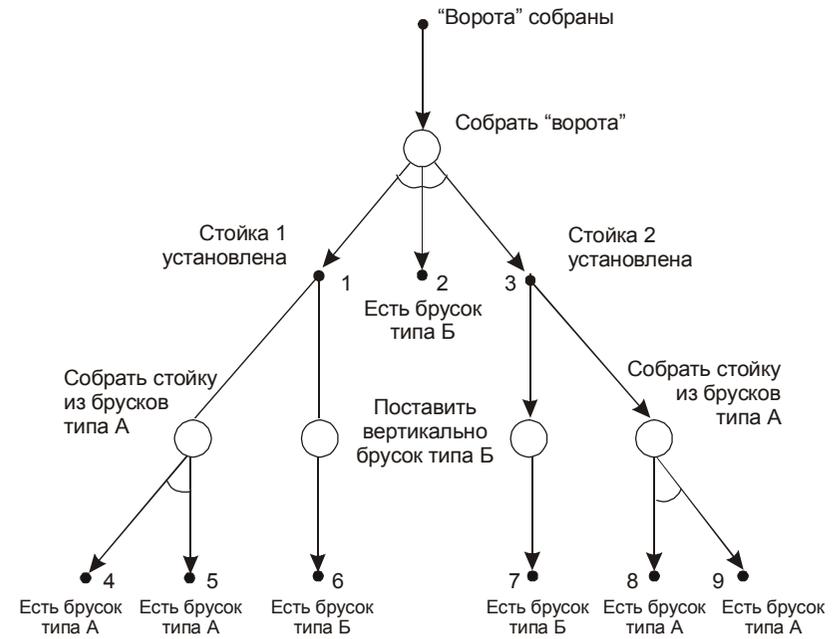


Рис. 3.6. Граф подцелей

информации, содержащейся в формулировке задачи, получила название *анализа средства – цели*.

В методологии анализа средства – цели важную роль играют средства выделения из модели целевой ситуации информации, которую можно было бы использовать для управления генерацией планов решения.

В связи с этим строится *граф подцелей* – ориентированный граф, в котором одна вершина без входящих дуг, называемая *корнем*, соответствует цели задачи, а остальные вершины – подцелям и действиям. Граф подцелей выделяется из понятийной модели проблемы путем прослеживания путей, исходящих от вершин, представляющих цель задачи.

Дуги направлены от вершин, соответствующих цели или подцелям, к вершинам, соответствующим действиям, в результате выполнения которых достигаются цель или подцели; от вершин, соответствующих

щих действиям к вершинам, соответствующим подцелям, к которым применяются эти действия; от вершин, соответствующих цели или подцелям, к вершинам, которые представляют составляющие подцели.

На рис. 3.6 приведен граф подцелей задачи «со-

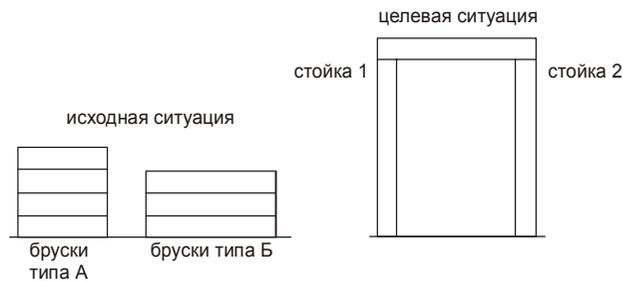


Рис. 3.7. Задача «собрать ворота»

брать «ворота» из брусков двух типов» (рис. 3.7). Варианты сборки показаны на рис. 3.8.

Процесс формирования моделей подцелей по модели целевой ситуации называется *редукцией задачи*.

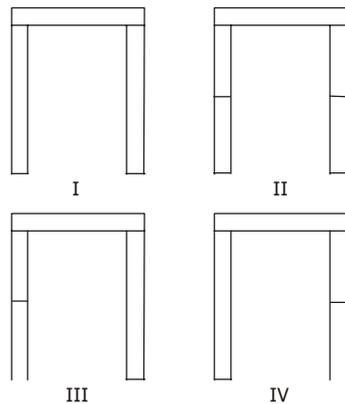


Рис. 3.8. Варианты сборки ворот

Граф подцелей отражает альтернативные и взаимодополняющие пути достижения целей и подцелей. Вершина, представляющая подцель, имеет несколько дочерних вершин, если данная подцель представляет собой объединение других подцелей или если она может быть достигнута путем выполнения различных последовательностей действий. Наличие нескольких дочерних вершин у вершины, представляющей действие, означает, что условием его применимости является достижение всех подцелей, представленных дочерними вершинами. Указанные ситуации выделяются в графе путем связывания дочерних вершин каждой вершины графа логическим отношением И, если они соответствуют подцелям, или логическим отношениям ИЛИ, если они соответствуют действиям. Графы с вершинами, связанными отношениями И, ИЛИ, называются И/ИЛИ-графами.

Множество подцелей, представленных дочерними вершинами вершины И/ИЛИ-графа, связанными отношением И, назовем И-множеством этой вершины. На рис. 3.6 заходящие дуги вершин, соответствующих подцелям одного И-множества, объединены. Дочерние вершины, заходящие дуги которых не объединены, связаны отношением ИЛИ.

Если задача имеет решение, построение графа подцелей рано или поздно приводит к формированию подцелей, описания которых отражают факты исходной ситуации. На рис. 3.6 такими подцелями являются подцели 4, 5, 6, 7, 8, 9.

План решения формируется в результате наращивания графа подцелей или в результате двунаправленного процесса, при котором помимо развития графа подцелей осуществляется прослеживание путей от исходной ситуации к целям и подцелям. При этом постоянно приходится решать, какую из ветвей графа подцелей наращивать дальше, какое из действий выбрать для достройки графа подцелей или преобразования ситуации. Выбор действия и варианта его применения для достройки дерева подцелей или преобразования ситуации осуществляется с по-

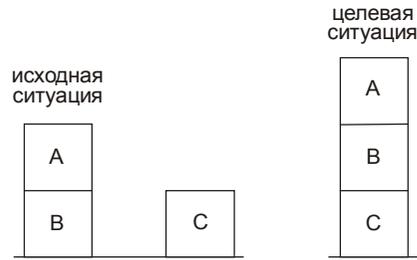


Рис. 3.9. Задача перестановки кубиков

мощью оценочных функций, определяющих степень уменьшения различий между подцелями и текущей ситуацией при выполнении действия.

Формирование плана решения существенно усложняется, если в графе подцелей имеются действия,

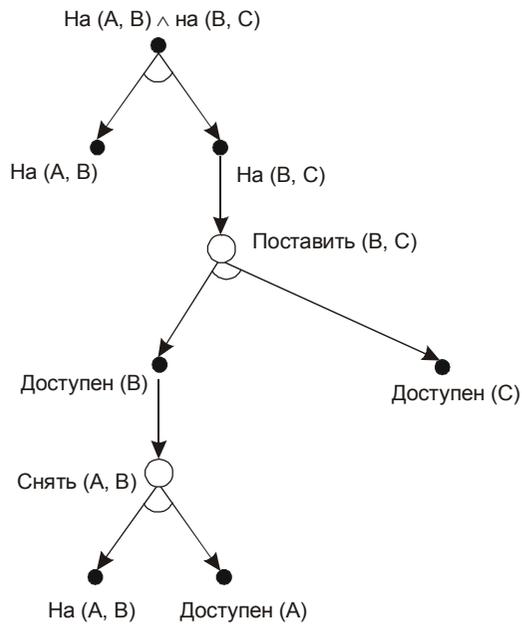


Рис. 3.10. Начальный граф подцелей задачи перестановки кубиков

выполнение которых препятствует выполнению других действий или приводит к разрушению уже достигнутых подцелей. Будем называть такие действия разрушающими. На каждом шаге построения плана необходимо стремиться выбирать неразрушающие действия. Возможен случай, когда в графе подцелей нет неразрушающих действий. Рассмотрим задачу перестановки кубиков, показанную на рис. 3.9. На рис. 3.10 показан граф подцелей этой задачи. Единственное применимое действие СНЯТЬ (A, B) приводит к ликвидации достигнутой подцели НА (A, B). При выборе этого действия возникает необходимость достройки графа подцелей. Граф подцелей, преобра-

