

УДК 519.6
ББК 22.18
С 51

Смолин Д. В. Введение в искусственный интеллект: конспект лекций. — М.: ФИЗМАТЛИТ, 2004. — 208 с. — ISBN 5-9221-0513-2.

В работе представлены базовые модели современного искусственного интеллекта, теоретические обоснования и практически полезные примеры построения разумных систем. Изложен авторский взгляд на основные достижения и пути дальнейшего развития программ с искусственным интеллектом. Рассмотрены практические аспекты применения интеллектуальных систем в предметных областях. Работа отличается простотой изложения — многие формулы дополнены или заменены словесным описанием, что, по мысли автора, должно послужить глубокому пониманию материала.

Для студентов информационных специальностей, аспирантов и специалистов в области применения современных информационных технологий.

Учебное издание

СМОЛИН Денис Валерьевич

ВВЕДЕНИЕ В ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: КОНСПЕКТ ЛЕКЦИЙ

Редактор *Е.Н. Глебова*

Оригинал-макет: *Е.В. Третьяков*

Оформление переплета: *А.А. Логунов*

ЛР № 071930 от 06.07.99. Подписано в печать 09.08.04. Формат 60×90/16.

Бумага офсетная. Печать офсетная. Усл. печ. л. 13. Уч.-изд. л. 13.

Заказ №

Издательская фирма «Физико-математическая литература»

МАИК «Наука/Интерпериодика»

117997, Москва, ул. Профсоюзная, 90

E-mail: fizmat@maik.ru; <http://www.fml.ru>

Отпечатано с диапозитивов

в ОАО «Чебоксарская типография № 1»

428019, г. Чебоксары, пр. И. Яковлева, 15

ISBN 5-9221-0513-2



9 785922 105132

ISBN 5-9221-0513-2

© ФИЗМАТЛИТ, 2004

© Д. В. Смолин, 2004

От автора

Книга, которую вы читаете, не претендует на полноту и энциклопедичность. Ее цель — дать первоначальное представление о предмете «Искусственный интеллект», его целях, задачах, методах, основных научных школах и достижениях. Автор излагает личное видение достаточно простых, давно известных концепций и методов, составляющих основу для понимания современных достижений искусственного интеллекта. Основу книги составляют символичные модели, новейшие достижения искусственного интеллекта в работе только упоминаются.

Книга написана по материалам курса «Интеллектуализация информационных систем», прочитанного в 1997–2001 гг. в АлтГТУ для студентов специальности 351400 — «Прикладная информатика (по областям)», а также ряда смежных специальностей.

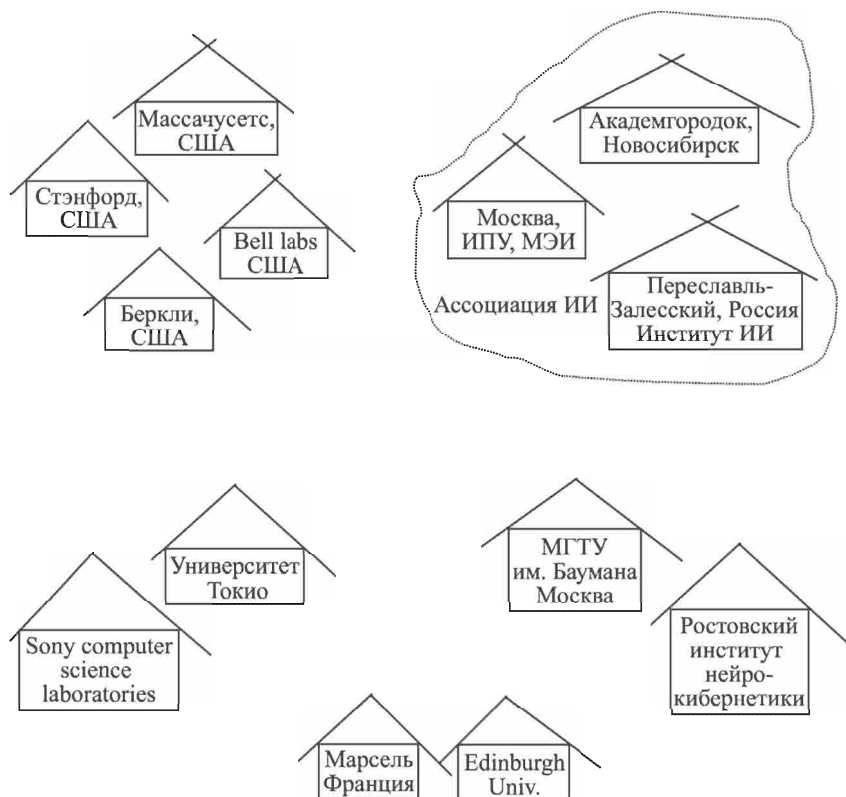
В ходе подготовки к лекциям автор активно пользовался источниками, приведенными в списке рекомендуемой литературы. Часть примеров, приводимых в тексте, дополненная и многократно опробованная на студентах, а также в практической инженерной деятельности, заимствована из этих источников.

Посвящаю работу памяти своего деда — Николая Никитовича Пинчука.

Барнаул, 1997–2003

Д. В. Смолин

Крупные центры создания искусственного интеллекта



Программа курса «Интеллектуальные информационные системы»

*Маленький одноступенчатый номер
на 6 этаже маленькой гостиницы
на La mote piquete grannelle,
шум дождя в открытом окне...*

13 февраля 2002

Примерная учебная программа разработана с учетом того, что количество часов может изменяться. Фактически в программе описаны два учебных курса.

1. «Принципы искусственного интеллекта» для специалиста-информатика.
2. «Возможности искусственного интеллекта и области его применения» для специалистов гуманитарного направления.

Ц е л ь к у р с а: подготовить обучаемого к практической деятельности в области создания, внедрения и эксплуатации систем искусственного интеллекта в качестве инженера по знаниям, пользователя или управленца (менеджера, ответственного за внедрение, рис. 1).

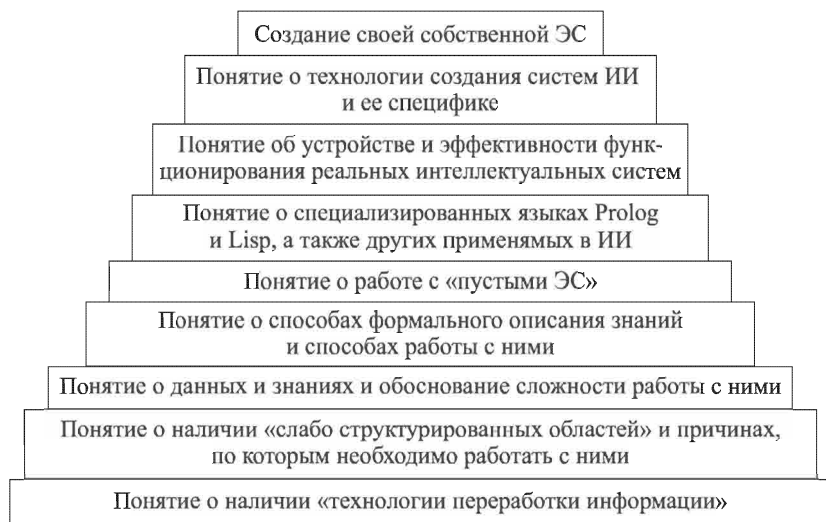


Рис. 1. Идеология учебного курса

В общей таблице темы, изучение которых является обязательным для минимального знакомства с предметом (т. е. на уровне возможностей), отмечены символом «X».

Задачи курса: рассмотреть краткую историю становления и развития искусственного интеллекта (ИИ); рассмотреть технические постановки основных задач, решаемых системами искусственного интеллекта; ознакомить с современными областями исследования по искусственному интеллекту; ознакомить с основными моделями представления знаний и некоторыми интеллектуальными системами; рассмотреть теоретические и некоторые практические вопросы создания и эксплуатации экспертных систем.

Организация практики: все занятия проводятся на компьютере; список необходимого программного обеспечения и копии программ — на отдельном компакт-диске.

Примерное содержание программы

Тема 1. Введение: краткая история и терминология.

Философские, технические, научные предпосылки для создания искусственного разума. История развития информационных технологий. Современные представления о структуре и функционировании человеческого мозга. Сила и слабость человеческого разума. Области разделения обязанностей между человеком и машиной. Возможность конфликта между человеком и машиной с особым вниманием не на философию, а на техническую точку зрения. Способы избежания конфликта, смягчения его. Практические примеры. Философы и поэты, работавшие над этими проблемами, и квинтэссенция их основных произведений. Данные и знания. Определения, интуитивные примеры, проблемы работы с данными, проблемы работы со знаниями. Выгоды, получаемые при использовании знаний. Единство и неразрывность данных и знаний. Основные термины и определения, относящиеся к ИИ.

Практика: «Правильная постановка целей».

Тема 2. Теоретические задачи, решаемые ИИ. Инженерные задачи, решение которых требует применения методов ИИ. Математическое описание инженерных задач — постановка абстрактных задач: выбор, поиск пути, генерация альтернатив, классификация. Абстрактные модели решения задачи: разновидности моделей, выбор наиболее подходящей для данной задачи.

Тема 3. Области практического применения методов ИИ.

Хорошо и плохо структурированные предметные области. Эффективность решения практических задач методами ИИ и критерии

Примерный тематический план курса

№	Тема	Примерное кол. часов	Прим.
1.	Введение: краткая история и терминология	2	Х
2.	Теоретические задачи, решаемые ИИ Практика: «Правильная постановка целей»	2 2	Х
3.	Области практического применения методов ИИ Тестирование: «Терминология и области применения ИИ» Тестирование: «Направления современных исследований в ИИ»	2 1 1	Х
4.	Модели представления знаний. Общий обзор	2	Х
5.	Логика Практика: «Применение элементов формальной логики для управления техническими устройствами» Тестирование: «Основы формальной логики»	2 2 1	Х
6.	Правила продукций	2	Х
7.	Языки описания продукционной модели Prolog и Lisp Практика: «Программирование на языке Prolog» Практика: «Программирование на Lisp» Тестирование: «Основы языка Prolog»	2 2 2 1	
8.	Семантические сети Практика: «Построение СС на основе фрагмента текста» Тестирование: «Основы теории множеств»	2 1 2	Х
9.	Фреймы Практика: «Создание фрейма — визуального образа» Практика: «Создание не визуального фрейма» Тестирование: «Основы объектной модели»	2 1 2 2	Х
10.	Новые модели представления знаний. Общий обзор	2	
11.	Модели теории принятия решений: критериальный и вероятностный подходы. Экспертные методы. Нейронные сети Практика: «Практическое применение отдельных экспертных методов»	2 2	
12.	Экспертные системы. Общий обзор Практика: «Работа с оболочкой ЭС» Практика: «Разработка простейшей ЭС на инструментальном языке» Тестирование: «Принцип работы, достоинства и недостатки отдельных ЭС»	2 2 1	
13.	Технология разработки экспертных систем. Тестирование: «Различия в разработке ЭС и традиционных программ»	2 1	
	Написание и защита рефератов	2	
	Итого часов	53	30

измерения эффективности. Принципы эффективного применения методов ИИ.

Тестирование: «Терминология и области применения ИИ».

Тестирование: «Направления современных исследований в ИИ».

Тема 4. Модели представления знаний. Общий обзор.

Общая схема моделей представления знаний. Основные сведения об основоположниках. Краткие исторические справки о развитии моделей. Основные решаемые задачи, область применимости и эффективность, опыт и специфика эксплуатации, примеры отдельных реальных систем, созданных на базе этих моделей, инструментальные средства для работы с этими моделями. Современные мировые модели-лидеры и причины их лидерства. Перечень ключевых публикаций.

Тема 5. Логика.

Краткая история развития логики от Аристотеля до нашего времени. Основные школы и решаемые ими задачи, система логических обозначений. Способы формальной записи логических выражений и правил. Технические приложения теории логики: практические примеры, достоинства и недостатки. Способы логических рассуждений и способы оценки истинности формул. Классификационные системы. Доказательства и софизмы. Логический квадрат. Логические имена. Исторические примеры применения логики и иллюстрации к самому процессу ее развития. Рассуждения с нечеткими и неясными именами. События, характеризующиеся вероятностью, и события, характеризующиеся степенью уверенности. Таксономические модели и история их развития, примеры таксономий. Математическая теория нечетких множеств. Примеры практических задач.

Практика: «Применение элементов формальной логики для управления техническими устройствами».

Тестирование: «Основы формальной логики».

Тема 6. Правила продукций.

Сложность («объемность») расчетов при выводе на основе уравнений Дж. Буля. Принцип резолюций как способ сократить количество уравнений. Представление задачи в виде «И/ИЛИ» графа. Другие способы «оптимизации затрат» на логический вывод при наличии большой системы уравнений: стратегия вывода и «бэктрекинг», способ алгоритмической организации «бэктрекинга». Запись «пути решения» задачи и методы хранения баз знаний (реляционная). Полезные эвристики для организации перебора. Заикливание и алгоритм борьбы с ним. Метод встречной волны. Метод ветвей

и границ. Метод ограничения числа дочерних вершин, метод динамического программирования и др. Наиболее известные задачи: «обезьяна и банан», «родственные отношения», «поиск кратчайшего пути». Методы построения программ на основе данной теории и специализированное аппаратное обеспечение: символьные машины и подробное описание механизма работы продукционной машины вывода на примере оболочки экспертной системы (например, «Экспр»).

Тема 7. Языки описания продукционной модели Prolog и Lisp.

Назначение, краткая история развития и современное состояние. Синтаксис. Сравнительный анализ. Примеры решения задач. Примеры типичных ошибок и затруднений.

Практика: «Программирование на языке Prolog».

Практика: «Программирование на языке Lisp».

Тестирование: «Основы языка Prolog».

Тема 8. Семантические сети.

Представление СС в виде графа с циклами. Теорема о возможности развязывания любого полносвязного графа в дерево. Определение СС. Очень краткая история развития. Типы узлов и типы отношений (теории категорий Канга, Локка, Бэкона, Аристотеля, современная теория лингвистики и ее авторы). «Поверхностность» и «глубинность» знаний как основные отличия модели СС и продукционной. Примеры «поверхностного» и «глубинного» описаний одной и той же задачи и указание областей применения поверхностных и глубинных знаний. Классификация СС. Предметные области, в которых СС получили распространение. Примеры. Достоинства и недостатки. Методы и алгоритмы вывода на СС. Основы теории множеств для описания СС.

Практика: «Построение СС на основе фрагмента текста».

Тестирование: «Основы теории множеств».

Тема 9. Фреймы.

Определение. История появления. Решаемые задачи. Практические системы, созданные на основе фреймов. Основные направления совершенствования сетевой модели. Фрейм как идеальный метод для описания внутренней структуры узлов сети. Определение. История появления. Типы фреймов и свойства фреймов (наследование, инкапсуляция, полиморфизм). Системы фреймов. Представление знаний об объекте при помощи фреймов, примеры. Примеры фреймов, применяемых в инженерной практике: объекты языков программирования, а также невизуальные фреймы, абстрактные фреймы-

образцы. Преимущества и недостатки фреймовой модели. Объектно-ориентированные языки программирования. Понятия об объектно-ориентированном анализе предметной области. Методы хранения объектных баз данных. Инструментальные средства описания и вывода на фреймовой модели.

Практика: «Создание фрейма — визуального образа».

Практика: «Создание фрейма — не визуального образа».

Тестирование: «Основы объектной модели».

Тема 10. Новые модели представления знаний. Общий обзор.

Причины неудовлетворенности возможностями «традиционных» моделей ИИ. Новые задачи, решение которых невозможно при помощи «традиционных» методов. Примерный перечень «неклассических» моделей ИИ и решаемых ими задач. Прагматический и идеалистический подходы к созданию советующих систем. Теория принятия решений как метод, позволяющий получать решения в слабо структурированных областях.

Тема 11. Модели теории принятия решений: критериальный и вероятностный подходы. Экспертные методы. Нейронные сети.

Принятие решений: определение выбора; языки описания задач выбора (критериальный, бинарные отношения, статистика (многомерный статистический анализ, многомерное шкалирование) и др.). Вероятностные методы осуществления выбора. Гибридные модели. Методы получения экспертных оценок. Персептроны. Нейронные сети как основной тип современных моделей ИИ.

Практика: «Практическое применение отдельных экспертных методов».

Тема 12. Экспертные системы. Общий обзор.

Необходимость ЭС в практических задачах человеческой деятельности. Определение ЭС. История развития и области применения. Задачи, решаемые ЭС. Технология применения ЭС и ее отличие от технологии применения «обычных» программ. Критерии необходимости применения ЭС. Типичные состав и структура ЭС. Языки представления знаний. Классификация знаний по глубине и жесткости. Классификация ЭС и современные тенденции в их развитии. Примеры практических ЭС (подробно): MYCIN, ELIZA, PROSPECTOR, CASNET, EXPOB, ESISP.

Практика: «Работа с оболочкой ЭС».

Практика: «Разработка простейшей ЭС на инструментальном языке».

Тестирование: «Принцип работы, достоинства и недостатки отдельных ЭС».

Тема 13. Технология разработки экспертных систем.

Этапы разработки ЭС и их отличие от разработки «обычного» ПО. Работа инженера по знаниям. Получение знаний. Выбор модели представления знаний. Коллектив разработчиков. Особенности разработки ЭС.

Тестирование: «Различия в разработке ЭС и традиционных программ».

Темы для рефератов.

1. Новые модели для представления знаний.
2. Практические примеры ЭС.
3. Интерфейсы ЭС.
4. Программные средства для построения ЭС.
5. Методы сортировки (сравнение текстовых строк, сравнение предложений).
6. Примеры задач, эффективно решаемых на Prolog или Lisp.
7. История развития искусственного интеллекта.
8. Сравнительный анализ эффективности различных моделей представления знаний (на примере прикладной задачи).
9. Современная литература по ИИ (обзор литературы за последние 3 года: области исследований, персоналии, прикладные системы, цели и задачи исследований).

ГЛАВА 1

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ КАК ВЕРШИНА РАЗВИТИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1.1. Предыстория теории ИИ

Искусственный интеллект, как и любое направление современной науки, имеет богатую предысторию. Знакомство с именами и трудами основоположников, их научными воззрениями, даже биографиями ученых имеет практический смысл, т. к. во многих случаях позволяет глубже понять смысл той или иной научной модели.

Широкое развитие автоматизированных технологий (см. табл. 1) началось, по-видимому, в 40–50 годах ушедшего XX века. Философская идея о замене человека умной машиной, возникшая в книгах Шелли¹ (если только не раньше), получила наибольшее распространение в 60–80 годах. Основные теоретические положения искусственного интеллекта, а искусственный интеллект сегодня — вершина развития информационных технологий, заложены как раз в этот период. Сказанное не означает, что сегодня исследования ведутся менее активно, напротив, наблюдается бум развития компьютерных технологий, в их разработку вкладываются деньги, сопоставимые с бюджетом ресурсодобывающих компаний. Тем не менее результаты скорее количественные. Впрочем, возможно, что современные достижения в этой области не рекламируются, поскольку имеют явную военную направленность.

Серьезных изменений в теории искусственного интеллекта следует ожидать, по-видимому, в 10–20 гг. нашего века. Это обусловлено необходимостью обобщения накопленных научных знаний, появления некоторой единой теории искусственного интеллекта.

История развивается циклично, и новый век, тем более начало нового тысячелетия, — это своеобразный старт для народов, государств, политиков, ученых.

Свидетелями каких представлений на сцене истории мы станем? Какое государство определит лицо современного мира? Какой ресурс станет главным для мировой экономики? Сегодня это не знает никто. Возможен только более–менее точный прогноз, и он состоит в том, что одним из самых важных, самым важным после нефти и угля, станет информационный ресурс, т. е. информация плюс технология ее обработки.

¹ Перси Шелли — английский поэт, брат Байрона. Идея о Франкенштейне принадлежит его жене Марии.

Т а б л и ц а 1
Этапы развития ИТ

Название этапа	Решаемые задачи	Ученые и инженеры	Техника
Домашний (XVII в.)	Автоматизация расчетов (в основном арифметических)	Готфрид Лейбниц, Блез Паскаль	Арифмометр, логарифмическая линейка, круг и т. п.
Механические и механико-электрические вычислительные машины (XIX в. — середина XX в.)	Учет (учет товаров, перепись населения, наука) Шифрование сообщений	Чарльз Бэббидж и Ада Лавлейс, Герман Голлерит, ² П. Чебышев	Машина Голлерита («Голлерит»), Арифмометры, в том числе с электроприводом, немецкая шифромашина «Энигма»
Первые ЭВМ (начиная с 40-х гг. XX в.)	Перебор значительного количества вариантов при относительно небольшом объеме вводимых данных Ввод, хранение, простейшая обработка значительных объемов данных	Фон Нейманн (теоретические основы) Эдгар Кодд (реляционные БД, 60-е гг.)	ENIAC (США, расчеты для ядерной программы) Мейфреймы (разные модификации) ЕС ЭВМ (70-е гг.)
Первые управляющие вычислительные машины (начиная с 50-х гг. XX в.)	Управление параметрами функционирования простейших технических объектов Управление параметрами функционирования больших технических систем, контроль за значениями параметров	Н. Винер (теоретические основы), Клод Шеннон (инженерная реализация) Д.А. Поспелов	Встроенные ЭВМ, иногда аналоговые — первые бортовые РЭС, системы пуска космических ракет (СССР) и др., релейные линии связи
Управляющие вычислительные комплексы (70-е гг. XX в.)	Автоматическая диагностика неисправностей в больших технических системах Управление функционированием больших технических систем, выбор не только параметров, но и режимов	Р. Белман Л.С. Поитрагин М. Мински	Измерительно-вычислительные комплексы на базе СМ ЭВМ (PDP-11) Минатомпрома, Миннавиапрома (СССР), Системы автоматизированного проектирования и гибкого производства, включая микро-ЭВМ, «беззольные заводы»
Персональные компьютеры, а также большие ЭВМ (70-е гг.; широкое применение с конца 80-х гг. XX в.)	Сложный анализ многомерных данных, поиск причин произошедшего события, объяснение этих причин с учетом их последствий Поиск путей выхода из сложившейся ситуации: генерация альтернатив, оценка альтернатив и выбор наилучшей	Аристотель Дж. Буль Л. Заде М. Мински С. Пейперт Р. Кини, Х. Райфа Р. Ковальски и др.	Микро-ЭВМ PDP-11, IBM/370, новые модификации СМ ЭВМ Серия ПК класса IBM 286, 386 и т. д.

² В конце XIX в. в Америке основал фирму, от которой произошла современная фирма IBM

1.2. Этапы развития программных средств

Необходимо отметить, что представленные этапы (см. табл. 2) не отменяют и не заменяют один другой. На каждом из них сформировалась своя специфическая технология обработки информации (например, технология баз данных). Каждая из них живет, развивается и служит основой для технологии следующего уровня.

Таблица 2

Некоторые этапы развития программных средств

Технология	Задачи
Технология программного управления физическим уровнем ЭВМ и внешних устройств	Разработка операционных систем Эвристическое программирование ³
Программно-аппаратная оптимизация функционирования физического уровня ЭВМ	Совершенствование операционных систем, разработка специальных утилит
Технология логической организации хранимых данных	Разработка файловых систем данных
Технология баз данных и вычислений	Разработка стандартных форматов и методов доступа к данным, численные алгоритмы
Технология искусственного интеллекта	Разработка универсальных методов сравнительного анализа хранимых данных, поиска противоречий, причин их появления и методов устранения

Каждая технология породила свои специфические понятия, термины и методы. Технология следующего уровня чаще всего их заимствует, однако возникают и новые понятия, образующие новую теорию. Для искусственного интеллекта таким понятием является понятие о знаниях.

Таким образом, Вы видите, что искусственный интеллект — новейшая информационная технология, имеющая специфический круг задач и базирующаяся на ранее существовавших технологиях.

³ Эвристика — правило, теоретически не обоснованное, но позволяющее сократить количество переборov в пространстве поиска. Эвристическое программирование — разработка стратегии действий на основе известных, заранее заданных правил.

1.3. Понятие «Искусственный интеллект»

О п р е д е л е н и е 1. *Интеллектуальной* называется система (рис. 2), способная целеустремленно⁴, в зависимости от состояния информационных входов, изменять не только параметры функционирования, но и сам способ своего поведения, причем способ поведения зависит не только от текущего состояния информационных входов, но также и от предыдущих состояний системы.

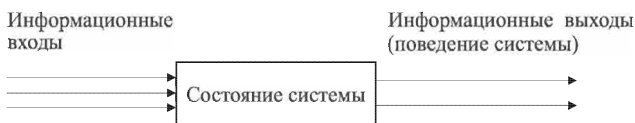


Рис. 2. Интеллектуальная система как «черный ящик»

Приведем несколько примеров.

Любой живой организм — интеллектуальная система. Он обладает долговременной памятью и способностью к самообучению. Ребенок, притронувшись к горячей плитке, уже не повторит ошибки. Щенок, впервые погнавшийся за кошкой, получит серьезный урок и вряд ли снова решит с ней поиграть. При следующей встрече он, скорее всего, убежит или покажет зубы, или проявит еще одну из тысяч возможных реакций.

Технические же системы чаще всего не являются интеллектуальными, т. е. их реакция на одно и то же событие не может измениться кардинально. Система автоматического управления давлением газа в трубе может открывать и закрывать заслонку (управлять параметрами), но она не может принять решение совсем вывинтить заслонку из трубы. Если аварии газопровода предшествует изменение давления (например, сначала резкое повышение, а затем резкое понижение), то автоматическая система воспримет это как нормальную ситуацию и попытается «отрегулировать» ее движением заслонки. Даже если после каждой аварии мы будем добавлять в систему управления новый блок, точно фиксирующий параметры предыдущей ситуации, ничего не изменится. Простое накопление данных не «обучит» систему.

Дело в том, что щенок, получивший урок от кошки, запомнил не только параметры ситуации (длину когтей и скорость реакции), но и правила поведения (не подходи, не подставляй нос, если лаять — она убежит).

⁴Различают целенаправленные и целеустремленные системы. Примером системы первого типа может служить артиллерийский выстрел, второй — самонаводящаяся ракета.

О п р е д е л е н и е 2. *Интеллектуальной* называется система, моделирующая на компьютере мышление человека.

Второе определение появилось в 60-х гг., когда считалось, что мозг человека можно смоделировать на компьютере. Клетки мозга — нейроны — программно описывались специальными математическими методами. Компьютерная программа, таким образом, представляла как бы кусочек мозга человека. На вход программы подавались некоторые данные (на вход клетки мозга в живом организме подается электрический сигнал), на выходе снимались результаты, которые сверялись с эталоном. В зависимости от того, насколько полученные результаты отклонялись от эталона, в расчетные коэффициенты вносились изменения. В зависимости от количества циклов такого «обучения» результаты работы программы постепенно все более приближались к результатам работы очень маленького элемента мозга человека.

Идея о возможности повторить мозг на компьютере к 80-м гг. XX в. потерпела полную неудачу, однако теория нейронных сетей, нейросетевой подход доказали свою полезность на целом ряде практических приложений. Хорошие результаты получены в первую очередь на задачах предсказания значений параметров и распознавания образов.

О п р е д е л е н и е 3. *Интеллектуальной* называется система, позволяющая усилить интеллектуальную деятельность человека за счет ведения с ним осмысленного диалога.

К концу 80-х гг. стало совершенно очевидно, что создать универсальный искусственный разум невозможно. Более того, выяснилось, что это совершенно не нужно. Следует создавать узкоспециализированные интеллектуальные системы, которые не заменяют человека, но дополняют его⁵. Человек имеет ряд уникальных способностей, но не свободен от недостатков. Не один человек не обладает реакцией кошки. Никто из нас не способен прочитать за минуту роман Л. Н. Толстого «Война и мир», редко кто из людей обладает энциклопедической памятью.

Компьютер обладает энциклопедической памятью, компьютер совершает миллионы операций в секунду, компьютер реагирует мгновенно.

⁵ Обратите внимание на новый акцент в постановке задачи создания ИИ. Если изначально выдвигалось требование к машине «мыслить», то теперь — «получать хорошие результаты». Другими словами, произошел переход от моделей, воспроизводящих процесс мышления человека или структуру головного мозга, к моделям, использующим какие-либо собственные принципы организации и методы обучения, но позволяющим получать результаты, «похожие на человека».

венно. Но компьютер — это просто «глупое железо», он не способен мыслить, не способен отвечать за собственные поступки.

Подчеркнем актуальность совместной деятельности человека и машины. Необходимо, чтобы компьютер служил советчиком человеку, быстро анализировал ситуацию, генерировал варианты действия на основе огромной памяти и предлагал их человеку, а человек рассматривал предложенные варианты и пояснял, почему тот или иной вариант плох. Компьютер, учитывая полученные разъяснения, вновь анализировал бы все варианты действия и выдавал новые, а человек выбирал подходящий вариант и нес ответственность за его реализацию.

Поясним примером: система автоматического наведения ракет обнаружила цель. Цель была обнаружена практически мгновенно, человек даже не успел ее заметить. Ракета была автоматически наведена на цель. Цели был послан запрос «свой—чужой». Цель появилась на пульте управления перед оператором, человек принял решение о поражении, выбрал тип оружия и нажал на кнопку «уничтожить». В случае полностью автоматического ведения цели существовала бы реальная опасность уничтожить свой самолет. В обратном случае, если бы наведением на цель, посылкой запроса занимался человек, могло быть упущено время.

Таким образом, сегодня искусственный интеллект — это самообучающийся инструмент, усиливающий деятельность человека по генерации и принятию решений.

1.3.1. Термины и определения. Создание реальных интеллектуальных программ, а такие программы сегодня уже широко распространены, столкнулось с рядом непреодолимых трудностей теоретического характера. Интеллектуальную программу невозможно создать, пользуясь только теорией процедурного программирования и баз данных. Ваша программа не станет интеллектуальной, если начнет заносить в свою базу данных все ситуации, с которыми она встречалась. Постоянно пополнять базы данных можно, и интеллектуальные программы это делают, однако это далеко не все.

Примечание⁶. В конце 40-х гг. была поставлена задача автоматического перевода с одного языка на другой, например, с русского на английский. Перед Вами сочетание: Read Only Memory. Здесь Read — читать, Only — только, Memory — память. Как правильно перевести эту фразу?

⁶ Знаменитый «джорджтаунский эксперимент» (07.01.1954; нью-йоркская контора IBM, электронная счетная машина IBM-701). Перевод 60-ти предложений типа «Обработка повышает качество нефти» с русского языка на английский.

1. С точки зрения компьютера следовало бы заменить английские слова русскими просто в порядке их следования, тогда получим: «Читать только память».
2. С точки зрения человека–переводчика следовало бы учесть грамматику английского языка, начать перевод с конца фразы и согласовать элементы: «Память только для чтения».

Очевидно, кроме непосредственно исходных данных (в данном случае это слова «читать, только, память») необходимо учитывать также и правила перевода. А что такое правила перевода? Как отразить их в программе-переводчике? Вот тут и возникает новое понятие — знания.

Знания — основной термин теории искусственного интеллекта. Очень интуитивно и довольно спорно знания можно определить как отношения между элементами данных. Например, машина получила на вход предложение: «Вася любит Катю». Здесь Вася и Катя — данные. Любит — отношение. Отношение «любит» не обладает симметричностью. Зная это, машина может сгенерировать вполне осмысленный ответ: «Совсем не обязательно, что Катя любит Васю».

Вернемся к примеру с переводом. Представим его графически (рис. 3). Вы видите, что схема перевода позволяет точно определить правильное направление перевода. Она неизменна для русского и английского вариантов. Вот эта схема, получаемая из каких-то внешних источников, не содержащаяся в исходной фразе, и есть знания.

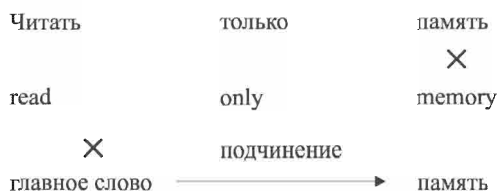


Рис. 3. Отношения между элементами данных

Другими словами, знания получаются в результате применения к исходным данным (их могут называть декларативной информацией⁷) некоторых методов обработки, подключения внешних процедур (их еще называют присоединенными процедурами или процедурной информацией⁸). Необходимость установления отношений между сущностями требует совершенно нового подхода и, соответственно, новых программных средств. Работа программы искусственного

⁷ to declare — объявлять, заявлять (англ.)

⁸ procedure — процедура (англ.)

интеллекта заключается в «выводе на знаниях» или «выводе новых знаний». Этот процесс схематично представлен на рис. 4.

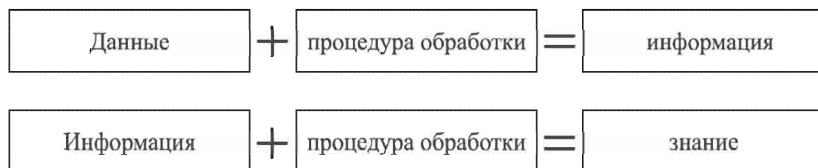


Рис. 4. Вывод на знаниях

Рассмотрим пример. Человек смотрит на часы. Что он видит? Данные? Информацию? Знания? Проследим, насколько это возможно, действия человека. Итак, глаза смотрят на циферблат. В мозг поступает электрический сигнал, в мозге формируется изображение стрелок на циферблате. Далее, сознательно или подсознательно человек прикладывает некоторые умственные усилия, чтобы понять (по положению стрелок) сколько же сейчас времени (т. е. соотносит полученные данные с некоторой шкалой). Получив в итоге декларативную информацию, например 17:20, человек подключает внешние знания, например свое рабочее расписание, понимает что опаздывает (знания) и ускоряет шаг (меняет параметры своего поведения, т. е. данные).

Характерная особенность знаний состоит в том, что они не содержатся в исходной системе. На циферблате часов не было написано «опаздываю». Слово «опаздываю» не содержалось и в расписании этого человека. Знания возникают в результате сопоставления информационных единиц, нахождения и разрешения противоречий между ними. Т. е. знания активны, их появление (или недостатка) приводит к реализации некоторых действий.

Для знаний характерны следующие свойства:

- внутренняя интерпретируемость (каждая информационная единица должна иметь уникальное имя и однозначно определяться);
- структурированность, т. е. между информационными единицами должны быть установлены отношения (например, «часть – целое», «род» – «вид» и др.); при этом возможна рекурсивность;
- знания образуют некоторое пространство, которое может оказаться как метрическим, так и не метрическим.

Установление имен для информационных единиц, само выделение этих единиц на конкретном объекте, определение отношений, имеющих место между единицами, — сложная задача. Для реально существующих объектов с относительно простой структурой иногда возможно единственное разбиение на информационные единицы

и отношения. Например, для системы «собака — хвост» трудно придумать иную модель кроме «хвост является частью собаки».

Для сложных систем, и в первую очередь абстрактных, не всегда можно сразу понять, где «целое» и где «часть» и как они между собой соотносятся. Рассмотрите, например, систему «курс доллара», для которой неясно, что входит в список важных параметров, а что нет. Или систему «жук», у которого нет отдельно головы и отдельно груди, а есть «головагрудь». Попробуйте однозначно объяснить многозначное отношение «задел» (А задел В) применительно к разным ситуациям.

Решением этих проблем занимается отрасль ИИ «инженерия знаний». Далее мы последовательно уточним эти вопросы.

Состояние — второй важнейший термин. Каждая информационная единица, как и вся система в целом, может находиться в одном из состояний. Например, лампочка может находиться в одном из состояний: включен — выключен. В данном случае мощность пространства состояний составляет 2 (см. рис. 5).



Рис. 5. Пространство состояний системы и таблица переходов

Есть системы, для которых пространство состояний бесконечно.

Пространство состояний удобно представлять в виде графа (рис. 5) или таблицы переходов.

Переход системы из состояния в состояние может обладать свойством симметричности. В этом случае система способна вернуться в исходное состояние (а компьютерная программа — заикнуться).

Цель — основная характеристика системы ИИ. Система искусственного интеллекта (интеллектуальная программа) отличается от традиционных программ свойством целеустремленности. Интеллектуальная программа в отличие от обычной не содержит алгоритм, а содержит только цель, которую нужно достичь, и правила движения к этой цели. Можно сказать, что программа ИИ — это программа «что нужно сделать», а традиционная — «как нужно делать». В традиционной программе алгоритм достижения цели задан изначально и выполняется с неизбежностью, в интеллектуальной программе алгоритм строится по ходу достижения цели с учетом складывающейся ситуации.

Для программиста, работающего на системы ИИ, очень важно уметь правильно ставить цели. От того, насколько точно и экономич-

но будет поставлена цель, зависит не только эффективность решения, но и ее принципиальная достижимость.

В качестве целей в интеллектуальных программах чаще всего задаются требования перехода системы в заданное состояние, а в качестве критерия эффективности — число шагов решения. Очевидно, достижение цели не всегда возможно без указания начального состояния (рис. 6).



Рис. 6. Пространство состояний с циклами

Модель представления знаний — основной тип моделей ИИ. Реализация конкретных систем, основанных на знаниях, происходит в рамках одной из моделей представления знаний или языка представления знаний. Таких моделей немного и ниже мы их рассмотрим. Реальные системы редко основаны на одной из моделей в ее классическом понимании. Чаще всего реальная система — это гибрид из классических моделей со значительной долей собственных догадок, ограничений и условностей. Наиболее полезные из них называют эвристиками.

Предметная область — это область человеческой деятельности, к решению задач которой применяется теория ИИ. Например, если Вы создаете экспертную систему, которая по набору результатов анализов ставит диагноз больному, то Ваша предметная область — медицина. Если Вы создаете интеллектуальную систему предотвращения столкновения кораблей, то предметная область — навигация, и т. д. Создание интеллектуальных программ, действительно полезных для практикующих специалистов, требует глубокого понимания специфики, присущей каждой предметной области⁹. Для практикующих специалистов «специфика» — это в наибольшей мере набор нечетных и неясных, выработанных годами правил разрешения сте-

⁹ В реальных задачах на построение интеллектуальных систем предметная область описывается гораздо хуже. Мир робота-сварщика может состоять, например, из стационарно закрепленных датчиков положения (границы мира), а область может ограничиваться: «электросварка стали толщиной 0,8–0,9 мм швами длиной 10–15 мм» и т. п.