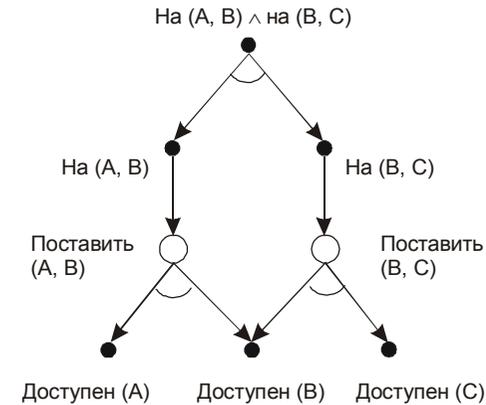


Рис. 3.11. Граф подцелей задачи перестановки кубиков после выбора действия СНЯТЬ (A, B)

зованный после выбора действия СНЯТЬ (A, B), показан на рис. 3.11. В этой новой ситуации нельзя выбрать действие ПОСТАВИТЬ (A, B), т.к. оно разрушает подцель ДОСТУПЕН (B), вследствие чего окажется невозможным действие ПОСТАВИТЬ (B, C). Выбор действия ПОСТАВИТЬ (B, C) не препятствует выполнению других действий графа подцелей и приводит к решению задачи.

При построении плана решения в результате неправильного выбора действий могут возникать ложные последовательности действий, не приводящие к решению задачи. Поэтому нужно иметь средства оценки промежуточных результатов формирования плана и отказа от развиваемой последовательности действий, если оценивание покажет ее бесперспективность. Для оценки прогресса в ходе решения используются оценочные функции, определяющие степень близости текущей ситуации к главной цели. При изменении хода решения построение плана решения продолжается с новой точки графа подцелей.

В результате выбора действия в понятийной модели проблемы возникают цепочки причинно-следственных изменений, зависящие от конфигурации связей исключаемых или добавляемых понятий. Для пояснения этого явления часто приводится следующий пример. Моделирование перестановки блюда состоит в исключении из модели проблемы понятий и связей, определяющих его прежнее местонахождение, и добавлении понятий и связей, определяющих его новое размещение. Но если на блюде стоит чашка, нужно аналогичные изменения выполнить относительно местонахождения чашки. Если в чашке находится ложка, необходимо сделать то же самое с определением расположения ложки и т. д. Проблема определения изменений в модели проблемы при моделировании выполнения действий получила название проблемы последствий (иногда применяется термин «проблема границ», более близкий к английскому эквиваленту «frame problem»).

Проблема последствий решается путем применения различного рода средств, обеспечивающих выполнение цепочек причинно-следственных преобразований. К числу таких средств относятся, например, процедуры – «демоны», для запуска которых достаточно выполнения условий их применимости. Совершенный аппарат для решения проблемы последствий предложен в языке ЭКСПРЕСС [Гладун, 1994]. С действиями связываются дополнительные

преобразования, выполнение которых зависит от истинности определенных фактов до или после выполнения действия.

Нелинейные планы

В предыдущем разделе были рассмотрены методы генерации планов решения, являющихся последовательностями действий. Такие планы называются линейными. В этом разделе рассматриваются методы генерации планов более сложной структуры: иерархических, ветвящихся, итеративных, параллельных.

Иерархическое планирование. В множестве действий, применяемых при решении задачи, практически всегда можно выделить основные, в результате выполнения которых достигаются этапные подцели основной цели, и детализирующие, вспомогательные действия. Эта возможность используется для упрощения процесса формирования планов, состоящих из большого числа действий. Формирование «длинного» плана разделяется на несколько этапов. Сначала строится обобщенный план из основных действий, затем с помощью этого плана формируется план, включающий детализирующие и вспомогательные действия, далее может быть построен еще более детальный план и т. д. В результате возникает иерархия планов.

План самого высокого уровня иерархии «короче» окончательного плана, формирование его обычно менее затруднительно, чем формирование окончательного плана без использования иерархического планирования. Составление остальных планов, в том числе и окончательного, упрощается за счет использования информации, содержащейся в планах более высоких уровней.

Планирование решений в условиях неопределенности. Под неопределенностью в данном случае понимается такое состояние модели проблемы, при кото-

ром система в процессе планирования решения не может однозначно сформировать множество применимых действий. Следует отличать неопределенность формирования множества применимых действий от неопределенности выбора действия из числа применимых, которая является принципиальной особенностью процесса планирования решений в сложных средах.

Неопределенность формирования множества применимых действий имеет место при неполноте, недостоверности или противоречивости данных, с которыми работает система. Причиной неопределенности может быть ошибка или неточность формализации среды или задачи, несовершенство средств сбора информации, динамичность среды, в которой решаются задачи, относительность или нечеткость используемых понятий, несовершенство методов классификации и логического вывода, применяемых при формировании моделей и т.д.

Есть несколько способов представления неопределенных данных. Для отражения неопределенности применяются выражения с переменными типа «объект X находится на объекте Y », « X соединяет Y и Z » и т.п. В таких выражениях отражается неполнота сведений о связях, свойствах и состояниях объектов. Иногда указывается набор возможных значений переменных. Практикуется пометка выражений, относительно истинности которых имеются сомнения. Вводятся различные вероятностные оценки степени достоверности фактов и результатов выполнения действий. Для выражения нечетких понятий применяются качественные модификаторы и квантификаторы, соответствующие таким словам, как «очень», «иногда», «недалеко» и т. п.

При обнаружении неопределенности в процессе планирования прежде всего должна быть предпринята попытка устранить ее. Информация, устраняющая неопределенность, может быть получена в ответ на запрос системы извне, от пользователя. Предложены методы логического вывода, устраняющие

неопределенность, причиной которой является противоречивость или неполнота данных.

Наличие неопределенности в модели проблемы вынуждает генерировать альтернативные планы, которые соответствуют разным значениям переменных в выражениях с переменными, значениям «истина» и «ложь» помеченных выражений, а также выражений, снабженных численными оценками достоверности фактов. Альтернативные планы с помощью условных переходов объединяются в один ветвящийся план. При выполнении плана ситуации, прогнозируемые в процессе планирования, становятся реальными, в результате чего многие виды неопределенности исчезают, и выбор ветвей ветвящегося плана для выполнения становится вполне детерминированным.

Особенно важна роль ветвящихся планов при решении задач в динамических средах, когда причиной неопределенности служит изменчивость среды, а отдельные альтернативные планы соответствуют прогнозируемым состояниям и действиям ее активных элементов.

Итеративные планы. Существует много задач, при решении которых приходится повторять одни и те же действия или последовательности действий. Для повышения компактности планов их повторяющиеся участки записываются в форме циклов. Планы, содержащие циклы, называются *итеративными*.

В [Дрючин, 1986] выделены три основных подхода к формированию итеративных планов.

1. Директивный подход, при котором в систему извне вводятся указания об организации циклов.

2. Трансформационный подход, при котором итеративные планы формируются путем преобразования предварительно составленных линейных планов. В циклы объединяются похожие, например, отличающиеся только операндами, участки линейного плана.

3. Индуктивный подход, при котором циклы формируются непосредственно в процессе генерации плана при наличии однотипных данных или появлении повторяющихся ситуаций.

При директивном подходе эффект применения автоматических средств планирования значительно снижается. Трансформационный подход предполагает предварительное построение линейного плана, что затруднительно, а иногда и невозможно, если число повторений одних и тех же действий велико. На фоне указанных недостатков директивного и трансформационного подходов особое значение приобретает индуктивный подход – единственный подход, позволяющий строить итеративные планы без включения человека в процесс планирования и предварительного многократного формирования однотипных участков планов.

Параллельные планы. Важной разновидностью планирования решений является планирование согласованных действий нескольких активных систем, например роботов. В этом случае множество действий делится на подмножества, соответствующие отдельным системам.

Планирование согласованных действий нескольких систем можно выполнять на основе графов подцелей, как описано в предыдущем разделе. Но при решении задачи несколькими системами возникает возможность, а иногда и необходимость сократить время выполнения плана решения путем его распараллеливания между отдельными системами. Выделение участков плана, которые могут выполняться одновременно, с целью минимизации времени выполнения всего плана представляет собой самостоятельную, иногда сложную оптимизационную задачу. Проблемы организации взаимодействия систем, решающих общую задачу, рассматриваются в [Chaib-draa, 1992].

Адаптация к классу задач

Адаптацией к классу задач называется процесс накопления и использования информации о классе задач, направленный на улучшение функционирования системы при решении задач этого класса.

Одной из характеристик плана решения является степень общности. Чем больший уровень обобщения достигнут при формировании плана, тем шире класс задач, для решения которых может быть применен этот план. Планы с высоким уровнем обобщения иногда называют *схемами решения*. Схема решения ограничивает область поиска путей решения задач некоторого класса. Информацию о классе задач, которым соответствует схема решения, будем называть *определением класса задач*. Определения классов задач и соответствующие схемы решения формируются по мере накопления опыта решения задач. На разных стадиях формирования определения $D(q)$ и схемы решения $S(q)$ класса задач q множество задач $U_{S(q)}$, пути решения которых находятся с помощью схемы решения $S(q)$, может не совпадать с множеством задач $U_{D(q)}$, определяемых определением $D(q)$. Для исключения неоднозначности выбора схемы решения по определению класса задач должно выполняться соотношение $U_{D(q)} \subseteq U_{S(q)}$.

Поскольку область применения схемы решения $S(q)$ определяется множеством $U_{D(q)}$, а не $U_{S(q)}$, нет смысла достигать в схеме решения большего уровня обобщения, чем в определении класса задач.

Системы планирования решений, в которых выполняется соотношение

$$U_{D(q)} = U_{S(q)} \quad (3.3)$$

называется *сбалансированными*.

Рассмотрим общую схему формирования схем решения и определений классов задач в процессе накопления опыта. Определения классов задач и схемы решений можно сформировать, подвергая формулировки и пути решений отдельных задач некоторым детерминированным преобразованиям. Функционирование адаптивной системы такого типа иллюстрируется схемой на рис. 3.12. Поступающие в систему задачи классифицируются путем сопоставления их формулировок с определениями классов задач, хранящимися в памяти системы. Если при этом удает-

ся найти определение класса задач, включающего новую задачу, для ее решения используется схема решения, связанная с найденным определением. Если новая задача не входит ни в один из классов, для которых уже сформированы определения, осуществляется поиск пути решения без использования

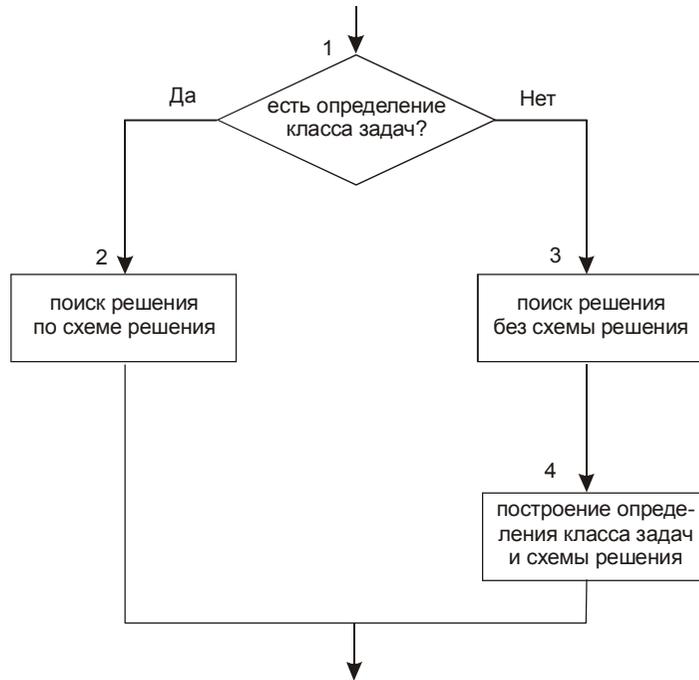


Рис. 3.12. Адаптация к классу задач

схемы решения. На основании формулировки задачи и пути ее решения строятся определение и схема решения нового класса задач.

Классификация задач на основе определений классов задач, полученных с помощью априорно заданных детерминированных преобразований, как правило, не отображает объективно существующее разделение пространства задач на классы по принципам решений, т.е. не отображает глобальные закономерности, присущие пространству путей решений.

Подобные трудности могут быть преодолены путем использования процессов индуктивного обучения при формировании по крайней мере одной из составных частей информации о классе задач. Например, можно задать преобразование, дающее схемы решения с нужным уровнем обобщения, а определения классов задач формировать в процессе обучения на множестве задач, добиваясь путем постепенных приближений выполнения условия сбалансированности (3.3). При этом в системе должны быть предусмотрены средства корректировки определений классов задач, возникающих на различных этапах обучения. Выбор преобразования, дающего схемы решения, в качестве априорно заданного обеспечивает классификацию задач по принципам решения.

Рассмотрим процесс индуктивной адаптации, обращаясь к схеме на рис. 3.13. Она отличается от схемы на рис. 3.12 прежде всего наличием блоков коррекции определений классов задач (блоки 4 и 9). Блок 4 запускается в том случае, если не удастся решить задачу с помощью найденной схемы решения. Это означает, что решаемая задача была ошибочно причислена к классу задач, которому соответствует схема решения. Определение этого класса должно быть скорректировано таким образом, чтобы в дальнейшем подобные задачи в него не включались. Такого рода корректировка называется *сужением класса задач*.

Если решение задачи было получено без использования схемы решения запускается блок построения схем решений (блок 6). Построенная схема решения сравнивается со схемами решений, хранящимися в памяти системы. Отсутствие в памяти схемы решения, совпадающей с вновь полученной, означает, что новая задача отличается по принципам решения от задач, решенных системой ранее. В связи с этим нужно начать формирование определений классов задач, для которых новая схема решения пригодна. В качестве начального определения класса задач берется формулировка решенной задачи. Информация о новом классе задач заносится в память системы.