реотипных ситуаций. Одна из первых задач инженера по знаниям (специалиста по ИИ) состоит как раз в том, чтобы формализовать (записать) эти правила. Проблема на этом этапе состоит в том, что предметные специалисты (эксперты) часто обладают только интуитивным видением предмета обсуждения и формально не способны обосновать свою точку зрения¹⁰. Попытка «выведать» подсознательные причины уверенности эксперта в некотором факте (например, «почему температура живого человека не может превысить 43°C?») часто вызывает недоумение и даже досаду. У эксперта начинает складываться мнение, что ему «подсунули некомпетентного в наших вопросах человека, и какую, собственно, «умную» программу он может написать, если не отличает крови от плазмы!» Поэтому на первом этапе, выделяя те самые важные понятия — «имена» предметной области, которые в дальнейшем предстоит детально изучить, инженер по знаниям должен продемонстрировать еще и способности тонкого психолога, способного «влезть в мозги» эксперта, не вызвав при этом резкой ответной реакции.

На втором этапе следует выделить отношения, которые связывают «имена», важные для данной предметной области, и состояния, в которых эти «имена» могут находиться. Например, для системы предотвращения столкновения кораблей важными именами будут: «вода», «корабль», «столкновение» и др. Имя «вода» будет иметь несколько состояний: «высокая», «мелкая», «спокойная», «бурная» и ряд других гораздо менее очевидных. Между именами «вода» и «корабль» будет находиться отношение «погружен». И вот тут то и возникает проблема непонимания. Что значит «погружен»? А если судно на воздушной подушке? А если это подводная лодка? Существует большое количество предметных областей (медицина — одна из показательных), где предметные специалисты уже не способны ответить на эти вопросы. Их образование не предусматривает детального рассмотрения основополагающих понятий и отношений, а требует только их знание и принятие «на веру». Важным также является вопрос о «силе» отношения: «насколько погружен?», «еще плавает или уже тонет?» и т. п.¹¹

За годы профессиональной деятельности специалист по ИИ сталкивается с десятками предметных областей и экспертов. Ясно, что он не может стать равным им врачом, навигатором, военным, игроком и т. п. Равно как и эксперты в предметной области, пусть даже компетентные еще и в вопросах программирования и разработки

¹⁰ Т. е. математически (логически) правильно, полно и однозначно.

¹¹ Весь этот процесс называют «извлечение знаний».

традиционных программ, не способны создать программу ИИ. Исюмина деятельности инженера по знаниям заключается не в личном детальном знании предметной области и современных способов создания программ, а в способности выделять общее в казалось бы различных областях, строить универсальные математические, а при невозможности — нечеткие логико-лингвистические модели.

Невозможность (в чистом виде) применить одну из ранее известных моделей — одно из основных ограничений, накладываемых предметной областью.

1.3.2. Современные области исследований в ИИ. В последнее десятилетие ушедшего века ясно обозначились следующие важные направления развития интеллектуальных систем.

1. Системы, имитирующие творческие процессы. Создание музыкальных произведений, решение игровых задач (шахматы, шашки, домино), автоматический перевод, доказательство теорем, распознавание образов, имитация мышления и т. п.
2. Информационные системы, основанные на знаниях (экспертные системы), т. е. консультирование малоопытных пользователей, настройка оборудования, обучение и др.
3. Интеллектуальные информационные системы — большие и очень большие программы, предназначенные для решения задач в предметной области на основе математических и алгоритмических моделей и обладающие способностью вести осмысленный диалог с пользователем с целью упростить управление, сократить объем работы человека, повысить качество и т. п.
4. Робототехника. С точки зрения «интеллектуальности» различают несколько поколений роботов. Первое поколение — роботы-манипуляторы, действующие по заранее утвержденной и неизменной программе (например, подающие заготовки к станку). Если в процессе работы по каким-либо причинам изменится расстояние до заготовки, робот ее потеряет. Второе поколение — адаптивные роботы. Члены таких роботов оснащены большим количеством датчиков: угломеров, тензометров, газовых анализаторов и др. Такие роботы применяются, например, для сварки кузовов автомобилей. Дело в том, что достаточно крупные технические изделия, даже выпускаемые серийно, все равно получаются уникальными. Длина двух автомобилей одной и той же марки различна. Адаптивный робот делает сварной шов не в абсолютных координатах, а относительно некоторой точки начала отсчета, которую сам же и находит на кузове автомобиля. При необходимости человек-сварщик берет управление

на себя, а робот, запоминая его действия в новой ситуации, «обучается». Первые два вида роботов — промышленные роботы, сконструированные для работы в заранее определенной среде — в цехе завода. Для ориентации и точного позиционирования в пространстве в цехе имеется большое количество контрольных точек — датчиков, координаты которых неизменны. Гораздо сложнее роботу, который находится в реальном мире, например, обследует затонувший корабль. Приходится отвечать на вопросы: «где я?», «что передо мной?», «можно ли пройти к?» и др. Интеллектуальным роботам приходится обрабатывать данные своих датчиков и команды человека в реальном масштабе времени, а если предусмотрен еще и прогноз событий, то в «сверхнатуральном» (применяются, например, для управления транспортными средствами). Схожие проблемы возникают у поисковых роботов — особого класса программ, предназначенных для индексирования документов в глобальной сети Интернет, что обеспечивает работу поисковых машин.

1.3.3. Современные теоретические проблемы ИИ.

1. Проблема представления знаний.

1.1. Проблемы нейронных сетей.

1.2. Многокритериальное принятие решений.

1.3. Стохастические модели принятия решений.

1.4. Создание моделей для очень узких предметных областей.

2. Разработка компьютерной лингвистики.

2.1. Разработки новых, более надежных языков программирования.

2.2. Разработка языка управления роботами на основе естественного языка.

3. Компьютерная логика.

3.1. Совершенствование архитектуры ЭВМ.

Для того чтобы получить более подробное представление об основных практических и теоретических задачах, решаемых сегодня в ИИ, следует ознакомиться с примерными планами работ, публикуемых на сайтах общеизвестных мировых центров, таких как, например, Лаборатория искусственного интеллекта Массачусетского технологического института (<http://www.ai.mit.edu>). В 2001 г. лаборатория занималась примерно следующим.

1. Проблема представления знаний.

1.1. Разработка новых моделей представления для узкоспециализированных предметных областей.

1.2. Биомашинны — машинны, имеющие своей частью живые существа либо структурно подражающие человеку:

1.2.1. подражание моторике человека (походка, пластика, бег, прыжки, создание двуногих роботов);

1.2.2. создание инженерных моделей для различных областей по аналогии с системами человеческого организма.

1.3. Многокритериальное принятие решений.

1.4. Принятие решений на основе статистических моделей.

1.5. Координация работы нескольких роботов.

1.6. Проблемы совершенствования нейронных сетей.

2. Проблема совершенствования компьютерной логики.

2.1. Разработка новых архитектур компьютеров (параллельные машинны, исследования в области так называемой интегрированной памяти, децентрализованные машинны, моделирование высокоскоростных электрических соединений).

2.2. Человекообразные роботы:

2.2.1. гибкие и портативные члены роботов (головы, руки, тела и т. д.);

2.2.2. распознавание роботами лиц, авторизованных для управления роботами;

2.2.3. разработка механизмов роботов (человекоподобный палец с сенсорами, человекоподобная модель мускулатуры, говорение роботов, создание роботов для детей);

2.2.4. численные методы для оптимизации вычислений.

2.3. Методы доступа к информации:

2.3.1. мультимедийные системы;

2.3.2. эвристический анализ текстов;

2.3.3. автоматическое извлечение знаний (ключевых слов) из текста;

2.3.4. анализ авторского права на текст на основе образцов текста.

2.4. Создание «интеллектуальных пространств»:

2.4.1. интеллектуальные обучающие среды и оболочки;

2.4.2. формирование запросов (к БД) на основе внимания, уделяемого пользователем различным элементам среды;

2.4.3. менеджеры ресурсов для интеллектуальных пространств;

2.4.4. автоматизация программирования и создания программной документации — интеллектуальная поддержка технологий разработки ПО (UML);

2.4.5. многопользовательские среды.

2.5. Машинное обучение:

2.5.1. марковские процессы;

2.5.2. машинное чтение и понимание текстов;

2.5.3. восстановление утраченных элементов данных;

2.5.4. очистка данных от шумов.

2.6. Медицинское зрение:

2.6.1. автоматический анализ анатомических структур;

2.6.2. чтение снимков (рентген и т. п.);

2.6.3. машинная геометрия и пространственные сцены;

2.6.4. восстановление изображений по их отражениям;

2.6.5. сегментация изображений (например, простейшая разбивка на растр).

2.7. Мобильные роботы.

3. Проблема практического применения теоретических моделей.

4. Проблема совершенствования компьютерной лингвистики.

4.1. Разработка языка управления роботами на основе естественного языка.

4.2. Создание моделей естественного языка.

4.3. Понимание речи.

4.4. Разработка языков программирования, позволяющих повысить надежность разрабатываемого программного обеспечения (динамические языки, адаптивные системы, системы, способные «выживать»).

На рис. 7 рассмотренные вопросы представлены обобщенно.

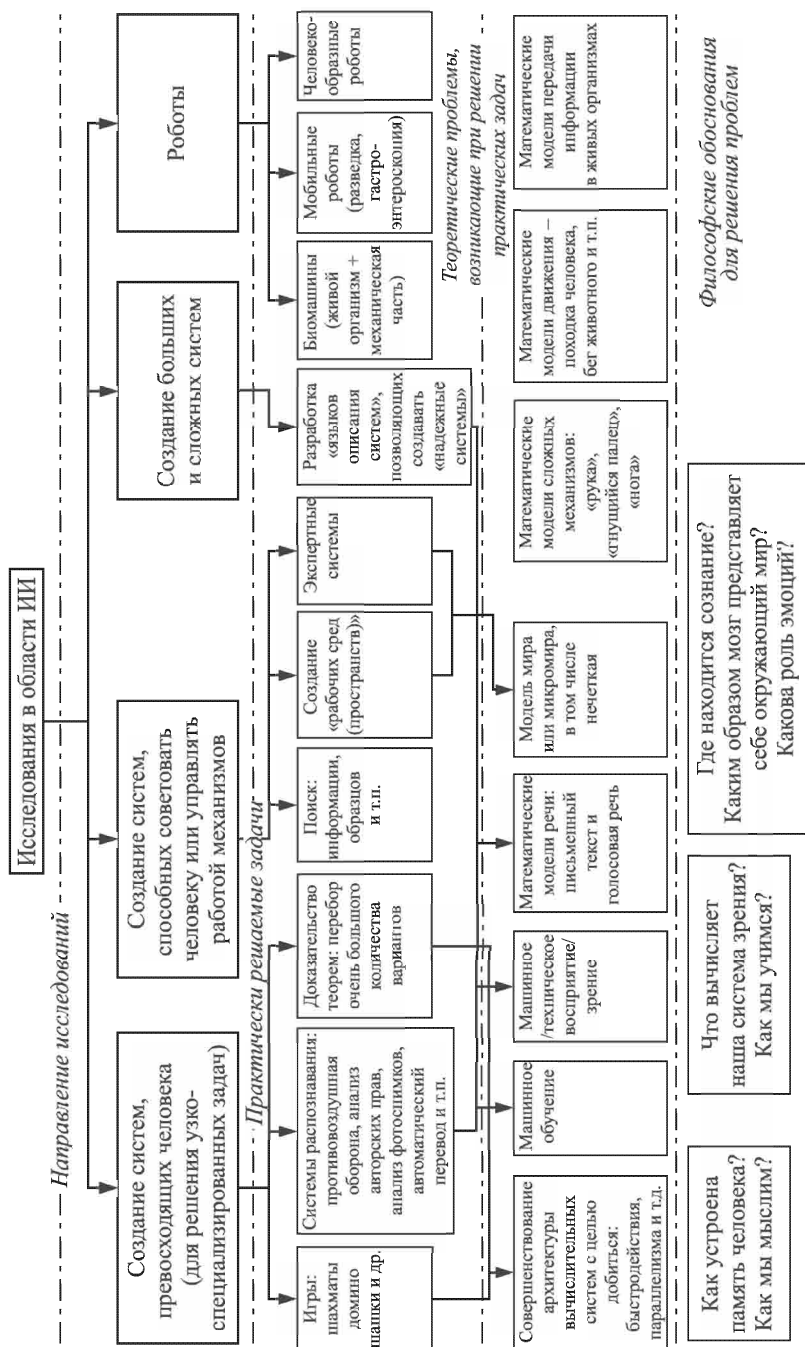


Рис. 7. Рассмотренные вопросы представлены обобщенно

Г Л А В А 2

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ЗАДАЧИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

2.1. ИИ — междисциплинарная область исследований

Каждая область академических знаний имеет достаточно узкий круг традиционных задач. Перечень этих задач и их формулировки становятся все более узкими по мере развития данной науки. В конечном счете область исследований разбивается на десятки частных задач, часть из которых поддается решению, часть в принципе не имеет решения, и последняя часть не может быть решена современными средствами, но и не существует доказательства невозможности ее решения.

Теория систем говорит нам о необходимости «междисциплинарного переноса», т. е. заимствования разными науками друг у друга формулировок задач и подходов к их решению. Очевидно, количество философских идей о способах и формах формулировки задач и подходов к их решению в этом случае возрастает, но все же остается конечным и исчисляется десятками. Проиллюстрируем эту мысль примером.

В традиционной медицине существует метод нормализации давления крови в сосудах путем введения специальных «сосудорасширяющих» препаратов. Однако их применение не всегда возможно. Кроме того, есть определенный предел расширения. Для снижения давления можно также попытаться снизить вязкость крови, но это связано с нежелательными химическими сдвигами.

Долгое время казалось, что задача не имеет решения. И действительно, в такой постановке «изобрести метод дальнейшего расширения сосудов» или «понижения вязкости крови» без побочных эффектов невозможно. Тем не менее необходимость в практическом решении не отпала. Инженеры—реологи, работавшие над проблемой движения жидкостей в технических системах (трубопроводы двигателей и т. п.), не знали о вышеописанных проблемах и, применительно к своей области, сформулировали задачу иначе: «Повысить пропускную способность трубопровода до определенного значения при условиях не изменяющегося во времени диаметра трубопровода и сохранения химического состава жидкости (заданная степень)». В результате был предложен метод, основанный на использовании специальных присадок (на основе высоких молекул — эффект Томса), изменяющих не диаметр трубопровода и вязкость жидкости, но характер течения — от турбулентного на ламинарный. «Обратный

перенос» разработанного метода в медицину дал качественный эффект¹.

В современном мире более сильным является тот ученый, который знает больше общих постановок задач, относящихся как к его области исследований, так и к смежным, и даже крайне отдаленным областям знаний. Для искусственного интеллекта такая концепция является просто определяющей, поскольку ИИ — это междисциплинарная прикладная область исследований. Термин «прикладная» подчеркивает тот факт, что понятие искусственного интеллекта само по себе, в отрыве от компьютерных наук и практически важных прикладных задач, не существует.

2.2. Перечень традиционных задач ИИ

Среди традиционных задач ИИ следует, конечно, выделить задачу классификации — отнесения некоторого объекта к одному из известных или неизвестных множеств объектов.

Такая задача возникает, например, в биологии при обнаружении нового биологического вида. В случае, если речь идет о крупном животном, как правило, возможно отнести обнаруженный вид к одному из известных классов, например, классу «млекопитающие»². Гораздо сложнее дело обстоит с мелкими организмами, например, червями. Вновь открытые виды часто имеют признаки, характерные для нескольких разных подвидов, и отнести их к тому или иному классу можно только с некоторой степенью уверенности на основе количественного сравнения степени выраженности некоторых признаков (рис. 8).

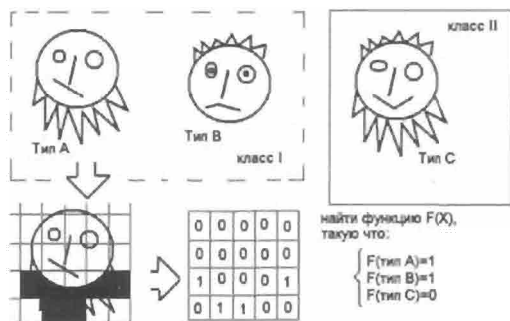


Рис. 8. Задача классификации

¹Пример предоставлен д.ф.-м.н. Г. В. Пышнограем, АлтГТУ. Впервые эффект Томса был использован в противокорабельных торпедах.

²Понятие «класс» использовано в логическом, а не биологическом смысле.

Пример: рассмотрим объект А. Он обладает признаком «цвет» и количественной характеристикой «темно-зеленый», а также признаком «лапы» с характеристикой «короткие». Известны также два класса: С1, в который входят все темно-зеленые объекты без лап, и класс С2, где все длинноногие объекты любого цвета. К какому же из классов следует отнести объект А?

Возможен и другой случай: имеются два класса С1, С2, о принципах формирования которых ничего не известно. Известно, что объект А отнесен к классу С1, а объект В — к классу С2. Необходимо выявить признаки, на основе которых построена классификация.

2.2.1. Более строгие формулировки задач.

1. Задача классификации: даны классы С1, С2; отнести вновь поступивший объект к одному из классов.
 - 1.1. Обратная задача: даны объекты А, В с известной принадлежностью к классам С1, С2; найти признаки классификации.
2. Задача составления плана действий.
 - 2.1. Задача поиска пути на графе.
 - 2.2. Задача объяснения произошедших событий.
 - 2.3. Задача прогнозирования возможных событий.
3. Задача генерации альтернатив.
 - 3.1. Всех возможных, которые могут произойти из некоторых посылок. Частным случаем можно считать задачу предсказания, например курса доллара на некоторую дату.
 - 3.2. Только альтернатив с заданными свойствами.
4. Задача выбора альтернативы, наилучшей в некотором смысле, из множества допустимых альтернатив.
 - 4.1. Задача случайного выбора.
 - 4.2. Критериальный выбор.
5. Задача управления некоторым техническим объектом, число учитываемых параметров в котором очень велико и-или параметры противоречивы. Последняя задача является как бы гибридом всех предыдущих: она комплексная.

2.2.2. Правила формулировки задач ИИ. Решение каждой задачи требует, во-первых, ее уточнения для конкретной предметной области; во-вторых — выбора необходимых параметров решения (точности и надежности³ результатов, затрачиваемого на решение времени и других).

³ Под надежностью понимается способность системы мало изменять значения своих выходов при малых изменениях входов в сериях испытаний.

Уточнение задачи — моделинезависимый этап; уточняя постановку задачи, следует думать о природе задачи, а не о способах ее решения. Очень важно так уточнить задачу, чтобы она, став «узкой», осталась достаточно «широкой». Т. е., до предела сузив указание «что требуется» или «какую цель необходимо достичь», необходимо оставить широкий диапазон «как делать».

Эти на первый взгляд противоречивые требования можно удовлетворить одновременно, если сформулировать абстрактную задачу. Примерами абстрактных задач могут быть: обработка списков, вычисление вероятностей, обоснование гипотез и др.

Пример постановки задачи. Существует известная шуточная задача, время от времени решаемая энтузиастами в различных странах: о доставке сырого яйца в целости и сохранности с высокой башни на землю. К решению применяется обязательное требование новизны. Казалось бы, явно курьезная задача, но она позволяет отработать целый класс идей, практически полезных в различных областях техники.

Студенты часто не могут решить ее (по-новому), оказываясь «заложниками» собственной неправильной постановки задачи. Ошибки в постановке касаются как формулировки самой цели — студенты непременно желают «бросать»⁴, а не «спускать» яйцо, так и внешних условий, состояний, и начального состояния — никто не предполагает, что яйцо может уже находиться внизу, падать не на твердое покрытие, быть замороженным и т. п. В последствии студенты поясняют, что они предполагали «наихудшие» условия, но ведь в исходной постановке этого не требовалось!

2.2.3. Выбор параметров решения задачи. Первым по важности параметром является точность решения. Для физических систем существует четкая и однозначная система единиц измерения: килограммы, метры и т. п.⁵ Для интеллектуальных систем общепринятой методики и единиц измерения нет. Говорят, например, о проценте правильных решений, о степени отклонения полученных решений от практически наблюдавшейся затем ситуации и др. Так, например, для медицинских систем хорошим считается результат в 70% и более правильных диагнозов при первом осмотре больного.

Второй параметр — устойчивость (требования к надежности) — касается поведения системы при ее широких испытаниях в различных ситуациях. Для интеллектуальных систем характерно, что при

⁴ Глагол «бросать», в свою очередь, значительно сужает способ реализации.

⁵ Ясно, что говорить о точности без адекватности не имело бы смысла. Определение адекватности дано ниже.

одних и тех же входных данных возможно получение различных выходных данных. Интеллектуальная система чаще всего неустойчива, и в этом ее сила. Так, например, на вопрос: «Каков будет курс доллара на первое января следующего года?» получен ответ: 31 рубль. Зададим вопрос повторно — получим ответ 31 рубль 10 копеек. Разница объясняется тем, что при каждом запросе система моделировала сотни и тысячи факторов, которые взаимодействуют нечетко. Польза от такого прогноза состоит в том, что «проиграв» один и тот же запрос сотни раз, мы сможем понять наиболее вероятное развитие событий.

Третьей, после адекватности⁶ и надежности⁷, важнейшей характеристикой качества предлагаемого решения является его экономичность. Решение является экономичным с некоторой точки зрения или по некоторому критерию. Наиболее часто говорят о времени решения задачи и стоимости решения задачи. Возможен также обратный вариант — стоимость ошибки. Стоимость решения задачи может складываться, например, из суммарной стоимости получения данных, без которых система не может решить задачу.

2.2.4. Выбор модели решения (представления знаний).

Избранные параметры решения (в наибольшей степени) и сама постановка задачи (в наименьшей степени) уже «подсказывают» нам способ решения задачи. Теперь инженер по знаниям должен определить, «на какую из решавшихся ранее задач это похоже больше всего и в чем принципиальное отличие?» Из опыта решения той «похожей» задачи он заимствует некоторый общий подход к решению, структуру, процедуры, данные или «модель». От правильного выбора модели зависит очень много. В теории ИИ сложились несколько типовых моделей, называемых «моделями представления знаний»⁸. Более подробно мы их рассмотрим ниже.

Важно понимать, что каждая модель представления знаний обладает некоторыми априорными ограничениями по надежности и степени адекватности. Необходимо соотносить требуемые для реальной задачи математические характеристики качества модели и те реальные характеристики, которыми на практике обладает избранная

⁶ Способность приводить именно к тому результату, который планировался (с заданной точностью).

⁷ Способность приводить именно к тому результату, который планировался всегда, во всех испытаниях, при условии, что исходные состояния «примерно одинаковы».

⁸ В русской терминологии — «ситуационное управление». Основоположник этой научной школы — профессор Д. А. Поспелов.

модель. Поясним мысль примером. Вероятностные модели ИИ, как правило, обладают лучшими по сравнению с детерминистскими моделями адекватностью и экономичностью. Но доверите ли Вы вероятностной системе управления наведение ракеты на цель, даже точно зная, что «по статистике» 99 из 100 ракет этой системы попадают точно в цель?

Важно также помнить, что предметная область, та самая среда, в которой затем будет действовать Ваша интеллектуальная система, обладает свойством постоянно изменяться. Одни модели учитывают свойство «изменчивости мира» в большей степени, другие — в меньшей.

Г Л А В А 3

ОБЛАСТИ ПРИМЕНЕНИЯ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

3.1. Хорошо и плохо структурированные предметные области

Мы обсудили список абстрактных задач, традиционных для теории ИИ. Какими же практически полезными примерами можно показать их применимость?

В первую очередь следует отметить, что методы ИИ эффективны для так называемых слабо структурированных предметных областей, т. е. областей, алгоритм действия в которых заранее неизвестен. Для таких областей характерна неясность и нечеткость во входных данных, наличие различного рода шумов. Вместе с тем решения требуется принимать однозначные, четкие и ясные, желательно также иметь возможность предсказать эффективность этих решений. К таким областям относятся: медицина, экономический менеджмент, управление сложными¹ техническими объектами, психология, лингвистика и еще много других.

Что же не относится к области эффективного применения методов ИИ? Это так называемые *хорошо структурированные предметные области*. К ним относятся в первую очередь точные и инженерные науки, такие как математика, физика, сопротивление материалов, геодезия, взрывное дело и целый ряд других. Применение методов ИИ менее эффективно в этих областях потому, что для решения задач в этих науках уже существуют свои собственные надежные алгоритмы и методы, опробованные в течение десятков или даже сотен лет.

Следует отметить, что при внедрении вычислительной техники даже эти старые и проверенные алгоритмы и методы видоизменяются, что приводит либо к возрастанию их вычислительной трудности, либо к значительному усложнению самого алгоритма. Феномен объясняется тем, что методы и алгоритмы, разработанные в докомпьютерную эпоху, ориентированы не столько на получение наилучшего результата, например расчета высокой точности, сколько на получение результата с минимальными вычислительными затратами. Эти

¹ Под «сложной» понимается система, принципы функционирования которой известны не полностью. Например, система «язык программирования» для первокурсника сложная, а для выпускника — просто большая (содержащая большое количество известных элементов).

алгоритмы предназначались для проведения расчетов вручную, и простое переложение расчетов на программу не дает выигрыша в качестве решения.

С другой стороны, усложнение алгоритмов и методов может дойти до такой степени, что конкретный пользователь окажется неспособен быстро разобраться не только в самом алгоритме, но даже в способе эффективного управления ходом этого алгоритма (программы). В этом случае некорректные или недопустимые значения входных и управляющих данных, введенные пользователем или полученные из базы данных, должны исправляться автоматически, но таким образом, чтобы добиться оптимального решения именно той задачи, которую имеет в виду конечный пользователь.

Пример: пользователь желает напечатать на листе бумаги ровно в центре строку «Поздравляю с Новым Годом!» Не обладая навыками форматирования текста, он набивает 16 символов «пробел» перед этой строкой. Интеллектуальный текстовый редактор, заподозрив, что пользователь является неопытным, спрашивает: «Вы пытаетесь центрировать текст?» В зависимости от ответа пользователя редактор либо заменяет пробелы символом выравнивания, либо выводит инструкцию, либо оставляет все «как есть», если выясняется, что пользователь имеет в виду именно 16 пробелов подряд, а не выравнивание текста.

Итак, выявление задачи возможно, например, на основе «разумного» диалога программы с пользователем, когда «умная» программа, последовательно выяснив уровень компетенции человека, задает ему либо профессионально-ориентированные вопросы, касающиеся значений параметров (например, у инженера-проектировщика можно спросить диапазон температур, допустимый для некоторого узла), либо более общие концептуальные вопросы о цели работы пользователя (как в примере выше), а подбор параметров осуществляет самостоятельно. Возможны и другие способы выявления целей, например стохастический (методом угадывания).

Таким образом, применение методов ИИ вполне допустимо и для традиционных, хорошо структурированных областей. В качестве примера можно привести химию. Существует программа искусственного интеллекта *Prospector*, позволяющая исследовать все химические соединения, полученные из заданного набора исходных. Методы исследования принадлежат самой химии и давно известны, но их практическое применение возможно только при помощи компьютера в виду огромного числа потенциальных соединений и свойств. По этой же причине управление «синтезом веществ внутри компьютера» требует применения методов ИИ.

Скорость решения является другой причиной применения методов ИИ в самых различных предметных областях. К решению задачи могут предъявляться специфические требования, например, требования соблюдения реального времени (режима реального времени), т. е. синхронизации процесса работы программы с некоторым другим физическим или программным процессом. Таких задач достаточно много, и в современной инженерной практике они выходят на первое место. Например, задача управления вооружением боевой машины. Современный солдат/самолет/танк должен уметь вести бой одновременно с несколькими объектами, заботясь при этом как о точности и надежности их поражения, так и о собственной безопасности. Но ни один летчик/солдат/наводчик не способен следить одновременно за несколькими целями, вести к цели 10 ракет и даже просто нажать на 10 курков одновременно. В этом случае человек должен только указать цель (целенаправить машину). А пуск ракет, их ведение до цели и поражение обеспечит система ИИ.

А нужен ли в принципе человек такой системе? На современном этапе — безусловно. Системы ИИ все еще «недостаточно разумны» для того, чтобы полностью доверить им управление². Человек несет ответственность за действия управляемой им системы ИИ.

Другим активно развиваемым направлением применения интеллектуальных машин является образование и, в частности, «обучение при помощи компьютера»³. Считается, что большую часть операций по предъявлению обучаемому информации и в особенности контролю за ее усвоением можно переложить на компьютер, причем не только без потери качества обучения, но с выигрышем.

В ряде областей обучение предполагает очень большое количество циклов повторений учебной информации и действий в специально сконструированной учебной ситуации с разными вариациями: управление движением самолета, заучивание английских слов и фраз применительно к сложившейся ситуации и т. п. При таком процессе присутствие человека-учителя совершенно не обязательно. Машина сама способна исправлять типовые ошибки, предъявлять контрольные и тренировочные упражнения и даже давать советы по методике обучения (на основе специальных правил, заложенных в нее опытным учителем). Обучение отдельным навыкам и умениям при помощи ЭВМ гораздо дешевле, а главное эффективнее, поскольку

² Некоторые специалисты считают подобный вопрос бессмысленным. По их мнению он должен звучать так: «Какова роль человека в такой системе?»

³ В американской терминологии CALL, CALI, CBI, CBM.

ку машине все равно, сколько раз придется исправлять одну и ту же ошибку, и у нее не наступит нервный срыв. Кроме того саму методику предъявления обучающего материала можно совершенствовать на основе объективных данных тестирования, протоколов работы, а далее оптимизировать, соотнося изменения в методике обучения с изменениями в показателях результата обучения. Это новая возможность, практически неосуществимая при безмашинной технологии.

Работу такой системы поясним примером. В 1997–2000 гг. в Лингвистическом центре Алтайского государственного технического университета была разработана оболочка информационно-образовательной среды «Chopin». Это специальный комплекс программ и электронных методических материалов, позволяющий организовать обучение иностранным языкам по принципу «один преподаватель — много студентов». В центре кроме обычных аудиторий имеются два компьютерных класса, соединенных ЛВС. Доступ в компьютерные классы не регламентируется, так что студент имеет возможность работать с электронными учебниками, электронной библиотекой, электронной фонотекой и банком данных тестов в любое время. Наиболее прилежные студенты проводят за этим занятием большую часть своего времени. В процессе работы с компьютером у студентов возникают некоторые типичные трудности. Так, например, у начинающих изучать французский язык имеется проблема с произнесением грассированной «р». У начинающих «англичан» имеется проблема «употреблять или не употреблять артикль» и «с каким предлогом сказать». Самая главная трудность при их самостоятельной отработке заключается в том, что обучаемый, выполняя упражнения, не знает, «правильно ли он, собственно, их делает». Если же компьютерная программа исправляет ошибки, то у студента возникает вопрос, «почему так», на который он не может получить ответ от человека-учителя немедленно и не знает, в каком разделе учебника найти его самостоятельно (или не может понять смысл этого раздела). Поскольку весь процесс происходит на компьютере, то вполне логично его полностью документировать при помощи базы данных. Просматривая затем такую БД по «своим» ученикам, каждый преподаватель может назначить индивидуальную консультацию каждому студенту. Со студентом Ивановым отработать звук «р», а с Петровым — неопределенный артикль. Чтобы построить такой план, придется решить нетривиальную задачу, более сложную, чем просто выборка из БД. В системе «Chopin» это делается при помощи специальных «деревьев целей», фактически являющихся моделью обучаемого (рис. 9).

могут оказаться и в разделах «выпечка», и в разделе «оформление праздников», а не только в разделах «торты» и «новый год». Какие либо интересные сведения по этому вопросу могут оказаться даже в разделе «история». Современные поисковые машины просто найдут все документы со словом торт, тортов, тортом и т. д., в том числе «тортов нет» и «рецепт неизвестен».

Иногда важны не скорость и четкость решения, как в медицине и военном деле, а само количество возможных решений; например, при принятии закона вычислить все возможные прорехи и противоречия; задача накопления опыта уходящих экспертов и ряд других⁶.

3.2. Модели представления знаний

Перечень реальных моделей, применяемых в системах ИИ, широк и трудно классифицируем. Большинство моделей в реальных системах являются гибридными, да еще и со значительной долей эвристики. В методических целях автор предлагает классифицировать наиболее известные модели в рамках приведенной схемы (рис. 11).



Рис. 11. Некоторые наиболее известные модели предоставления знаний

⁶ После прочтения этой лекции рекомендуется выполнить практическую (лабораторную) работу «Правильная постановка целей» и сдать тест «Некоторые основные термины ...»

ГЛАВА 4

ЛОГИЧЕСКАЯ МОДЕЛЬ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

4.1. Виды логических моделей, общие термины и определения

В современной теории ИИ используются несколько основных групп моделей: логика, фреймы, семантические сети, нейронные сети и др. Логическая модель по своей практической результативности и степени внедрения в реальные технические устройства сегодня занимает центральное место. Сама по себе модель не однородна и включает в себя уже около десятка существенно разных теорий¹ (рис. 12, а).



Рис. 12, а. Элементы логики, наиболее часто применяемые в системах ИИ

Исторически первой моделью для представления знаний (среди моделей ИИ вообще) стала классическая (формальная) логика Аристотеля, заложившая фундамент всех последующих логических теорий.

¹ В искусственном интеллекте говорят о «моделях представления знаний», а в логике — о «формальных теориях». Применительно к данному контексту это одно и то же.

Аристотель (учитель Александра Македонского) родился в 384 г. до н.э. в деревне Стагире в Халкидике (север Греции). В 17 лет пришел в Афины (чтобы учиться у Платона), где и стал известен. Позднее основал библиотеку и лицей (философскую школу). Оказал значительное влияние на дальнейшее развитие всех наук. Сочинения относятся ко всем областям знания того времени: «Физика», «О происхождении и уничтожении», «О небе», «О метеорологических вопросах», «Механика». Первичными качествами материи считал две пары противоположностей («теплое–холодное» и «сухое–влажное»). Основными (низшими) элементами, или стихиями, — землю, воздух, воду и огонь. Элементы мира рассматривал как комбинации первичных качеств: соединению холодного с сухим соответствует земля, холодного с влажным — вода, теплого с сухим — огонь. Пятым, наиболее совершенным элементом считал эфир. В средние века Аристотелевское учение о Вселенной было канонизировано церковью, что сильно тормозило развитие науки (до Николая Коперника).

Сегодня памятники Аристотелю установлены по всей Греции, и Вы можете смело спорить, что неизвестная улица называется либо Аристотэл, либо Алесандрос Мегалес.

Дальнейшее развитие модели связано с именами Канта (семантический анализ на основе категорий), Буля («математизация» логики), Фреге, Заде (ввел понятие «нечеткой логики»). Последние десятилетия активно развиваются «неклассические» направления: модальная, временная логики и др.²

Смысл построения любой формальной теории состоит в том, чтобы выразить мыслительные процессы (а в идеале — создать универсальную модель мира) формально (т.е. записать формулами). Очевидно, необходимо, чтобы система уравнений, образуемая такими формулами, решалась какими-либо методами, причем так, чтобы результаты вычислений совпадали с картиной, которую мы наблюдали бы в реальном мире при практическом осуществлении тех действий, которые моделируем. Если удастся (применительно к какому-либо реальному миру) сформулировать такие формулы и правила, то, очевидно, мы сможем предсказывать последствия тех или иных действий (в данном мире) без практического осуществления самих этих действий. Важно, что в процессе вычислений мы не будем думать о предметном смысле преобразований над переменными и формулами, а только о точном соблюдении формальных правил. Интерпретировать (наполнять смыслом применительно к реальному, скорее всего очень маленькому миру) мы будем только конечные результаты. Следовательно, сам процесс формального логического вывода можно будет поручить машине. За человеком же останутся интерпретация и оценка полезности результатов.

² Активно работающие в этой области ученые есть в России, США, Франции. Французская школа автору представляется наиболее известной.

Для того чтобы задать формальную логическую теорию, необходимо определить алфавит (множество символов, используемых для записи), правила синтаксиса (правила записи формул), аксиоматику (особое подмножество формул) и правила вывода (множество отношений на множестве формул). Правила вывода должны быть заданы так, чтобы на любых исходных данных обеспечить правильность логических заключений. Алфавит и аксиоматика должны быть заданы так, чтобы гарантированно обеспечить осмысленность (семантику) получаемых заключений и промежуточных следствий. Таким образом, основное различие в логических теориях, представленных на рис. 12, а, состоит в выборе аксиом и определении правил вывода. Далее мы последовательно рассмотрим аксиоматику и правила вывода логических моделей.

Разные логические теории имеют свои преимущества при реализации моделей в разных микромирах. Так, например, семантика силлогистики Аристотеля очень близка к естественному языку, поэтому результаты формального вывода легко интерпретировать. Выводы, получаемые на основе исчисления предикатов, уже не в полной мере совпадают с семантикой естественного языка, их труднее интерпретировать, однако модели на основе исчисления предикатов получаются гораздо более компактными и, как следствие, обозримыми. Модели на основе нечеткой логики Л. Заде позволяют оперировать размытыми понятиями, однако такие результаты и интерпретировать труднее и даже не всегда возможно.

4.2. Формальная (Аристотелева) логика

Аристотель стал известен благодаря своим исследованиям рассуждений³, получившим в последствии название формальной логики⁴. Он полагал, что мышление человека происходит на языке и, следовательно, мысль только тогда становится мыслью, когда выражена словами. В тоже время он полагал, что мысль, выраженная словами, — это уже не более чем графическая запись, строка символов, некоторое грамматически правильное предложение, имеющее свое подлежащее, сказуемое и т. д. Если не нарушать правил грамматики,

³ В данном случае слово «рассуждение» — это термин. Под рассуждением понимается определенная, внутренне обусловленная связь утверждений (посылок). Например, имеются две посылки: «крокодил зеленый» и «крокодил злой»; вывод «зеленый — значит, злой». Метод получения вывода — это и есть рассуждение. Как видите, возможны неправильные или правильные, но бессмысленные рассуждения.

⁴ Наука о законах и операциях правильного мышления (логического вывода, рассуждения). Основная задача логики — отделение правильных способов рассуждения от неправильных.

то ответ (равно как и вопрос к этому предложению) вполне можно построить только на основе правил грамматики, даже не понимая смысла предложения. Логика Аристотеля потому и называется формальной, что с ее точки зрения правильность рассуждения зависит только от логической формы этого рассуждения и не зависит от его содержания. Содержание мысли не оказывает никакого влияния на правильность рассуждений.

В мире существуют самые разные языки. Способы записи (от привычных нам символьных до графических — пиктограмм) различаются настолько разительно, что иногда трудно сказать, где язык, а где, например, художественное изображение (рис. 12, б). Ведь и художественная картина является некоторым уникальным языком, позволяющим обмениваться информацией.

Как же определить, что такое язык? С точки зрения классической логики язык — это некоторая система, позволяющая обозначать сущности и отношения между ними. Понятия «сущность» и «отношение» являются центральными понятиями логики. Под сущностями обычно понимаются объекты реального мира или абстрактные, не существующие в реальности объекты, например: «автомобиль», «черт», «глупость». Отношения — это способ связи сущностей, например: «едет», «путает», «раньше»⁵. С точки зрения логики любую фразу на языке можно рассмотреть в виде схемы «сущность — отношение — сущность». Например, в предложении «Иван съел яблоко» две сущности («Иван», «яблоко») и одно отношение «съел».

Впрочем, с точки зрения современной теории ИИ вполне допустима и обратная интерпретация: одна сущность «съел» и два отношения («Иван» и «яблоко»). Решение вопроса о том, что выбрать в качестве сущности, а что в качестве отношения в общем случае является произвольным и зависит от специфики решаемой задачи.

Итак, Аристотель выделил в языке сущности и отношения⁶. Эту мысль можно изобразить графически (рис. 13).



Рис. 12, б. Сущности, существующие в любом языке (иероглифы китайские)

⁵ До появления теоретико-множественного и математического подходов в логике понятие «отношение» являлось неопределимым. Сегодня этот термин определяют, например, как подмножество (область истинности) n -местного предиката, заданного на множестве (всех возможных отношений) $J_p = \{(a_1, a_2, \dots, a_n) | a_i \in A_i \text{ и } P(a_1, a_2, \dots, a_n) = \text{Истина}\}$.

⁶ Различают унарные, бинарные, тернарные, n -арные отношения. Например, отношение «живет» является унарным, живет (X), видит (X, Y) — бинарным, треугольник (угол1, угол2, угол3) — тернарным.



Рис. 13. Логическая схема предложения

Как видите, такой схеме могут соответствовать десятки самых разных предложений: «Паша любит Машу», «собака гонит кошку», «девять больше семи» и т. д.

Каков же практический смысл схемы? Она позволяет на основе чисто формальных правил сгенерировать осмысленное третье предложение из двух заданных. Рассмотрим два предложения: «Петр отец Ивана» и «Иван отец Коли» (рис. 14).



Рис. 14. Отношение «отец» не транзитивно

Исходя из этой схемы, вполне можно сделать вывод, что Петр «отец — отец» Коли, т. е. дед. Можно также сделать вывод, что 4 «меньше — меньше» шести, т. е. «много меньше», а вот вывод Петя «любит — любит» теннис (потому что «Петя любит Машу» и «Маша любит теннис») сделать нельзя, хотя предложения вполне могли бы уложиться в приведенную схему. В чем же дело?⁷

Аристотель, а затем Бэкон, Локк, Кант тоже задумывались над этим вопросом.

Джон Локк (29.09.1632–28.10.1704) — английский философ, основатель либерализма. Выпускник Вестминстерской школы (Лондон), а затем и Оксфорда. В университете изучал медицину, естествознание (у Роберта Бойля), философию, языки и литературу. Наибольших успехов добился в медицине и языках. Увлекался философией Декарта. По окончании университета преподавал древние языки, а затем занял место домашнего врача и воспитателя двух мальчиков в богатой семье. Локк первым выдвинул и обосновал принцип разделения властей на законодательную (выборную) и исполнительную. Написал книгу «Мысли о воспитании».

Иммануил Кант (22.04.1724–02.02.1804) — основатель философского критицизма. По образованию теолог. До 45 лет занимался физикой и астрономией. Создал оригинальную физико-астрономическую теорию мироздания. Издал труды: «Об огне», «О вращении земли вокруг ее оси», «Об одряхлении земли», «О землетрясениях». В 1755–1797 гг. читал курсы логики, метафизики, математики, физики, естественного права, этики, физической географии, антропологии и рационального богословия в Кенигсбергском университете. Лекции вынужден был прекратить после сочинения о «Религии в пределах одного разума», вызвавшего недовольство короля. В конце жизни начал труд, содержащий энциклопедию всех наук, но не смог закончить из-за ослабления ума. Логика приняла современный вид именно после

⁷В данном случае отношение «любит» не транзитивно.

работ Канта. До этого момента сильное влияние на логику оказывало искаженное церковное истолкование Аристотеля.

Учение Канта о познании предполагает, что ум может познавать предметы потому, что все познаваемое в них создается тем же умом по присущим ему правилам или законам. «Первичные качества предметов» (например, протяженность, субстанциальность, причинность) обусловлены познающим умом, но не в его эмпирических состояниях как чувственные (вторичные) качества (например, цвета, звуки, запахи), а его априорными (трансцендентальными) актами, создающими предметы в сознании. *A priori* — до опыта. *A posteriori* — после опыта. Познание состоит из суждений, получаемых на основе мысленных актов соединения понятий. Возможны два вида суждений.

- Аналитические — достоверные суждения, получаемые еще до проведения опыта (*a priori*); связь субъекта и предиката представляется всеобщей и необходимой. Пример: «тело есть нечто протяженное». Однако здесь нет нового знания, поскольку предикат протяженности уже содержится в самом понятии «тело».
- Синтетические — связь субъекта и предиката полагает нечто новое, не содержащееся в понятии субъекта, как его признак (суждения *a posteriori*). Например, длина улицы — 377 метров. Суждение сообщает нечто новое, так как число метров не может быть выведено аналитически из представления данной улицы; но зато это суждение выражает только единичные эмпирические факты, лишенные всеобщего и необходимого значения и потому не составляющие истинного познания.

Значение познания (такого нового, достоверного знания, которое не содержалось уже в исходных посылках) имеют суждения, являющиеся одновременно и аналитическими и синтетическими. Пример: суждение $789 + 567 = 1356$ есть истина всеобщая и необходимая, поскольку сумма чисел всегда, в применении ко всем предметам остается той же самой, причем мы можем быть уверены в этом еще до опыта. Результат суждения — число 1356 не содержится в понятии чисел 789 и 567. Суждения происходят следующим образом:

- ум приводит все данные ощущения в некоторый наглядный (воззрительный) порядок в формах времени и пространства (создает мир чувственных явлений);
- связывает чувственные явления умственно, по известным основным способам понимания (категории рассудка), создает мир опыта, который и подлежит познанию.

Всего возможны четыре формы суждений (связи субъекта S с предикатом P), в каждой из которых возможны три случая.

1. Предикат выражает объем субъекта — категории *количества*.
 - 1.1. Единство (субъект находится под своим предикатом как единичный экземпляр) — единичное суждение.
 - 1.2. Множественность (S находится под P как часть рода) — особенные суждения.
 - 1.3. Всеобщность (S находится под P как целый род) — всеобщее суждение.
2. Предикат выражает признак, входящий в содержание субъекта, — категории *качества*.
 - 2.1. Реальность (утверждается наличие некоторого признака, « A есть B ») — утвердительные суждения.
 - 2.2. Отрицание (отрицается наличие некоторого признака, « A не есть B ») — отрицательные суждения.

- 2.3. Ограничение (« A есть не B » — признак исключается так, что за субъектом остаются всякие другие признаки, кроме одного этого) — бесконечные суждения.
3. Предикат и субъект соотносятся определенным образом — категории *отношения*.
 - 3.1. Субстанция и принадлежность (предикат безусловно принадлежит субъекту) — безусловные (изъявительные, категорические) суждения.
 - 3.2. Причина и действие (субъект указывается как условие предиката, «если есть A , т. е. B ») — условные (гипотетические) суждения.
 - 3.3. Взаимодействие или общение (предикат представляется разделенным на несколько видов, через один из которых связывается с ним субъект, « A есть или B_1 или B_2 ») — разделительные высказывания.
4. Предикат и субъект имеют силу связи — категории *модальности*.
 - 4.1. Возможность (сочетание S и P представляется как только возможное, « A может быть B ») — проблематичные (сомнительные) суждения.
 - 4.2. Действительность или существование (сочетание S и P действительно существует, « A есть B ») — ассерторические (утвердительные) суждения.
 - 4.3. Необходимость (сочетание S и P представляется как необходимое, « A должно быть B ») — аподиктические (обязательные) суждения.

Из перечисленных основных понятий (категорий) выводятся некоторые другие общие понятия — *величина*, *сила* и т. п. На их основе Кант определяет коренные истины (основоположения чистого рассудка), обуславливающие опытное познание или естественную науку. Поскольку все сводится к закономерному соединению или сложению представлений, ясно, что кроме правил соединения требуется еще само соединяющее действие.

Соединяемые представления должны удовлетворять следующим требованиям.

1. Выделены или схвачены в своей особенности («аппрегенированы»).
2. Обладать воспроизводимостью при новом акте их соединения.
3. Быть «узнаваемыми». Необходимо быть уверенным, что воспроизводимые представления суть те же самые, какие были прежде схвачены. Акт узнавания возможен только в случае, если субъект схватывающий, воспроизводящий и узнающий представления остается одним и тем же или себе равным, т. е. субъект неизменно сохраняет свое внутреннее безусловное тождество: $Я = Я$.

В этом смысле интересно посмотреть на природу времени. Можете ли Вы доказать, что секунды равны между собой? Является ли время линейным? Меняется ли масштаб времени? Рассматривая природу времени, Кант приходит к понятию «схема». Схема — это нечто среднее между чувственным образом и чистым понятием. Время является чистым воззрением, но вместе с тем в нем заключены четыре рода мысленных определений, дающих соответственные схемы для всех категорий.

1. Схема *количества* (продолжительность, величина, т. е. число моментов или равных единиц).
2. Схема *качества* (содержание или само временное бытие) — то, что наполняет время и дает схему качества. Наполненное время — схему реальности, пустое время — схему отрицания.
3. Схема *отношения* (явления находятся в различном временном порядке относительно друг друга). Если одно явление пребывает, когда другие проходят — схема субстанции и акциденции. Если одно следует за другим — схема причи-

ны и следствия. Если все явления существуют в одно время — схема взаимодействия или общности.

4. Схема *модальности* — такая схема, при которой явление во времени существует или когда-нибудь (схема возможности), или в определенный момент (схема действительности), или во всякое время (схема необходимости). Представляя чувственные явления по этим схемам, чистое воображение в каждом случае указывает рассудку на приложимость той или другой из его категорий.

В результате возникла теория семантических категорий⁸ — учение о возможных типах сущностей, типах отношений и правилах вывода, позволяющих при осуществлении формальных преобразований не терять осмысленность получаемых результатов.

Можно построить такую логическую теорию, что в ней на основе формальных правил окажутся возможны выводы типа: «треугольность бьет рекорды», «если дует ветер, то победа». Важно понимать, что с точки зрения формальной логической записи такие выражения вполне могут оказаться истинными, но осмысленностью-то они не обладают! Выражение получается бессмысленным, если смешать языковые выражения разных типов (категорий). Таким образом, два выражения считаются относящимися к одной и той же семантической категории рассматриваемого языка, если замена одного из них другим в произвольном осмысленном предложении не превращает предложение в бессмысленное. Каждое правильно построенное выражение языка принадлежит одной и только одной из семантических категорий.

Считается, что множество категорий бесконечно и иерархично⁹. Все категории делятся на основные и функторные. Основные — это имена и предложения (высказывания). Упрощенно можно считать, что категорий три: имена, предложения (высказывания) и функторы (выражения, служащие для образования новых имен или высказываний из уже имеющихся).

4.2.1. Имена. *Имя* — выражение языка, означающее отдельный предмет, совокупность сходных предметов, свойства и т. п. Имена делят на единичные и общие в зависимости от того, сколько предметов они обозначают. Различают пустые (черт, призрак) и непустые (отсылает хотя бы к одному реальному объекту) имена.

Содержание имени — совокупность тех свойств, которые присущи всем предметам, обозначаемым данным именем и только им. Например, в содержании имени «благоустроенная квартира» мы понимаем помещение с водопроводом, канализацией и телефоном, предназначенное для проживания людей¹⁰.

⁸ Или, что то же самое, логическая грамматика.

⁹ Единого мнения на счет «бесконечности» множества категорий не существует.

¹⁰ В теории ИИ сформировались понятия экстенционала и интенционала понятия. Экстенционал — набор конкретных фактов, соответствующих данному понятию. Интенционал — определение или описание некоторого понятия через его свойства.

Объем имени — совокупность или класс тех предметов, которые обладают признаками, входящими в содержание имени. В объем имени «благоустроенная квартира» войдут «квартира в современном городском доме», «квартира в благоустроенном загородном коттедже» и т. п.

Различают однозначные и многозначные, точные и неточные, ясные и неясные имена. В логике существует так называемый «принцип однозначности», согласно которому в процессе коммуникации не следует использовать языковые выражения с несколькими разными значениями. Для неточных имен не всегда ясно, какие именно вещи они включают, а какие нет (объем). Неясные имена «размыты» или недостаточно определены в отношении своего содержания или смысла. Рассуждения, включающие неточные имена, имеют некоторые важные особенности.

- (а) Неточность имеет контекстуальный характер. Бессмысленно спорить о предмете, принимая во внимание только сам предмет (процесс, состояние), в одних ситуациях имя означает одно, с других точек зрения — другое.
- (б) Употребление неточных имен может привести к парадоксальным заключениям. Иногда неточность имени удастся устранить (молодой — совершеннолетний — 18 лет).

Имена находятся в различных отношениях друг к другу. Между объемами произвольных имен, которые есть смысл сопоставлять друг с другом, имеет место только одно из отношений: равнозначность, пересечение, подчинение и исключение (рис. 15).

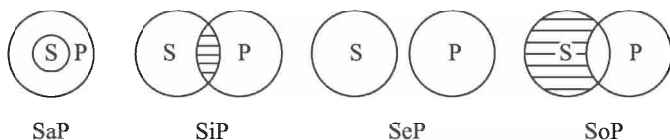


Рис. 15. Диаграммы Эйлера для иллюстрации отношений между именами

Равнозначными являются два имени, объемы которых полностью совпадают. Равнозначность не означает совпадение содержаний. Например, имена «благоустроенная квартира» и «благоустроенное помещение» на множестве двух перечисленных альтернатив можно считать равнозначными, но их содержание не совпадает (проживать в офисном помещении нельзя).

Деление — операция распределения на группы тех предметов, которые входят в объем исходного имени. Полученные в результате деления группы — члены деления. Признак, по которому производится деление, — основание деления. Требования к делению:

- 1) деление должно производиться только по одному основанию;
- 2) деление должно быть соразмерным (исчерпывающим), т. е. сумма членов деления должна равняться объему делимого понятия;
- 3) члены деления должны взаимно исключать друг друга;
- 4) деление должно быть непрерывным.

Частным случаем деления является дихотомия. Дихотомическое деление опирается на крайний случай варьирования признака. С одной стороны, выделяются предметы, имеющие этот признак, а с другой — не имеющие его.

Классификация — многоступенчатое разветвленное деление. Результатом классификации является система соподчиненных имен: делимое имя является родом, новые имена — видами, видами видов (подвидами) и т. п.

Класс — множество предметов, каждому из которых присущи признаки, отраженные в содержании соответствующего понятия. Слово или словосочетание, выражающие это понятие, служат именем класса. Классы могут быть простыми и сложными.

Простой класс — это класс, члены которого характеризуются только одним общим признаком, выражаемым именем данного класса. Пример: самолеты, пассажиры, реактивные двигатели и т. п. Простой класс нельзя расчленить на еще более простые.

Сложный класс — такой класс, члены которого имеют более чем один общий признак. Сложный класс всегда можно расчленить на простые, а при необходимости однозначно восстановить из простых.

Классификационные системы (КС) применяются для структурирования и обобщения знаний. С одной стороны, все сущности разбиваются по определенным признакам на некоторое число классов; с другой стороны, группируются вместе. Наблюдателю дается набор объектов, которые можно описать некоторым множеством признаков. Каждый объект принадлежит одному или более классам из некоторого фиксированного множества.

С помощью КС решают три основные задачи.

- Классификация образов: наблюдатель должен применить установленное ранее правило и отнести вновь поступивший объект к некоторому классу.
- Распознавание образов: правило классификации вырабатывается на основе исследования множества объектов с известной принадлежностью к различным классам (обучающей выборке).
- Формирование образов: объекты представляются наблюдателю без указания их принадлежности классам. Наблюдатель должен сам построить соответствующее деление классов.

Общая задача классификации эквивалентна задаче выяснения: является ли некоторая цепочка допустимым предложением в некотором формальном языке (грамматике). Некоторые КС применяются при представлении декларативных знаний, например для построения таблиц ключевых трансляторов алгоритмических языков.

В иерархической системе (рис. 16) между классифицируемыми группировками установлено отношение подчинения (например, род — вид). Множество объектов делится по некоторому признаку (основанию деления) на крупные группировки, затем каждая группировка разбивается на последующие группировки и т. д., постепенно конкретизируя объект классификации. Необходимо соблюдать правила деления, изложенные выше.

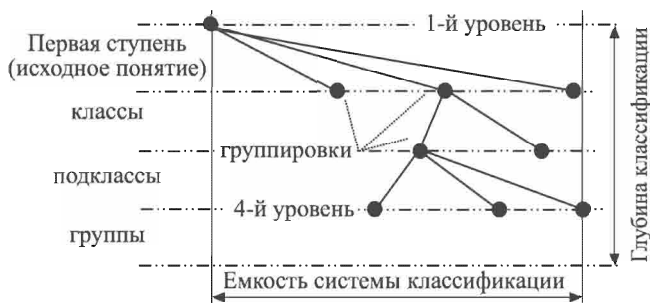


Рис. 16. Иерархическая классификация

Так, например, при классификации собак по породам ошибочно разбивать понятие собаки на колли, догов и овчарок. Если Вы соберете вместе всех колли, догов и овчарок, то не получите класс «собаки». Сколько бы Вы не дополняли список пород, Вы все равно что-либо упустите. Правильный вариант: собаки — это «доги и «не доги», «не доги» — это «овчарки и «не овчарки» и т. д.

Если Вы имеете профессиональный опыт в одной из предметных областей (например, в бухгалтерии), то Вам должно быть очень интересно проверить классификаторы (особые списки, применяемые в разных предметных областях) на соблюдение этих правил.

Писатель Х. Л. Борхес приводит пример, связанный, как сказал бы современный бухгалтер, с учетом основных фондов в Древнем Китае. Чиновники полагали, что животные подразделяются на: а) принадлежащие императору; б) бальзамированные; в) прирученные; г) молочные поросята; д) сирены; е) сказочные; ж) бродячие собаки; з) заключенные в настоящую классификацию; и) буйствующие как в безумии; к) неисчислимы; л) нарисованные очень тонкой кисточкой из верблюжьей шерсти; м) прочие; н) только что разбившие кувшин; о) издалека кажущиеся мухами.

Иерархическая классификация применяется во многих областях. В докомпьютерную эру она являлась основой для учета и систематизации (например, в биологии, см. рис. 17).



Рис. 17. Фрагмент классификации животных по Ч. Дарвину

Изучением разнообразия животных занимается систематика. Главная ее задача — это распределение животных по группам (классификация). Основная единица классификации — вид животных. Под видом понимается совокупность организмов или особей, имеющих сходное строение, образ жизни, способных к скрещиванию с образованием плодовитого потомства и населяющих определенную территорию. Близкородственные виды животных объединяют в особую группу, называемую родом. Близкие, сходные роды животных относятся к одному семейству. Близкие, сходные семейства объединяют в отряд, отряды — в класс, классы — в тип, типы — в подцарство, подцарства — в царство. Подцарств животных два: простейшие (одноклеточные) и многоклеточные животные. Главное различие состоит в том, что у простейших каждая клетка — это самостоятельный организм. Клетки же многоклеточных животных входят в состав организма и выполняют различные функции: одни — защитные, другие — по добыванию пищи, третьи — ее перевариванию и т. д. Вне организма эти клетки жить не могут.

В автоматизированных системах сейчас большое применение имеют тезаурусы (табл. 3) и фасетные классификации (рис. 18).



Рис. 18. Фасетная классификация

Основное преимущество иерархической системы классификации — большая информационная емкость и простота поиска (возможность ручного поиска). Недостатки — малая гибкость структуры,

невозможность агрегировать объекты по любому произвольному сочетанию признаков.

Алфавитно-предметная классификация — это система классов (каждый из которых соответствует одному виду предметов или факторов), расположенных в алфавитном порядке их имен (телефонный справочник). Пример такой классификации — англо-русский словарь. Слова расположены в строгом алфавитном порядке.

Иногда удобен другой принцип — расположить рядом все слова, начинающиеся на разные буквы, но относящиеся к одной ситуации или предмету. Например, слова «полиция», «больница», «страховка» логично поместить в раздел «авария». В этом случае имеем дело с предметной классификацией.

Тезаурус — центральная точка (тема) документа или информационного запроса. Выражается в виде некоторого перечня простых слов и словосочетаний естественного языка, обычно являющегося именем простых классов. Такие слова — координаты документа в n -мерном пространстве предметно-тематических признаков. Для координатного индексирования документов или информационных запросов могут использоваться полнзначные (неслужебные) слова, выбираемые непосредственно из индексируемых текстов. Такие слова и сочетания — ключевые. Процесс построения тезауруса представлен в таблице 3.

Порядок построения тезауруса

Т а б л и ц а 3

Название классификационной единицы	Операции
Исходный текст или фрагмент исходного текста	Понск (выделение в тексте) ключевых слов
Нормативный словарь	Приведение ключевых слов к единой морфологической форме и написанию. Учет синонимии, полисемии ¹¹
Дескрипторный словарь	Строится класс из близких ключевых слов, выбирается обобщенное имя (дескриптор)
Нормативный словарь-справочник (информационно-поисковый тезаурус)	Дополнение дескрипторного словаря сведениями о предметно-тематической области

Информационно-поисковый тезаурус предназначен для координатного индексирования документов и информационных запросов, в которых приведены в алфавитном порядке все дескрипторы и синонимичные им ключевые слова, а также отражены важнейшие

¹¹ Пример: мосты (строения), мосты (ходовая часть).

отношения между дескрипторами. В зависимости от функции лексически единицы тезауруса делятся на дескрипторы и условные синонимы.

Дескриптор — однозначное ключевое слово или код, обозначающий класс условной эквивалентности, в который включены эквивалентные слова и близкие им по смыслу ключевые слова. Элементарная структурная единица тезауруса — словарная статья.

Пример. Вычислительные машины:

⟨компьютеры, ЭВМ, вычислительные комплексы⟩

⟨персональные, большие ЭВМ, малые ЭВМ, супер-ЭВМ⟩

⟨техника вычислений, вычислительная техника, теория инженерных расчетов⟩

⟨расчеты, Интернет, игры, экономика⟩.

Здесь тезаурус $d_i\langle M_1, M_2, M_3, M_4\rangle$, где d_i — заглавный дескриптор;

M_1 — упорядоченное по алфавиту множество условных синонимов данного заглавного дескриптора, образующих вместе с ним класс условной эквивалентности;

M_2 — упорядоченное множество дескрипторов, каждый из которых связан с заглавным отношением род-вид;

M_3 — упорядоченное множество дескрипторов, каждый из которых связан с заглавным отношением вид-род;

M_4 — упорядоченное множество дескрипторов, каждый из которых связан с заглавным отношением целое-часть, часть-целое, причина-следствие и др.

Затруднения с классификацией имеют объективную причину. Окружающий мир сложен, в нем отсутствуют жесткие границы и ясно очерченные классы. Всеобщая изменчивость усложняет картину.

Проблемами выделения жестко очерченных границ занимается таксономия. В тех случаях, когда однозначно классифицировать предмет не удастся, его относят к некоторому классу по совокупности выраженности признаков, оцененных числами. В этом случае говорят о числовой или численной таксономии (см. также 11.2.3).

4.2.2. Высказывания. *Высказывание* (утверждение) — грамматически правильное предложение, взятое вместе с выражаемым им смыслом и являющееся истинным или ложным. Простое высказывание (неразложимое, атомарное, атом) не включает других высказываний в качестве своих частей¹². Сложное высказывание получается из простых с помощью логических связок: отрицание, конъюнкция (и),

¹² Но само по себе также имеет внутреннюю структуру.

дизъюнкция (или), исключаяющая дизъюнкция. Пример простого высказывания: «Все медведи любят мед», сложного — «Некоторые медведи любят мед и (некоторые медведи) молодые побеги бамбука».

Простые высказывания могут разбиваться на составные части по-разному¹³. Аристотель рассматривал одну из разновидностей простых высказываний — *категорические* высказывания (суждения), т. е. такие, в которых утверждается или отрицается наличие какого-то признака у всех или некоторых предметов рассматриваемого класса.

П р и м е р: « S есть P » и « S не есть P », где S — имя предмета (субъекта), P — имя признака (предикат). «Земля есть планета».

Рассматривают также высказывания об *отношениях*, в которых устанавливаются отношения между двумя или большим числом предметов. П р и м е р: «Три меньше пяти», «Москва больше Барнаула». Высказывания об отношениях не могут быть сведены к категорическим высказываниям.

Особенность именно категорических высказываний в том, что не просто устанавливается связь предмета и признака, но дается количественная характеристика субъекта, т. е. «все», «некоторые»¹⁴ (табл. 4).

Т а б л и ц а 4

Формы категорических высказываний

Функторы	Обозначение	Пример интерпретации
«все ... есть ...»	a	SaP — все жидкости упруги
«некоторые ... есть ...»	i	SiP — некоторые животные говорят
«все ... не есть ...»	e	SeP — все дельфины не есть рыбы
«некоторые ... не есть ...»	o	SoP — некоторые металлы не есть жидкости

Каждое из этих четырех выражений (a , i , e , o) Аристотель рассматривал как логические постоянные, не имеющие самостоятельного содержания и позволяющие из двух обладающих содержанием имен получить содержательные, являющиеся истинными или ложными простые высказывания. При этом имена S , P не должны быть единичными или пустыми.

При составлении сложных высказываний, осуществлении логического вывода (получения следствия из двух посылок) необходимо

¹³Результат разложения высказывания зависит от цели, ради которой оно осуществляется, т. е. той теории логического вывода, в рамках которой оно анализируется.

¹⁴В своем дальнейшем развитии логика пришла к понятию «исчисление предикатов» — особой области математической логики, рассматриваемой отдельно. Стали рассматривать предикаты первого и второго порядков. На основе теории созданы специальные машинные языки, например Planner и Prolog (соответственно).

заботиться не только об истинности заключения, но и его осмысленности. Осмысленность можно проанализировать формально на основе так называемого «логического квадрата» (рис. 19). На логическом квадрате можно выделить особые фигуры (логического вывода), следуя которым гарантированно получим осмысленный результат.



Противоречание высказывания — не могут быть одновременно истинными и ложными.

Противные — могут быть вместе истинными, но не могут быть вместе истинными.

Подпротивные — не могут быть одновременно ложными, но могут быть одновременно истинными.

Подчиненные — из истинности подчиняющего высказывания автоматически следует истинность подчиненного.

Рис. 19. Логический квадрат

Фактически рис. 19 задает правила вывода (отношения между формулами). Таким образом, определив алфавит, правила вывода, синтаксис, мы задали некоторую логическую теорию, в данном случае — исчисления высказываний (ИВ). Отличительная особенность ИВ, в отличие от появившегося позднее исчисления предикатов (ИП), состоит в том, что в качестве терминов (S, P) могут выступать именно логические переменные, в то время как в ИП на месте термина может оказаться целый предикат. Логический вывод на категорических высказываниях может происходить индуктивно или дедуктивно.

В *дедуктивном* умозаключении связь посылок и заключения опирается на логический закон. Заключение с логической необходимостью вытекает из принятых посылок, от истинных посылок всегда ведет к истинному заключению. Для дедукции характерен переход от общего знания к частному.

В *индуктивном* умозаключении связь посылок и заключения опирается не на логический закон, а на некоторые фактические или психологические основания, не имеющие формального характера. Индукция дает вероятные (правдоподобные) заключения. Максимум, о котором можно здесь говорить, — определенная степень вероятности выводимого утверждения. Для индукции характерен переход от множества частных случаев к общему знанию — обобщение. Основное отличие индукции от дедукции состоит в том, что дедукция — это логический переход от одной истины к другой, а индукция — переход от достоверного знания к вероятностному.

Правильный вывод от истинных посылок всегда ведет к истинному заключению. Позволяет получить новое знание с помощью

чистого рассуждения, без обращения к опыту и интуиции. Если хотя бы одна из посылок является ложной, правильное рассуждение может давать в итоге как истину, так и ложь. Неправильный вывод от истинных посылок может вести как к истинным, так и ложным заключениям.

Аристотель рассматривал так называемые категорические силлогизмы (или просто силлогизмы, соответственно теория в целом — силлогистика) — дедуктивные умозаклучения, в которых из двух категорических высказываний выводится новое категорическое высказывание.

Все жидкости упруги

Вода — жидкость

Вода упруга

Все M есть P

Все S есть M

Все S есть P

В каждом силлогизме должно быть 3 термина: меньший (субъект S), больший (предикат P) и средний (термин M , присутствующий в посылках, но отсутствующий в заключении). В зависимости от положения среднего термина различают 4 фигуры силлогизма (рис. 20).

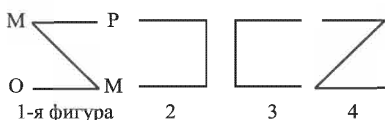


Рис. 20. Фигуры силлогизма

Модусами силлогизма называются разновидности фигур, отличающиеся характером посылок и заключения.

Всего возможно $4 \times 64 = 256$ различных сочетаний посылок и заключений (4 фигуры по 64 модуса в каждой фигуре). Из всех возможных модусов силлогизма правильных только 24 (по 6 в каждой фигуре), из которых пять являются ослабленными — заключения в них являются частноутвердительными или частноотрицательными высказываниями, хотя в других модусах эти же посылки дают общеутвердительные или общеотрицательные заключения. Таким образом, остается 19 правильных модусов силлогизма. Некоторые из силлогизмов имеют традиционно принятые имена (табл. 5).

Таблица 5
Традиционные имена некоторых силлогизмов

1-я фигура	Barbara, celarent, darii, ferio, barbari, celaront
1-я фигура	Cesare, camestres, festino, baroco, cesaro, camestros

В каждом из этих названий содержатся 3 гласные буквы. Они указывают, какие именно категорические высказывания используются в модусе в качестве посылок и заключения. Например, Celarent — большая посылка (SeP), меньшая — общеутвердительное (SaP), заключение — общеотрицательное высказывание (SeP). Для оценки правильности силлогизма удобно использовать диаграммы Эйлера.

Пример 1. (Barbara): Все металлы (M)ковки (P) + железо (S) есть металл (M) = железоковки (P). Т.е. если все M входят в объем P , то S войдет в объем P (рис. 21).



Рис. 21

Пример 2. Все рыбы (P) не имеют перьев (M) + у всех птиц (S) есть перья (M) = ни одна птица (S) не является рыбой (P). Т.е. если все птицы входят в объем M , а M ничего общего не имеет с P , то и у S нет ничего общего с P (рис. 22).

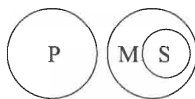


Рис. 22

В силлогизме не может содержаться информация, отсутствующая в посылках, заключение только развертывает информацию, содержащуюся в посылках, но не может привносить новую информацию, отсутствующую в них.

Энтимема — силлогизм, в котором одна из посылок не выражена явно: «Щедрость заслуживает похвалы, как и всякая добродетель», «он ученый, поэтому любопытство ему не чуждо». В первом случае опущена меньшая посылка «щедрость — это добродетель», во втором большая «всякому ученому не чуждо любопытство». Для оценки правильности энтимемы следует восстановить в полный силлогизм.

4.2.3. Процедуры доказательства и опровержения. Практические программные системы ИИ на основе логической модели ориентированы в основном на решение двух задач.

1. Доказательство логических гипотез (которые в данной конкретной области называются теоремами). Гипотеза выражается некоторой логической формулой, и необходимо доказать ее истинность на некотором наборе исходных данных. Информационной системе на вход подается формула, система ее проверяет, и если формула истинна, отвечает «Да». Если формула ложна, то ответ системы «Нет», однако в некоторых ситуациях система не может ответить ни «Да» ни «Нет». В этом случае запрос переформулируют, пытаясь доказать отрицание этой формулы, т.е. опровергнуть ее. В ряде случаев опровержение оказывается гораздо проще доказательства ложности. Например: «все медведи бурые» — достаточно предъявить одного белого медведя, т.е. подтвердить теорему «не все медведи бурые».

2. Объяснение доказательства теоремы — предоставление пользователю всех промежуточных выводов. В этом случае система поясняет: «Я сделала такой вывод, потому что...» Если речь идет о 10–15 пар «причина–следствие», то вполне возможно отследить логику работы машины, более 20–30 пар — объяснения машины становятся похожи на речь шизофреника.

Безусловно, на основе логической модели можно решать и другие задачи, например задачу вывода всех возможных (как правильных, так и неправильных и неосмысленных) следствий из некоторых посылок. Практически это очень полезно, так как позволяет, например, «обыграть» все возможные неправильные истолкования некоторого документа, уточнить терминологию, исключить дублирование определений, найти противоречия, тавтологии в определениях.

Пример: «ЛЕСНОЙ КОДЕКС РОССИЙСКОЙ ФЕДЕРАЦИИ» от 29.01.1997 N 22-ФЗ (ред. от 25.07.2002, с изм. от 24.12.2002; принят ГД ФС РФ 22.01.1997).

Общая часть: ... с учетом представлений о лесе как о совокупности лесной растительности, земли, животного мира и других компонентов окружающей природной среды, имеющей важное экологическое ... значение.

Статья 87: Целями воспроизводства лесов являются своевременное воспроизводство лесов на не покрытых лесом землях, улучшение породного состава лесов, увеличение производительности лесов, обеспечение рационального использования земель лесного фонда ...

Статья 88: Повышение продуктивности лесов осуществляется в результате реализации системы научно обоснованных рубок, воспроизводства лесов, улучшения их породного состава ... ухода за лесами и проведения других лесохозяйственных работ.

Выделяем имена: 0 — лес, 1 — растительность, 2 — животные, 3 — цель, 4 — рубка. Строим посылки: 1 — лес есть растительность, 2 — лес есть животные, 3 — рубка есть лес (рубят лес). Для автоматизации построения следствий воспользуемся программой Переверзева Е. А. (Барнаул, АлтГТУ, 1998). Построим только правильные модусы. Система генерирует возможные выводы (сорит) и представляет их графически (рис. 23, 24):

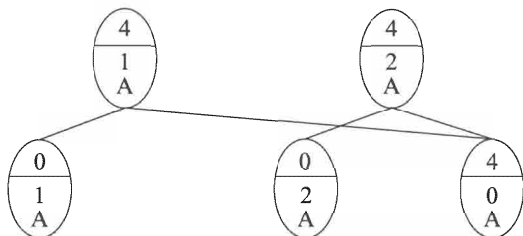


Рис. 23. Сорит

Как видите, из текста закона следует, что «всякая рубка есть растительность» ($A, 4, 1$) и «всякая рубка есть животные» ($A, 4, 2$). Очевидно, термин «рубка» в применении ко всему понятию «лес» недопустим, следовало бы указать «рубка лесной растительности». Теперь построим, в том числе и слабые, модусы.

Вариантов толкования гораздо больше. Появились, например, «некоторые животные есть растительность» ($i, 2, 1$).

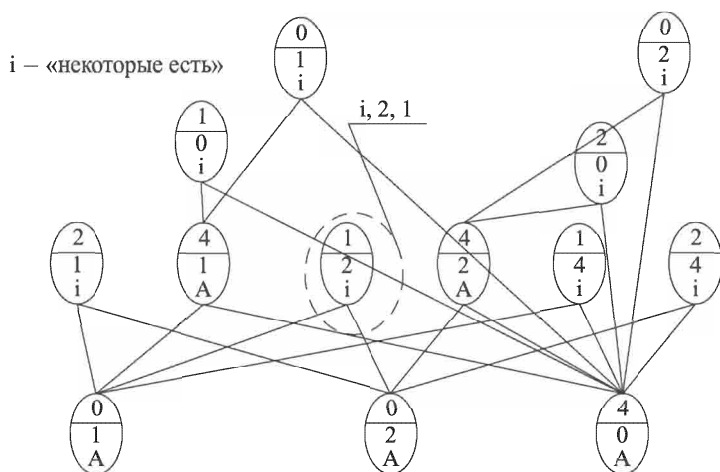


Рис. 24. Сорит

Логика высказываний, с которой мы сейчас имеем дело, является не самой выразительной. Разные программные системы ИИ используют различные процедуры доказательства и опровержения, рассмотрим наиболее общую теорию.

Доказательство — дедуктивная процедура установления истинности некоторого утверждения путем приведения других утверждений, истинность которых уже известна и из которых с необходимостью вытекает первое. Состоит из тезиса, основания (аргументы) и логической связи. По структуре доказательства делят на прямые и косвенные (устанавливают справедливость путем вскрытия ошибочности антитезиса). Рассмотрим виды косвенных доказательств.

1. Приведение к абсурду (вывести из утверждения противоречие). Например, древнегреческий философ Протагор высказал утверждение: «Истинно все, что приходит кому-то в голову», на что Демокрит ответил, что «не все высказывания истинны».

2. Разделительное косвенное доказательство (если число рассматриваемых возможностей более 2-х). Применяется, если есть уверенность, что доказываемое положение входит в число всех рассматриваемых возможностей. Взаимная несовместимость возможностей и то, что ими исчерпываются все мыслимые ситуации, определяются не логическими, а фактическими обстоятельствами. Типичная ошибка таких доказательств — выдвинутые возможности, вместе взятые, не исчерпывают всех возможных альтернатив.

Опровержение — рассуждение, направленное против выдвинутого тезиса, имеющее целью установление его ложности или недоказанности. Рассмотрим приемы опровержения.