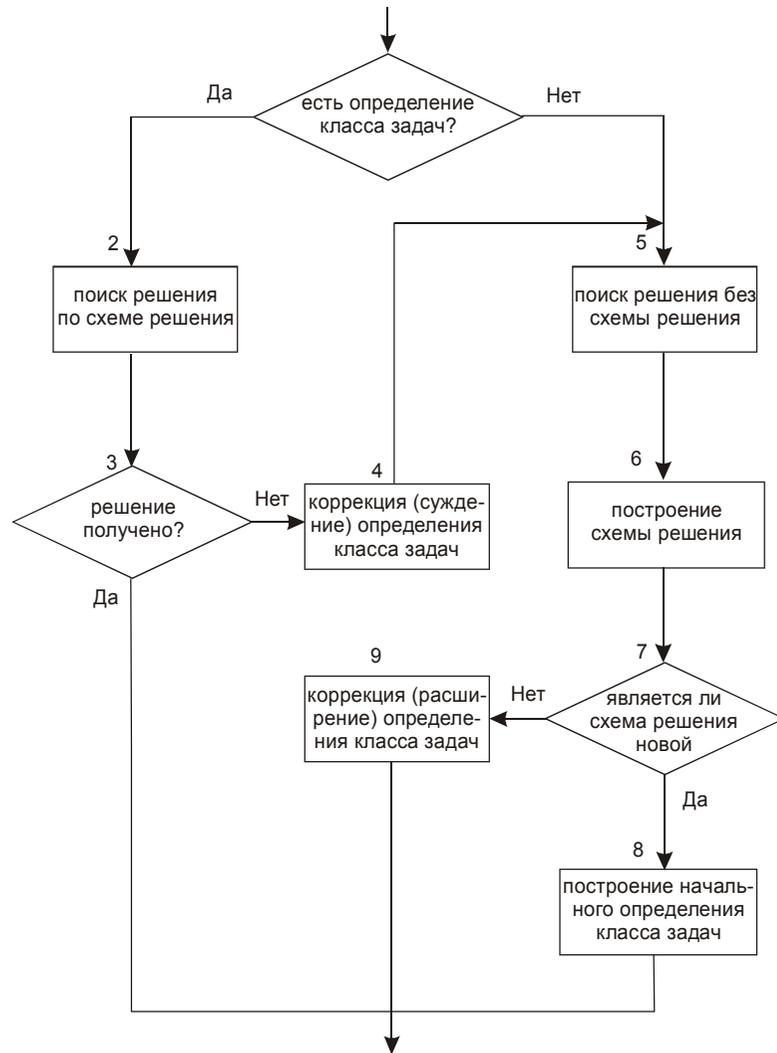


Рис. 3.13. Процесс индуктивной адаптации

Наличие в памяти системы схемы решения, совпадающей с вновь составленной, означает, что новая задача по принципам решения входит в один из классов, уже выделенных в системе. Необходимо скор-

ректировать определение этого класса таким образом, чтобы в дальнейшем в него включались подобные задачи (блок 9). Корректировка такого типа называется *расширением класса задач*.

Блоки корректировок интенсивно используются в начальный период работы системы, когда происходит процесс накопления схем решений и совершенствование определений классов задач. Посредством корректировок для каждого из выделенных классов задач строятся последовательные приближения к определениям, удовлетворяющим соотношению сбалансированности. В обученной системе в основном работают блоки 1,2,3.

Системы с индуктивной адаптацией составляют важный класс кибернетических систем с развитыми средствами обобщения и использования накопленного опыта.

Рассмотрим принципы формирования схем решения и определений классов задач.

Схема решения может быть эффективно использована в целях управления процессом поиска пути решений, если в ней выделены основные этапы решения и задана последовательность этих этапов.

Простейший способ построения схемы решения по пути решения конкретной задачи состоит в задании этапов указанием имен действий пути решения. Решение задачи по такой схеме решения сводится к выбору варианта применения каждого действия последовательности..

Более высокий уровень обобщения может быть достигнут, если каждый этап решения схемы задавать указанием наименования класса действий. Все действия должны быть разделены на классы, объединяющие действия одного типа. На каждом шаге формирования пути решения поиск действия осуществляется в пределах класса, указанного в схеме решения.

Рассмотрим еще один способ построения схем решений, обеспечивающих более высокий уровень обобщения.

В путях решения конкретных задач можно выделить промежуточные и конечные результаты, кото-

рые являются общими в пределах некоторого класса задач, характерного для заданной проблемной среды, и результаты, присущие только анализируемой задаче, появление которых вызвано спецификой ее условия. Промежуточные результаты первого типа назовем *ключевыми подцелями*.

Схема решения строится в виде последовательности $S(q)=G_1G_2...G_m$, где $G_i(i=1, 2, \dots, m)$ – ключевые подцели, G_m – главная цель.

Опишем два подхода к выделению ключевых подцелей.

Первый основан на знании семантики предметной области. Выделяются действия, выполнение которых всегда является важным этапом в решении задачи. Для построения схемы решения в пути решения конкретной задачи отмечаются в качестве ключевых подцелей результаты выполнения выделенных действий. Имена объектов в описании ключевых подцелей заменяются переменными. Главная цель формируется в результате подстановки переменных вместо имен объектов описаний цели, достигнутой с помощью анализируемого пути решения. Такой подход удобен для решения задач, в которых целью является создание новых объектов, например, для решения задач сборки конструкций. В задачах сборки в качестве действий, с помощью которых достигаются ключевые подцели, выделяются действия, обеспечивающие присоединение деталей к конструкции.

При втором подходе схема решения строится полностью автоматически на основе анализа информации, содержащейся в описании цели. В качестве ключевых подцелей выделяются промежуточные результаты пути решения анализируемой задачи, состоящие в изменении свойств, состояний и отношений объектов, указанных в описании цели. Имена объектов в описаниях ключевых подцелей заменяются переменными. Главная цель формируется так же, как и в первом подходе.

При использовании схемы решения для решения конкретных задач переменные в описаниях ключе-

вых подцелей заменяются константами описания цели. Исходная задача сводится к последовательности подзадач, в которых целями служат ключевые подцели и главная цель схемы решения. Суммарный объем областей поиска путей решений подзадач меньше области поиска пути решения основной задачи. Ключевые подцели схемы решения, ориентируя поиск в нужном направлении, играют роль вех в процессе поиска пути решения.

В формировании определений классов задач (блоки 4, 9 на рис. 3.13) главную роль играет процесс индуктивного обучения, описанный в разделе 2.2.

3.3. Формирование многокритериальных решений

Задача нормализации параметров

Процессы выбора действий изучаются в рамках тематики, связанной с принятием решений. Получили широкое распространение исследования, нацеленные на поиск оптимальных решений с использованием методов математического программирования [Кигель, 1999]. В рамках этого направления выбор решений из множества альтернатив осуществляется на основе известного отображения между альтернативами и их критериальными оценками. В практических применениях такой подход не всегда возможен, т.к. отображение между альтернативами и критериями часто неизвестно или нестабильно. В таких ситуациях приходится применять эвристические методы, использующие информационные модели предметных областей, в которых осуществляется процесс выбора решений. Для организации процессов решения, ориентированных на удовлетворение жизненных потребностей системы, могут быть использованы задачи в которых целью является выпол-

нение множества ограничений (constraint satisfaction problems). Приведем формальную постановку задачи такого типа.

Имеется объект управления

$$\langle X, P, Y, F, L \rangle, \quad (3.4)$$

где $X = \{x_i\}$, $i = 1, 2, \dots, n$ – множество управляющих параметров (управляющих воздействий), путем изменения которых меняются состояния объекта управления;

P – множество производных параметров, определяемых через другие параметры;

Y – множество выходных параметров, $Y \subseteq P$;

F – множество функций и продукционных правил типа «если, ...то», определяющих зависимости производных параметров от других параметров (управляющих или производных);

L – множество ограничений типа $a_y \leq y' \leq b_y$, где a_y , b_y – числовые константы, y' – значение параметра $y \in Y$.

Множество выходных параметров может включать жизненные потребности объекта управления.

Целью является нахождение в пространстве управляющих параметров точки $x = (x_1, \dots, x_n)$, в которой выполняются все ограничения множества L .

Наборы значений управляющих параметров служат альтернативами, среди которых выбирается решение. Ограничения из множества L играют роль критериальных оценок.

Выходной параметр называется *нормализованным*, если его значение соответствует заданным ограничениям. Выходные параметры, не соответствующие своим ограничениям, называются *ненормализованными*.

Состояние объекта, в котором удовлетворяются все ограничения, называется *нормативным*. Задачу достижения нормативного состояния будем называть далее задачей *параметрической нормализации*.

Если бы отображение $X \rightarrow Y$ было задано некоторой системой уравнений, для выбора решения мож-

но было бы использовать методы математического программирования. В (3.4) отображение $X \rightarrow Y$ непосредственно не задано. Не задана также априорная информация (вероятности исходов действий, полезности, приоритеты действий и состояний), которая используется в теории решений для выбора решений в условиях неопределенности. Решение должно быть построено только на основе семантической информации, характеризующей среду и цели [Russell, 1995].

Связи между элементами множеств X и Y нужно сформировать, используя информацию о непосредственных связях между параметрами. Решение может быть получено методом последовательных приближений. Очередное приближение к решению формируется путем изменения отдельных управляющих параметров и тестируется на соответствие критериальным оценкам.

Общая схема метапроцедуры такова:

1. Выделение подмножества релевантных управляющих параметров, т.е. таких управляющих параметров, которые влияют на значения ненормализованных выходных параметров.

2. Определение направления изменения релевантных управляющих параметров, приближающего ненормализованные выходные параметры к их нормативным значениям.

3. Выбор наилучшего релевантного управляющего параметра в соответствии с некоторым критерием.

4. Моделирование выполнения выбранного управляющего воздействия, т.е. определение новых значений выходных параметров, соответствующих новому значению выбранного управляющего параметра.

5. Повторение пп. 1 – 4, если решение задачи не получено и есть возможность формировать новое приближение.

Для реализации метапроцедуры нужно иметь возможность:

– осуществлять сложные поисковые операции;

– строить цепочки преобразований, связывающие управляющие параметры с выходными;

– моделировать многошаговые последствия применения управляющих воздействий.

Эффективность выполнения этих действий зависит от качества компьютерной модели, представляющей задачу информацию. Лучший рецепт поддержки поисковых операций состоит в использовании сетевых моделей. В следующем разделе будет описана сетевая модель объекта управления типа (3.4).

Балансные модели

Балансной сетью называется ориентированный граф $\langle V, U \rangle$, в котором:

1) V – множество вершин, соответствующих параметрам всех типов. Левым и правым частям производственных правил соответствуют отдельные параметры.левой части соответствует предикат, принимающий значение 1 только при выполнении условия применимости правила. Параметр, определяемый в правой части, вычисляется с помощью указанного в правиле выражения лишь при единичном значении предиката, соответствующего левой части. В остальных случаях его значение считается неопределенным;

2) U – множество связей: каждая вершина соединяется заходящими дугами с вершинами, которые представляют аргументы функции, используемой для вычисления соответствующего параметра;

3) заходящие дуги специально отмечают, если функции являются возрастающими или убывающими по отношению к параметрам, представленным вершинами, от которых эти дуги исходят.

Управляющие и выходные параметры представляются соответственно управляющими и выходными вершинами сети.

Балансная сеть, не содержащая циклов, называется иерархической (рис. 3.14). Среды, в которых

решаются задачи принятия решений, обычно представляются иерархическими балансными сетями.

Наряду с такими структурами, как «деревья решений», «И/ИЛИ графы», «сети полезности», балансные сети предназначены для управления процессами поиска решений. В системах принятия решений, использующих балансную модель, важное значение имеет блок, обеспечивающий автоматическое построение балансной сети по информации (3.4), а также оперативную перестройку сети при изменениях среды принятия решений. При решении разнородных комплексных задач балансная сеть может быть частью сети понятий.

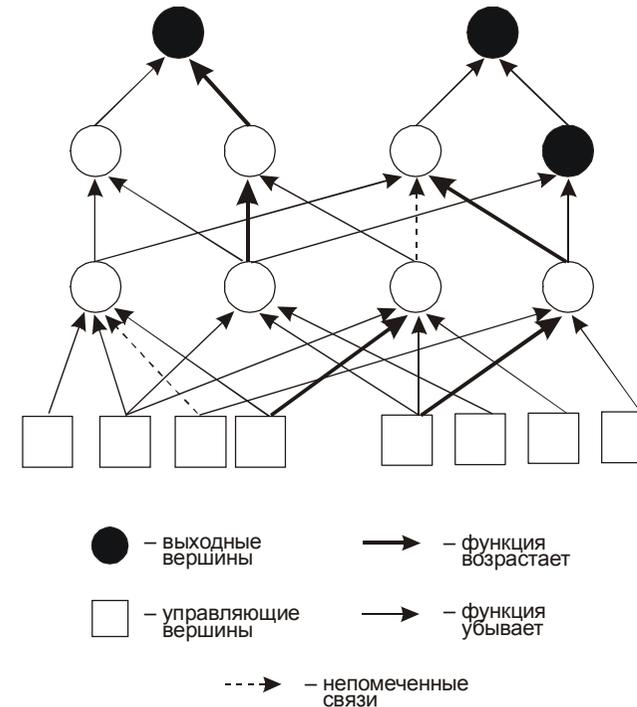


Рис. 3.14. Балансная сеть

Решающая процедура

Процесс решения представляет собой цикл, на каждом шаге которого выбирается «наилучшее» управляющее воздействие и формируется модель новой ситуации, возникающей в результате его применения.

Если имеется возможность использовать некоторый обобщенный критерий для выбора «наилучшего» управляющего воздействия, обеспечивающий минимизацию числа шагов при преобразовании начальной ситуации в нормативную, общая схема управления сводится к схеме динамического программирования. Но если такого критерия нет, необходимо заменить его эвристической оценочной функцией, которая была бы некоторым приближением к нему. Естественно, в этом случае минимизация числа шагов при преобразовании начальной ситуации в нормативную не гарантируется.

Возможность достижения нормативной ситуации и необходимое для этого число управляющих воздействий зависит от качества эвристической функции.

Для описания процесса выбора управляющего воздействия введем следующие обозначения:

α_1 – метка, отмечающая заходящую дугу вершины, если функция, с помощью которой вычисляется соответствующий параметр, является возрастающей по отношению к параметру-аргументу, представленному вершиной, от которой эта дуга исходит;

α_2 – метка, отмечающая заходящую дугу вершины, если функция с помощью которой вычисляется соответствующий параметр, является убывающей по отношению к параметру-аргументу, представленному вершиной, от которой эта дуга исходит;

Y' – множество ненормализованных параметров;

R – множество релевантных управляющих параметров;

h_r^y – переменная, обозначающая необходимое «направление» изменения управляющего параметра $r \in R$ для нормализации выходного параметра y ;

H_r – переменная, обозначающая «направление»

изменения релевантного управляющего параметра r , с учетом всех ненормализованных выходных параметров.

Используются следующие операции.

Операция 1. Поиск релевантных управляющих параметров.

Для ненормализованных выходных параметров с известными значениями определяется необходимое направление изменения. Предположим, для y задано ограничение $a_y \leq y' \leq b_y$. Если $y' < a_y$, необходимо увеличить y' , если $y' > b_y$, – уменьшить. Для каждого ненормализованного параметра y прослеживаются заходящие пути соответствующей вершины до вершин, представляющих управляющие параметры. Перед прослеживанием пути, соединяющего управляющий параметр $r \in R$ с ненормализованным выходным параметром y , устанавливается $h_r^y = 1$, если значение y следует увеличить, и $h_r^y = -1$, если значение y следует уменьшить. В дальнейшем в процессе прослеживания каждого пути знак значения переменной h_r^y сохраняется при прохождении дуги, отмеченной α_1 , и изменяется при прохождении дуги, отмеченной α_2 . При прослеживании неотмеченной дуги переменная h_r^y принимает значение 0. Это значение сохраняется до конца прослеживания соответствующего пути.