1. Выведение из опровергаемого утверждения следствий, противоречащих истине.
2. Доказательство истинности отрицания утверждения (утверждение и его отрицание не могут быть одновременно истинны). Пример: медведи бывают только бурыми. Показываем одного белого медведя — опровержение получено.
3. Если тезис выдвигается с обоснованием, то опровергнуть можно эти обоснования. При этом, однако, само положение может быть и верным, но подкреплено слабыми аргументами.
4. Опровергнуть можно саму связь аргументов и тезиса, если показать, что тезис не вытекает из доводов. Из этого, однако, не следует ошибочность ни аргументов, ни тезиса.

Пример. Юморист начала прошлого века В. Билибин писал: «Если бы на свете не существовало солнца, то пришлось бы постоянно жечь свечи и керосин. Если бы пришлось постоянно жечь свечи и керосин, то чиновникам не хватало бы их жалования, и они брали бы взятки. Следовательно, чиновники не берут взятки потому, что на свете существует солнце».

Ошибки в доказательстве.

1. Ошибки в отношении аргументов:
 - 1.1. содержательная ошибка — попытка обосновать тезис с помощью ложных аргументов;
 - 1.2. круг в доказательстве — попытка доказать положение через него же, но сформулированное в другой форме: «Опиум усыпляет, потому что обладает снотворным действием, а его снотворная сила проявляется в том, что он усыпляет».
2. Подмена тезиса — замещение его в ходе доказательства близким по форме или содержанию положением. Тезис при этом может, например, «сужаться». Большим любителем подобных вещей был Диоген. В споре его оппонент утверждал, что в мире, как он представляется нашему мышлению, движение невозможно. Диоген встал и начал не спеша ходить, за что и был бит.
3. Потерянная логическая связь.

Несмотря на кажущуюся простоту, поиск логических ошибок — занятие трудное и нетривиальное. Во времена Древней Греции, Китая, в средневековой Европе существовали целые придворные философские школы, где дипломатов учили правилам составления и разрешения софизмов и паралогизмов.

Софизм — рассуждение, кажущееся правильным, но намеренно содержащее скрытую логическую ошибку. Служит для придания видимости истинности ложному заключению.

Примеры: «что не терял, то имеешь; рога ты не терял, значит у тебя рога», «сидящий встал, кто встал, тот стоит; следовательно, сидящий стоит», «этот пес твой, он отец, значит, он твой отец», «компания, получившая кредит, ничего банку не должна, поскольку она стала иной: в правлении не осталось никого из просивших ссуду».

Паралогизм — непреднамеренная ошибка в рассуждении.

4.3. Математическая реализация формальной логики

В XIX веке (≈ 2000 лет после Аристотеля) английский математик Джордж Буль усовершенствовал логическую систему обозначений.

Джордж Буль (2.11.1815–8.12.1864) — отец математической логики. Профессор математики, самоучка (закончил только школу). Знал греческий, французский, немецкий и итальянский языки. Учился по книгам Ньютона, Лапласа, Лагранжа. Был отмечен как талантливый математик профессором Августом Де Морганом. В 1854 г. опубликовал работу «Исследование законов мышления, базирующихся на математической логике и теории вероятностей». Первым показал, что существует аналогия между алгебраическими и логическими действиями, попытался сформулировать общий метод вероятностей, с помощью которого из заданной системы вероятных событий можно было бы определить вероятность последующего события, логически связанного с ними.

Он предложил записывать истинность как 1, а ложь как 0^{15} . Булю не интересовало все возможное многообразие отношений естественного языка. Он рассмотрел только отношения «+» объединения, «*» пересечения, исключения.

Усовершенствованная система записи Буля позволила поставить абстрактную логическую задачу в виде системы уравнений:

$$X_1 + X_2 = 1$$

$$X_2 * X_3 = 0.$$

Кто съел шоколад? Кошка Лиза (X_1) или («+») щенок Кинг (X_2)? Если хозяин (X_3) не ел, а гулял в это время с щенком, значит это была Лиза.

Буль¹⁶ ввел специальные обозначения (табл. 6, 7).

Исчисление предикатов — некоторая аксиоматическая система, предназначенная для моделирования некоторой среды и проверки каких-либо гипотез относительно свойств этой среды при помощи

¹⁵ Другими словами, высказывание только тогда можно считать высказыванием, когда его переменные конкретизированы, причем либо значением «Истина», либо «Ложь». Конкретизация «неизвестно» не допускается.

¹⁶ Сегодня все это направление носит название «булева алгебра», хотя целый ряд ее важнейших положений открыт задолго до Дж. Буля.

разработанной модели. Гипотезы¹⁷ при этом утверждают наличие или отсутствие некоторых свойств у некоторых объектов и выражаются в виде логической формулы. Обоснование гипотезы сводится, таким образом, к оценке выводимости и выполнимости логической формулы.

Таблица 6

Некоторые обозначения булевой логики

Операция	Обозначение
Конъюнкция	«и», « $\wedge$ », « $\times$ »
Дизъюнкция	«или», « $\vee$ », « $+$ »
Эквивалентность	« A тогда и только тогда, когда B », « $\leftrightarrow$ » или « $\equiv$ »
Импликация	«если A , то B . A влечёт B », « $\rightarrow$ » ¹⁸
Отрицание	«не», « $\neg$ »

Таблица 7

Таблицы истинности основных логических операций

X	Y	$\neg X$	$Xg\wedge Y$	$Xg\vee Y$	$X\rightarrow Y$	$X\leftrightarrow Y$
Л	Л	И	Л	Л	И	И
Л	И	И	Л	И	И	Л
И	Л	Л	Л	И	Л	Л
И	И	Л	И	И	И	И

Противоречивое исчисление — исчисление, в котором существуют формулы, выводимые в этом исчислении вместе со своими отрицаниями.

Формула α является выводимой из $\alpha_1, \alpha_2, \dots, \alpha_n$, если любая, но одна и та же модель этих формул (набор значений) является также моделью формулы α . Символ выводимости пишется как « $\vdash$ ».

Модель формулы — та предметная среда, значениями которой конкретизируется формула. Очевидно, что одна и та же абстрактная логическая формула может быть основой для построения модели некоторой среды. Поэтому говорят об абсолютной и относительной выполнимости, о тождественной (для всех моделей) и относительной (выполняется только для конкретной модели) истинности.

Семантика формулы — смысловая интерпретация формул формальной логики, т.е. сопоставление переменных в формулах со

¹⁷Гипотезы часто называют «теоремами», отсюда и термин «автоматическое доказательство теорем». Встречая этот термин, студенты путаются, полагая, что должно происходить автоматическое порождение некоего нового аналитического доказательства. На самом деле подразумевается обыкновенный рекурсивный логический вывод, чаще всего при помощи резолюций. Результатом такого вывода являются «Да» или «Нет» в зависимости от того, удалось ли за разумное число шагов получить верное равенство или пустую формулу.

¹⁸Иногда обозначают « $\supset$ ».

свойствами объектов среды и отношений между этими формулами со свойствами объектов. Семантика логики высказываний и фраз на естественном языке практически совпадают. Семантика логики предикатов такова, что не всегда можно точно сопоставить формуле правильно построенное предложение на естественном языке.

Выполнимая формула — существуют наборы значений ее аргументов, на которых она принимает значение «Истина».

Невыполнимая формула — на всех наборах значений своих аргументов принимает значение «Ложь».

Общезначимая формула (она же тавтология, обозначается « $\models$ ») — формула, абсолютно (т. е. на всех моделях) истинная на всех наборах значений своих аргументов. В логике высказываний известно много общезначимых формул (законов, аксиом, логики высказываний).

Элементарные (атомарные) высказывания — простые высказывания, например: «светает», «дом белый» и т. п.

Конstituента (полная конъюнкция) — элементарная конъюнкция, в которую по одному разу входит каждая переменная, определяющая состояние среды. Например, среда «слепой дождь» — идет дождь и светит солнце ($x_1 \wedge x_2$), или «буря» — сильный ветер и «света белого не видно» ($x_3 \wedge \neg x_4$).

Квантор — некоторый способ приписать наличие каких-либо свойств целому множеству объектов: $\forall$ (квантор общности), $\exists$ (квантор существования). Переменные, которые записываются за знаком квантора в скобках, — связанные; переменные, которые ни один квантор не связывает, — свободные. Наличие кванторов, а также возможность оперировать в качестве аргументов не только с переменными, но и с высказываниями — одно из основных отличий исчисления предикатов от исчисления высказываний. В свою очередь исчисление предикатов первого порядка отличается от ИП второго порядка тем, что если при ИП первого порядка мы устанавливаем кванторы над переменными, то в ИП второго порядка мы устанавливаем кванторы над самими отношениями (т. е. предикатами). Так, например, в логике первого порядка мы можем сформулировать запрос: «Какие переменные X , Y , Z и т. д. находятся в некотором отношении R ?» Именно такой пример приведен в главе «Краткие сведения о Prolog». В случае исчисления второго порядка возможен запрос: «В какие отношения входят некоторые переменные X , Y и т. д.» Применительно к тому же примеру получили бы ответ «likes»¹⁹.

¹⁹Большинство современных систем ИИ, основанных на логической модели, реализуют ИП первого порядка, и только отдельные системы (Prolog) реализуют отдельные элементы ИП второго порядка.

4.3.1. Методы автоматического доказательства теорем (исчисление предикатов). Некоторые, наиболее важные аксиомы.

Коммутативность: $\alpha_1 \& \alpha_2 \equiv \alpha_2 \& \alpha_1$, $\alpha_1 \vee \alpha_2 \equiv \alpha_2 \vee \alpha_1$.

Дистрибутивность: $\alpha_1 \& (\alpha_2 \vee \alpha_3) \equiv (\alpha_1 \& \alpha_2) \vee (\alpha_1 \& \alpha_3)$.

Ассоциативность: $\alpha_1 \& (\alpha_2 \& \alpha_3) \equiv (\alpha_1 \& \alpha_2) \& \alpha_3$,
 $\alpha_1 \vee (\alpha_2 \vee \alpha_3) \equiv (\alpha_1 \vee \alpha_2) \vee \alpha_3$.

Законы Де Моргана:

$\neg(\alpha_1 \& \alpha_2) \equiv (\neg \alpha_1) \vee (\neg \alpha_2)$,

$\neg(\alpha_1 \vee \alpha_2) \equiv (\neg \alpha_1) \& (\neg \alpha_2)$,

$\neg(\neg \alpha_1) \equiv \alpha_1$.

Аксиомы классического исчисления высказываний (ИВ):

$\alpha \supset (\beta \supset \alpha)$,

$(\alpha \supset \beta) \supset (\alpha \supset (\beta \supset \gamma)) \supset (\alpha \supset \gamma)$,

$(\alpha \wedge \beta) \supset \alpha$,

$(\alpha \wedge \beta) \supset \beta$,

$\alpha \supset (\alpha \vee \beta)$,

$\beta \supset (\alpha \vee \beta)$,

$\alpha \supset (\beta \supset (\alpha \vee \beta))$,

$(\alpha \supset \gamma) \supset ((\beta \supset \gamma) \supset ((\alpha \vee \beta) \supset \gamma))$,

$(\alpha \supset \beta) \supset ((\alpha \supset \neg \beta) \supset \neg \alpha)$,

$\neg \neg \alpha \equiv \alpha$.

Правила вывода.

1. Из истинности условия импликации и истинности самой импликации следует истинность следствия импликации (модус поненс²⁰): $\alpha, \alpha \supset \beta \vdash \beta$.
2. Из формулы $\alpha(p)$ выводима формула $\alpha(P)$, получающаяся подстановкой P вместо p (правило подстановки): $\alpha(p) \vdash \alpha(P)$.

При создании интеллектуальных программ, основанных на модели математической логики, кроме этих двух правил широкое распространение получили еще шесть. Это специальные правила вывода, позволяющие оперировать не самими логическими законами в классической записи, а некоторыми ценными следствиями:

- из истинности конъюнкции следует истинность любого ее конъюнкта (исключение конъюнкта);
- из списка истинных формул следует истинность их конъюнкций (введение конъюнкции);
- из истинности формулы следует истинность ее дизъюнкции с любыми другими формулами (введение дизъюнкции);

²⁰ Modus ponens (лат.) — «правило отделения».

- из истинности двойного отрицания формулы следует истинность ее самой (исключение двойного отрицания);
- из истинности дизъюнкции и отрицания одного из ее дизъюнктов следует истинность формулы, получающейся из дизъюнкции удалением этого дизъюнкта (простая резолюция или удаление дизъюнкта): $\alpha \vee g, \neg\beta \vdash \alpha$;
- из истинности двух дизъюнкций, одна из которых содержит дизъюнкт, а другая его отрицание, следует формула, являющаяся дизъюнкцией исходных формул без упомянутого дизъюнкта его отрицания (резолюция): $\alpha \vee \beta, \neg\beta \vee \gamma \vdash \alpha \vee \gamma$ или, эквивалентно, $\neg\alpha \supset \beta, \beta \supset \gamma \vdash \neg\alpha \supset \gamma$.

Последние два принципа в свое время (1966 г.) совершили «небольшой переворот» в прикладной логике. Оценка истинности формулы в современных программах ИИ происходит одним из следующих методов:

- 1) оценка через преобразование, упрощение и приведение к нормальным формам;
- 2) оценка путем логического вывода из системы аксиом;
- 3) оценка методом редукции;
- 4) оценка методом опровержения.

В ряде случаев для определения тождеств удобен так называемый алгоритм редукции. Алгоритм основан на доказательстве путем приведения к абсурду. Метод особенно хорош, когда формула содержит много импликаций. Согласно принципу дедукции, вопрос о выводимости (не выводимости) некоторой формулы сводится, в конечном счете, к анализу невыполнимости множества ее дизъюнктов. Множество дизъюнктов невыполнимо тогда и только тогда, когда логическим следствием из него является пустой дизъюнкт. Т. е. невыполнимость формулы можно проверить, порождая логические следствия из нее до тех пор, пока не получится пустой дизъюнкт. Метод, позволяющий получить логические следствия из множества дизъюнктов, основан на применении принципа резолюций.

П р и м е р. Даны формулы A, B, C . Дизъюнкты: (1) $(A + C)$ и (2) $(B + \bar{C})$, будем считать их истинными.

1. Предположим, $C = \text{«Истина»}$, тогда первый дизъюнкт $(A + C) = I$ при любом A . Подставив $C = I$ во второе выражение, имеем $B + I$, следовательно $B = I$, поскольку принято, что весь дизъюнкт истинен (пытаемся доказать истинность формулы).

2. Предположим, $C = I$, тогда второй дизъюнкт истинен при любом B , а первый принимает вид $A + I$, следовательно $A = I$.

Таким образом, вне зависимости от интерпретации C : $(A + B) = I$. Новый дизъюнкт (резольвента) добавляется в базу знаний. Он позволяет исключить контрарные формулы C и $\bar{C}$.

П р и м е р. Дизъюнкты $(p + q + \bar{r})$ и $(t + r)$, резольвента $(p + q + t)$.

П р и м е р. Дизъюнкты (p) и $(\bar{p})$ дадут в сумме пустой дизъюнкт.

П р и м е р. Доказать выполнимость системы $H = (p + q, p + r, \bar{q} + \bar{r}, \bar{p})$.

Гипотезы:

$$1) p + q, \quad 3) \bar{q} + \bar{r},$$

$$2) p + r, \quad 4) \bar{p}.$$

Доказательство:

$$(p + q) + \bar{p} = q,$$

$$(p + r) + \bar{p} = r,$$

$$(\bar{q} + \bar{r}) + r = \bar{q},$$

$$q + \bar{q} = L(\text{ож}).$$

Следовательно, система невыполнима. То же самое можно было доказать на формуле p (а не q). Алгоритм (похож на решение системы алгебраических уравнений, не правда ли?) легко формализуется на ЭВМ. Проблема при этом состоит в возможности закливания машины вследствие порождения некоторого одного и того же дизъюнкта неограниченное число раз. Поэтому в машинные языки вводят специальную операцию «отсечение».

Во времена Дж. Буля математическая логика не имела того важнейшего прикладного значения, которое ей придается сейчас. С изобретением в 1918 г. Бонч-Бруевичем принципа триггерного кольца или просто триггера появилась возможность реализовать уравнения Буля, а следовательно, и часть логики Аристотеля в электрических и электронных устройствах.

Михаил Александрович Бонч-Бруевич (09.02.1888–07.03.1940). Изобретатель триггерного кольца (1918). Независимо от него открытие повторили английские ученые У. Экклз и Ф. Джордан (1919).

В 1909 г. окончил Инженерное училище в Петербурге, в 1914 г. — Офицерскую электротехническую школу. С 1922 г. профессор Московского высшего технического училища, с 1932 г. профессор Ленинградского института инженеров связи, ныне носящего его имя. В 1916–19 гг. создал отечественное производство электронных ламп. В 1918 г. возглавил Нижегородскую радиолaborаторию, объединившую лучших радиоспециалистов того времени. По заданию В. И. Ленина спроектировал первую в мире мощную (12 кВт) радиостанцию им. Коминтерна (Москва, 1922). В 1919–25 гг. создал конструкцию мощной генераторной радиолампы с водяным охлаждением и схемы радиотелефонных станций. В 1924–30 гг. руководил разработкой первой в мире коротковолновой направленной антенны и коротковолновых линии дальней радиосвязи. Занимался вопросами физики верхних слоев атмосферы и их практическим применением, в том числе в области радиолокации.

Примеров такого применения очень много, самый главный — это, конечно же, создание компьютеров. Набор микросхем, на которых они строятся, называют *элементарной логикой* или *элементной базой*.

Компьютер — это очень большая и сложная система логических уравнений. Системы попроще нашли свое применение в управлении техническими объектами, например, лифтами, разумно реагирующими на примерно одновременное нажатие кнопок вызова на разных этажах, панелях управления и индикации различных приборов, системах «защиты от дурака», когда, например, технологический режим работы устройства не допускает работы в двух режимах одновременно, равно как и полной остановки устройства: $X_1 * X_2 = 0$.

Развитие этой ветви систем ИИ активно шло до 60–70 гг. и позволяло получать новые научные результаты. Дальнейшее развитие потребовало модификации логики Буля, введения двух новых важных понятий: состояния системы и степени уверенности²¹.

4.3.2. Элементы теории нечетких множеств Л. Заде. Как Вы помните, Аристотель полагал, что любое рассуждение либо истинно, либо ложно, т. е. система «Маша → Паша» может находиться только в одном из двух возможных состояний: либо «Маша любит Пашу», либо «Маша не любит Пашу».

Разработчики микросхем элементарной логики обнаружили, что возможно, по крайней мере, три состояния. В самом деле, если состояние 1 (любит) кодируется высоким уровнем напряжения, а состояние 0 (не любит) низким, то как быть, если на данной схеме в некоторый момент времени вообще нет напряжения, либо его уровень ровно в середине между 0 и 1? Это третье состояние можно интерпретировать как «неизвестно, как Маша относится к Паше» или «Маша не знает, как ей относиться к Паше». А если уровень напряжения ближе к «любит», чем «не любит»?

Можно возразить, что это очевидная ошибка в работе устройства, долгое время так и считалось. Но как показал Л. Заде в своей работе, вышедшей в свет в 1965 г. в США, это очень полезная ошибка. Эта «ошибка» позволяет приблизить логику работы машины к логике рассуждений живого человека, научить машину «сомневаться» и «разносторонне оценивать варианты». Для этого необходимо не просто выделить возможные состояния системы, но и оценить степень уверенности в том, что система находится именно в этом состоянии.

²¹ После прочтения этой лекции рекомендуется выполнить практическую (лабораторную) работу «Элементы формальной логики ...»

Лотфи Заде (Lotfi A. Zadeh) — американец азербайджанского или иранского происхождения (род. 04.12.1921). В 1956 г. приглашен исследователем в институт современных исследований в Принстон (штат Нью-Джерси). В 1959 г. стал членом департамента инженерной электрики в университете Беркли. Впервые сформулировал понятие нечеткой логики в 1965 г. в статье «Нечеткие множества» (журнал «Информатика и управление»).

Существует легенда о том, что теория «нечетких множеств» была придумана Заде в результате спора со своим другом относительно того, чья жена более привлекательна. Термин «привлекательная» является неопределенным, что и вынудило Заде сформулировать концепцию, которая выражает нечеткие понятия в числовой форме.

Приведем пример. Какого цвета крокодил? Большинство из Вас, не задумываясь, ответит: «Зеленый». Подумав некоторое время, Вы скажете, что, возможно, бывают крокодилы и другого цвета, например коричневые или желтые. На худой конец Вы скажете, что в магазине «Детский мир» видели даже красного крокодила.

Итак, у системы «крокодил» характеристика «цвет» может находиться в одном из четырех возможных состояний: «зеленый», «коричневый», «желтый», «красный». Можете ли Вы объяснить, что такое «зеленый крокодил»? Если автоматическая система ИИ увидит два объекта, различающихся интенсивностью окраса, какой из них считать крокодилом? Оба? А если один из объектов будет красным? А где граница цвета красный?

Основная проблема применения логики Аристотеля и Буля как раз и состоит в том, что в реальном мире состояния «ложь» и «истина» часто недостижимы. Аристотель исследовал наиболее простые, элементарные отношения. Существуют и более сложные, которые могут принимать состояния «истина» и «ложь» в частном случае. Для того чтобы оценить сложное отношение как истинное или ложное, всегда приходится делать некоторые упрощения и допущения.

Например, отношение «любит». Истинность или ложность этого отношения на множестве двух элементов невозможно ни доказать, ни опровергнуть. Можно только приписать ему некоторую степень уверенности. Так, по мнению автора, крокодил является зеленым со степенью уверенности 0,8. Это значит, что каждый крокодил является темно-зеленым и большая часть его тела именно такого цвета. У крокодила длинные лапы с уверенностью 0,4 — это значит, что у каждого крокодила довольно короткие лапы.

Если крокодил красный, то это частный случай игрушечного крокодила со степенью уверенности 1, равно впрочем, как и если крокодил синий, белый, в полосочку или даже в горошек. Какого же цвета игрушечный крокодил? Цвет такого крокодила многозначен. Он может быть любым, даже темно-зеленым, как настоящий. Тогда

какую же степень уверенности сообщить системе ИИ? Сможет ли робот на основе ИИ отличить игрушечную собачку от игрушечного крокодила, а обоих вместе от живых объектов?

В данном случае мы говорим о степени вероятности или просто вероятности некоторого события. Из ста игрушечных крокодилов 70 будут зелеными, 20 желтыми, 5 красными и еще пять всех других цветов и оттенков. Значит, и вероятность цвета зеленый — 0,7, желтый — 0,2, красный — 0,05. Зеленый — 0,7 означает, что объект может быть любого цвета, но, скорее всего, он зеленый.

Итак, мы рассмотрели основное отличие понятий *степень уверенности* и *вероятность* события. Вероятностные модели подробнее будут рассмотрены в следующих лекциях. Остановимся на нечетких множествах Лотфри Заде.

Нечеткое множество задается парами вида $x_1 | \mu_X(x_1)$, где x_1 — элемент нечеткого множества X , а $\mu_X(x_1)$ — степень принадлежности элемента x_1 к нечеткому множеству X . Значение $\mu_X(x_1)$ изменяется в интервале $[0..1]$. Например, некоторое нечеткое множество A будет задаваться выражением вида

$$A = \{(x_1|0,1), (x_2|1), (x_3|0,6)\},$$

$$x_1 = 17 \text{ (лет)}, \quad x_2 = 24, \quad x_3 = 49$$

$$\text{или сразу } \{(17|0,1), (24|1), (49|0,6)\}.$$

Множество A (рис. 25) могло бы, например, иметь смысл ответа на вопрос: «Сколько лет молодому человеку?» Люди из разных поколений ответят на него по-разному, вкладывая в понятие «молодой» разные объем и содержание. Для подростков «молодой человек» — это тот, кому «мало лет», для людей постарше — тот, кому «немного лет» и т. д. Пенсионеры, в противоположность школьникам, отнесли бы шестнадцатилетнего юношу к категории «подростки», а безнадежно устаревшего на взгляд школьника тридцатилетнего преподавателя — к категории «довольно молодые люди». Мнения разных поколений сошлись бы абсолютно только относительно «молодого человека в возрасте 18–25 лет». Подобная нечеткость всегда имеет место в реальных инженерных задачах, поскольку предельные понятия «истинность — 1» и «ложь — 0» существуют, скорее, только в теории. Метод получения «меры принадлежности» некоторого

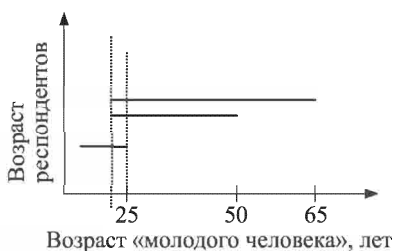


Рис. 25. Лингвистическая функция

элемента к множеству — тема отдельного разговора. Четких математических методов здесь нет, применяются так называемые методы принятия решений.

Другим важным вопросом является способ оперирования с нечеткими (лингвистическими) оценками (мерами принадлежности). Предположим, у Вас имеется некоторая цель A , достижение которой зависит от достижения подцелей A_1 и A_2 , причем полное достижение подцелей возможно только в частном случае. Как вычислить меру достижения цели A ? Подобная ситуация возникла, например, при разработке деревьев целей информационно-образовательной среды «Chopin» (рис. 9, 10, 86). Как сравнить два нечетких множества? Множества A и B равны, если $\forall x \in M \mu_A(x) = \mu_B(x)$, где M — множество элементов, которые есть и в A и в B . Множества A и B дополняют друг друга, если $\forall x \in M \mu_A(x) = 1 - \mu_B(x)$. Пересечение A и B : $\forall x \in M \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x))$. Объединение: $\forall x \in M \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x))$.

Введенные операции дополнения, пересечения и объединения удовлетворяют законам коммутативности, ассоциативности, идемпотентности, дистрибутивности, действия с константами де Моргана, двойного отрицания (дополнения) и поглощения, т. е. выполняют все основные законы теории множеств. Достаточно интересно выглядит геометрическая интерпретация операций на нечетких множествах (диаграммы Эйлера). На рис. 26 представлено множество оценок возраста для понятия «молодой человек» и его дополнение.

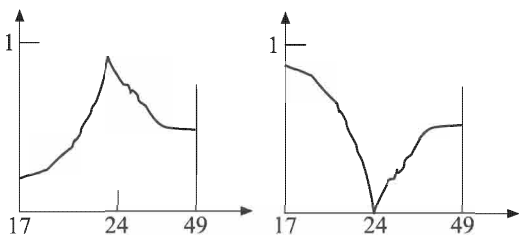


Рис. 26. Нечеткое множество и его дополнения

На нечетких множествах определены также операции алгебраического произведения и суммы:

$$\forall x \in M \quad \mu_{A+B}(x) = \mu_A(x) \bullet \mu_B(x);$$

$$\forall x \in M \quad \mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \bullet \mu_B(x).$$

Применительно к задаче оценки степени достижения учебных целей мы использовали (см. рис. 9, гл. 1) операцию пересечения, в то время как в системе тестирования при оценке ответа на вопрос

применили операцию объединения. При принятии решения о «самом лучшем», «самом плохом» студенте воспользовались операцией дополнения. Современные экспертные системы все чаще вместо классической (Аристотелевой) логики используют логику Заде²².

Математическая логика с течением времени развилась в самостоятельно ценную науку, исследования в области троичной и многозначной логики заложили фундамент развития информатики на десятки, если не сотни лет вперед²³.

²² Многим читателям логика Заде окажется уже знакомой по английскому «fussy logic» или американскому «fuzzy».

²³ Здесь рекомендуется сдать тест «Основные элементы логической модели . . .»

ГЛАВА 5

ПРОДУКЦИОННАЯ МОДЕЛЬ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

5.1.1. Описание предметной области правилами и фактами¹.

Для того чтобы реализовать описанные выше принципы логического вывода, в автоматизированной системе необходим специальный машинно-ориентированный язык². Одним из наиболее простых и эффективных машинно-ориентированных языков для описания логических задач являются правила продукций.

«Продукция» представляет собой выражение вида:

если A (условие), то B (действие), постусловие C.
ядро

Под условием понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под действием — действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и целевыми (терминальными), завершающими работу системы). Постусловие описывает действия и процедуры, которые необходимо выполнить после реализации действия. Например, после покупки некоторой вещи в магазине необходимо в описи товаров уменьшить количество вещей такого типа на 1.

Продукционная система, таким образом, представляет собой базу знаний (правил продукций) и машину вывода — специальную программу «сопоставления по образцу». В зависимости от того, какие использованы продукции и каковы правила вывода, получаются различные продукционные системы.

Ядра продукций можно классифицировать по различным основаниям. Прежде всего различают детерминированные и недетерминированные ядра. При актуализации детерминированного ядра и выполнимости условия правая часть (действие) выполняется с неизбежностью (и степенью уверенности), в недетерминированных — с вероятностью.

Если A , то, возможно, B с вероятностью α^3 .

¹ Перед прочтением этой лекции рекомендуется выполнить практическую (лабораторную) работу «Программирование на языке Prolog».

² Безусловно, программная реализация логического вывода возможна на любом алгоритмическом языке, однако это будет очень сложная реализация. Необходимо, чтобы программист думал в первую очередь о природе решаемой задачи, а не деталях ее реализации, а для этого необходим универсальный абстрактный язык описания.

³ В классической логике $\alpha = 1$ либо $\alpha = 0$.

Если A , то B с коэффициентом уверенности α (детерминированная продукция).

Продукции также могут быть однозначными и альтернативными. Для альтернативных правил в правой части ядра указываются «альтернативные возможности выбора», которые оцениваются «весами выбора» — $\alpha_1, \alpha_1, \dots, \alpha_n$.

Если A , то «чаще всего» надо делать B_1 , «реже» B_2 (вероятностные оценки).

$$\text{Если } A, \text{ то } \begin{cases} B_1 & \text{с уверенностью } \alpha_1; \\ B_2 & \text{с уверенностью } \alpha_2; \\ B_n & \text{с уверенностью } \alpha_n. \end{cases}$$

В качестве $\alpha_1, \alpha_1, \dots, \alpha_n$ используются вероятностные оценки, лингвистические оценки, экспертные оценки и т. п. Метод их получения — тема для отдельного изучения.

Отдельно выделяют прогнозирующие продукции, в которых описываются последствия, ожидаемые при актуализации A , например: если A , то с вероятностью p можно ожидать B .

Продукционная база знаний (БЗ) состоит из набора фактов и правил. Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения — к данным). Данные — это исходные факты, на основании которых запускается машина вывода, т. е. программа, перебирающая правила из базы.

А б с т р а к т н ы й п р и м е р БЗ (рис. 27).

Раздел «Факты»:

дорога (A, B);

дорога (B, C);

дорога (B, D);

дорога (C, D).

Раздел «Правила»:

прямая дорога (X, Y), если дорога (X, Y)
или дорога (Y, X);

транзитная дорога (X, Y), если дорога (X, Z) и дорога (Z, Y);

есть дорога (X, Y), если прямая дорога (X, Y) или транзитная дорога (X, Y).

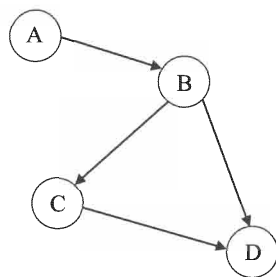


Рис. 27. Абстрактная карта дорог: графическая интерпретация отношений между именами

Пример задачи, решаемой при помощи такой БЗ: «Установить, есть ли дорога между пунктами A и D ».

Ход решения (рис. 28).

1. На вход машины вывода (далее МВ) пользователь подаст цель: «Есть дорога (A, D)».

«Есть дорога (A,D)»	«Прямая дорога (A,D)»	«Дорога»	«Транзитная дорога (A,D)»
	«Есть дорога (A,D)»	«Прямая дорога (A,D)»	«Есть дорога (A,D)»
		«Есть дорога (A,D)»	
Шаг 1	Шаг 3	Шаг 4	Шаг 6

Рис. 28. Работа стека при решении логической задачи

2. МВ осуществит поиск текстовой строки «Есть дорога» в файле, хранящем БЗ.
3. МВ втолкнет цель «Есть дорога» в стек⁴ целей и приступит к поиску цели «Прямая дорога».
4. МВ втолкнет цель «Прямая дорога» в стек целей и приступит к поиску цели «Дорога».
5. МВ просмотрит все текстовые строки «Дорога (A, B)», «Дорога (B, C)» и т. д. и обнаружит, что искомой цели «Дорога (A,D)» среди них нет.
6. МВ вытолкнет из стека цели «Дорога», «Прямая дорога» и «Есть дорога». Раскроет цель «Есть дорога» и приступит к поиску цели «Транзитная дорога», предварительно втокнув цель «Есть дорога» в стек.
7. Цель «Транзитная дорога» будет удовлетворена, поскольку действительно есть дорога «A — B — D».

В данном примере показан так называемый «обратный вывод» — некоторая цель подтверждена при помощи имеющихся правил и данных.

Прямой вывод имел бы место, если бы требовалось породить все возможные на данной карте маршруты, в этом случае цель могла бы выглядеть как «Дорога (X, Y)».

Как видите, алгоритмически поиск решения (имитация мышления человека) представляет собой не более чем примитивный поиск подстроки в строке в цикле до первого решения, либо в бесконечном цикле, либо до первого повторения полученного решения. Для экономии времени и ресурсов ЭВМ важно оптимально управлять этим поиском. Графическое представление пространства состояний значительно упрощает эту задачу (рис. 29).

Как видите, возможны два решения «A — B — D» и «A — B — C — D»⁵. Первое явно короче, но ведь находясь в пункте «A», мы об этом не знаем!

⁴ Стек называется структура для хранения данных, принимающая и выдающая элементы данных по принципу LIFO: «первым вошел — последним вышел».

⁵ Такая запись решения называется «путь решения задачи».

Вершину «А» называют «корнем дерева», «целевой вершиной» или «глобальной целью».

Вершины «В» и «С» — вычисляемыми (раскрываемыми, промежуточными) вершинами.

Вершину «D» — терминальной⁶, т. е. завершающей.

Ребра, соединяющие вершины, имеют смысл присоединенных процедур, которые необходимо выполнить, чтобы совершить переход в следующее состояние (вершину).

Процесс применения присоединенных процедур называют порождением вершин или перебором вариантов. При порождении новой вершины обязательно запоминается указатель на старую. В конце перебора совокупность этих указателей образует путь решения задачи, который записывается вместе с именами выполненных присоединенных процедур, например, так:

$$A = (r1(r2(r3))), \quad \text{где } r \text{ — ребра графа; или } A(B(C(D))).$$

В зависимости от того, раскрывают ли каждую возможную «веточку» дерева целей сразу до терминальных фактов, либо только на некоторую заданную глубину, различают две основные стратегии слепого⁷ перебора: «в ширину» и «в глубину».

В первом случае вершины раскрываются в том же порядке, в котором они порождаются (*breadth — first process*). Во втором случае на каждом шаге первой раскрывается вершина, которая была построена последней (*depth — first process*).

В случае, если существует некоторая дополнительная информация о предметной области, которая позволяет делать суждения о характере графа пространства состояний и расположения цели, говорят об эвристическом⁸ поиске. Эвристическая информация, опирающаяся, как правило, на предыдущий опыт, позволяет выполнять поиск в наиболее перспективных направлениях.

Говоря о графе, будем рассматривать только один наиболее простой его тип — граф типа «дерево». Дерево — граф, каждая вершина которого имеет только одну вершину, непосредственно предшествующую

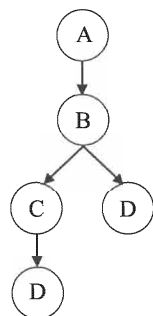


Рис. 29. Абстрактная карта дорог: графическая интерпретация отношений между именами

⁶ Для медика «терминальные факты» — это несколько иное, следует быть осторожным при общении с экспертом.

⁷ В этих процессах расположение целевой вершины не влияет на порядок раскрытия.

⁸ Что означает «служащий открытию».

ющую ей (родительскую), за исключением вершины-корня, которая предшествующих вершин не имеет.

5.1.2. Метод полного перебора в ширину. Вершины раскрываются в том порядке, в котором они строятся (рис. 30). Основным алгоритм состоит в выполнении следующих действий.

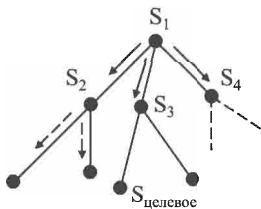


Рис. 30. Фрагмент дерева перебора в ширину

1. Начальная вершина раскрывается до тех пор, пока ее можно раскрыть, применяя один и тот же (или разные, смотря по условию) оператор. При этом образуются вершины первого уровня: $S_2, S_3 \dots$ Они раскрываются в свою очередь и образуются вершины второго уровня и т. д.

2. Расставляются указатели⁹, ведущие от новых вершин к корню.

3. Проверяется, нет ли среди полученных вершин целевой. Если есть, то формируется решение на основе соответствующего оператора. Если целевых вершин нет, то рассматривается первая порожденная вершина и к ней применяется тот же алгоритм. После чего переходят ко второй и т. д., пока среди получаемых вершин не окажется целевой.

Полный перебор в ширину гарантирует нахождение целевой вершины как раз потому, что перебор полный. Путь достижения цели, вообще говоря, может быть много. В этом случае имеется возможность выбрать наикратчайший (самый дешевый/самый легкий/быстрый и т. п.) путь. Но может быть случай, когда граф поиска окажется бесконечным, и тогда этот алгоритм никогда не кончит работу. Имеется класс задач, для которых метод эффективен. Классическим примером является задача «лабиринтного поиска»¹⁰, впервые решенная К. Шенноном. Здесь пространство различных вариантов действия невелико: «повернуть вправо», «влево», «вперед», и решение лежит на некоторой заданной (обычно небольшой) глубине.

5.1.3. Метод полного перебора в глубину. При переборе в глубину (рис. 31) прежде всего следует раскрывать те вершины, которые были построены последними. Первой, а следовательно, и последней раскрываемой вершиной является корневая. Процесс всегда будет

⁹ Условные имена, буквы, цифры, имена операторов, расстояния, стоимость, вес и т. д.

¹⁰ В учебниках часто приводится в виде задачи про «обезьяну и банан», реже про «мышь в лабиринте».