В результате выполнения операции 1 формируется множество R и определяется направление изменения релевантных управляющих параметров.

Операция 2. Вычисление значений переменных H_r .

Для каждого $r \in R$ вычисляется значение переменной H_r следующим образом:

$$H_r = \sum_{y \in Y'} h_r^y \quad (3.5)$$

Если $H_r > 0$, целесообразно увеличить значение управляющего воздействия r , если $H_r < 0$ – уменьшить.

Операция 3. Выбор «наилучшего» управляющего воздействия.

В качестве «наилучшего» может быть выбрано управляющее воздействие, имеющее $\max |H_r|$. Примене-

ние для выбора «наилучшего» управляющего воздействия функции (3.5) позволяет максимально использовать особенности балансной сети в целях сокращения объема поисковых операций. Однако функция (3.5) не учитывает влияние релевантных управляющих воздействий на нормализованные выходные параметры, что может стать причиной затягивания или неудачи поиска решения. Учет влияния управляющих воздействий на все выходные параметры осуществляется путем пробных изменений значений релевантных управляющих параметров. При каждом таком тестировании пересчитываются значения всех параметров, представленных в сети вершинами, которые связаны заходящими путями с вершиной тестируемого управляющего параметра. В качестве наилучшего выбирается управляющее воздействие, которое вызывает максимальное уменьшение числа ненормализованных параметров. Стратегия обеспечивает достижение ситуации, когда каждое из релевантных управляющих воздействий вызывает как позитивные, так и негативные изменения множества ненормализованных параметров. Это означает, что достигнуто подпространство недоминируемых альтернатив (множество Парето). В этом случае дальнейшее улучшение выбираемых альтернатив возможно на основе дополнительной информации о приоритетах альтернатив или критериев.

Операция 4. Моделирование применения управляющего воздействия.

Значение выбранного управляющего параметра r в зависимости от знака H_r увеличивается или уменьшается на некоторую фиксированную величину, которая $\alpha \cdot \text{pr}i0i$ указывается для каждого управляющего параметра.

Все параметры, соответствующие вершинам, связанным заходящими дугами с измененным управляющим параметром, пересчитываются.

Результатом применения управляющего воздействия является нормализация имеющихся и, возможно, появление новых ненормализованных вершин.

Для нового состояния сети, если в ней еще остаются ненормализованные параметры, все вышеперечисленные операции повторяются.

Таким образом, общий алгоритм формирования решения выглядит следующим образом:

1. Является ли состояние объекта управления нормативным? Если «нет», переход к п. 2, если «да», переход к «Конец».

2. Операция 1.

3. Операция 2.

4. Операция 3.

5. Операция 4. Переход к п. 1.

Конец.

В случае успеха алгоритм формирует последовательность управляющих воздействий, т.е. некоторый план достижения нормативной ситуации.

Предпочтения лица, принимающего решение, могут быть приняты во внимание с помощью весовых коэффициентов, которые вводятся для управляющих или выходных параметров и используются при выборе управляющих параметров.

Описанный выше подход реализован в инструментальной системе MANAGER. MANAGER сигнализирует о выходе параметров за пределы ограничений, рекомендует возможные управляющие воздействия для ликвидации этих нарушений, принимая во внимание их многошаговые последствия. Имеется опыт применения системы для принятия решений в нестандартных ситуациях при управлении сложными техническими объектами, для выбора параметров, обеспечивающих основные выходные характеристики сложных технических устройств, в целях технической диагностики, для управления административными регионами, для подготовки менеджеров.

Помимо уже состоявшихся применений, указанных выше, описанный метод рекомендуется применять в стратегических информационных системах для поддержки наивысшего уровня управления на предприятиях и в административных районах.

Стратегические информационные системы должны поддерживать решение следующих задач:

1. Анализ возможностей, опасностей или ограничений, которые существуют во внешней окружающей среде.

2. Анализ сильных и слабых сторон организации.

3. Формулирование стратегий с учетом сильных и слабых сторон организации, а также возможностей и опасностей окружающей среды.

Балансная сеть является удобным инструментом для решения задач такого рода.

ЗАКЛЮЧЕНИЕ

Мы рассмотрели принципы организации и основные метапроцедуры целеустремленного поведения, показанные на рис. 1.3. Мне кажется, материал книги дает основание считать, что изложена концепция нового типа компьютерных систем – целеустремленных человеко-машинных систем. Это – многогранная научная проблема, автор осознает, что некоторые важные вопросы не рассмотрены в книге в должной мере. Например, вопросы организации естественно-языкового интерфейса. Постараемся заполнить эти пробелы в будущем.

Ввод в состав функций компьютера функции целеполагания дает повод для генерации весьма смелых футуристических гипотез. Изменится роль компьютеров во многих сферах человеческой деятельности. Повысится уровень автоматизации процессов решения сложных «интеллектуальных» задач. Компьютерная поддержка процессов целеустремленного поведения увеличит эффект использования компьютеров в управлении фирмами, производством, регионами.

Сближение функций компьютера и человека должно сопровождаться повышением доверия к компьютеру, повышением роли компьютера в решении персональных человеческих проблем и, следовательно, расширением использования компьютера в сферах быта, услуг, индивидуального бизнеса.

Естественно, прорыв в решении проблемы организации целеустремленного поведения компьютерных систем породит новые проблемы, новые трудности.

...А там еще другая даль,
Что обернется далью новой.
А та, неведомая мне,
Еще с иной, большой, суровой,
Сомкнется...

А. Т. Твардовский
«За далью—даль»

СПИСОК ЛИТЕРАТУРЫ

Акофф Р., Эмери Ф. О целеустремленных системах. – М: Наука, 1974. – 269 с.

Аркадьев А. Г., Браверман Э. И. Обучение машины классификации объектов. – М: Наука, 1971. – 220 с.

Бонгард М. М. Проблемы узнавания. – М: Наука, 1967. – 320 с.

Валькман Ю. Р. Интеллектуальные технологии исследовательского проектирования. – Киев: Port-Royal, 1998. – 249 с.

Вагин В. Н. Дедукция и обобщение в системах принятия решений. – М: Наука, 1988. – 383 с.

Войшвилло Е. К. Понятие. – М.: Изд-во МГУ, 1967. – 285 с.

Гладун В. П., Ващенко Н. Д. Локально-статистические методы формирования знаний // Кибернетика и системный анализ. – 1995. – N2. – С. 62-74.

Гладун В. П., Величко В. Ю., Киселева Н. Н., Москалькова Н. М. Вывод гипотез о составе и свойствах объектов на основе аналогии//Искусственный интеллект. – 2000. – №1. – С. 44-52.

Гладун В. П., Галаган Н. И., Сасина С. П. Об одной стратегии поиска решений задач преобразования// Кибернетика. – 1974. – N5. – С.134-141.

Гладун В.П. Гипотетическое моделирование: методология и применение // Кибернетика и системный анализ. – 1997. – N1. – С.10-20.

Гладун В. П. Планирование решений. – Киев: Наукова думка, 1987. – 168с.

Гладун В. П. Процессы формирования новых знаний. – София: Педагог, 1994. – 190с.

Горский Д. П. Обобщение и познание. – М: Мысль, 1985. – 208с.

Дрючин Ю. Л. Формирование ветвящихся и итеративных алгоритмов при решении прикладных задач в систе-

мах планирования действий: Автореф. дис. канд. техн. наук. – Киев, 1986. – 18с.

Загоруйко Н. Г., Елкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей.-Новосибирск: Наука, 1985.-108с.

Закревский А. Д. Логика распознавания.-Минск: Наука и техника, 1988.-117с.

Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами.-Киев: Техника, 1975.-311с.

Киселева Н. Н., Ващенко Н. Д., Гладун В. П., Леклер С. Р., Джексон А. Г. Прогнозирование неорганических соединений, перспективных для поиска новых электрооптических материалов//Перспективные материалы. – 1998. – N3. – С.28-32.

Кигель В. Р. Математические методы принятия решений в эффективном предпринимательстве (укр). – Киев: Ин-т экономики, управления и хозяйственного права, 1999. – 269с.

Коваль В. Н., Кук Ю. В. Целенаправленные системы планирования решения//Искусственный интеллект. – 1999. – №2. – С. 158-165.

Маркин О. Ю. Дискретизация количественных признаков при предобработке информации для обобщения// Управл. системы и машины. – 1988. – N2. – С.88-92.

Морозов А. А., Яценко В. А. Интеллектуализация ЭВМ на базе нового класса нейроподобных растущих сетей. – Киев: ИПММС НАН Украины, 1997. – 125с.

Нильсон И. Принципы искусственного интеллекта. – М.: Радио и связь, 1985. – 373с.

Поспелов Д. А. Ситуационное управление. Теория и практика. – М: Наука, 1986. – 278с.

Рабинович З. Л. О концепции машинного интеллекта и ее развитии// Кибернетика и системный анализ. – 1995. – N2.-С.163-173.

Штерн И. Б. Выбранные топики и лексикон современной лингвистики. – Киев: АртЕк,1998. – 335с.

Chaib-draa B., Mandian R., Millot P. Distributed Artificial Intelligence: An Annotated Bibliography//Sigart Bulletin. – 1992(v.3). – N2. – С.24-26.

Gladun V., Rabinovich Z. Formation of the World Model in Artificial Intelligence Systems//Machine Intelligence, 9.-

Ellis Herwood Ltd., Chichester. – 1980. – P.299-309.

Gladun V., Vaschenko N. Local Statistical Methods of Knowledge Formation//Cybernetics and System Analysis. – 1995. – v.31. – N2. – P.207-217.

Hall R. Computational Approaches to Analogical Reasoning: a Comparative Analysis//Artif. Intellig. – 1989. – 3.– N39.– P. 39-120.

Michalski S., Carbonell G., Mitchell M.(edit.) Machine Learning, an Artificial Intelligence Approach.-Morgan Kaufmann, San Mateo, California. – 1986. – v.1,2.

Piatetsky-Shapiro G., Frawley W. (edit.) Knowledge Discovery in Databases//AAAI Press, Menlo Park, California.– 1991.

Quinlan R. Induction of decision trees//Machine Learning–1.– 1986. – P.81-106.

Russell S. Norvig P. Artificial Intelligence. A Modern Approach.-Prentice Hall, New Jersey. – 1995.-932p.

Schank R. Conceptual Information Processing. – New York: An Elsevier Publ., 1980. – 300p.

Wooldridge M., Jennings N., Agent Theories, Architectures and Languages:A Survey//Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August, 1994. – Springer Verlag, 1994. – P.3-39.

СОДЕРЖАНИЕ

Предисловие 5

Глава 1

Целеустремленное поведение

- 1.1. Процессы выбора в поведенческом акте 7
- 1.2. Методологические основы моделирования процессов выбора..... 12
 - Гипотетическое моделирование 12
 - Сетевые модели..... 13
- 1.3. Метапроцедуры поведенческого акта 16

Глава 2

Аналитические процессы

- 2.1. Сетевые структуры для аналитических задач..... 18
 - Атрибутивный анализ 18
 - Пирамидальные сети 19
- 2.2. Методы, включающие индуктивный вывод..... 25
 - Обнаружение знаний 25
 - Аналитические процессы в пирамидальных сетях 34
 - Недетерминистский процесс формирования понятий 43
 - Применения методов анализа на основе пирамидальных сетей 46
- 2.3. Методы, содержащие вывод по аналогии 55

Сеть «объект-свойство»	55
Проектирование состава составных объектов	61
2.4. Дискретный анализ распределений объектов по шкалам признаков	65
Глава 3	
Процессы выбора	
1.1. Формирование системы знаний целеустремленной системы	70
Формирование сети понятий	70
Концептуальные умозаключения	76
3.2. Планирование решений	81
Формирование оптимального плана решения	84
Направление поиска плана решения	86
Граф подцелей	88
Нелинейные планы	95
Адаптация к классу задач	98
3.3. Формирование многокритериальных решений	105
Задача нормализации параметров	105
Балансные модели	108
Решающая процедура	110
Заключение	115
Список литературы	116

PARTNERSHIP WITH COMPUTERS Man-Computer Task-Oriented Systems

Creation of computers was stimulated by the need to have a tool to support mental processes. According to the modern views, capabilities of computer systems are not limited to the functions of that of a tool. In principle modern computer systems are capable to perform all components of a task-oriented behavior act including the formation of goal and the choice of actions. So one can say that such new type systems are, namely, the man-computer task-oriented systems. As a result, a new type of a man-computer interaction appears, which could be defined by the word «partnership». In a task-oriented computer system, general control is realized by the highest level metaprocedures that work on the basis of a common conceptual knowledge.

The book deals with the conception of task-oriented man-computer systems. For the major metaprocedures, such as knowledge discovery, formation of a concept, classification, diagnostics, projection, decision-making and planning, the network knowledge representation is used. Original methods to process effectively large volumes of data have passed a long-term testing. Suggested areas of application are as follows: projection of new chemical compounds and materials with the predefined properties, discovery of characteristic trends in particular industries and regions, medical and technical diagnostics, forecasting in geology, biology and so on.



АССОЦИАЦИЯ СОЗДАТЕЛЕЙ И ПОЛЬЗОВАТЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

предлагает

бизнесменам, научным сотрудникам, инженерам,
медикам, социологам, административным работникам –
всем, кто работает с базами данных, –

сотрудничество в формировании аналитической информации.

- ♦ Ассоциация имеет большой опыт взаимодействия с исследовательскими коллективами, работающими в разных областях **науки и техники**. С помощью методов и программных систем, созданных в Ассоциации, выявлены закономерности, характеризующие химические соединения и материалы разных типов. В сотрудничестве с химиками и материаловедом России и США выполнено несколько тысяч высокоточных прогнозов существования новых химических соединений и материалов.
- ♦ **Бизнесменам** Ассоциация может помочь в формировании условий успешного инвестирования с учетом особенностей регионов и отраслей, в выявлении тенденций изменения пользовательских требований к техническим характеристикам производимых или продаваемых изделий.
- ♦ Для **медиков** создаются системы, помогающие ставить диагноз и выбирать метод лечения, формируются многопараметрические модели, характеризующие состояние здоровья населения по регионам и социальным группам.
- ♦ **Социологам, политикам, административным работникам** Ассоциация помогает в создании обобщенных многопараметрических “портретов” социальных групп, регионов, групп предприятий которые используются для прогнозирования результатов голосований, тенденций развития, следствий реализации различных решений.
- ♦ Помощь Ассоциации необходима в **технической диагностике, экологии, геологии, генетике**.

АСПИС обладает богатым арсеналом оригинальных высокоэффективных программных и методических средств для решения основных аналитических задач, таких как обнаружение закономерностей (knowledge discovery), классификация, диагностика, прогнозирование.

Разработанные АСПИС программные продукты превосходят по многим своим характеристикам другие известные инструментальные системы.