идти по самой левой ветви вершин. Чтобы как-то ограничить перебор, вводится понятие глубины вершины в дереве перебора. Глубина корня дерева равна нулю, а глубина любой последующей вершины равна единице плюс глубина вершины, непосредственно ей предшествующей. Наибольшую глубину всегда будет иметь та вершина, которая должна быть в этот момент раскрыта. Если образующийся путь оказывается бесполезным, т.е. при заданной глубине раскрытия целевой вершины не получилось, необходимо вернуться в вершину, предшествующую раскрытой, и попытаться еще раз применить к ней операцию раскрытия¹¹. И так до тех пор, пока не будет получена целевая вершина.

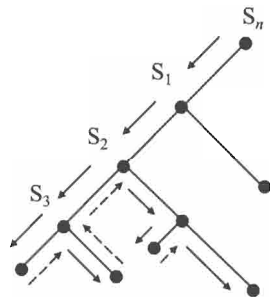


Рис. 31. Дерево переборов в глубину с указанием бэк-трекинга

Возврат осуществляется с помощью указателей. Как только в процессе порождения вершин достигается заданная граничная глубина, раскрывается вершина наибольшей глубины, не превышающая этой границы¹². Общая схема перебора в глубину показана на рис. 31.

Алгоритм перебора в глубину состоит в следующем.

1. Раскрывается начальная вершина, соответствующая начальному состоянию.
2. Раскрывается первая вершина, получаемая в результате раскрытия начальной. Ставится указатель.
3. Если она раскрывается, то следующей будет раскрываться вновь порожденная вершина. Если вершина не раскрывается, то процесс возвращается в предыдущую вершину.
4. По получении целевой вершины процесс раскрытия заканчивается и по указателям строится путь, ведущий к корню. Соответствующие дугам операторы образуют решение задачи.
5. Если для заданной глубины раскрытия целевая вершина не находится, то весь процесс повторяется снова, а в качестве новой вершины рассматривается самая левая из полученных на предыдущем этапе.

5.1.4. Эвристические методы поиска в пространстве состояний. Методы полного перебора гарантируют решение задачи, если

¹¹ Этот процесс (возврата на несколько шагов назад) называется откатом назад или бэктрекингом.

¹² Так называемое «программирование с обратным слежением» (back – track programming).

оно существует, а при наличии нескольких решений гарантирует оптимальное. Однако на практике эти методы используются для решения лишь небольших по размерности графа состояний задач. Для реальных случаев чаще всего используется дополнительная информация, основанная на предыдущем опыте или полученная на основании теоретических выводов. Такая информация называется

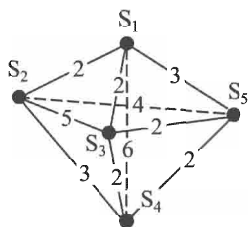


Рис. 32. Граф состояний задачи коммивояжера

эвристической, а организованная в правила — *эвристическими правилами* или *эвристиками*. Эвристическая информация носит сугубо специальный характер и может применяться только в рамках данной задачи, в лучшем случае в рамках задач данного класса. Эвристическая информация делает перебор упорядоченным¹³.

В качестве примера рассмотрим известную задачу о коммивояжере. Бродячий торговец должен побывать в каждом из N городов в точности по разу и возвратиться в исходный город. Желательно, чтобы маршрут был минимальным по протяженности (рис. 32).

Фрагмент пространства состояний представлен на рис. 33. Начинаем перебор в ширину, и на первом же уровне получаем возможные пути разной длины: 2, 2, 6, 3. Если исходить из эвристики «на каждом шаге выбирать путь минимальной длины», следует сделать шаги $S_1 - S_2$ и $S_2 - S_3$, затем $S_3 - S_4$ или $S_3 - S_5$, затем $S_4 - S_5$ и т.д. При желании читатель сам может достроить дерево и убедиться, что путь, найденный таким методом, не всегда будет самым коротким.

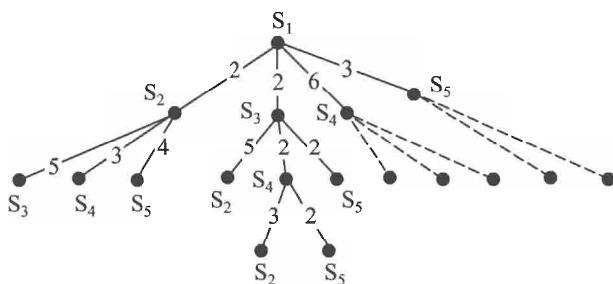


Рис. 33. Фрагмент дерева перебора для задачи коммивояжера

Более правильной была бы эвристика «выбирать так, чтобы минимальным был суммарный путь». Пояснение этого принципа (принцип Р. Беллмана) не входит в курс ИИ.

¹³ Речь идет, прежде всего, о нескольких хорошо известных методах: «ветвей и границ», «методе динамического программирования Беллмана», методах на основе принципа максимума Л. С. Понтрягина и др.

Граф полного перебора, включающий все возможные пути коммивояжера, будет содержать $(N - 1)!$ вариантов. Впрочем, если отбросить обратные пути, то всего $(N - 1)!/2$. На реальных задачах подобного типа, часто решаемых при создании электронных карт (разместить предатчики пейджинговой связи так, чтобы расстояние от любой точки заданной зоны не превышало X , разместить банкоматы так, чтобы минимизировать путь инкассатора (не обязательно по протяженности) и т.п.), N обычно равняется нескольким десяткам объектов, что для современных ЭВМ создает серьезную проблему.

Эвристические алгоритмы чаще всего применяются не для поиска единственно правильного (оптимального решения), а для поиска первого решения, удовлетворяющего некоторому критерию при заданных ограничениях. Так, например, человек, отправляясь в магазин, не ставит себе задачи купить самое дешевое молоко, он готов купить молоко не ниже заданного качества и не дороже некоторой (возможно нечеткой) оценки, и твердо уверен в том, что не станет тратить на покупку более 5 минут (обходить несколько магазинов).

В 60–70 гг. прошлого века разработаны десятки действительно полезных эвристик, не всегда являющихся очевидными и доказуемыми, но позволяющих значительно сократить перебор или получить выигрыш в качестве решения. Их применение возможно не только в продукционной модели. Известен, например, «метод встречной волны» или «двунаправленный перебор от цели к фактам и обратно». Последний метод очень хорош для анализа противоречивости построенных деревьев.

5.1.5. Решение задач методом разбиения на подзадачи. Графическое представление является очень важным для понимания как самой задачи, так и метода ее решения.

По мнению автора, в стандартной теории программирования на языках ИИ имеется существенный пробел, связанный с недостаточным осознанием важности графического (визуального) программирования. Применительно к языкам ИИ слово «визуальный» должно означать совсем не «визуальное проектирование форм и экранов», как в обычных алгоритмических языках, а визуальное проектирование дерева целей программы с последующим автоматическим генерированием ввода. На момент написания книги широко известных систем, реализующих этот принцип, не существовало. В 1997–1999 гг. совместно со студентами специальности ИСЭ и ПОВТ автором был разработан специальный редактор, позволяющий автоматизировать написание программ на Prolog — подобном языке EDL — язык описания путей достижения учебных целей в информационно обучающей среде Chopin (Барнаул, АлтГТУ, 1996–2002).

Графы, подобные представленным выше, являются простейшими с той точки зрения, что отражают взаимосвязи только между объектами предметной области и не отражают взаимосвязи между действиями. Другими словами, на рассмотренных ранее графах никак не заданы отношения между дугами (присоединенными процедурами). Такими отношениями могут быть, например, отношения последовательности и одновременности.

5.1.6. Представление задачи в виде И-ИЛИ графа. Между полученными при разбиении подзадачами могут быть отношения согласованности (одновременности) их решения (отношение «И»), или отношение альтернативности (отношение «ИЛИ»), рис. 34.

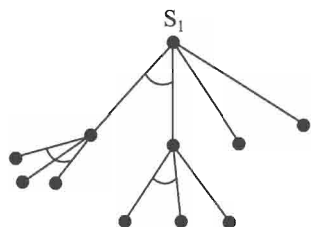


Рис. 34. Фрагмент «И-ИЛИ» дерева

Отношения типа «И» отмечают дугой (иногда двойной), связывающей ребра графа. Для единообразия представления можно ввести дополнительные (фиктивные) вершины, которые группировали бы подзадачи типа «И» и «ИЛИ» под своей собственной родительской вершиной.

Таким образом, исходная задача представляется подзадачами, имеющими альтернативный ИЛИ-характер, а сами подзадачи, в свою очередь, — подзадачами с отношениями типа И. Альтернативные вершины называют «ИЛИ»-вершинами.

Основная цель поиска на «И-ИЛИ» графе — показать разрешимость вершины S_i . Вершина является *разрешимой*, если выполняется одно из следующих условий:

- 1) вершина S_i является заключительной (терминальной);
- 2) следующие за S_i вершины являются вершинами типа «ИЛИ» и при этом хотя бы одна из них разрешима;
- 3) следующие за S_i вершины являются вершинами типа «И» и при этом каждая из них разрешима.

Решающим графом называется подграф, состоящий из разрешимых вершин с корнем в начальной вершине.

В случае если у вершины «И-ИЛИ» графа, не являющейся заключительной, нет следующих за ней вершин, такая вершина называется *нерешимой*.

Порождение новых вершин (редукция задачи) выполняется путем применения обобщенного оператора (т. е. каких-либо операторов из множества допустимых; обозначим, например, $g \in G$). Применение оператора g к описанию задачи порождает всю структуру «И-ИЛИ» графа (графа редукции).

5.1.7. Управление системой продукций. При выполнении условия применимости ядер продукции для группы продукции возникает дилемма выбора той продукции, которая в данной ситуации будет активизирована. Решение этой задачи возлагается на систему управления системой продукций. Если ИС реализована на ЭВМ с параллельной архитектурой, то из фронта готовых продукций может выбираться не одна продукция, а столько, сколько параллельных ветвей может одновременно в данной ситуации выполнять ЭВМ. Эта задача характерна не только для систем продукций, но и для всех систем, где необходимо выполнение параллельных процессов. Возможны два способа ее решения: централизованный и децентрализованный. При первом решение об актуализации принимается специальной системой управления, при втором — складывающейся в данный момент ситуацией. Существуют некоторые основные стратегии управления выполнением продукций и поиском.

5.1.8. Достоинства и недостатки продукционной модели. Продукционная модель (все еще) чаще других применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода. Имеется большое число программных средств, позволяющих реализовать продукционный подход:

- языки Prolog, Lisp¹⁴, LOGO¹⁵, OPS;
- «оболочки» или «пустые» ЭС, а именно EMYCIN, EXSYS, ESISP, ЭКСПЕРТ;
- инструментальные системы ПИЭС, СПЭИС и др.

Среди основных достижений советской науки в данной области — программа, автоматически доказывающая теоремы¹⁶.

Продукционные модели имеют, по крайней мере, два серьезных недостатка. При большом числе продукций становится сложной проверка непротиворечивости системы продукций. Это заставляет при добавлении новых продукций тратить много времени на проверку. Системе присуща недетерминированность (неоднозначность выбора выполняемой продукции из фронта активизируемых продукций),

¹⁴ Разработан проф. Джоном Маккартни (США, 1961 г.). LISP — List processing language — язык обработки списков.

¹⁵ Создан проф. математики М.И.Т. Сеймуром Пайпертом на основе LISP, 1969 г.

¹⁶ «АЛИЕВ ЛОМИ» — Ленинградское отделение математического института им. В.А. Стеклова. Она основана на оригинальном обратном выводе С.Ю. Маслова, аналогичном методу резолюций Робинсона.

возникают принципиальные трудности при проверке корректности работы системы. Считается, что если в ИС число продукций достигнет тысячи (по другим источникам, полутора тысяч), то мало шансов, что система продукций во всех случаях будет правильно функционировать.¹⁷

¹⁷ Для закрепления материала рекомендуется выполнить практическую (лабораторную) работу «Работа с оболочкой ЭС» и сдать тест «Основы языка Prolog».

ГЛАВА 6

ФРЕЙМЫ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Проблема представления знаний¹ является ключевой во всех направлениях исследований по искусственному интеллекту, начиная от задач распознавания образов и до задачи понимания естественного языка. Первопричина сложности создания машинной модели мира кроется в многообразии этого мира. Одну из наиболее психологически обоснованных и практически ценных моделей предложил Марвин Мински. В 1974 г. вышла его книга «Фреймы² для представления знаний», ставшая в определенном смысле «подведением черты» под работами практиков и крупных теоретиков как в области психологии, так и компьютерного моделирования психики (рис. 35).

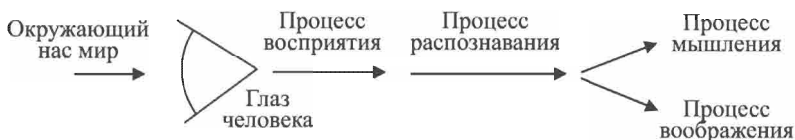


Рис. 35. Психика с точки зрения теории фреймов

Marvin Minsky — профессор компьютерных наук в Массачусетском технологическом институте, США. Специалист в области искусственного интеллекта, когнитивной психологии, программирования для машины Тьюринга. Работал также в области вычислительной геометрии, вычислительной семантики, машинного восприятия, символического обучения. Привлекался к космическим проектам. Один из основателей робототехники и удаленного управления: лично разработал тактильные и зрительные датчики, механические части первых роботов. В 1951 г. создал одну из первых нейронных сетей. В 1959 г. основал лабораторию ИИ — M.I.T. Много работал над проблемами представления знаний для понимания языков и визуального восприятия. В 70-е гг. создал собственную психологическую теорию интеллекта. Всего опубликовал около 10 книг, каждая из которых имела успех, но наибольший следует признать, видимо, за монографией «Фреймы для представления знаний», 1974 г. Учился в Нью-Йорке, степени получил в Гарварде (1950) и Принстоне (доктор философии, 1954). Одним из преподавателей Марвина был Клод Шеннон — основоположник теории информации и один из первых математиков, работавших в области ИИ. Шеннон, например, изобрел первую программу для игры в шахматы (компьютер MANIAC, Los Alamos, 1956) и электронную мышь для решения задачи лабиринта. Мински и Шеннон — легенды истории вычислительной техники.

¹ То есть проблема организации памяти. Кроме памяти необходимы еще механизмы преобразования, например мышление, воображение и др.

² Frame (англ.) — кадр, каркас, рамка, скелет. Структура данных для представления стереотипной ситуации. С каждым фреймом ассоциирована информация разных видов. Одна ее часть указывает, каким образом следует использовать данный фрейм, другая — что предположительно может повлечь за собой его выполнение, третья — что следует предпринять, если эти ожидания не подтвердятся.

В отличие от семантических сетей, разработанных первоначально для узкой задачи анализа текстов и потому охватывающих только отдельные этапы психической деятельности, теория фреймов является достаточно полной. В ней рассматриваются не только процессы мышления (их также рассматривают силлогистика Аристотеля, семантические сети), но и восприятие, распознавание, мышление и воображение (как одна из высоких форм мышления). Истоки теории фреймов лежат, по-видимому, в области гештальт — психологии.

Современный тезис о параллельности мышления («природной» параллельности процессов в мозге человека в силу огромного числа нейронов) в понимании М. Мински не является однозначным. Он признает параллельность за процессами высокого уровня (очень общего, и потому неприменимого к реальности без детализации³), оперирующими некоторыми «глубинными структурами»⁴. К таким процессам он относит процесс воображения. Мышление и распознавание считает процессами по своей природе последовательными, оговаривая, впрочем, ряд исключений. Логический подход к представлению знаний считает априорно ограниченным и применимым только к учебным задачам⁵.

Создание своей теории М. Мински, работавший тогда над проблемами робототехники, начал с рассмотрения процессов восприятия и распознавания пространственных сцен: куба, комнаты, предметов в комнате. Затем усложнил задачу, добавив движение (робота по комнате). Для этого понадобилось учесть время и динамику сцен, т. е. изменение геометрических образов предметов из-за движения наблюдателя. Необходимость в соблюдении реального времени или хотя бы приближения к нему потребовала введения «ожиданий» — предопределения некоторых элементов знания еще до начала процесса распознавания. В ходе исследования М. Мински пришел к выводу, что процесс распознавания (у человека) гораздо более длителен,

³ В современных языках программирования это положение воплотилось в понятие «абстрактного класса».

⁴ Это понятие из лингвистики. Если вы скажете по-русски «идет дождь», то это будет означать ссылку на «ничто» — дождь идет сам по себе. Та же самая фраза, сказанная по-французски, будет звучать примерно так: «Он дождит». В данном случае «ничто» и «Бог» — это глубинные структуры.

⁵ Следует отметить, что к моменту написания «Фреймы для представления знаний» еще не существовало языка Prolog, оперирующего (с некоторыми исключениями) с предикатами второго порядка, а только язык Planner, оперирующий с предикатами первого порядка. Впрочем, по мнению автора данной работы, несмотря на очевидный прогресс логической модели, замечания М. Мински остались актуальными.

чем мы предполагали, он начинается еще до появления объекта («ожидания») и в принципе не имеет четкого окончания (воплотилось в идею «отсутствия заданий»). Этот процесс по своей природе последователен, поскольку требует детального изучения микроситуаций и их согласования с другими микроситуациями⁶. Далее, «обкатывая» и подтверждая свою теорию, он рассмотрел процесс повторного распознавания — возвращение робота в комнату, которую тот только что прошел. Здесь М. Мински привел выразительный пример о человеке, который, выйдя за дверь на минуту, обнаружил совершенно иную комнату⁷.

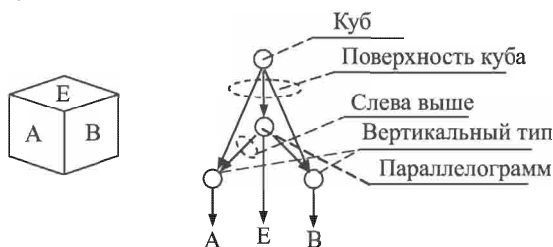


Рис. 36. Фрейм «Куб»

Сами же процессы распознавания, мышления и воображения М. Мински представил как типовые операции над фреймами: выбор некоторого фрейма из сети подобных, его заполнение (в том числе пустыми значениями) и переход к новому фрейму (при помощи некоторой связи). Работая над проблемой зрения человекоподобных роботов, он пришел к выводу, что в сознании и подсознании человека существует набор некоторых «главных» фреймов. Для распознавания пространственных сцен он определил «комнату» или более абстрактно «коробку», у которой есть «линия горизонта», «верх», «низ», «право» и «лево». Такую коробку можно вращать (с некоторой точки зрения, т. е. относительно некоторой линии горизонта) без опасения, что стенка, находившаяся справа от заданной, вдруг окажется слева⁸. Представление знаний фреймами позволяет избежать вычисления местоположения предметов при изменении позиции наблюдателя⁹.

На рис. 36, 37, 38 приведены оригинальные фреймы (М. Мински).

⁶ Известная проблема выделения «объект–фон». Выделив на фотографии объект «крыльцо», вы должны вернуться к объекту типа «дом» и уточнить его описание.

⁷ В шутку надо заметить, что русский народ предвосхитил это открытие в сказании о баране.

⁸ Попробуйте воспользоваться другой системой отношений, например «вперед — сзади», «ближе — дальше», и мысленно поверните коробку.

⁹ Считается, что пространственное мышление (система фреймов) ребенка полностью формируется к 8–9 годам.

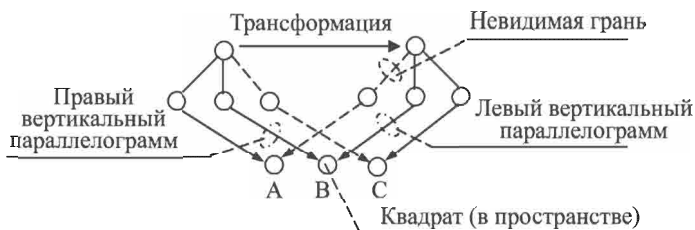


Рис. 37. Процесс трансформации фреймов

При перемещении наблюдателя (относительно объекта) происходит трансформация фреймов (рис. 37). Группа близких фреймов, соответствующая разным позициям наблюдателя, образует сеть.

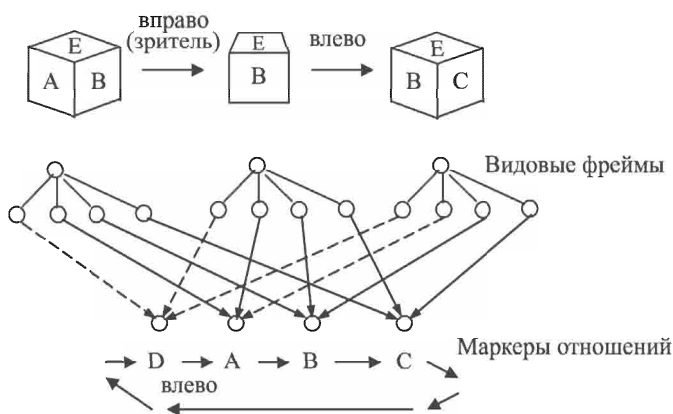


Рис. 38. Сеть фреймов

«Фрейм можно представлять себе в виде сети, состоящей из узлов и связей между ними. Верхние уровни фрейма четко определены, поскольку образованы такими понятиями, которые всегда справедливы по отношению к предполагаемой ситуации. На более низких уровнях имеется много особых *вершин-терминалов* или ячеек, которые должны быть заполнены *характерными примерами или данными*. Каждым терминалом могут устанавливаться условия, которым должны удовлетворять его *задания*. Простые условия определяются *маркерами*, например, в виде требования, чтобы заданием терминала был какой-либо субъект или предмет подходящих размеров, или указатель на субфрейм определенного типа. Более сложными условиями задаются отношения между понятиями, включенными в различные терминальные вершины. Группы семантически близких друг к другу фреймов объединены в *систему фреймов*» [14].

Итак, как же работает система фреймов? Человек смотрит на некоторый предмет и сознательно или бессознательно подбирает наиболее близкий из известных ему фреймов. Это называется процессом *согласования*. Согласование проходит в несколько стадий, останавливаясь, если получается удовлетворительный результат, ко-

гда терминалы фрейма заполняются заданиями, удовлетворяющими ограничениям. В простейшем случае находится фрейм, точно соответствующий ситуации, происходит *сопоставление* (имеющихся в реальной ситуации терминалов и фреймов, к которым эти терминалы могли бы «подойти»).

Если это невозможно, можно попробовать *оправдание*, т.е. пояснить, что фрейм подходит, но объект имеет изъяны, например, сломан. Если фрейм снова «не подошел», происходит *советование*, т.е. выработка четких указаний на то, что делать. И в последнем случае, при полной неудаче делается *резюме* — отказ от фрейма и четкое указание причины отказа.

По окончании процесса согласования у человека появляется некоторое абстрактное представление о предмете, конкретизированное пока только ожиданиями и ограничениями. Большинство узлов фрейма не определены. Свою работу начинает процесс *детализации* (конкретизации) — определяются субфреймы, конкретизируются задания. Детализация может столкнуться с рядом непреодолимых трудностей, и тогда активизируется процесс *изменения*, который в предельном случае способен породить новый фрейм на базе двух предыдущих¹⁰. В процессе детализации терминалы фрейма полностью заполняются, и если при повторном применении фрейма это вызывает противоречия, активизируется процесс *обновления*. Полученный таким образом экземпляр фрейма будет обладать некоторой ценностью, в зависимости от которой либо будет уничтожен сразу после использования, либо помещен в долговременную память как ценный образец. Последний процесс называется *обучением*.

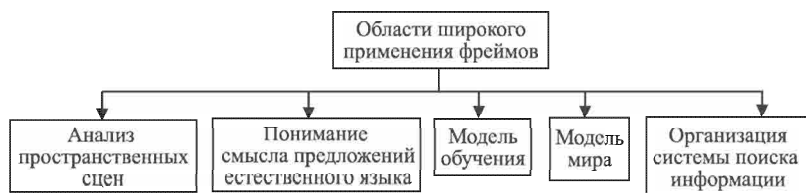


Рис. 39. Области широкого применения фреймов

Применительно к различным задачам человек в каждый момент времени имеет различные глобальные системы фреймов (рис. 39), позволяющие ему ориентироваться в пространстве, двигаться по карьерной лестнице, понимать смысл поэзии, искать необходимую ему информацию и др.

¹⁰ Это очень похоже на механизм множественного наследования в современных языках программирования, не правда ли?

Переход от одного фрейма к другому (трансформация), выполняемый с некоторой точки зрения и направленный на достижение согласования, называется процессом *управления*. Теоретически управление может быть централизованным, когда некоторый суперфрейм (центральный фрейм) управляет процессом заполнения заданий субфреймов. Опять же теоретически управление может быть децентрализованным, когда сначала делается попытка самостоятельно заполнить терминалы субфреймов. Ни один из этих процессов не может считаться решением задачи управления, потому что его завершение требует взаимодействия двух этих процессов. В самом деле, если Вы распознали некоторый объект как «крыльцо», Вам необходимо присоединить его к надфрейму «дом», что нельзя сделать в рамках фрейма «крыльцо». Распознать все типы возможных крылец в рамках фрейма «дом» тоже невозможно.

6.1.1. Анализ пространственных сцен. При анализе изображений, полученных на фотографической бумаге, на светодиодной матрице глаза робота и т. п., возникает ряд специфических проблем, связанных с измерением расстояний между объектами и различением границ объектов: проблема перспективы (искажения перспективы, перекрытия и заслонения), проблема анализа мерности пространства (2-х мерное или 3-х мерное), проблема отделения объекта от фона. Люди неплохо (но не безошибочно) решают такие задачи (рис. 40). Вам не удастся избавиться от ощущения, что на рисунке изображена пирамидка, хотя это не так. Сработал механизм согласования по принципу «оправдание» — «пирамидка сломана».

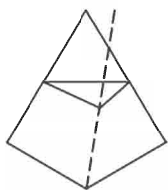


Рис. 40. «Сломанная пирамида?»



Рис. 41. Структура верхнего уровня фрейма «комната»

Другие важные вопросы: «Является ли рисунок трехмерным?», «Можно ли рассмотреть его как набор двухмерных и затем правильно согласовать?», «С чего начинается процесс рассмотрения рисунка?» М. Мински рассмотрел эти вопросы на примере фрейма «комната» (рис. 41).

Заходя в комнату, человек ищет «линию горизонта», относительно которой он затем сможет определить «лево», «право», «верх», «низ» по правилам:

- граница e является горизонтальной линией;
- граница e расположена ниже уровня глаз;
- граница e разделяет пол и стену.

Дальнейшее рассмотрение стен происходит обособленно, но при необходимости активизируется и фрейм «комната» (рис. 42, 43). Рассмотрение прямоугольника, расположенного не то на стене, не то перпендикулярно стене, невозможно без рассмотрения фрейма «комната». Что это могло бы быть? Открытое окно? Рисунок на стене¹¹?

Ответ на этот вопрос мог бы быть получен путем трансформаций — перехода к другой точке зрения. Наблюдатель поменяет позицию, и проекции стен изменятся (рис. 43). Неизвестный объект при этом должен будет подчиниться законам преобразования одной из стен, следовательно, он ей параллелен (например, лежит на ней).



Рис. 42. Открытая форточка или окно неправильной формы?

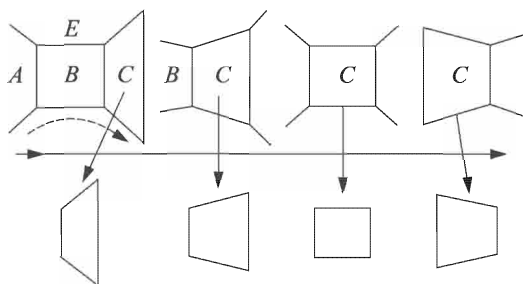


Рис. 43. Движение робота по комнате

Наполним комнату вещами. Возникает проблема заслонений — изменение геометрии объектов при движении по комнате не будет подчиняться только законам проекции. Постоянно будут возникать новые предметы, которых мы не видели раньше¹². Теоретически проблема разрешима путем рассмотрения локальных групп предметов, т. е. достаточно обособленных, таких как, например, стол и стул, стоящие посреди комнаты. Мы легко восстанавливаем недостающие проекции, поскольку имеем такие фреймы в памяти. Сложнее, если стол и стул стоят в углублении стены, тогда решение невозможно

¹¹ В определенном смысле это и есть анализ перспективы.

¹² В Японии есть очень интересный «сад камней», состоящий из 15 объектов, расположенных так, что ни с одной точки зрения нельзя увидеть их все одновременно. При любом подсчете получается 14 камней.

без использования глобальной системы фреймов. Терминалы такой системы должны соответствовать отдельным элементам внутреннего представления пространства сцены, а трансформации — определять выявленные законы перспективы, согласно которым одни элементы по-разному заслоняют другие с различных точек наблюдения. Решение проблемы заслонений открывает перспективы к моделированию воображения.

6.1.2. Понимание смысла предложения. Логическая модель не позволяет анализировать смысл, поэтому в принципе отказывается это делать. Теория фреймов позволяет моделировать как форму (синтаксис), так и смысл (семантику) предложений. Одной из основных проблем при анализе смысла предложения (автоматический перевод, поиск ошибок в тексте и т. п.) является проблема «замыкания». При анализе короткой группы простых предложений с четко определенными терминами задача анализа грамматики в значительной мере эквивалентна задаче анализа смысла предложения. Однако при нарастании числа предложений отчетливо видна разница: «Джуди собирается подарить Мартину воздушного змея, однако зная, что у него уже есть такой, опасается, что придется отнести его обратно в магазин». Кто уже есть у Мартина? Кого придется отнести в магазин? Путем просто подстановки текстовых строк «он», «его» разумно ответить на вопросы не удастся. Необходима «глубинная структура», которая «замкнет» местоимение «него» на «Мартин» и местоимение «его» на «змея». При этом часто возникает необходимость в невидимых объектах. Волк обвинил ягненка в том, что тот «замутил воду». Доказать противоречивость этого высказывания невозможно без определения фрейма «течение» и фрейма «щепка», относительно которых можно получить противоречие «нельзя замутить воду выше себя по течению».

Изучение иностранных языков невозможно без так называемой «языковой догадки» — способности угадывать смысл новых слов по контексту или звучанию. У некоторых людей она поразительно развита. Переводчики часто угадывают речь на несколько предложений вперед (по смыслу) и практически всегда окончание фразы дословно. Это возможно только в рамках некоторой темы и ограниченного круга лиц-фигурантов предложения. Для автоматического перевода используют многоуровневые сети фреймов, состоящие из поверхностных синтаксических, семантических, тематических и повествовательных фреймов.

Задачи слушания и понимания речи являются, вообще говоря, разными по своей природе. В первом случае происходит пассивное заполнение терминалов некоторого фрейма (видение), во втором —

осмысление (согласование — активная форма). Замыкание происходит по-разному. В рамках активной формы оно происходит в рамках некоторого сценария (табл. 8).

Таблица 8

Фрейм сценарий

Сценарий: поход в ресторан	Роли: посетитель, официантка, шеф, кассир Цель: получить пищу, чтобы утолить голод
Сцена I. Вход	Глаза направить туда, где есть пустые столы Выбрать, где сесть Направиться к столу, сесть
Сцена II. Заказ	Получить меню Прочитать меню Решить, что хочешь заказать Сделать заказ официантке
Сцена III. Еда	Получить пищу Съесть пищу
Сцена IV. Уход	Попросить счет, получить чек Направиться к кассиру Заплатить деньги Выйти из ресторана

Для читателя не должно составить трудности восстановить внешний вид графа «поход в ресторан» по приведенному описанию.

В каждом сценарии средства выполнения действий могут варьироваться по обстоятельствам. Например, в сцене II заказ можно сделать письменно, устно и даже (в другой стране при незнании языка) жестами. В сцене IV выплата денег может быть осуществлена кассиру, официанту или словами «Включите в мой счет». Возможно также, что обычная последовательность действий может нарушаться. Имеется по крайней мере три характерных случая такого нарушения. Первый — это отклонение, которое представляет собой прерывание последовательности действий сценария другим сценарием. Другие два случая называются препятствием и ошибкой. В принципе после каждого элементарного действия могут возникнуть препятствия и ошибки, поэтому в сценарий вводятся различные наборы вопросов типа «а что если...», ответ на которые необходимо получать после каждого элементарного действия. При положительном ответе на один из них в сценарии предусматриваются новые действия, устраняющие препятствия и ошибки. Например, в сцене II, если официантка не замечает посетителя, он попытается встретиться с ней взглядом или окликнуть ее.

Таким образом, сценарий — не просто цепь событий, а скорее связанная каузальная¹³ цепочка действий. Он может разветвляться на

¹³ От английского *cause* — причина.

множество возможных путей, которые сходятся в особо характерных для сценария точках — элементарных действиях. Для сценария в ресторане такими действиями являются «прием пищи» и «уплата денег». Для того чтобы знать, когда пользоваться сценарием, нужны заголовки. Эти заголовки определяют обстоятельства, при которых обращаются к данному сценарию¹⁴.

6.1.3. Практическая реализация фреймовой модели. На практике наибольшее распространение модель, основанная на фреймах, получила в объектно-ориентированном программировании и теории объектных баз данных. Современное понятие объекта языка программирования довольно точно соответствует классическому понятию фрейма. Основными свойствами фреймов, используемыми в современных языках, являются: инкапсуляция, наследование и полиморфизм объектов.

Инкапсуляция — единство данных и методов в рамках объекта.

Наследование — способность объекта пользоваться методами и данными, определенными в рамках одного из его предков.

Полиморфизм — способность объекта в разные моменты времени вести себя по-разному, то как объект своего типа, то как объект типа любого из предков.

Различают фреймы-образцы (прототипы, абстрактные классы), хранящиеся в базе знаний, и фреймы-экземпляры (объекты), которые создаются для отображения реальных ситуаций на основе поступающих данных. Для простоты поиска нужного фрейма их типизируют:

фреймы-структуры (матрица сканера, конструкция дома);

фреймы-роли (операционист, покупатель, директор);

фреймы-сценарии (поход в ресторан, день рождения).

Записывают фреймы в виде таблиц (см. табл. 9, 10, 11).

Т а б л и ц а 9

Имя фрейма: абстрактный технический объект			
Имя слота	Значение слота	Присоединенная процедура	Тип слота
Тип объекта	Технический	Нет	АКО
Положение	$X = 20; Y = 20; Z = 50$	Двигаться	То же
Назначение	Перемещение массы в пространстве	Нет	То же

В сети фреймов понятие «абстрактный легковой автомобиль» наследует свойства фреймов «абстрактный технический объект» и

¹⁴ В этом месте рекомендуется выполнить практическую (лабораторную) работу «Создание фрейма визуального образа».

Т а б л и ц а 10

Имя фрейма: абстрактный автомобиль			
Имя слота	Значение слота	Присоединенная процедура	Тип слота
Предок	Абстрактный технический объект	Нет	АКО
Положение	$X = 20; Y = 20; Z = 50$	Ездить	То же
Назначение	Перемещение массы в пространстве	Нет	То же
Ограничение скорости	120 км/час	Нет	То же

Т а б л и ц а 11

Имя фрейма: абстрактный легковой автомобиль			
Имя слота	Значение слота	Присоединенная процедура	Тип слота
Предок	Абстрактный автомобиль	Нет	АКО
Положение	$X = 20; Y = 20; Z = 50$	Ездить	То же
Назначение	Перемещение пассажиров	Грузить	То же
Ограничение скорости	120 км/час	Нет	То же

«абстрактный автомобиль», которые находятся на более высоком уровне иерархии. На вопрос: «Способен ли автомобиль перемещаться в пространстве?» следует ответ: «Да», так как этим свойством обладает класс-предок. Наследование свойств может быть частичным, как свойство «назначение».

Область применения объектного программирования для решения интеллектуальных задач очень широка. С теми или иными затратами модель пригодна для решения любых интеллектуальных задач (решение которых в принципе возможно). Наиболее выраженной областью эффективного применения модели являются сложные недетерминированные процессы, т. е. такие, исход которых зависит от взаимодействия большого и очень большого числа различных компонент, объектов, условий, которые, в свою очередь, могут вести себя по-разному, в зависимости от сложившейся на момент ситуации. Особенность моделирования таких ситуаций именно на основе фреймовой модели состоит в том, что процесс можно «проиграть в реальном времени», а не последовательно, как это пришлось бы делать при помощи, например, правил продукций.

В качестве примера рассмотрим задачу о проектировании безопасной дорожной развязки: «Требуется спроектировать геометрию и размеры пересечения двух дорог, находящегося на выезде из города,

таким образом, чтобы наиболее вероятное количество столкновений автомобилей в год не превышало заданное. Параметры пересекающихся дорог — ширина, покрытие, разрешенный и реальный скоростной режимы, сведения о потоке автомобилей и др. — известны».

Инженеры-проектировщики предложили три варианта развязки (рис. 44). В первом случае строится классический перекресток. Основная забота о безопасности движения возлагается на самих водителей. При полном соблюдении ими скоростного режима и знака «Уступи дорогу» столкновения возможны только по причине неисправности техники или наличия непредвиденных помех (например, коров на дороге). Априорно проектировщики оценили опасность такого решения как «удовлетворительно».

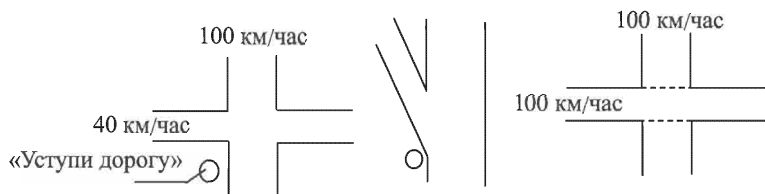


Рис. 44. Варианты дорожной развязки

Во втором случае предполагается, что за счет правильного выбора угла примыкания и наличия полосы разгона водители смогут при необходимости «уходить от прямых столкновений». Далее по трассе предполагается строительство кольцевой развязки. Разработчики оценили степень безопасности также на «удовлетворительно», а стоимость решения как «высокую».

Третий вариант предполагает строительство моста над одной из дорог. Разработчики оценили последний вариант как «самый безопасный, хотя и дорогой» и сочли его предпочтительным.

Модельные испытания на компьютере включали следующие шаги.

1. Выбор из БЗ необходимых фреймов-образцов: «корова», «лошадь», «москвич 412», «волга 3110», «тойота ленд круизер», «запорожец 1974 г. выпуска» и др.
2. На заданную геометрию развязки были помещены указанные фреймы, исходя из условий:
 - 2.1 соблюдения правил движения/слабого соблюдения правил движения;
 - 2.2 благоприятных/неблагоприятных погодных условий;
 - 2.3 хорошего/плохого технического состояния транспортных средств.

3. Фреймам отсылается сообщение «начать моделирование», которое достигает их одновременно. Каждый фрейм начинает вести себя некоторым образом: автомобили движутся с некоторой скоростью, коровы внезапно перебегают дорогу, птицы пролетают перед машинами и пугают водителей. В результате достаточно продолжительного моделирования ситуация приходит к некоторому почти стабильному состоянию, которое мы, скорее всего, и будем наблюдать в реальной жизни. На последнем этапе с модели снимают статистические характеристики: количество аварий в сутки, реальная пропускная способность и т. д.

Проведенные статистические испытания показали, что при полном соблюдении правил ДД и отличном техническом состоянии наивысшие характеристики безопасности и пропускной способности имеет вариант 3.

При полном соблюдении правил ДД и плохом техническом состоянии наивысшие характеристики безопасности имеет вариант 1.

Вариант 2 имеет более высокую (по сравнению с вариантом 1) пропускную способность, но безопасность движения сильно зависит от скорости движения, при превышении порога в 60 км/ч вариант 2 приводит к большему (по сравнению с вариантом 1) количеству аварий.

Принимая во внимание тот факт, что в данной местности основной автомобильный парк имеет возраст более 8 лет и представлен, в основном, низкоскоростными автомобилями, следует рекомендовать к строительству вариант 2.

Фреймовая модель имеет ряд безусловных достоинств. В первую очередь следует сказать, что это едва ли не единственная глубинная модель, хорошо обоснованная теоретически. Ее практическая реализация — построение микромиров — позволяет сосредоточиться на свойствах и поведении участников микромира, предоставив им самим организовать свое взаимодействие на основе правил, привычек, сценариев и просто случайностей. Благодаря этим замечательным свойствам, фреймовая модель позволяет значительно отодвинуть тот предел применимости по размерности системы, который имеет каждая модель (см., например, правила продукций).

Фреймовая модель позволяет хорошо систематизировать иерархию объектов реального мира, справляясь даже с задачами численной таксономии, за счет введения абстрактных классов. Модель реализована во многих языках программирования и специальных языках представления знаний (FRL — Frame Representation

Language), успешно использована в ряде известных экспертных систем: ANALYST, МОДИС и др.¹⁵

К недостаткам модели следует отнести, по-видимому, некоторые затруднения, возникающие при обмене большими порциями данных между двумя объектами. Собственно, фрейм в классическом виде, описанный М. Мински, вообще не предполагает никакого обмена (см. проблему отказа от неправильно подобранного фрейма с сохранением значений терминалов для их последующего использования с другим фреймом).

¹⁵ Для закрепления материала рекомендуется выполнить практическую (лабораторную) работу «Создание не визуального фрейма» и сдать тест «Основы объектной модели . . . »

ГЛАВА 7

СЕМАНТИЧЕСКИЕ СЕТИ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Термин «семантическая (смысловая) сеть» (или «концепция слов») возник в 1968–69 гг. в работах Р. Куиллиана¹. Он подразумевает целый класс общих подходов, для которых характерно использование графических схем с узлами, соединенными дугами (графов). Особенность состоит в том, что в теории семантических сетей рассматриваются не только графы-деревья, но циклические и полносвязные графы. Кроме того отдельно и особо исследуются возможные типы отношений (традиционные: АКО — a kind of, ISA, has part и ряд новых).

Первоначально семантические сети были разработаны для анализа естественных языков и построения психологических моделей человеческой памяти (задача автоматического перевода — положение Аристотеля о том, что «человек мыслит на языке», задача подбора синонимов к заданному слову и др.). На этом этапе считалось, что в предложении есть некая «центральная тема», «раскрутив» которую, машина может «понять» смысл (семантику) предложения (рис. 45).

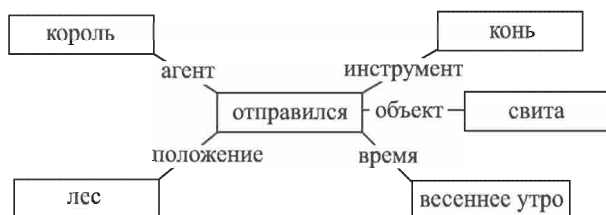


Рис. 45. «Весенним утром король взял слуг и отправился в лес на конную прогулку»

Семантика — наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков. Семантическая сеть — это ориентированный граф, вершины которого — понятия, а дуги — отношения между ними.

Экспериментальную проверку теоретических положений об устройстве человеческой памяти и организации логического вывода Р. Куиллиан провел совместно с психологом А. Колинз (Allan Collins) в 1969–70 гг. на основе иерархической сети². Для такой сети

¹ Ross Quillian — социолог, работал в области ИИ, открыватель «семантической памяти».

² Если считать, что узлы «корабль», «океанский лайнер» и др. не имеют внутренней структуры (например, в некоторой задаче нас не интересует структура этих понятий), то сеть рис. 42 можно назвать простой, а не иерархической.



Рис. 46. Иерархическая сеть с использованием отношения «является»

характерны отношения вида класс — элемент класса; свойство — значение; пример элемента класса (рис. 46, 48).

Дальнейшее развитие модели связано с именами Р. Симмонса (Robert Simmons), С. Шапиро (Stuart C. Shapiro, обратил внимание на наличие «отношений разного рода»), Хендрикса (Hendrix, ввел понятие «разделенной сети/partitioned network») и Шенка (Roger C. Schank, ввел понятия «концептуальной зависимости» и «концептуальных примитивов»³).

В сетях Р. Симмонса нет строгой иерархии, поскольку он решал несколько другую задачу: понять значение предложения путем «развертывания/дополнения» глаголов. Т.е. от глагола строил собственную «дополнительную сеть», включающую в себя «варианты использования» (cases) данного глагола. Соответственно получена и классификация узлов: Агент, Реципиент⁴, Объект.

Давать

Агент: {Яго, мальчик, Отелло}

Реципиент: {Дездемона, Родриго}

Объект: {яблоко, цветы, платок}

Узлы в семантической сети обычно соответствуют объектам, концепциям, событиям или понятиям. Дуги могут быть определены разными методами, зависящими от метода представления знаний. Любой фрагмент сети, например одна вершина, две вершины и соединяющие их дуги, называют подсетью (рис. 47, 48). Логический вывод (поиск решения) на семантической сети заключается как раз в том, чтобы найти или сконструировать подсеть, удовлетворяющую некоторым условиям. Для того чтобы формализовать этот процесс, вводят типизацию семантических сетей и затем разрабатывают методы решения для сетей конкретного вида (на основе анализа математических свойств отношений, входящих в сеть).

Сети конкретного вида в совокупности с методами вывода на них иногда называют именами первооткрывателей.

³ Например, «23 февраля», «8-е марта» и т.п. Более прижилось понятие «концепт». В сетях Шенка концептам обычно соответствуют узлы. Существуют целые диссертации, посвященные описанию концептов реального мира, например, «народ», «вещь», «делать», «быть». Различают концепты высокого уровня, например «корабль», и низкоуровневые.

⁴ To receive (англ.) — получать.

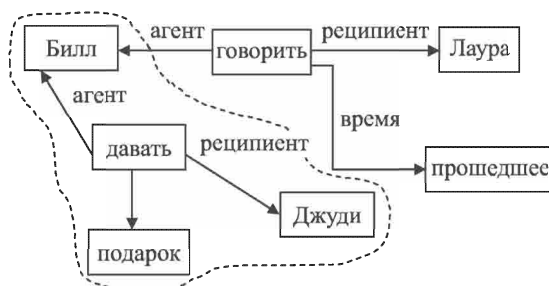


Рис. 47. Представление семантической сетью сложного предложения: «Билл сказал Лауре, что (он) дал Джуди подарок»; подсеть «Билл дает Джуди подарок»



Рис. 47. Семантическая сеть «автомобиль» и подсеть «двигатель передает усилие колесам»

Сети различного вида получаются и в зависимости от того, какие ограничения накладывают на вершины и дуги. Если вершины сети не обладают внутренней структурой, то такие сети называют простыми сетями, если вершины обладают внутренней структурой, то такие сети называют иерархическими. Динамические семантические сети (сценарии) — сети с событиями. Одно из основных отличий иерархической сети от простой сети — в возможности разделять сеть на подсети и устанавливать отношения не только между вершинами, но и между подсетями. Различные подсети, существующие в сети, могут быть упорядочены в виде дерева подсетей, вершины которого — подсети, а дуги — отношения видимости. Понятие подсети аналогично понятию скобок в математической нотации, понятие видимости — понятию переменных, находящихся внутри и снаружи скобок.

Опишем при помощи СС технический объект «Арка» (рис. 49) — перекладина, опирающаяся на 2 опоры, которые не должны соприкасаться друг с другом. Формально эту сеть можно описать так:



Рис. 49. Семантическая сеть понятия «Арка»

Как видите, описание сети ничем принципиально не отличается от описания деревьев в продукционной модели. Фактически для описания сетей и деревьев используются одни и те же машинные языки. Но вывод на сетях, безусловно, сложнее, более того, многие типы отношений не исследованы и формального алгоритма не существует.

Одна из значительных трудностей для понимания, а следовательно, и возможности логического вывода на сети заключается в том, что пространства предметной области, с которыми приходится иметь дело программам ИИ, часто являются неметрическими, т. е. не удовлетворяют одной из аксиом:

1. $\rho(a, b) = 0$ только при $a = b$,
2. $\rho(a, b) = \rho(b, a)$,
3. $\rho(a, b) < \rho(a, c) + \rho(b, c)$,

где ρ — мера (для простоты понимания можно считать, что мера соответствует понятию расстояния между элементами).

В обыденной жизни человек практически всегда имеет дело с метрикой. Мы никогда не задумывались над тем, почему кратчайшим расстоянием между двумя точками следует считать прямую, равноценны ли выражения «2 + 3» и «3 + 2», равен ли объект сам себе?

Обратите внимание на отношение «ведет к». Дорога ведет из пункта *A* в пункт *B*. Значит ли это, что она же ведет из пункта *B* в пункт *A*? Некоторое здание на карте города обладает свойством «иметь подъезд для грузовика», значит ли это, что грузовик может подъехать с любой улицы? Нет? Тогда выходит, что одно и то же здание является относительно разных улиц разным? Если некоторого мальчика зовут Миша, значит ли это, что с точки зрения разных людей его зовут по-разному?

Важным является также вопрос об установлении соответствия и взаимно-однозначного соответствия между элементами двух сетей. С математической точки зрения модель семантики является одной из наиболее интересных и сложных. Исследования по этой теме ведутся в основном в математической теории множеств и теории чисел.

Очень полезным свойством, которым могут обладать отношения СС, является транзитивность. Существует «бородатый пример», иллюстрирующий эту пользу.

П р и м е р. Рассмотрим предложения «Куин Мери является океанским лайнером» и «Каждый океанский лайнер является кораблем», рис. 46. Используются дуги «является». Поскольку мы знаем свойства дуг, связывающих узлы (отношение «является» —

ISA – транзитивно⁵), мы можем вывести из сети третье утверждение: «Куин Мери является кораблем», хотя оно и не было сформулировано в явном виде.

Отношение «является» и другие (вроде отношения «имеет часть») устанавливают свойство иерархии наследования в сети. Это означает, что элементы более низкого уровня в сети могут наследовать свойства элементов более высокого уровня. Это экономит память, поскольку информацию о сходных узлах не нужно повторять в каждом узле. Вместо этого она может размещаться в одном центральном узле сети. Например, для семантической сети, представляющей корабль, такие его части, как двигательная установка, корпус, котельная, включены один раз на уровне корабля, вместо того чтобы повторяться на более низком уровне иерархии, вроде типа корабля или конкретного корабля. Это позволяет сберечь огромный объем памяти, даже если мы работаем только с сотнями кораблей и их частей. По сети можно затем осуществить поиск, используя знания о смысле отношений, обозначаемых дугами сети, чтобы установить факты вроде «Куин Мери имеет котельную».

Можно ввести несколько классификаций семантических сетей. Например, по количеству типов отношений:

- однородные (с единственным типом отношений);
- неоднородные (с различными типами отношений).

По типам отношений:

- бинарные (в которых отношения связывают два объекта);
- n -арные (в которых есть специальные отношения, связывающие более двух понятий, например «треугольник»).

Наиболее часто в семантических сетях используются следующие отношения:

- связи типа «часть – целое» («класс – подкласс», «элемент – множество» и т. п.);
- функциональные связи (определяемые обычно глаголами «производит», «влияет», ...);
- количественные (больше, меньше, равно, ...);
- пространственные (далеко от, близко от, за, под, над, ...);
- временные (раньше, позже, в течение, ...);
- атрибутивные связи (иметь свойство, иметь значение, ...);
- логические связи (и, или, не) и др.

⁵ Транзитивность: для любых $a, b \in R$ таких, что $a < b$ и $b < c$, справедливо соотношение $a < c$. Пример нарушения транзитивности (пример предоставлен д.ф.-м.н. Г. В. Пышнограем): A сын B , C сын B , но C не сын A .

частью другого). Этот язык используется для работы с большими грамматиками. PLNLP работает по следующим правилам:

- 3.1. с помощью правил декодирования производится синтаксический анализ линейной языковой цепочки и строится сеть;
- 3.2. с помощью правил кодирования сканируется сеть, порождается языковая цепочка или другая трансформированная сеть.
4. Создан специальный сетевой язык NET.
5. Созданы широко известные экспертные системы, использующие семантические сети в качестве языка представления знаний — PROSPECTOR, CASNET, TORUS.

Основной недостаток модели — сложность поиска вывода; на семантической сети хорошие математические модели разработаны только для нескольких типов отношений.⁷

⁷ Для закрепления материала рекомендуется выполнить практическую (лабораторную) работу «Построение семантической сети . . . »

Г Л А В А 8

НОВЫЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Основные модели представления знаний, перечень ценных задач и методы их решения сформировались, в основном, к началу 80-х гг. прошлого века. В развитии теории и практики ИИ наступил некоторый застой, связанный с тем, что дальнейшее совершенствование традиционных моделей не давало качественного скачка.

В начале 90-х гг. философское понимание задач и методов ИИ несколько изменилось, что сразу же привело к появлению ряда новых моделей: критериального языка описания выбора, нейронных сетей, ряда вероятностных моделей. Следует сказать, что далеко не все специалисты по ИИ готовы признать их полноценными моделями для представления знаний, хотя бы в силу того, что эти модели не способны обосновывать свои выводы¹. Кроме того, «новыми» их можно назвать с большой натяжкой; критериальному языку, например, несколько сот лет.

Итак, новый взгляд на понятие «модель для представления знаний» состоит в том, что имеется в виду некоторая модель, позволяющая получать «хорошие решения». При этом не важно, моделирует ли она способ мышления человека и способна ли пояснять свои выводы².

8.1. Критериальные методы

На практике применение интеллектуальных, нечетких (в первую очередь) моделей часто необходимо для оценки альтернатив и выбора наилучшей в некотором смысле. Необходимость их применения обусловлена большим количеством альтернатив, учитываемых параметров, требованием «проиграть несколько сценариев развития событий»; например, «благоприятный, обычный и наихудший».

Выбор в этом случае может осуществляться осмысленно, на основе учета всех «за» и «против», либо случайным образом (например, по аналогии с кем-то). Следует заметить, что для больших систем логические обоснования, которые способен предоставить человек

¹ Имеется в виду механизм бектрекинга, присутствующий в логических и продукционных моделях и отсутствующий в новых моделях представления знаний. Считается, что экспертная система без подсистемы объяснения своих выводов менее ценна.

² В приложении (пункт 11.2) приведен краткий обзор некоторых известных и применяемых на практике методов, как критериальных, так и вероятностных.

эксперт, не намного надежнее случайного выбора, а часто менее надежны. Это связано, прежде всего, с тем, что человек не способен одновременно удерживать в памяти большое количество фактов, причин, целей, а современные задачи (например, экономические) требуют именно этого.

Предположим, что некоторый служащий в банке принимает решение о кредитовании предприятия, находящегося в неустойчивом положении. В случае, если в кредите будет отказано и предприятие преодолест трудности самостоятельно, банк не получит прибыль. Если же предприятие разорится, банк понесет убытки. Если кредит будет выдан под значительный залог — предприятие откажется его брать. Достаточно ли интуиции служащего и имеющейся бухгалтерской документации для принятия решения?

Для описания и решения подобных задач создана специальная модель — языки описания выбора. Задачи выбора чрезвычайно многообразны, различны и методы их решения. Прежде всего введем понятия, общие для всех задач выбора.

8.1.1. Задача выбора. Метод критериальной оценки, ранжирования и последующего выбора является, возможно, одним из самых старых методов, применяемых в ИИ.

В самой простой постановке он предполагает сравнение двух альтернатив (бинарное отношение). Например, $3 < 5$, $6 \in \{3, 5, 6\}$ и т. п. Подобные отношения были известны задолго до Аристотеля. В данном случае сравнение происходит по одному параметру (критерию). Более сложная ситуация в следующем примере.

Дано: альтернатива $A(3, 3) = A(a_1, a_2)$ и альтернатива $B(5, 5) = B(b_1, b_2)$. Установить вид связывающего их отношения R , т. е. $A R B$.

Если придерживаться традиционной точки зрения, то $B > A$, поскольку $b_1 > a_1$ и $b_2 > a_2$. Однако такое решение не бесспорно, поскольку неясно, что означают a_1, a_2, b_1, b_2 и допустимо ли устанавливать на множествах $\{a_1, b_1\}$ и $\{a_2, b_2\}$ отношения строгого порядка (« $<$ », « $>$ »). Предположим, что альтернативы A и B имеют значения «Иванов» и «Петров» соответственно. Параметры $\{a_1, b_1\}$ и $\{a_2, b_2\}$ — количество неудовлетворительных оценок в первом и втором семестрах соответственно. Кто из них лучший студент? Очевидно, $A = B$ — они оба отвратительные студенты. Если же предположить, что параметры имеют смысл «количество ошибок в очень сложном студенческом чертеже», то, очевидно, $A > B$ (A лучше B). Как видите, решение зависит от точки зрения («нуля измерения») и самого вида отношения R .

Сложный случай — сравнение более чем двух альтернатив (попарно), либо определение отношения, связывающего более двух аргументов. Отношение R в этом случае может оказаться тернарным (например, R — треугольник связывает стороны a , b , c) или n -арным (семья). Число n может оказаться конечным, счетным или континуальным. Так, например, Вам может быть поставлена задача «выбрать турбазу для отпуска», исходя из критериев «уровень комфорта» и «цены». В случае, если для выбора Вы воспользуетесь справочником баз Горного Алтая, множество альтернатив будет конечно. Если же предпочтете «дикий туризм», а в качестве «базы» будете рассматривать «места для кемпинга» — континуальным³. В самом деле, между любыми двумя кемпингами всегда можно найти чудное местечко. Очевидно также, что принимать решение, в каком кемпинге остановиться, еще до самой поездки не имеет смысла. Равно, впрочем, как и чрезмерно дотошно оценивать каждый кемпинг. Тот, в котором Вы остановитесь, все равно окажется хуже того, на который наткнетесь следующим утром. Случайный выбор (с некоторым законом распределения случайной величины) — вполне разумное решение для континуального множества альтернатив.

Таким образом, критериальная задача выбора характеризуется: наличием многомерного пространства параметров; описанием желательных областей в этом пространстве («идеальных точек» или, если невозможно задать точку, «уровней притязаний»); мерой (критерием выбора), которая позволяет определить, насколько избранная альтернатива близка к желательной; набором пороговых значений (т. е. таких предельных различий в мерах между двумя альтернативами, которые позволяют обоснованно сказать, что альтернативы действительно различны с некоторой точки зрения); а также математическими методами, применяемыми для определения наиболее выгодного направления движения в пространстве состояний (рис. 51, 52, 53).

Тот же самый абзац другими словами: имеется некоторая функция $f(x_1, x_2, \dots, x_n)$ от n независимых переменных. Необходимо найти $\max^4 \{f(x_1, x_2, \dots, x_n)\}$ при наличии ограничений одного из видов, представленных в табл. 12.

Здесь все индексы являются целыми числами 1, 2, 3 ...

То же самое можно наглядно показать несколькими частными геометрическими примерами на множестве двух параметров x_1 и x_2 (рис. 51, 52, 53).

³ То есть множество чисел, соответствующее отрезку $[0..1]$.

⁴ Или $\min$ в зависимости от того, задана «мера близости» или «мера удаленности».

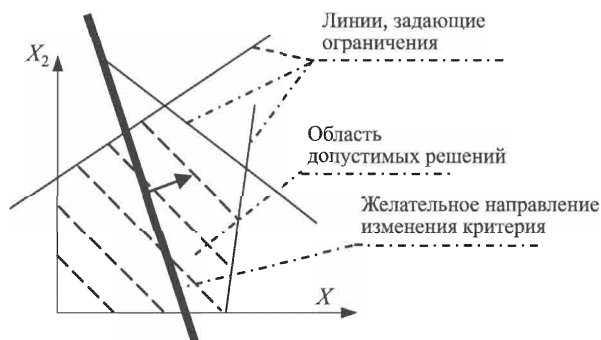


Рис. 51. Линейный критерий на множестве линейных непрерывных ограничений

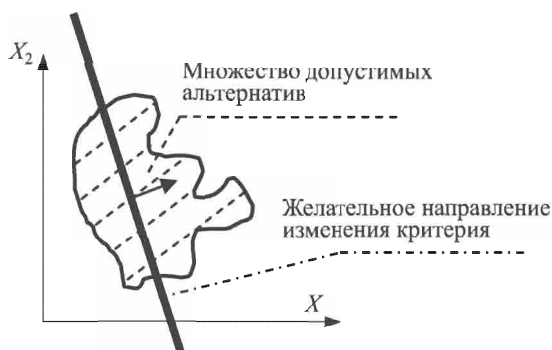


Рис. 52. Линейный критерий на невыпуклом непрерывном множестве нелинейно ограниченных альтернатив

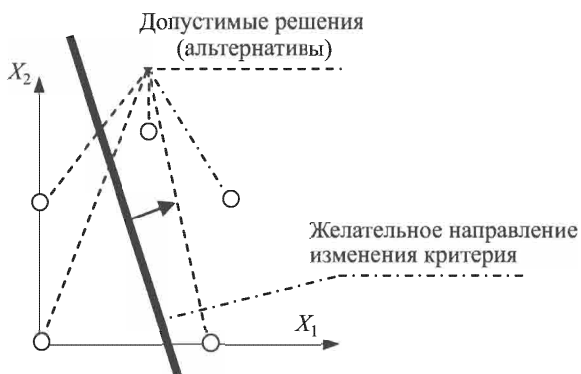


Рис. 53. Линейный критерий на дискретном множестве альтернатив

Т а б л и ц а 12

Виды ограничений в критериальных задачах

Линейные ограничения на непрерывном множестве значений параметров x_i	Нелинейные ограничения на непрерывном множестве значений параметров x_i	Линейные или нелинейные ограничения на дискретном множестве значений параметров x_i
$C_0 \geq X_1 \geq C_1$ $C_2 \geq X_2 \geq C_3$...	$C_0 \geq X_1 \geq C_1$ $C_2 \geq X_2 \geq C_3$...	$X_1 \in \{x_{11}, x_{12}, \dots, x_{1s}\}$...
$C_{n+2} \geq X_n \geq C_{n+3}$ $a_{1m}x_1 + a_{2m}x_2 + \dots$ $\dots + a_{nm}x_n \leq b_1$...	$C_{n+2} \geq X_n \geq C_{n+3}$ $a_{1m}x_1^{k1} + a_{2m}x_2^{k2} + \dots$ $\dots + a_{nm}x_n^{kn} \leq b_1$...	$X_n \in \{x_{11}, x_{12}, \dots, x_{1d}\}$...
$a_{1m}x_1 + a_{2m}x_2 + \dots$ $\dots + a_{nm}x_n \leq b_m$	$a_{1m}x_1^{k1} + a_{2m}x_2^{k2} + \dots$ $\dots + a_{nm}x_n^{kn} \leq b_m$	Линейные или нелинейные ограничения

Вы видите, что задачи критериального принятия решений фактически являются задачами линейного или нелинейного математического программирования⁵. Частным случаем любой из этих задач может оказаться, например, задача целочисленного программирования, когда требуется найти такую наилучшую альтернативу, оценки x_1 и x_2 которой — целые числа.

В общем случае метод критериального выбора (принятия решения) предполагает следующие шаги.

1. Постановка очень общей, концептуальной задачи: «Что нужно сделать» или «Что нужно в конце и что есть сейчас». Т.е. определение цели (целей) принятия решения.
2. Генерация альтернатив. На этом этапе применяются так называемые экспертные методы: «мозговой штурм», «синектика», «матрицы изобретений» и др.
3. Выбор критериев оценки степени достижения цели (целей).
 - 3.1. Выбор точки зрения: «стремление к идеалу» или «минимизация убытков».
 - 3.2. Выбор вида оценочной функции для каждого критерия и вида свертки (для всех в целом).
 - 3.3. Определение «пороговых значений» для каждого критерия (шкала «хорошо — плохо»)⁶.
4. Оценка альтернатив по выбранным критериям.
 - 4.1. Измерение значений параметров каждой альтернативы.

⁵ И решаются различными модификациями соответствующих математических методов, таких, например, как симплекс-метод.

⁶ Решение задач пункта 3 описывает так называемую «структуру предпочтений эксперта».

- 4.2. Учет предыстории (для случая многократного выбора).
- 4.3. Вычисление значения критерия.
5. Ранжирование альтернатив по полученным оценкам степени достижения цели (целей).
6. Отбрасывание явно худших альтернатив.
7. Выбор одной альтернативы из множества недоминирующих (например, случайным образом) или возврат к п. 1.

Еще одним очень важным моментом является характеристика выпуклости (вогнутости) всех рассматриваемых функций и замкнутость множеств допустимых значений. Эти определения даются в функциональном анализе и повторять их мы не будем.

Вернемся к рассмотренным шагам решения критериальной задачи. Шаг 1 подробно описан в первых лекциях. Второй пункт достаточно обширен, а его рассмотрение не входит в курс ИИ. Выбор критериев (шаг 3) в значительной мере является искусством. Все критериальные методы (принятия решений) можно разделить на методы «идеальной точки» и «компромиссного программирования». В первом случае все альтернативы рассматриваются в многомерном пространстве своих признаков, причем существует «идеальная точка» (рис. 54). Оценка любой из альтернатив получается как расстояние до идеальной точки⁷. Выигрывает та, у которой это расстояние минимально. Иногда, напротив, рассматривают не «меру близости», а «меру удаленности». В этом случае могут, например, создать «абсолютно нежелательную точку» и определять расстояния до нее (от каждой из альтернатив). Очевидно, наилучшей будет альтернатива, наиболее удаленная от «плохой» точки. Многомерные пространства, допускающие введение понятий «плохой — хороший», «сильный — слабый» и т. п. для указания желательного направления развития событий, называют «пространствами с семантической метрикой» или «семантическими пространствами»⁸. Впервые широкую известность рассмотрение таких пространств приобрело после концептуальной книги Чарльза Осгуда и Суси⁹ «Измерение значений». Эта книга — своеобразный пролог к появившейся через 10 лет теории семантических сетей. В книге рассматриваются конструкции английского

⁷ Для удобства шкалы конструируют часто так, чтобы «идеальной» оказалась точка начала координат.

⁸ Реже то же самое называют «пространством Осгуда».

⁹ И еще нескольких десятков аспирантов и студентов, имена которых по сложившейся на Западе традиции в книгу профессора не вносят [2]. Впервые опубликована в 1947, США.

языка и психологические структуры, пригодные для их представления и понимания. Лингвисты считают книгу своей, психологи своей, а специалисты по ИИ точно уверены, что она написана для них¹⁰.



Рис. 54. Идеальная точка и реальная альтернатива (точка зрения Маши)

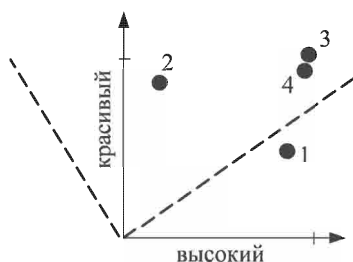


Рис. 55. Доминирующие и несравнимые альтернативы

Итак, построение некоторого пространства (рис. 55), в котором мы далее намерены принимать решения, предполагает определение целей¹¹ (желаемого состояния, идеальной точки), направления изменения свойств пространства, а также определение вида меры и пороговых значений. В качестве меры в простейшем случае может использоваться простая разность (сумма разностей). Так, например, «Вася неидеален с мерой $(180-170) + (\text{очень красивый} - \text{симпатичный})$ ». Как видите, попытка введения даже самой простой меры тут же наталкивается на проблему наличия измерительных шкал разного вида. Количество возможных типов шкал, конечно, неплохо изучено в математической теории измерений. Далее посвятим шкалам отдельный параграф и продолжим рассмотрение мер. С геометрической точки зрения следовало бы рассматривать Евклидову меру (расстояние) $\Delta = \{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 + (f - f_1)^2\}^{1/2}$. Пример намеренно приведен для 4-х мерного пространства, чтобы показать, что показатели степени не изменяются и в пространстве выше 2-х или 3-х мерного. В случае, если вместо вторых степеней мы будем использовать степень p , то при $p \rightarrow 0$ получим классическую балльную оценку (применяется, например, в школах), при которой любое отклонение реального состояния от

¹⁰ Ч. Осгуд рассматривал массовые процессы. Например, по результатам контрольной по математике у некоторой группы студентов можно построить пространство Осгуда. Л. Заде рассматривал единичные понятия, поэтому модели, которые у них получились, безусловно, имея некую «общую философию», все же различны.

¹¹ В западной терминологии целям часто соответствует термин «goal», в то время как направлению изменения свойств — термин «objectives». В неспециализированных англо-русских словарях «goal» и «objectives» трактуются как синонимы, что часто вызывает ошибки в переводных книгах.

идеального засчитывается за 1; при $p \rightarrow \infty$, напротив, получим ситуацию, когда малые отклонения (меньше 1) нивелируются, в то время как значительные отклонения увеличиваются (фактически, получим оператор нахождения максимального элемента). В результате правильного выбора степени для конкретной задачи мы получаем меры, которые «хорошо растаскивают решения», т. е. позволяют сформировать множество явно доминирующих альтернатив (альтернатива 3) и множество недоминирующих (относительно друг друга) альтернатив (1 и 2). При неправильном выборе меры получим проблему с альтернативой 4, которая, очевидно, хуже 3, но насколько? Является ли это отклонение достаточным, чтобы предпочесть альтернативу 4 альтернативе 3?

И все же практически при любом виде меры рано или поздно придется выбирать между 1 и 2. Но как? Геометрически это возможно методом поворота осей и проецирования точек на оси. Очевидно, что альтернатива 1 предпочтительнее 2, причем значительно (проекция на ось абсцисс). Математически операция поворота эквивалентна умножению на некоторый коэффициент (вес параметра). Такие меры называют взвешенными: $\Delta = \{w_1(x - x_1)^p + w_2(y - y_1)^p + w_3(z - z_1)^p + w_4(f - f_1)^p\}^{1/p}$, очевидно, что $\sum w_i = 1$. Набор весов еще называют «структурой предпочтений эксперта», а конкретный вид меры — сверткой. Более строго ту же формулу следовало бы записать $\Delta = \{w_1(x - x_1)^p/a_1 + w_2(y - y_1)^p/a_2 + w_3(z - z_1)^p/a_3 + w_4(f - f_1)^p/a_4\}^{1/p}$. Здесь коэффициенты a_n служат для нормирования шкал, чтобы Вы не начали складывать, например, рубли с долларами. Однако эта формула тоже не универсальна, поскольку для ряда шкал необходимо не только привести масштаб «к единому», но и «подвинуть начало отсчета».

Еще более сложной ситуация становится в случае многократного принятия решения. Рассмотрим пример классификационной задачи постановки медицинского диагноза по одному параметру с учетом времени (рис. 56).

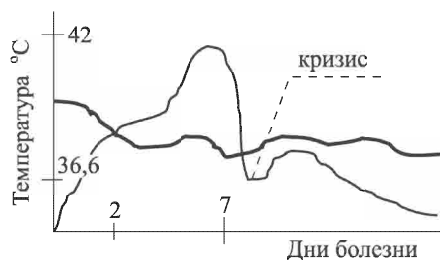


Рис. 56. Непрерывные температурные кривые бронхита и пневмонии

Воспаление легких (предположим, что оно началось вследствие переохладения и упадка сил) в общем случае характеризуется резким повышением температуры и ухудшением самочувствия. Если не предпринимать мер, то температура растет и состояние ухудшается примерно неделю, после чего наступает кризис, т. е. температура резко поднимается и затем падает. В начале прошлого века, до создания антибиотиков, пациент либо переживал кризис и шел на поправку, либо умирал. При бронхите же температурная кривая ведет себя иначе — состояние пациента стабильное, температура невысокая или даже пониженная. Небольшой подъем наблюдается обычно в начале заболевания. Необходимо поставить диагноз больному на второй день. Очевидно, это нельзя сделать без учета первого дня. Если в первый день у больного наблюдалось небольшое повышение температуры, то предпочтительнее бронхит, иначе — пневмония. Но достаточно ли двух точек для окончательного решения? А сколько достаточно? И что делать, если точки не будут «точно ложиться на модельную кривую»? Учтите, что в данном примере учитывается только один параметр, а для более-менее реальной диагностики необходимо минимум 7–8 параметров.

Однозначных ответов на поставленные вопросы Вы не найдете в современной литературе по принятию решений. В классических книгах на эту тему вообще редко ставится задача рассмотрения явления в динамике. Те методы, которые Вы встретите, будут касаться в основном однократного принятия решения¹². Задачи медицинской диагностики в современных условиях редко рассматривают критериальными методами. Чаще всего предпочитают использовать нейронные сети в силу их объективной эффективности (далее посвятим нейросетям параграф). Но нейронная сеть в значительной мере наследует все перечисленные проблемы. Она действительно эффективна для вычисления весов, но это не решает проблему. Предположим, что Вы вычислили веса и рассчитали взвешенную разницу между пациентом и каждой из модельных кривых на отрезке 1–14 дней. Очень может оказаться, что ни одна из альтернатив не будет доминировать, поскольку начало болезни будет «более похоже» на одну модель, а «окончание» на другую. Тогда Вам придется вводить

¹² К классикам в данной области следует отнести Кини Р. Л., Райфа Х. И, а также Дейвисона М. Некоторые их книги были переведены на русский язык в 80-х гг. прошлого века. Следует обратить внимания на библиотечные разделы «многокритериальное принятие решений» и «многомерное шкалирование», иногда их относят к разделу «математическая статистика». Из более поздних работ следует ознакомиться с книгами В. Roy, которые, насколько мне известно, на русский язык не переводились.

веса для дней болезни, делить графики на «стадии» (начало, течение, выздоровление) и вводить «свертку сверток». В конечном счете Вы рискуете получить гигантский суперкритерий с непонятной логикой построения¹³.

Автор, конечно, не может оспаривать очень большой опыт разработчиков такой системы как MYCIN, которые рассматривали десятки или даже сотни параметров, но по личному опыту решения задач, аналогичных приведенной, советует ограничиться 7–8 важными параметрами. При дальнейшем увеличении количества параметров точность диагностирования не будет значительно изменяться. Важно просто выбрать значимые параметры. Собственно, критериальная задача — это и есть задача обоснованного выбора весов, при котором значимые параметры получают «большие» веса.

Одним из самых важных следствий развития критериальных методов, появившихся в 60-х гг., является понятие о «разделяющем пространстве/гиперплоскости/функции». Разделяющая гиперплоскость (она же решающее правило) — это некоторая геометрическая интерпретация, появившаяся изначально для задачи дифференциальной медицинской диагностики. Т.е. это функция такого вида, что она позволяет однозначно сказать, принадлежит ли исследуемая точка $A(a_1, a_2, a_3, \dots, a_n)$ некоторой замкнутой области или нет (см. рис. 83). Вид и сама возможность построения такой функции сильно зависят от конфигурации тех самых областей, которые нужно разделить, и их взаимного положения. Если между областями можно провести некоторую прямую так, что все точки разных областей окажутся по разные стороны от прямой, то говорят о «линейной разделимости». Первые перцептроны (прообраз нейронных сетей) были основаны на этом принципе.

Задача становится более сложной, если одна из областей выпуклая, а вторая вогнутая (см. рис. 85). Еще более задача усложняется, если области пересекаются, например представляют собой два облака. Одно облако — результат горения леса, второе — испарения с поверхности реки. Необходимо найти границы их взаимного проникновения. Очевидно, решить такую задачу с помощью одной разделяющей прямой невозможно. Необходима либо некоторая кривая, либо ее кусочная аппроксимация на основе касательных.

¹³ В 60-х гг. были популярны исследования, направленные на выявление некоторого наилучшего (для некоторой области) суперкритерия. Есть целые монографии, в которых приводятся десятки критериев с результатами их применения, например, к задаче медицинской диагностики. Общий результат в том, что никакого суперкритерия найти не удалось.

В данном случае вид разделяющей функции становится нелинейным второго, третьего (в максимуме обычно до 4-го) порядка. Еще большее усложнение задачи может быть связано с тем, что непрерывная по своей физической природе функция (картинка на сетчатке глаза) может быть представлена дискретным приближением (битовой матрицей на сетчатке робота). Основные математические теоремы о возможностях распознавания на основе такой гиперплоскости были доказаны в теории нейронных сетей М. Мински и С. Пейпертом.

Теперь кратко повторим ряд важных моментов, относящихся к рассмотренным вопросам, далее обсудим измерительные шкалы (8.1.2) и в конце (более подробно) — некоторые из часто применяемых критериальных методов (8.1.3). Итак, следуя приведенной схеме, следует учесть, что

- множество альтернатив может быть конечным, счетным или континуальным;
- оценка альтернативы может осуществляться сразу по нескольким критериям (каждый из которых, возможно, многомерен¹⁴), имеющим как количественный, так и качественный характер и измеренным на шкалах разного типа; без приведения шкал к общему виду сворачивать их в суперкритерий нельзя;
- режим выбора может быть однократным (разовым) или повторяющимся (например, допускающим обучение¹⁵ на опыте);
- последствия выбора могут быть известны заранее (выбор в условиях определенности), иметь вероятностный характер, когда известны только вероятности возможных исходов после сделанного выбора (выбор в условиях риска), или иметь неоднозначный исход, не допускающий введения вероятностей (выбор в условиях неопределенности).