АСПИС объединяет творческий потенциал высококвалифицированных ученых и инженеров.

Адрес для контактов:

Институт кибернетики им.В.М.Глушкова НАН Украины
просп. акад. Глушкова, 40, 03680 ГСП Киев 187, Украина
Тел. (380+44) 526-22-60 Факс (380+44) 526-33-48
e-mail: glad@aduis.kiev.ua



Институт кибернетики им. В. М. Глушкова
Национальная академия наук Украины

DISCRET

СИСТЕМА ДИСКРЕДИЗАЦИИ ЧИСЛОВЫХ ДАННЫХ

- ◆ Система предназначена для сравнительного анализа числовых характеристик объектов разных классов, а также для преобразования числовых признаков в номинальные на этапе предобработки данных при решении дискретных аналитических задач (извлечение знаний из данных, классификация, диагностика, прогнозирование).
- ◆ Система может быть применена в различных предметных областях для:
 - * сравнительной оценки технических и рыночных характеристик изделий, выпускаемых разными фирмами;
 - * дискретизации и сопоставления физико-химических характеристик различных материалов;
 - * сопоставления числовых показателей, характеризующих предприятия или регионы в разные периоды времени и т.п.
- ◆ Система осуществляет разделение шкал числовых параметров на интервалы путем сопоставления распределений значений этих параметров, характеризующих разные классы объектов. В результате выделяются интервалы, в наибольшей мере характеризующие исследуемые классы объектов. Применение системы в целях предобработки данных в комплексе со средствами решения аналитических задач значительно повышает точность классификации, диагностики или прогнозирования.
- ◆ Система стыкуется по выходным данным с системой интеллектуального анализа данных **CONFOR**.
- ◆ Отличительной чертой системы по сравнению с другими системами аналогичного назначения является:
 - * полная автоматизация процесса дискретизации;
 - * возможность выделения характерных интервалов с учетом распределений объектов разных классов, что позволяет формировать более эффективные классификационные правила при решении дискретных аналитических задач для этих классов.

ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ для системы представляются в виде текстовых файлов.

СИСТЕМНЫЕ ТРЕБОВАНИЯ. Операционная система WINDOWS
процессор - PENTIUM
15 Мб свободного пространства на диске.

просп. акад. Глушкова, 40, 03680, ГСП, Киев-187, Украина
Тел. (380+44) 526-22-60. Факс (380+44) 526-33-48
e-mail: glad@aduis.kiev.ua



Институт кибернетики им. В. М. Глушкова
Национальная академия наук Украины

CONFOR

СИСТЕМА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

(ИЗВЛЕЧЕНИЕ ЗНАНИЙ ИЗ ДАННЫХ, КЛАССИФИКАЦИЯ,
ДИАГНОСТИКА И ПРОГНОЗИРОВАНИЕ)

- ◆ В системе **CONFOR** (CONcept FORmation) реализованы наиболее эффективные методы интеллектуального анализа данных.
- ◆ Основными функциями **CONFORa** являются:
 - * вывод закономерностей, характеризующих классы объектов, которые представлены наборами значений признаков;
 - * использование выведенных закономерностей для решения задач классификации, диагностики и прогнозирования.
- ◆ Для выполнения этих функций используются оригинальные методы извлечения знаний на основе пирамидальных сетей, многолетнее применение которых в различных сферах науки и техники подтвердило их решающие преимущества по сравнению с другими известными методами.
- ◆ С помощью **CONFORa** химики и материаловеды выполнили свыше тысячи высокоточных прогнозов новых химических соединений и материалов. **CONFOR** применяется в целях анализа ценовых ситуаций и покупательского спроса на рынке информационных технологий. Сферой применения **CONFORa** является также медицина, экономика, экология, геология, техническая диагностика, социология и т.п.
- ◆ Применение пирамидальных сетей позволяет во много раз сократить объемы поисковых операций, что дает возможность избежать эффекта «информационного взрыва» при решении аналитических задач на больших объемах данных.

- ◆ Для выполнения классификации, диагностики и прогнозирования формируются логические модели классов объектов, которые позволяют учитывать влияние на диагностируемый или прогнозируемый параметр не только отдельных наблюдаемых признаков, но и их различных сочетаний. Кроме того, учитывается влияние «исключающих» признаков, т.е. таких признаков, при наличии которых диагностируемые или прогнозируемые значения невозможны. Эти свойства моделей классов объектов способствуют повышению точности диагноза и прогноза. **CONFOR** выделяет закономерности любой логической сложности.

ИСХОДНЫЕ ДАННЫЕ для **CONFORa** представляются в виде текстовых файлов, таблиц dBASE и PARADOX.

СИСТЕМНЫЕ ТРЕБОВАНИЯ. Операционная система WINDOWS , процессор-

PENTIUM

20 Мб свободного пространства

на диске.

просп. акад. Глушкова, 40, 03680, ГСП, Киев, 187, Украина

Тел. (380+44) 526-22-60. Факс (380+44) 526-33-48

e-mail: glad@aduis.kiev.ua



Институт кибернетики им. В. М. Глушкова
Национальная академия наук Украины

АНАЛОГИЯ

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РЕШЕНИЯ ЗАДАЧ НА ОСНОВЕ АНАЛОГИИ

Комплекс **предназначен** для:

- ♣ проектирования состава сложных объектов с заданными свойствами;
- ♣ определения неизвестных свойств сложных объектов;
- ♣ поиска в базе знаний объектов, аналогичных заданным.

Комплекс может быть адаптирован к различным предметным областям путем заполнения базы знаний. **База знаний** комплекса содержит:

- ♣ описание известных свойств сложных объектов;
- ♣ описание состава сложных объектов;
- ♣ описание свойств элементов, входящих в состав сложных объектов;
- ♣ описание правил конструирования сложных объектов

Комплекс **обеспечивает**:

- ♣ наполнение базы знаний в процессе диалога с пользователем на ограниченном естественном языке;
- ♣ наполнение базы знаний из файлов фиксированного формата;
- ♣ ознакомление пользователя с содержимым базы знаний в процессе диалога;
- ♣ настройку диалога с пользователем на предметную область применения комплекса;
- ♣ решение задач на основе аналогии;
- ♣ вывод результатов решения задачи в виде отчета, подготовленного к печати, а также в виде текстового файла;
- ♣ возможность параллельной работы с несколькими базами знаний.

Комплекс позволяет производить **анализ знаний**:

- ♣ выявление и оценивание степени сходства известных объектов;
- ♣ выделение общих и специфических характеристик в описаниях объектов;
- ♣ контроль полноты и непротиворечивости.

Области применения комплекса: химия, материаловедение, генетика, экономика.



- ♣ ПЭВМ, совместимые с IBM PC;
- ♣ операционная система Windows;
- ♣ 3Mb свободного места на жестком диске;
- ♣ Borland Database Engine 5.01 или не менее 7Mb дополнительно на жестком диске.

просп. Акад. Глушкова, 40, 03680, ГСП, Киев-187, Украина
Тел. (380+44) 526-22-60. Факс (380+44) 526-33-48
e-mail: glad@aduis.kiev.ua

Научное издание

Виктор Поликарпович Гладун
ПАРТНЕРСТВО С КОМПЬЮТЕРОМ

Заведующий редакцией *А. В. Жадан*
Художественный и технический редактор *Л. Козеко*
Корректор *И. Й. Судзинская*.

Подписано к печати 01.08.2000. Формат 84x108/32. Объем 4 п.л.
Гарнитура SchoolBook. Бумага офсетная. Печать офсетная.

Тираж 1 000 экз.

Издательство «Пор-Рояль»
Киев, ул. Гарматная, 8.
Печать АТ «Академ-пресс»