Различные сочетания перечисленных вариантов приводят к разным математическим задачам выбора, изученным не в одинаковой степени¹⁶.

8.1.2. Измерительные шкалы. Сравнение альтернатив происходит на основе измерения характеристик объекта. Измерение — это алгоритмическая операция, которая данному наблюдаемому состоя-

¹⁴ Таковы, например, полное электрическое сопротивление, состоящее из активной и реактивной частей, медицинский симптом «сыпь», характеризующийся размером и цветом, и др.

¹⁵ В данном случае это равноценно «пересчету весов».

¹⁶ Существуют целые специальные разделы математики, посвященные этим вопросам, например, «теория игр», «исследование операций» и др.

нию объекта, процесса, явления ставит в соответствие определенное обозначение: число, номер или символ.

Важно отметить, что допустимые в дальнейшем способы обработки полученных данных сильно зависят от способа, которым они были измерены. Одним из наиболее важных моментов в измерении является тип измерительной шкалы. В теории измерений различают 6 основных типов шкал (в порядке возрастания мощности): номинальная, порядковая, интервальная, циклическая, отношений, абсолютная.

Номинальная шкала (*шкала наименований, классификационная шкала*) позволяет рассматривать объекты, про любые два состояния которых можно сказать только различимы они или нет, и только такие алгоритмы измерения, которые различным состояниям ставят в соответствие разные обозначения, а неразличимым — одинаковые обозначения.

- | | |
|--|---|
| 1. Либо $A = B$, либо $A \neq B$. | Здесь символ = обозначает отношение эквивалентности; в том случае, когда A и B — числа, это знак равенства. |
| 2. Если $A = B$, то $B = A$. | |
| 3. Если $A = B$ и $B = C$, то $A = C$. | |

Для обозначения классов на шкале¹⁷ используются слова, символы, цифры, их различные комбинации (например, шкала цветов — красный, оранжевый, желтый и т. д.); порядок их следования не имеет значения¹⁸ (рис. 57).



Рис. 57. Эквивалентные номинальные шкалы

Над обозначениями на номинальной шкале, даже если эти обозначения числовые, нельзя производить арифметические операции, а только операцию проверки их совпадения. С результатами операции проверки совпадения можно выполнять более сложные преобразования: считать количество совпадений, вычислять и сравнивать относительные частоты, выполнять статистические процедуры и др.

П р и м е р. В начале некоторого эксперимента вы нумеруете испытуемых. В конце упорядочиваете испытуемых по полученному результату. В общем случае никакой взаимосвязи между номером в начале и рангом в конце не будет.

¹⁷ Далее будем использовать термин «квантификатор», не наделяя его, однако, тем смыслом, который он может иметь в математике.

¹⁸ Если Вы цвету сопоставите длину волны, то это уже не будет номинальная шкала.

А вот шкала номеров квартир в доме позволяет установить некоторые закономерности. Бессмысленно делить номер 80 на 40, но можно отнять и сделать вывод о взаимном расположении квартир. Поэтому в зависимости от интерпретации такую шкалу можно отнести и к номинальной, и к порядковой. Рассмотрим шкалу цифр на дисковом телефоне. Какова она? Как видите, на практике бывает достаточно трудно понять, что некоторая шкала является номинальной.

Дальнейшим усилением номинальной шкалы является порядковая (ординальная, ранговая) шкала, которая позволяет некоторым образом сравнить классы, указать порядок следования квантификаторов.

В зависимости от того, могут ли совпадать квантификаторы¹⁹ на шкале, различают шкалы сильного (простого) порядка, слабого и частичного порядков. Дополнительно к аксиомам номинальной шкалы должны выполняться следующие аксиомы.

Сильный порядок:	Слабый порядок:	Частичный порядок:
если $A > B$, то $B < A$;	$A \leq B$, либо $A \geq B$;	имеются пары классов,
если $A > B$ и $B > C$,	если $A \geq B$ и $B \geq C$,	несравнимые между со-
то $A > C$.	то $A \geq C$.	бой, т.е. ни $A \leq B$, ни
		$B \leq A$.

Шкалы частичного порядка часто возникают в социологических исследованиях. Например, при изучении покупательского спроса субъект часто не в состоянии оценить, какой именно из двух разнородных товаров ему больше нравится.

Отношение порядка ничего не говорит о дистанции между сравниваемыми классами (см. рис. 58). Порядковые экспериментальные данные, даже если они изображены цифрами, нельзя рассматривать как числа, над ними нельзя выполнять действия, которые приводят к получению разных результатов при преобразовании шкалы, не нарушающем порядка. Например, нельзя считать среднее, делить значение на значение или коэффициент.

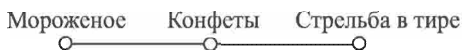


Рис. 58. Порядковая шкала предпочтений мальчика Паши

Экспериментально установлено, что Паша откажется сначала от мороженого, а затем и от конфет ради того, чтобы попасть в тир, но совершенно нельзя утверждать, что ради тира он откажется от двух коробок конфет или четырех порций мороженого.

¹⁹ У термина «квантификатор» имеется четкий математический смысл. Пользуясь им, всегда следует оговаривать, что мы под этим понимаем в данном контексте. В данной работе квантификатор — это деление на шкале.

Практически все психологические шкалы являются порядковыми. Если результаты некоторого психологического теста для двух субъектов равны a_1 и a_2 , то можно говорить о том, что первый субъект «более способен» к некоторой деятельности, либо «менее способен», либо «способности примерно одинаковы», но совершенно ошибочно говорить, что первый субъект «в a_1/a_2 раз более способен». Напротив, если один штангист поднял 100 кг, а другой — 200 кг, то вполне справедливо заявить, что «второй в 2 раза сильнее», однако здесь речь идет о другом типе шкалы.

Стремление получить возможность «оперировать с числами» приводит к дальнейшему усилению до интервальной шкалы, для которой известны расстояния между любыми двумя квантификаторами. Расстояния выражают в единицах, одинаковых по всей длине шкалы. Длина интервала не зависит от места его расположения на шкале. Арифметические операции допустимы над длинами интервалов, но не над самими значениями квантификаторов, поскольку «0» в таких шкалах не единственен.

Пример²⁰. Воду нагрели от 9° до 18°C. Можно ли утверждать, что температура воды увеличилась в два раза? Те же самые измерения по шкале Фаренгейта дали бы результат 37°–42°F. (Связь между шкалами Фаренгейта и Цельсия выражается формулой $F = 5/9^\circ C + 32$).

Некоторые величины по своей физической природе либо не имеют абсолютного нуля, либо допускают свободу выбора в установлении начала отсчета и поэтому измеряются в интервальных шкалах: температура, время, высота местности. В статистике центральные моменты (в том числе дисперсия) имеют объективный физический смысл, а начальные моменты (в том числе среднее значение) являются относительными наряду с началом отсчета.

Приведем пример типичной ошибки в коде интеллектуальной программы, которая следит за состоянием здоровья больного. Программа должна реагировать на «существенное изменение» температуры тела. Возникла ситуация 40°C при норме 36,6°C. Программа вычислила относительное отклонение $(1 - 36,6/40) = 1 - 0,915 = 0,085$; поскольку отклонение «явно малое²¹», никаких действий предпринято не было.

²⁰ Заимствован из известного учебника *Перегудов Ф.И., Тарасенко Ф.П.* Введение в системный анализ: учебное пособие для вузов. — М.: Выс. шк., 1989. 367 с.

²¹ Вопрос об установлении пороговых значений в различных предметных областях сам по себе сложен. В инженерной практике существуют чаще экспериментально, реже аналитически вычисленные допустимые отклонения в 1%, 5%, 10% и очень редко в 50% от идеальных значений параметра.

Правильные вычисления выглядели бы так:

$$1 - |40 - 36,6|/|36,6 - 43| = 1 - 0,45 = 0,55.$$

Здесь 43°C — известный на сегодня предел температуры тела живого человека.

Шкала отношений является первой шкалой, над значениями квантификаторов которой допустимо выполнять арифметические действия «как над числами». Должна удовлетворять (дополнительно) аксиомам аддитивности:

если $A = P$ и $B > 0$, то $A + B > P$;

$$A + B = B + A;$$

если $A = P$ и $B = Q$, то $A + B = P + Q$;

$$(A + B) + C = A + (B + C).$$

Примерами величин, природа которых соответствует шкале отношений, являются длина, вес, деньги.

На ряде измерительных приборов шкала замкнута (разностная, циклическая, периодическая шкала). В этом случае речь идет об интервальной шкале с некоторым периодом $y = x + nb$, $n = 0, 1, 2, \dots$

Последней и самой «сильной» является абсолютная шкала. Она имеет абсолютный нуль и абсолютную единицу. Является единственной, уникальной. Важной особенностью абсолютной шкалы по сравнению со всеми остальными является отвлеченность (безразмерность) и абсолютность ее единицы. Указанная особенность позволяет производить над показаниями абсолютной шкалы такие операции, которые недопустимы для показаний других шкал, употреблять эти показания в качестве показателя степени и аргумента логарифма²². Такой шкалой является числовая ось.

В таблице 13 собраны²³ основные сведения о рассмотренных шкалах. Можно сказать, что чем сильнее шкала, с помощью которой производятся измерения, тем больше сведений об изучаемом объекте, явлении, процессе дают измерения. Естественно стремление каж-

²² Подобная необходимость часто возникает, например, при обработке психологических экспериментов. Для того чтобы определить взаимосвязи между исследуемыми величинами, применяется простая корреляция. Такой расчет допустим только для исходных данных, распределенных по нормальному закону. Реальные данные могут отклониться от нормального распределения. Поэтому в некоторых случаях (так называемое *log-нормальное* распределение) работают с логарифмами от этих данных, которые гарантированно распределяются нормально.

²³ Таблица заимствована из известного учебника Перегудова Ф.И., Тарасенко Ф.П. (см. список литературы).

Таблица 13

Типы измерительных шкал

Название шкалы	Определяющее отношение	Эквивалентное преобразование шкал	Допустимые операции над данными	Вторичная обработка данных
<i>номинальная</i>	эквивалентность	перестановка наименований	вычисление символа Кронекера	вычисление относительных частот и операций над ними
<i>порядковая</i>	эквивалентность предпочтения	монотонное (не изменяющее порядка)	вычисление символа Кронекера и ранга	вычисление относительных частот и выборок квантилей над ними, и операций над ними
<i>интервальная</i>	эквивалентность предпочтения сохранение отношения интервалов	линейное преобразование $y = ax + b$, $a > 0$, $b \in \mathbb{R}$	вычисление символа Кронекера, ранга и интервалов (разностей между наблюдениями)	арифметические действия над интервалами
<i>циклическая</i>	эквивалентность, предпочтении, сохранение отношения интервалов, периодичность	сдвиг $y = x + nb$, $b = \text{const}$, $n = 0, 1, 2, \dots$	то же, что для интервальной	то же, что для интервальной
<i>отношений</i>	эквивалентность, предпочтении, сохранение отношения интервалов, периодичность, сохранение отношения двух значений	растяжение $y = ax$, $a > 0$	все арифметические операции	любая подходящая обработка
<i>абсолютная</i>	эквивалентность, предпочтении, сохранение отношения интервалов, периодичность, сохранение отношения двух значений, абсолютная и безразмерная единица, абсолютный ноль	шкала уникальна	все арифметические операции; использование в качестве показателя степени, осциллограммы и аргумента логарифма	любая необходимая обработка

дого исследователя провести измерения с помощью более сильной шкалы. При этом необходимо, чтобы исследователь точно мог доказать, что шкала именно того типа, каким он ее полагает. В противном случае результаты исследования могут оказаться ошибочными. Иногда исходные наблюдения делают на слабой шкале, а затем за счет введения каких-либо допущений переходят к более сильной. Полученные в результате обработки результаты исследования зачастую еще раз преобразуют, теперь уже для третьего типа шкалы, обычно с целью обеспечения их наглядности. Например, в результате принятия решения по 7–8 параметрам строят трехмерное изображение.

Приведем пример шуточного исследования, с акцентом на типы шкал. Маша и Катя собираются на день рождения к Паше. Что подарить?

1. Генерируются варианты: 1 — фотоаппарат, 2 — бутылка коньяка, 3 — сервиз, 4 — набор CD-дисков. В ходе выдвижения альтернативы не оцениваются и не критикуются, в противном случае «мы вообще ничего не придумаем». Совокупность альтернатив образует номинальную шкалу, предпочтения на которой неизвестны (рис. 59). Маша и Катя проводят опрос экспертов — специалистов по Паше. Что из перечисленного больше любит Паша? На номинальной шкале измеряются частоты. За первую и вторую альтернативу высказались по 2 человека, за третью — четверо, шестеро за последнюю.

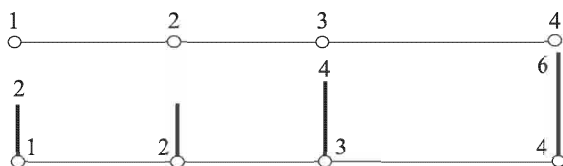


Рис. 59. Простейший частотный анализ на номинальной шкале

2. Маша и Катя строят порядковую шкалу предпочтений Паши (рис. 60).

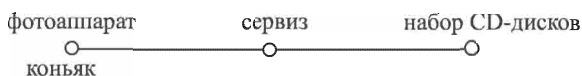


Рис. 60. Шкала слабого порядка

3. Катя и Маша задумываются о затратах, которые им придется понести — они оценивают варианты в некоторой валюте. Любительский фотоаппарат стоит меньше, чем хороший коньяк, следовательно, порядковая шкала становится «сильной», а с введением денежных обозначений ее тип изменяется на относительный (рис. 61).

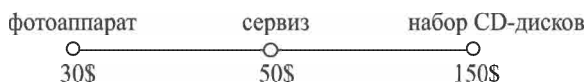


Рис. 61. Шкала отношений

4. Теперь у девочек есть две шкалы, одна из которых характеризует предпочтения Паши, другая — предпочтения Маши и Кати (рис. 62). Оценка вариантов с точки зрения Паши выражена в некоторых абстрактных единицах (шкала «приведена к 0» — от оценок вычтено 2). Строго говоря, приписывать квантификаторам «0, 2 и 4» смысл чисел нельзя²⁴, но мы сделаем такое допущение в целях простоты примера. Для этого приведем интервальную шкалу к относительной.

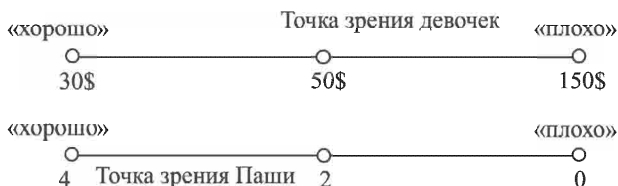


Рис. 62. Приведение шкалы к «0»

5. С точки зрения Паши набор дисков в 2 раза предпочтительнее сервиса. Маша и Катя полагают сервис в 3 раза предпочтительнее набора CD! Конечно, фотоаппарат в 5 раз предпочтительнее, но он может вызвать неудовольствие Паши (рис. 63).

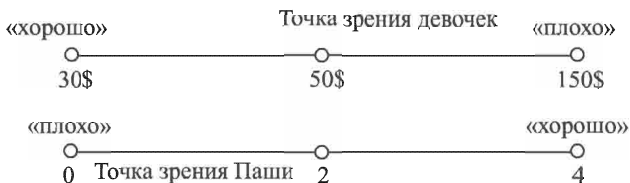


Рис. 63. Сопоставление предпочтений Паши и предпочтений девочек

Таким образом, девочки сходятся на альтернативе 50\$, оценка Паши — 2.

Решение задачи было бы более интересным (но более длинным), если бы мы рассмотрели еще несколько альтернатив.

Очевидно, что последний шаг — принятие окончательного решения в примере — остался несколько недосказанным. Это связано с большим разнообразием методов, которые могли бы быть применены.

²⁴ Этой проблеме посвящены целые диссертации по психологии.

8.1.3. Некоторые методики принятия решения.²⁵ В первую очередь следует отметить, что выбор может быть обусловлен одним критерием (или множеством критериев, свернутых в один), либо совокупностью критериев (несводимых к одному). Результат выбора может «быть как можно лучше», «соответствовать заданному уровню», «быть не хуже некоторого заданного уровня». Результат выбора может быть четко определен или неясен. Если удастся установить «явно лучшую» с некоторой точки зрения альтернативу, то ее называют доминирующей. Оставшиеся варианты, в свою очередь, называют множеством «явно плохих альтернатив». В случае, если среди нескольких вариантов нельзя указать ни лучших, ни худших (например, $q(X_1) = (3,5)$, $q(X_2) = (2,6)$ — оценки двух альтернатив по двум критериям), говорят о множестве несравнимых (не доминирующих) альтернатив или множестве Парето.

Сложившиеся методики осуществления критериального выбора называют «техниками» или «методами программирования». Говорят, например, о целевом программировании и компромиссном программировании.

Эффективность решений, принимаемых при помощи критериальных методов, зависит от предметной области. Существуют области человеческой деятельности, в которых критериальный выбор «показывает хорошие результаты». Например, эколого-экономические задачи, задачи упорядочивания объектов и ряд других. Эффект от применения критериальных методов иногда можно оценить, например, через процент «правильных» решений: «Компьютер ставил диагноз больным. Из 100 больных у 65 этот диагноз подтвердился». По мнению автора, на практике наилучшим (для критериального языка вообще) является показатель в 70%.

Рассмотрим метод главного критерия. В случае, если альтернативы (среди которых следует выбрать наилучшую в некотором смысле) оценены одним критерием или несколькими (но такими что имеется один «наиважнейший», а влиянием остальных можно пренебречь) либо существует возможность свернуть все критерии в один «суперкритерий»²⁶, тогда логично осуществить выбор по принципу «самый

²⁵ Вообще, сам вопрос «следует ли относить теорию принятия решений (ТПР) к методам ИИ?» достаточно спорный. Наиболее согласованное мнение, по-видимому, состоит в том, что эти методы имеют вспомогательное значение на этапах сбора исходной «экспертной» информации, а также на этапе анализа результатов работы системы — представление и интерпретация, например, геометрическая интерпретация.

²⁶ Вопрос о конкретном виде критериальной свертки решается каждый раз отдельно. Важно знать, что существуют наборы критериев, которые в принципе не могут быть сведены к одному.

лучший»:

$$x_{\text{опт}} = \arg [\max q(x)], \quad x \in X.$$

Здесь X — множество рассматриваемых альтернатив, $q(x)$ — значение критерия.

В случае, если среди критериев существует один (или группа, которая может быть свернута в один) самый важный, но пренебречь остальными нельзя (равно как и свернуть их в один), говорят о максимизации главного критерия с условием, что дополнительные критерии остаются на заданных им уровнях (условная максимизация):

$$x_{\text{опт}} = \arg [\max q_0(x) | q_i(x) \leq C_i, \quad i = 1, 2, 3, \dots, p], \quad x \in X.$$

Метод уступок (разновидность условной максимизации) заключается в том, что мы определяем некоторую небольшую величину, на которую мы готовы «поступиться» главным критерием. За счет этой «уступки» мы пытаемся улучшить значения второстепенных критериев. Второстепенные критерии могут быть упорядочены по важности.

Поиск альтернативы с заданными свойствами применяется для выбора любой альтернативы, значения частных критериев которой лежат в заданном диапазоне. В частном случае можно получить пустую альтернативу.

В случае условной максимизации или поиска альтернативы с заданными свойствами часто говорят об «уровнях притязаний». Эти уровни, диапазоны изменения значения критерия (критериев), как раз и задают нам варианты развития событий: «наихудший», «наилучший», «компромиссный» и т. п. Очевидно, что если уровни притязаний разных критериев пересекутся, то мы получим идеальную точку. Геометрически уровни притязаний очень похожи на линии равного уровня на картах.

Ниже приведена общая схема (рис. 64) некоторых наиболее часто применяемых методов [38].

8.1.4. Учебный пример многокритериального принятия решений. Рассмотрим пример постановки медицинского диагноза при помощи критериальных методов. Все данные, приводимые в примере, получены на реальных больных.

Имеются данные наблюдений за пациентом (табл. 14), предположительно болеющим одной из разновидностей желтухи (либо здоровым). Колонки таблицы содержат результаты лабораторных исследований, например, характеристики анализа крови, мочи и т. п. Первая ячейка — номер дня наблюдения за больным. Расшифровка названий не приводится, поскольку с точки зрения ИИ она ничего не добавит

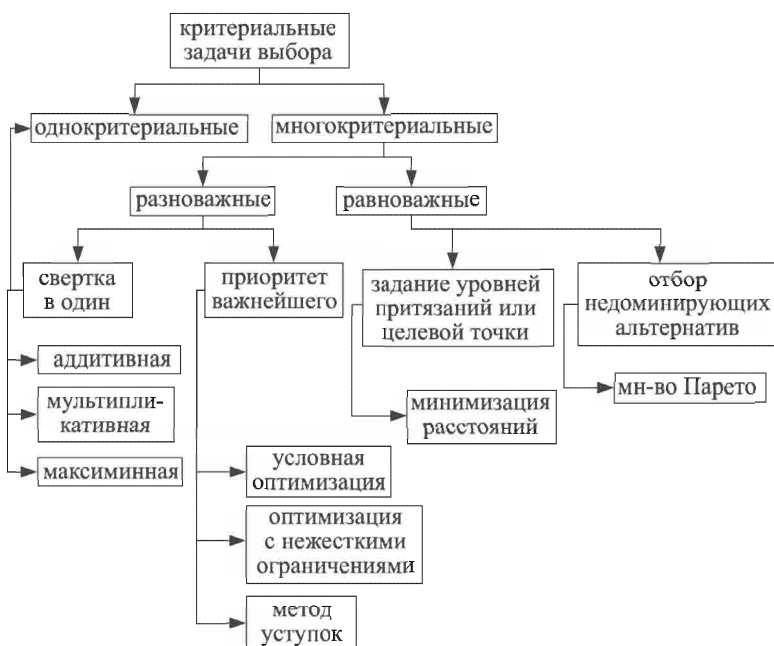


Рис. 64. Некоторые наиболее важные критериальные методы

к постановке задачи — инженер по знаниям не разбирается в медицине. Интерес представляет только сам тип измерительной шкалы. Определить тип придется самостоятельно, исходя из принимаемых значений. Если Вы попытаетесь поговорить об этом с доктором, то, пожалуй, Вы его напугаете.

Таблица 14

Исходные данные для диагностики

3	0,00	95,00	50,00	1,50	0,50	1,00	1,00	1,00	1,90	50,40
---	------	-------	-------	------	------	------	------	------	------	-------

При наблюдениях за здоровыми людьми по тем же параметрам получена таблица 15.

Таблица 15

3	0,00	4,80	10,00	0,40	0,30	3,60	0,00	0,00	0,00	0,00
---	------	------	-------	------	------	------	------	------	------	------

Наблюдения за пациентами, заведомо больными одной из разновидностей желтухи, дали следующие результаты.

Таблица 16

Паренхиматозная желтуха

3	0	10,52	28,00	1,30	2,00	4,69	0,00	0,00	1,00	0,00
---	---	-------	-------	------	------	------	------	------	------	------

Т а б л и ц а 17

Механическая желтуха

3	1,00	18,48	10,80	0,48	0,40	19,18	1,00	1,00	0,00	0,00
---	------	-------	-------	------	------	-------	------	------	------	------

Т а б л и ц а 18

Гемолитическая желтуха

3	0	4,80	30,00	0,30	0,40	3,60	0,00	0,00	0,00	1,00
---	---	------	-------	------	------	------	------	------	------	------

Изучение медицинской литературы и общение с экспертами позволили установить «существенные отклонения» для перечисленных параметров.

Т а б л и ц а 19

Гемолитическая желтуха

3	0,10	0,30	6,00	0,01	0,01	0,00	неизвестно	неизвестно	неизвестно	неизвестно
---	------	------	------	------	------	------	------------	------------	------------	------------

Т а б л и ц а 20

Здоровье

3	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,00
---	------	------	------	------	------	------	------	------	------	------

Т а б л и ц а 21

Механическая желтуха

3	0,10	8,00	2,00	0,03	0,01	3,20	неизвестно	неизвестно	неизвестно	неизвестно
---	------	------	------	------	------	------	------------	------------	------------	------------

Т а б л и ц а 22

Паренхиматозная желтуха

3	0,10	0,40	5,00	0,12	0,20	0,40	неизвестно	неизвестно	неизвестно	неизвестно
---	------	------	------	------	------	------	------------	------------	------------	------------

В качестве меры удаленности (реальной альтернативы от модели болезни) выберем функцию

$$D = \sum [w_i (X_{\text{модели}} - X_{\text{пациента}}) / (2 \times X_{\text{допустимое отклонение}})],$$

здесь $i = 1, \dots, 10$ — номера параметра; w_i — вес i -го параметра.

Конкретный вид «разделяющей функции» сильно влияет на правильность решения. К каждой задаче он подбирается индивидуально и часто имеет довольно сложный и объемный вид. Функцию D практически всегда можно определить несколькими различными способами.

$$D_{\text{(пациент – паренхиматозная желтуха)}} = 366,7;$$

$$D_{\text{(пациент – механическая желтуха)}} = 290,2 \text{ (наиболее близкая альтернатива);}$$

$$D_{\text{(пациент – гемолитическая желтуха)}} = 470,5;$$

$$D_{\text{(пациент – здоровье)}} = 418.$$

Важным также является вопрос о выборе весов. В данном расчете веса положены равными, что на практике встречается редко. Если

читатель попытается повторить расчет, то столкнется с ситуацией «деление на 0» или «деление на неизвестность». Реальные данные всегда содержат подобные сюрпризы, их разрешение — работа инженера по знаниям. Например, «0» заменяют некоторым малым значением (в данном примере 0,1). Вторая ситуация («деление на неизвестность») хуже. При полной невозможности установить это самое «неизвестное значение» его заменяют некоторой переменной. Значения функции D получаются в виде «число/неизвестное». По таким оценкам также можно принять решение.

В данном примере пациент действительно болел механической желтухой.

Получив набор оценок (в данном случае расстояний), необходимо выбрать наилучшую (в данном случае наименее удаленную). В ряде случаев это проблематично, поскольку оценки могут быть достаточно близкими. Вводят специальные «пороговые значения» в абсолютных цифрах или в процентах от оценки. Например, оценки 366,7 и 290,2 различаются на 21%. Часто используют «пороги» в 1%, 5%, 10% и даже 50%. Очевидно, что на оценку сильно влияют наборы весов « w_i », поэтому их и подбирают таким образом, чтобы «попасть» в требуемое пороговое значение. Этот процесс называют обучением. Набор весов « w_i » называют структурой предпочтения (эксперта или предметной области). Рассмотрение структуры предпочтений, полученной в результате обучения (а не опроса экспертов), само по себе интересно, поскольку позволяет судить о степени важности тех или иных параметров для принятия окончательного решения.

Один из наиболее интересных примеров практического применения аппарата многокритериального принятия решений автор наблюдал на лекциях австрийского профессора Г.П. Нахтнебела, прочитанных в 1999–2001 гг. в Алтайском государственном техническом университете имени И.И. Ползунова. Особое удивление вызывает тот факт, что методы ИИ, в частности критериальные, в его стране не являются «игрушкой для ученых», а широко применяются на практике, обеспечивают объективную, не зависящую от политики оценку общественно значимых проектов.

Итак, на реке Эрлаф в Нижней Австрии (1994) находилась малая гидроэлектростанция обводного типа²⁷. Т. е. станция была поставлена не на самой реке, а на обводном канале длиной 1100 м, при этом обводимая часть реки составила 2200 м. Река являлась, как принято

²⁷ В современной Европе, в противоположность современной России, малые станции очень популярны. Одна из причин — строительство такой станции не приводит к глобальному изменению местной экосистемы.

сейчас говорить, «экологически чистой» и в ней с избытком водилась форель. Хозяева станции, преследуя чисто экономические выгоды, стремились как можно больше воды забрать в обводной канал. При этом обводимая секция реки в летние месяцы вообще пересыхала (рис. 65). В гидрологии в качестве меры забора воды используется величина попуска (стока) ниже плотины, которая измеряется в $\text{м}^3/\text{с}$. Законодательные акты о природопользовании как раз и устанавливают минимальный попуск, который обеспечивает сохранение экосистемы. Проблема в том, что для больших и малых рек эта величина будет разной. Хозяева станции не нарушали закон, они держали попуск в $0,05 \text{ м}^3/\text{с}$ (что привело к деградации экосистемы). Общественность подняла шум, требуя изменить закон. Причем изменить так, чтобы найти разумный компромисс между экологическими и экономическими критериями. К основному экономическим следует отнести рабочую мощность станции и стоимость энергии. К экологическим: температуру воды ниже плотины, скорость течения, глубину ниже плотины, концентрацию растворенного кислорода и др. Очевидно, что критерии являются противоречащими. На основании каждого из рассмотренных параметров были построены уравнения. Ограничениями для задания области допустимых решений послужили известные из биологии, экономики и техники пороговые значения. Например, падение мощности станции ниже некоторого заданного значения вообще лишало смысла ее существование. Далее каждому параметру экспертными методами были назначены веса. В качестве суперкритерия использовали взвешенное Евклидово расстояние. Далее предстояло определить, при какой величине попуска удастся полностью сохранить экосистему реки и максимизировать выработку электроэнергии. Сложность задачи состояла в том, что река не постоянна. В разные сезоны и в разную погоду исходные данные для расчетов получаются разными. И, следовательно, пришлось провести не один расчет, а целую серию — 300. В результате было установлено, что минимальная величина попуска ниже плотины должна составить около $2 \text{ м}^3/\text{с}$. Закон был изменен, и хозяева станций выполнили эти требования.



Рис. 65. Идеальная точка

8.2. Вероятностные методы

Широкий класс подходов использует в качестве меры близости альтернатив вероятностные оценки. В настоящее время методы, позволяющие оценить вероятные исходы того или иного решения, его правильность, исследуют в рамках так называемой теории рисков. Теория носит прикладной характер, поэтому обычно говорят о рисках в предметной области, например в природопользовании. В математическом плане это обыкновенная теория вероятностей, причем в ее самых простых проявлениях. Суть теории рисков составляют не математические, а алгоритмические процедуры, связанные с особенностями измерения и вероятностной обработки данных в конкретной предметной области, а также интерпретации полученных результатов.

Рассмотрим простейший, но полезный и практически ценный пример применения методов теории вероятностей к задаче постановки медицинского диагноза по данным предыдущего параграфа. В качестве «решающего правила» возьмем формулу Байеса²⁸:

$$P(A/B) = [P(A) \times P(B/A)] / [\sum (P(A) \times P(B/A))].$$

Формула условной вероятности позволяет вычислить вероятность события « A » при условии, что произошло событие « B », т. е. $P(A/B)$ на основании имеющихся до опыта (априорных) вероятностей $P(A)$ и $P(B/A)$. Суммирование в знаменателе формулы проводится по всем гипотезам в полной группе событий.

П р и м е р 1. Необходимо вычислить вероятность того, что пациент болеет ангиной при условии, что наблюдается повышение температуры. Если пациент не болеет ангиной, то он болеет «не ангиной», например здоров.

$$\begin{aligned} P(\text{Ангина} / \text{Температура}) &= \\ &= \frac{P(\text{Ангина}) \times P(\text{Температура} / \text{Ангина})}{P(\text{Ангина}) \times P(\text{Температура} / \text{Ангина}) + P(\text{Не ангина}) \times P(\text{Температура} / \text{Не ангина})} \end{aligned}$$

Априорные вероятности $P(\text{Ангина})$ и $P(\text{Температура} / \text{Ангина})$ могут быть получены из анализа медицинской статистики для данной местности. Если, согласно записям в поликлинике, из 10000 больных в этот период года 7000 перенесли ангину, то априорная вероятность

²⁸ Bayes Tomas (1702–1761) английский математик, член Лондонского королевского общества. Поставил и решил одну из основных задач элементарной теории вероятностей (теорема Байеса). Занимался также богословием.

$P(\text{Ангина}) \approx 0,7$. Изучение медицинской литературы позволит установить также и $P(\text{Температура/Ангина}) \approx 0,8$.

$P(\text{Не ангина}) \approx 0,3$. Несколько сложнее с $P(\text{Температура/Не ангина})$, в худшем случае придется оценить экспертно. Аналогично получим оценки для знаменателя.

Пример 2. Под событием «В» будем понимать не одно событие, а целую группу событий, происходящих одновременно. Т. е. какова вероятность, что пациент болен паренхиматозной желтухой (он болен либо одной из 3-х желтух, либо здоров) при наличии симптомов «температура», «изменение цвета кожных покровов», «боли в боку» и т. п.? Максимально упрощая, можно считать, что вероятность синдрома равна произведению вероятностей отдельных симптомов.

Таким образом, для данных из критериального примера имеем:

$P(A_1) = P(A_2) = P(A_3) = P(A_4)$, т. е. полагаем все болезни равновероятными (A_i — имена болезней: паренхиматозная, механическая и гемолитическая желтухи, здоровье).

$P(B_1, B_2, \dots, B_{10}/A_1) = P(B_1/A_1) \times P(B_2/A_1) \times \dots \times P(B_{10}/A_1)$, очевидно, это даже «весьма заниженная» оценка.

Под вероятностями $P(B_1/A_1)$, $P(B_2/A_1)$, ... понимаем вероятность для симптома B_1 , B_2 и т. д. отклониться от «нормального для данной болезни» значения. Например, отклонение параметра «температура» от нормального для здоровья значения $36,6^\circ\text{C}$ до значения 42°C дает значение $P(42^\circ\text{C}/\text{здоров}) \rightarrow 0$, отклонение же $P(35,5^\circ\text{C}/\text{здоров}) \approx 0,8$ (весьма вероятно, что здоров).

Оценка в 0,8 может быть получена из анализа медицинской статистики для данного региона. Возможно, например, полагать, что допустимые значения симптома распределены нормально вокруг некоторого среднего (рис. 66).

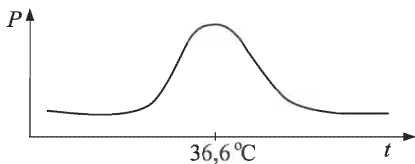


Рис. 66. Исходные данные могут быть распределены нормально

Исследования по определению параметров (вида распределения, математического ожидания, дисперсии и др.) такого распределения для разнообразных симптомов и различных заболеваний широко велись в Советском Союзе в 60-х гг. прошлого века.

Применительно к рассмотренным симптомам (на основе построенных более 3000 распределений²⁹) получены оценки (1 — механи-

²⁹ Работа проводилась в конце 80-х–середине 90-х гг. на базе Алтайской краевой клинической больницы г. Барнаул, станция санитарной авиации. Созданная экс-

ческая желтуха, 2 — паренхиматозная, 3 — гемолитическая, 4 — здоров) и сведены в таблицу 23.

Таблица 23

Значения для подстановки в формулу Байеса

	$P(0,0/A_i)$	$P(95,0/A_i)$	$P(50,0/A_i)$	$P(1,50/A_i)$	$P(0,50/A_i)$
$i = 1$	0,60	0,90	0,20	0,05	0,05
$i = 2$	0,05	0,10	0,90	0,85	0,85
$i = 3$	0,01	0,01	1	0,02	0,02
$i = 4$	1,00	1	1	0,50	0,70

$P(1,00/A_i)$	$P(1,00/A_i)$	$P(1,00/A_i)$	$P(1,90/A_i)$	$P(50,4/A_i)$	$\Pi \times 10^{-6}$
0,95	0,60	0,90	0,20	0,01	0,277
0,10	0,15	0,03	0,90	0,30	0,395
0,01	0,01	0,01	0,10	0,50	0
0,50	0,01	0,01	0,01	0,01	0,00175

$$P(A_1) = (0,25 \times 0,277 \times 10^{-6}) / (0,25 \times 0,277 \times 10^{-6} + 0,25 \times 0,395 \times 10^{-6} + 0,25 \times 0,0 \times 10^{-6} + 0,25 \times 0,00175 \times 10^{-6}) \approx 0,41$$

$$P(A_2) \approx 0,58$$

$$P(A_3) \approx 0,0$$

$$P(A_4) \approx 0,002$$

8.3. Нейронные сети

Одним из наиболее активно развиваемых сегодня направлений в ИИ являются нейронные сети — модель, представляющая собой в некотором смысле гибрид критериальных, вероятностных и логических методов³⁰. Можно также провести аналогию с правилами продукций. Нейронная сеть — это в некотором смысле «числовая запись» правил «если/то»³¹.

пертная система оценивала степень опасности болезни для жизни и рекомендовала или не рекомендовала вызов санитарного самолета в отдаленный район. Система показала высокий (до 70%) и по отдельным диагнозам очень высокий (до 98%) процент правильных решений на основе однократной консультации (даже без учета динамики состояния больного).

³⁰ Нейронную сеть при большом желании можно создать на основе правил продукций. Другое дело, что продукционная система не позволит создать достаточно большую сеть. Конечно, сделав это, мы рискуем получить замечание относительно «параллелизма» сети и «последовательного характера правил», но это тоже вопрос не однозначный. С логической же точки зрения возможно однозначное преобразование сети — правила.

³¹ Разрабатывавшуюся в 80-х гг. ЭВМ VI поколения японцы называли «нейро-компьютер» или «транспьютер», говоря проще — параллельная ЭВМ. К этому

Развитие теории началось с работ У. Макклоха (W. McCulloch) и У. Питтса (W. Pitts), рассматривающих функционирование головного мозга (1943 г.). В 1957 г. психолог Фрэнк Розенблатт (Корнелл, США) создал «персептрон» — электромеханическое устройство, моделирующее глаз улитки и его взаимодействие с мозгом. Модель различала буквы алфавита, но оказалась чувствительна к написанию. Буквы А, А и А для персептрона — три разных знака.

В 1969 г. вышла классическая работа «Персептроны», в которой М. Мински и С. Пейперт (М.И.Т.) доказали ряд теорем об ограниченных возможностях распознавания с использованием персептрона. Фактически они ввели в ИИ требование строгого математического обоснования используемых методов. Далее начала развиваться современная теория нейросетей, персептроны в классическом виде вышли из применения. В России в настоящий момент сильной считается красная школа нейроинформатики (например, Горбань А. Н., 1990). Рассмотрение нейросетей в задачу данной работы не входит. Ограничимся историческим примером персептрона.

Пример персептрона³². Построим персептрон, способный распознавать образы «буква А» и «не буква А» (рис. 67). Веса w_{ij} на первом шаге положим равными 1. Множество значений элементов x_{ij} положим состоящим из двух элементов $[0, 1]$. В качестве формулы выберем: $F(X, W) = \sum x_{ij} \times w_{ij}$, пороговое значение 10. Персептрон имеет вид $\sum x_{ij} \times w_{ij} \geq 10$. Очевидно, что $\sum x_{ij} \times w_{ij} \leq 16$. Обучим при помощи образца (см. рис. 67). Получаем $w_{11} = 0$, $w_{12} = 1$, $w_{13} = 1$, $w_{14} = 0$, $w_{21} = 1$ и т. д.

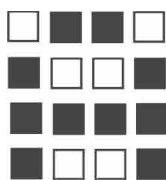


Рис. 67. Растровое представление буквы «А»

Теперь перейдем в режим распознавания и попытаемся предъявить, например, букву «R», изображение которой отличается тем, что точка (1,1) тоже черная (т. е. кодируется как 1). Поскольку неравенство выполняется: $\sum x_{ij} \times w_{ij} \geq 10$, то буква «R» распознана как «А»! Можно, конечно, установить строгое равенство, но тогда мы сами себя лишаем возможности распознавать «А», или «А», или «а» и т. д. Для того чтобы обучить персептрон распознавать еще и «А», очевидно, потребуется пересчитать веса w_{12} , w_{13} , w_{14} и, возможно, уменьшить пороговое значение. Классическая

же вопросу относятся исследования по «пороговой логике», «вычислительной геометрии» и др.

Среди значимых результатов, полученных СССР, — алгоритм «Кора», моделирующий деятельность человеческого мозга при распознавании образов (М. Бонгард, 60-е гг.).

³² Разработан совместно с к.т.н. Гарколь Н. С., АлтГТУ, 2003.



Рис. 68. Буква может быть написана по-разному

книга М. Мински и С. Пейперт как раз и посвящена этим вопросам. Применительно к примеру следует поступить следующим образом.

1. Определить минимальное количество точек, необходимых для однозначной идентификации буквы «А». Очевидно, что для этого понадобится определить, с каким алфавитом (набором символов) в принципе предстоит работать (рис. 68).
2. Определить особенности, присущие только распознаваемому символу. Например, неизбежной особенностью буквы «А» является наличие замкнутой верхней части. А вот наличие «завитушек в основании» не обязательно.
3. Описать первые два пункта логически, сопоставив каждому из утверждений пункта 2 n -местный предикат вида $f(x_{sp}, x_{gh}, x_{nm}, \dots)$, принимающий значение 1 (истина), если искомый элемент обнаружен, и 0 в противном случае. Индексы здесь — это любые целые числа, причем первый индекс — ширина сетки, второй — высота. Так для распознавания буквы «А» нам понадобятся одноместные, двухместные и трехместные предикаты (рис. 69).

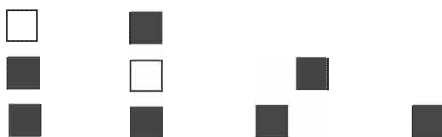


Рис. 69. Два трехместных, один двухместный и одноместный предикаты

4. Описать сам персептрон, который теперь приобретет гораздо более длинный и непонятный вид: $\sum f(x_i) \times w_i + \sum f(x_j, x_k) \times w_s + \sum f(x_j, x_k, x_l) \times w_h \leq Q$. Здесь буквами w с индексами обозначены веса предикатов, а за знаком суммы стоят одноместные, двухместные и трехместные предикаты. Пороговое значение Q как раз и задает минимально необходимое число соответствий. Например, предикатов первого порядка должно быть (смотрим на самый первый рисунок) не менее 10, второго — не менее 6, третьего — 4. Итого Q в самом лучшем случае может иметь значение 20. Можно было описать другие предикаты и получить другие Q . Персептрон в чистом виде чаще всего используется для учебных задач.

В медицине нейроном (рис. 70) называется клетка коры головного мозга. Ее отличительная особенность в том, что, возбуждаясь

под воздействием приходящего по специальным связям (от других нейронов) электрического сигнала, она генерирует собственный сигнал, который передает окружающим и т.д. Через некоторое время наступает процесс торможения — электрическая активность недавно возбужденных нейронов утихает, в то время как «отдохнувшие» нейроны вновь возбуждаются³³.

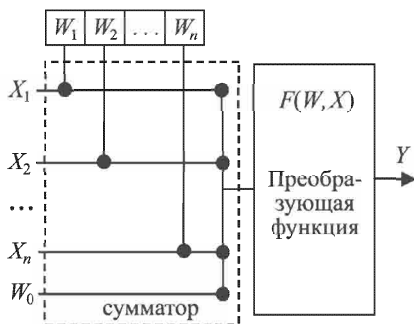


Рис. 70. Нелинейный преобразователь

На основе исследования карты электрической активности коры головного мозга (электроэнцефалография) можно делать, например, очень приблизительные предположения о роде мыслей, занимающих в данный момент испытуемого. Известно, что различные доли коры (лобная, затылочная и т.д.) отвечают за различные виды мышления, разным эмоциональным состояниям соответствуют различные варианты электрической активности. Таким образом, с точки зрения электроэнцефалографии мышление — это процесс изменения электрической активности в различных долях коры мозга.

В современных работах по этой теме, особенно написанных не инженерами, часто встречаются аналогии «искусственный нейрон — нейрон головного мозга». Ряд авторов вообще прямо указывают, что единственное отличие нейросети от мозга человека — в количестве нейронов. По их мнению, как только удастся создать сеть с таким же количеством нейронов, получится искусственный мозг. В этой связи очень полезно вспомнить введение к книге «Перцептроны» (М. Мински и С. Пайперт). В те времена подобные мысли высказывались относительно способностей перцептрона распознавать, а М. Мински и С. Пайперт показали, что в силу логических законов ряд геометрических фигур в принципе не может быть распознан этой моделью.

По мнению автора данной работы, используя нейронные сети, конечно, не следует забывать о том, что они создавались как «аналог клетки мозга», но и не следует придавать этой аналогии большого значения. Нейронную сеть следует рассматривать, прежде всего, как систему приближенных математических уравнений. Решая конкретную задачу с помощью сети, либо конструируя новую сеть, следует думать именно о математическом обосновании полученной системы, а уж никак не уповать на «способность самоорганизации сети».

С математической точки зрения нейрон — это нелинейный преобразователь, функция, на вход которой подано некоторое значение

³³ Именно на этом эффекте основаны картинки-иллюзии, когда Вы не можете определить направление перспективы. В первую секунду кажется, что изображение выпукло в сторону наблюдателя, в следующую наоборот.

(возбуждающий сигнал), а с выхода снято другое значение (величина возбуждения), связанное с входным некоторым сложным образом или даже вероятностно:

$$Y(X, W) = w_0 + \sum f(w_i \times x_i), \quad i = 1, 2, \dots, n;$$

здесь x_i — значения информационных входов, w_i — веса, причем w_0 — смещение.

Функционирует эта схема следующим образом. В сумматоре вычисляется возбуждение, полученное нейроном ($w_0 + \sum f(w_i \times x_i)$). На основе этой величины преобразующая функция вычисляет возбуждение самого нейрона — Y . Конкретный вид преобразующей функции подбирается с учетом специфики решаемой задачи. Если выбрать такую функцию, что Y будет принимать только значения 0 или 1, то, очевидно, получим правила продукций, только в несколько непривычной записи.

Практически полезных видов преобразующей функции не так много. При объединении множества нейронов в сеть³⁴ различают «слои» (рис. 71), связь между которыми может быть односторонней или с «обратным распространением». Очевидна аналогия с фронтом «готовых продукций» при управлении перебором правил в продукционной модели.

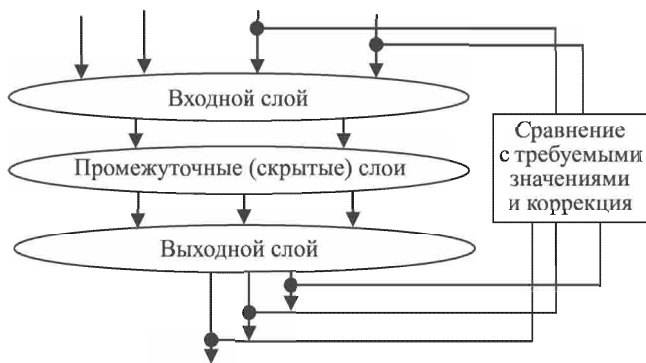


Рис. 71. Многослойная нейронная сеть

Под обучением сети понимается итеративный процесс вычисления весов w_i , позволяющих получать выходные значения на сети с заданной точностью. Очевидно, чтобы вычислить такие веса, необходимо иметь тестовые наборы значений входов и соответствующих им выходов. Такие наборы используются и в других моделях и назы-

³⁴ Речь может идти даже о миллионах нейронов, в противоположность правилам продукций, где оперируют только тысячами правил.

ваются обучающей выборкой³⁵. Изменение значений весов³⁶ может происходить с некоторым заданным шагом, либо по некоторому закону распределения случайной величины. Изменение происходит в направлении градиента в пространстве весов. Очевидно, при этом возможны две ситуации, когда через несколько шагов расчета сеть пришла в устойчивое состояние (веса w_i шаг от шага больше не изменяются), либо «зациклилась»³⁷.

Практическое применение имеют также однослойные сети, например, полносвязная сеть Хопфилда, в которой все нейроны связаны друг с другом.

Теоретически нейронная сеть с обратным распространением (сигнала между слоями) позволяет аппроксимировать любую функцию с любой точностью. На практике это возможно, если количество независимых входов X не превышает нескольких тысяч.

Если вектор выходных значений представляет собой сжатый или искаженный образ входного вектора, то говорят об автоассоциативной нейронной сети, в противном случае — о гетероассоциативной.

Исследования по нейроинформатике являются сегодня одними из фаворитов в области ИИ. Практическое применение нейронных сетей широко в военных, экономических, производственных системах. Нейронные сети «хорошо» реализуются аппаратно, в виде выполненных по специальному заказу микросхем. Такие микросхемы являются, например, «мозгом» всемирно известных систем ПВО, где решают задачу распознавания «неопознанного объекта на экране радара» на расстояниях свыше 300 км.

Одно из главных препятствий на пути дальнейшего широкого внедрения нейронных сетей, реализованных в виде специальных микросхем, в реальные системы управления состоит в том, что принципиальные возможности нейронных сетей математически не доказаны. По этой причине военные, например, не могут быть полностью уверены в надежности такой системы наведения.

Фактически теория, созданная в нейроинформатике, объединяет элементы всей предыдущей теории ИИ. Подробное знакомство с этой теорией не входит в задачу данной работы.

³⁵ Существуют разновидности алгоритмов обучения, которые не требуют предварительного обучения сети на обучающей выборке, а обучаются прямо в процессе решения реальной задачи. Таковы, например, самоорганизующиеся сети (карты) Кохонена.

³⁶ Очевидно, просматривается аналогия с методом уступок при критериальном принятии решений.

³⁷ Поведение сети должно быть хорошо понятно тем, кто имеет представление об оптимизационном методе градиентного спуска. Его изложение не входит в курс ИИ.

ГЛАВА 9

ЭКСПЕРТНЫЕ СИСТЕМЫ

9.1. Определение термина «экспертная система»

Широкую практическую известность методы ИИ получили при создании особого класса программ, называемых «Экспертными системами»¹.

Первые такие программы появились в конце 60-х гг. прошлого века и предназначались для создания искусственного «сверхума» в некоторой предметной области. Например, программа MYCIN мыслилась как «сверхдоктор» в области гематологии, CASNET — в области глазных болезней; различные модификации программы ELIZA претендовали на роль «суперпсихолога» и т. д.

В 60–70-х гг. большинство ЭС мыслились как некоторый программный механизм, позволяющий аккумулировать опыт лучших людей-экспертов (в некоторой предметной области), а затем консультировать менее опытных пользователей. Классические базы данных (основная теория которых к этому моменту уже сформировалась) не подходили на эту роль, потому что их язык запросов не являлся достаточно гибким, чтобы описать все многообразие запросов реального языка (мира); кроме того языки запросов к БД не допускали *рекурсивности*, т. е. возможности вызова запросом самого себя. В рамках такого подхода сформировались основные черты современной технологии применения и построения ЭС.

В современном понимании экспертная система — это узкоспециализированный программный комплекс, позволяющий либо очень быстро принимать стандартные решения (чаще всего по управлению техническими объектами), либо на основе длительного диалога с пользователем помогать в выборе некоторого решения (путем оценки вариантов, предлагаемых пользователем, и их коррекции с учетом множества факторов, возможно неизвестных пользователю, а также предложения собственных вариантов решения и их коррекции в соответствии с целями пользователя (рис. 72)).

¹ Некоторые специалисты четко разделяют термины «интеллектуальная система» и «экспертная система». В качестве отличия отмечают, что ЭС выросли из задач теории игр и до сих пор сохранили основные черты этого направления. Автор данной работы не склонен считать это замечание слишком важным. В конце книги приведена таблица некоторых наиболее заметных экспертных систем.



Рис. 72. Технологии взаимодействия предметного специалиста с ЭВМ с использованием ЭС и традиционная

9.2. Технология работы с экспертными системами

Решение прикладной задачи с помощью ЭС — это всегда целеустремленный² итеративный процесс (рис. 72), имеющий целью выработку решения, наилучшего с некоторой точки зрения.

Решение задачи без использования ЭС (другими программными средствами) чаще всего является разовым расчетом даже в случае, если для расчета используется итеративный алгоритм.

Основная разница в работе экспертных и «не экспертных» систем состоит в том, что первые в каждой своей итерации используют новый алгоритм действий в предметной области, отличающийся от алгоритма предыдущего шага не только параметрами, но и структурой. На рис. 72 представлены сравнительные схемы этапов работы с ЭС и без них.

² То есть в ходе достижения поставленной цели эта цель и ее параметры могут изменяться.

9.3. Структура ЭС

На рис. 73 приведена очень общая структура абстрактной экспертной системы. Скорее всего не существует реальных экспертных систем, полностью соответствующих приведенной структуре. Реальные ЭС содержат большое количество дополнительных блоков, специфичных для каждой предметной области. Тем не менее общая технология работы и назначение основных структурных элементов ЭС, исходя из приведенной схемы, должны стать понятны.

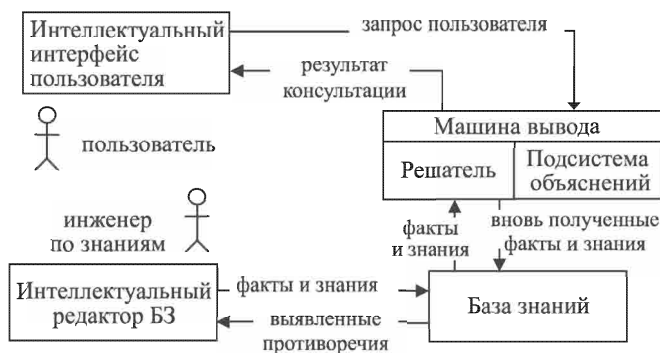


Рис. 73. Структура абстрактной экспертной системы

Главным структурным отличием ЭС от всех других типов программ является наличие *базы знаний* и, как следствие, способность к обучению и самообучению. Ее конкретный вид сильно зависит от избранной модели представления, но в наиболее общем виде она всегда будет содержать фактуальную и алгоритмическую части.

Так для системы правил продукций мы будем говорить о фактах и правилах (логического, чаще всего дедуктивного вывода).

Пример. Высказывание: крокодил (лапы, зубы, хвост, цвет).

Правило: крокодил (X, Y, Z, U), если X = «короткие», Y = «большие», Z = «длинный», U = «зеленый».

Для системы ИИ, основанной на фреймовой модели представления знаний, речь идет о фактах и методах (присоединенных процедурах).

Пример. Фрейм «Линия на экране»:

координаты начала 100, 100;

координаты конца 200, 200;

координаты соседней линии неизвестны.

Методы:

плавно вращаться вокруг своей середины;

исчезать;

появляться.

Возможно, у Вас возникает вопрос: «А что интеллектуального в линии, нарисованной на экране?» Ответ состоит в том, что фреймы в ИИ чаще всего используются как средство моделирования некоторого процесса или явления, поэтому линия на экране вполне может являться моделью живого организма и т. п. В процессе вращения она столкнется с другими линиями, в результате чего будет заполнено, например, поле «координаты соседней линии» и т. д. (система обучится).

Для сетевой модели (сетевых моделей великое множество) БЗ в общем случае содержит: состояние системы (например, значение некоторого параметра в узлах сети), избранную активную подсеть и правила вывода (включающие в себя и описание отношений, присутствующих в сети).

Машина вывода — это программа, ответственная за порядок и способ актуализации алгоритмической части БЗ³. Для разных МПЗ она, естественно, различна.

Для логической модели, где каждый раз происходит полный перебор всех логических уравнений, вне зависимости от уже имеющихся данных она имеет простейший вид последовательного автомата.

Для системы продукций (машина логического вывода), где выбор правила, которое следует актуализовать следующим, неоднозначен, она имеет систему предпочтений, основанную на некоторой детерминированной или вероятностной оценке, зависящей от ситуации⁴, сложившейся к этому моменту.

Для фреймовой модели машина вывода состоит как бы из двух частей. Единая общая часть отвечает за то, чтобы события, происходящие в системе фреймов, становились известны всем фреймам одновременно или в некоторой последовательности. Вторая часть — это обработчики событий самих фреймов, которые решают, какой из методов следует актуализовать в ответ на полученное сообщение о произошедшем событии.

Подсистема объяснений — это один из важнейших элементов ЭС, который в значительной мере определяет эффективность ее практического использования. Практическая реализация подсистемы объяснений часто использует те же процедуры, что и решатель, только в другом порядке. Поэтому часто можно считать, что машина вывода состоит из решателя и подсистемы объяснений. Специфика

³ Иногда в термин «машина вывода» включают еще и подсистему объяснений, как показано на схеме.

⁴ Поэтому в русском научном варианте термин «искусственный интеллект» до середины 70-х гг. звучал как «ситуационное управление».

задач, решаемых интеллектуальными системами, такова, что очень часто ценность имеет не само решение, полученное человеком или машиной, а процесс — ход решения, полученного человеком в диалоге с машиной. Подсистема объяснений — это программный механизм получения ответа на вопрос «А почему так?» или «А почему так, а не так?» относительно решения в целом и его элементов (шагов) в отдельности.

Не все интеллектуальные системы обладают такой способностью (имеют подсистему объяснений). Ее отсутствие может быть связано с трудностью программирования, но чаще с недостатками самой модели ПЗ. В классическом смысле подсистема объяснений существует только у систем, основанных на правилах продукции и семантических сетях.

Пример. Запрос (пользователя через интеллектуальный интерфейс к машине вывода) к интеллектуальной системе: «Каков будет курс доллара на завтра?»

Диалог системы.

1. Какова политическая стабильность? Ответ оператора: «Высокая».
2. Каков экономический прогноз? Ответ оператора: «Благоприятный».
3. Какова активность на бирже? Ответ оператора: «Низкая».

Ответ системы: «Предполагаемый курс 31 p.70 к.»

Объяснение системы.

Правило 1: так как политическая стабильность = высокая и прогноз = благоприятный, то параметр 1 = стабильность.

Правило 2: так как параметр 1 = стабильность и сезон = низкая деловая активность, то параметр 2 = рост низкий.

Правило 3: так как параметр 2 = рост низкий, то курс = курс_на_сегодня + малое случайное отклонение.

Заметьте, что при повторном запуске ответ системы мог бы несколько измениться, например, 31 p. 69 к.

Интеллектуальный редактор БЗ — это программа, позволяющая добавлять, удалять, модифицировать факты и правила, содержащиеся в БЗ. Определение «интеллектуальный» применительно к редактору БЗ стало активно использоваться в конце XX в. Под «интеллектом» применительно к данному классу программ понимается способность обеспечить целостность, корректность и непротиворечивость БЗ. Аналогичная проблема в теории баз данных решается при помощи ключей и нормализации, однако для концепции знаний

такое решение не подходит. Основная причина состоит в том, что знания представляют собой нечто единое, целое и многозначное. Отдельно вырванный из контекста элемент знаний не имеет смысла, поэтому, чтобы осуществить любую операцию над БЗ (добавление, удаление, модификация записи), необходимо проверить соблюдение целостности, корректности и непротиворечивости всей БЗ, причем на некоторую заданную глубину.

Поясним примером. Существует таблица — телефонный справочник. Первичный ключ «Ф.И.О. + телефон» (табл. 24).

Таблица 24

Ф.И.О.	Телефон	Адрес
Сидоров И.П.	233456	Ленина 46–80
Сидорова Р.О.	567845	Юрина 204–56
...	...	...
Сидоров И.И.	233456	Ленина 46–80

Предположим теперь, что мы хотели бы внести новую запись «Сидоров И.П./233455/Ленина 46–80». В случае ошибки ввода (например, телефон «233456») мы получим отказ от СУБД — нарушен первичный ключ. А если мы наберем десять разных фамилий, ошибаясь каждый раз (запишем всех на телефон «233456»), то не получим никаких сообщений, но ведь не могут в одной квартире жить 11 человек! При добавлении нового в БЗ должна активизироваться так называемая «присоединенная процедура», проверяющая корректность вводимой информации с точки зрения той информации, которая уже имеется в системе.

Рассмотрим подробнее, что означает «некоторая заданная глубина». Дело в том, что знания по своей природе обладают рекурсивностью — «вложенностью». Находясь в хранилище (в БЗ), элементы знаний образуют причудливые и часто трудно понимаемые человеком смысловые цепочки. Любой новый добавленный элемент становится как бы «мостом» между ними, что приводит к образованию новых, более крупных цепочек. На конце каждой такой цепочки — некоторый новый вывод относительно предметной области. Этот новый вывод вполне может прийти в противоречие с тем, что уже содержалось в базе ранее. Такие противоречия не только не исключение, а основной режим работы интеллектуальной системы, поскольку их появления заставят проверить и уточнить факты и правила, а также критически оценить правильность и точность того нового знания (моста), которое только что было внесено в систему.

Пр и м е р 1. Предположим, в системе содержатся записи⁵:

«Джон отец Мэри»;

«Мэри мать Билла».

Вы пытаетесь внести запись: «Билл внук Тома». Система обнаруживает противоречие на глубине 2, т.к. согласно имеющимся сведениям «Билл внук Джона»! Система требует уточнить, «точно ли это один и тот же Билл», и если выясняется, что разные, то модифицирует базу знаний:

«Джон отец Мэри»;

«Мэри мать Билла Джоскинза»;

«Билл О'Нил внук Тома».

Или исправляет ошибку:

«Том отец Мэри»;

«Мэри мать Билла»;

«Билл внук Тома».

Пр и м е р 2. Имеется технический объект, представленный на схеме (рис. 74).

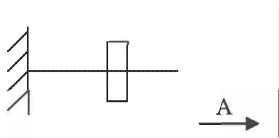


Рис. 74. Чертеж объекта, описанного в ЭС

На жесткой заделке над поверхностью «А» установлена штанга с гайкой.

Вы пытаетесь внести запись: «После аварии на поверхности «А» обнаружены многочисленные фрагменты цилиндрической формы, предположительно — отлетевшие гайки».

В системе схема описана приблизительно так:

соединены (жесткая заделка №1, штанга №1);

соединены (штанга №1, гайка №1);

маленькая (гайка №1);

шестиугольная (гайка №1);

цилиндрическая (штанга №1);

длинная (штанга №1);

пустотелая (штанга №1);

рядом (штанга №1, «А»).

Новая запись вызовет противоречие.

1. «Должно быть небольшое количество мелких фрагментов, принадлежащих одной разрушенной гайке». Подтверждаете?

⁵ Надо сказать, что пример с родственными отношениями - самая распространенная иллюстрация применения ЭС. Он приводится в учебниках так часто и в таком примитивном виде, что скорее убеждает в абсолютной бесполезности ЭС. В результате студенты оказываются неспособны самостоятельно придумать аналогичный пример, но без отношений типа «отец, сын и т.п.»

2. Иначе «фрагменты должны принадлежать разрушенной штанге». Подтверждаете разрушение штанги?
3. Иначе «фрагменты внесены извне».

Возможно читателю покажется, что выводы очевидны. Совершенно нет. Представьте себе, что речь идет о реальном техническом устройстве, например автомобиле. Вы сможете сказать, фрагменты каких деталей обнаружены при замене масла?

Остается только добавить, что в рассмотренных примерах отношения обладали свойством транзитивности, что бывает не всегда.

Интеллектуальный интерфейс пользователя — это программа, ответственная за организацию такого диалога с пользователем, который оптимальным образом приводит к достижению цели общения человека с ЭС. Так же как и в случае с редактором БЗ, определение «интеллектуальный» стало активно применять в конце XX в. «Интеллектуальность» интерфейса заключается в том, что в ходе диалога система помогает пользователю уточнить или совсем переформулировать цель его консультации с экспертной системой. Такая необходимость действительно имеет место быть, поскольку современные объекты, процессы, явления, моделируемые при помощи ЭС, настолько большие и сложные, что конечный пользователь не может помнить (а иногда и знать вообще) даже их имена (названия) и, следовательно, просто не способен грамотно сформулировать запрос и получить ответ.

9.4. Классификация ЭС

Экспертные системы можно классифицировать по различным основаниям: по типу используемой МПЗ, по области применения, по назначению, по показателям эффективности и др. (табл. 25, рис. 75).

Глубинной называется модель, которая позволяет обоснованно доказывать свои выводы. К глубинным знаниям относят абстракции, образы, аналогии, в которых отражается понимание структуры предметной области, назначение и взаимосвязь отдельных понятий, известные законы и теоретические основания.

Поверхностные знания касаются в основном внешних проявлений объекта исследования. Они позволяют делать некоторые (часто очень вероятные) предположения. Выводы, полученные на основе поверхностных знаний, возможно, являются столь же надежными и ценными, как и выводы, полученные с помощью глубинных, но они не обоснованы. Поверхностные знания обычно касаются лишь совокупности эмпирических ассоциаций и причинно-следственных отношений между понятиями предметной области. В принципе,

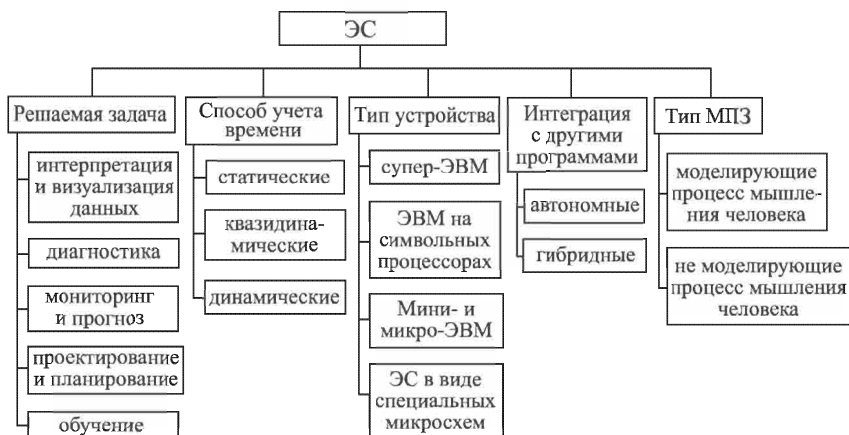


Рис. 75. Варианты классификации ЭС

почти любая модель ИИ изначально зарождается как поверхностная, и затем, доказав свою «полезность», получает твердое математическое обоснование, способ интерпретации и объяснения результатов работы модели.

П р и м е р. Мы пользуемся телефоном на основе поверхностных знаний о том, что, сняв трубку и набрав номер, можно соединиться с нужным абонентом. Подавляющее большинство владельцев телефонов не имеет и не нуждается в глубинных представлениях о структуре телефонной связи, схеме телефонного аппарата, которыми, безусловно, пользуются специалисты телефонной станции.

Т а б л и ц а 25

Классификация ЭС по типу МПЗ

Модель представления знаний	ЗНАНИЯ			
	Классификация по глубине		Классификация по жесткости	
	глубинные	поверхностные	мягкие	жесткие
Логика	—	+	—	+
Продукции	—	+	+	+
Фреймы	+	—	+	+
Семантические сети	+	—	+	+
Объектно-ориентированные языки	+	—	+	+
Стохастические модели	Эти модели не моделируют процесс мышления			
Критериальные языки выбора				
Нейронные сети				

В большинстве ЭС сейчас используются поверхностные знания. Введение глубинных представлений позволяет создавать БЗ большей мощности, так как глубинные знания более гибки и адаптивны, чем достаточно жесткие поверхностные. Классическим примером может служить медицина, где глубинные знания опытных врачей позволяют им порождать разнообразные способы лечения одной и той же болезни в зависимости от состояния больного, его возраста, наличия лекарств и т. д. Молодой или неквалифицированный врач часто действует по поверхностной модели: «Если кашель — то пить таблетки от кашля, если ангина — то эритромицин» и т. п.

Глубинные знания образуются как результат обобщения первичных понятий предметной области в некоторые более абстрактные структуры. Часто эти структуры не имеют вербального описания. Так в медицине введено понятие «синдрома» — сочетания симптомов, которым пользуются для обобщенного и углубленного описания болезни. Отметим, что степень глубины и уровень обобщенности знаний прямо связаны с опытом экспертов и могут служить показателем профессионального мастерства последних.

«Жесткость» и «мягкость» знаний указывает на возможность получать однозначные, четкие рекомендации при заданных начальных условиях, либо множественные, расплывчатые решения и рекомендации. Согласно этой классификации можно и сами предметные области относить к жестким и мягким. Современная тенденция в развитии ЭС состоит в том, чтобы от широкого применения жестких поверхностных моделей в большинстве задач переходить к мягким глубинным.

9.4.1. Классификация по решаемой задаче. ЭС интерпретации и визуализации чаще всего применяются в экономических системах. Дело в том, что расчетные коэффициенты, используемые, например, директором в ежедневном управлении предприятием⁶, трудно не только проанализировать, но даже просто удержать в голове. Сам анализ значений коэффициентов часто не очень сложен и заключается, например, в том, что значение некоторого коэффициента не должно выходить за определенные рамки при условии, что значение другого коэффициента находится в других определенных рамках, и так несколько сот, а то и тысяч взаимосвязанных коэффициентов. В этой ситуации чрезвычайно важно представить данные наглядно, т. е. визуализировать. Простейший вариант — создать диаграммы (столбиковые, круговые, точечные и др.).

⁶ Например, «коэффициент ликвидности» или «теневые цены» и еще несколько сот коэффициентов.

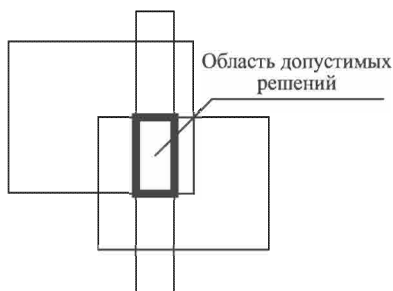


Рис. 76. Выбор решения, удовлетворяющего нескольким ограничениям

Однако в случае, если коэффициенты находятся в определенной зависимости, необходимо использовать более сложную визуализацию (рис. 76). Если бы рис. 76 был представлен тремя столбцами цифр, то вряд ли Вы смогли бы быстро принять решение.

Визуализация не всегда возможна. Простейший вариант — истинная мерность пространства⁷ превышает 3. В этом случае столбцы

цифр следует заменять развернутыми пояснениями, например, в текстовом виде. Важно, чтобы пояснения были краткими (снижали объем данных, удерживаемых в голове) и содержали уже выводы о причинах и следствиях, зависимостях, возможных решениях и их последствиях. Для решения задач интерпретации имеется целая теория математического и статистического анализа данных.

ЭС диагностики были разработаны одними из первых. Совместно с ЭС мониторинга они решают задачи непрерывной диагностики технических объектов, например ракет в момент запуска и в процессе полета. Более общо можно сказать, что такие ЭС выполняют процесс отнесения объекта к некоторому классу и обнаружения места и вида неисправности. Неисправность — это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии.

Особенность ЭС мониторинга состоит в том, что часто необходимо учитывать очень большое число параметров, непрерывно поступающих от сотен и даже тысяч датчиков с различным типом шкал. При этом часто необходимо придерживаться реального масштаба времени для того, чтобы вовремя дать сигнал тревоги.

ЭС планирования и проектирования имеют много общего с системами визуализации и интерпретации. Например, такая система, проанализировав чертеж электрооборудования нового автомобиля, может предсказать места возможного механического повреждения

⁷ Если у Вас есть, например, 4 столбца цифр, то это совсем не значит, что Вы имеете дело с 4-х мерным пространством. Вполне возможно, что среди столбцов имеются зависимые, либо такие, что их влиянием на принимаемые решения можно пренебречь. На практике значительная часть исходных данных не является «самостоятельно значимой».

проводов в процессе эксплуатации. Для решения этой задачи будет использована и специальная «когнитивная» графика, и методы перебора вариантов, и ряд специфических методов, которым систему обучат опытные инженеры-проектировщики. В современных ЭС последнее выполняется чаще всего при помощи модели сценариев (прецедентов). Если проект осуществляется в области экономики, то вместо чертежей мы будем иметь бизнес-план.

ЭС прогнозирования логически выводят вероятные следствия из заданных ситуаций. В прогнозирующей системе обычно используется параметрическая динамическая модель, в которой значения параметров «подгоняются» под данную ситуацию.

ЭС обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильные решения. Они аккумулируют знания о гипотетическом «ученике» и его характерных ошибках, затем в работе они способны диагностировать слабости в познаниях обучаемых и находить соответствующие средства для их ликвидации.

В общем случае все системы, основанные на знаниях, можно разделить на системы, решающие задачи *анализа*, и на системы, решающие задачи *синтеза*. Основное отличие задач анализа от задач синтеза заключается в том, что если в задачах анализа множество решений может быть перечислено и включено в систему, то в задачах синтеза множество решений потенциально и строится из решений компонент или подпроблем. Задачами анализа являются: интерпретация данных, диагностика; синтеза: проектирование, планирование. Комбинированные задачи: обучение, мониторинг, прогнозирование.

Автономные ЭС работают непосредственно в режиме консультаций с пользователем только для «экспертных» задач, при решении которых не требуется привлекать традиционные методы обработки данных (расчеты, моделирование и т. д.)

Гибридные ЭС представляют программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирование или системы управления базами данных) и средства манипулирования знаниями. Это может быть интеллектуальная надстройка над пакетом прикладных программ или интегрированная среда для решения сложной задачи с элементами экспертных знаний. Несмотря на внешнюю привлекательность гибридного подхода, следует отметить, что разработка таких систем являет собой задачу на порядок более сложную, чем разработка автономной ЭС. Стыковка не просто разных пакетов, а разных методологий (что происходит в гибридных системах) порождает комплекс теоретических и практических трудностей.

Если говорить о современном состоянии разработок ЭС, то тенденции за рубежом таковы:

задачи диагностики, интерпретации, проектирование →

→ мониторинг, обучение;

статические → динамические;

автономные → гибридные.

Статические ЭС разрабатываются в предметных областях, в которых база знаний и интерпретируемые данные не меняются за время решения задачи. Они стабильны. Например, диагностика неисправностей в автомобиле.

Квазидинамические ЭС интерпретируют ситуацию, которая меняется в течение некоторого фиксированного интервала времени. Например, микробиологические ЭС. Лабораторные измерения снимаются во время технологического процесса один раз в 4–5 часов (производство лизина, например) и анализируется динамика полученных показателей по отношению к предыдущему измерению.

Динамические ЭС работают с изменяющимися во время решения задачи данными, часто в сопряжении с датчиками объектов, иногда в режиме реального времени с непрерывной интерпретацией поступающих данных. Пример: гибкие производственные системы, мониторинг в реанимационных палатах и др.

Некоторые из хорошо известных ЭС представлены в табл. 38.

9.5. Технология разработки ЭС

9.5.1. Основные этапы разработки. Технология создания традиционных типов ПО предполагает следующие этапы: анализ требований, проектирование, программирование и отладка, тестирование. Технология создания интеллектуальных систем имеет ряд особенностей в их реализации (табл. 26).

Как Вы видите, появляется ряд новых стадий. Рассмотрим все стадии подробнее.

1. Анализ требований предполагает выявление основных сущностей, с которыми должна оперировать ЭС, а также возможных типов взаимосвязей между этими сущностями. На этом этапе оговариваются типичные запросы, на которые должна уметь отвечать система, а также границы применимости этих запросов. Например, ЭС анализа родственных связей на основе базы данных паспортного стола должна уметь выявлять «сводных братьев».

2. Анализ предметной области предполагает рассмотрение всех возможных сущностей и отношений между ними, ограничение самой

Т а б л и ц а 26

Этапы разработки ЭС

Наименование этапа	Решаемые задачи	Сроки
1. Анализ требований	Выбор проблемы; определение режимов работы будущей ЭС; определение требований к адекватности и точности результатов работы будущей ЭС	2–4 недели
2. Анализ предметной области	Выделение сущностей в предметной области, непосредственно относящихся к решаемой задаче; предварительное выделение фактов и правил, непосредственно относящихся к решаемой задаче	До полугода
3. Проектирование: 3.1. разработка прототипа; 3.2. оценка	Предварительный выбор МПЗ; предварительная разработка логической структуры; предварительная разработка программной структуры; предварительное кодирование и испытания на возможно большем материале (в короткие сроки); возврат к проектированию или анализу требований	4–8 недель
4. Программирование и отладка: 4.1. кодирование; 4.2. доработка до промышленной; 4.3. стыковка ЭС	Выбор языка программирования; кодирование, создание интерфейса, службы помощи и документации; интеграция ЭС в технологический процесс предприятия; предварительное обучение путем заполнения баз знаний результатами работы других ЭС	4–8 недель
5. Обучение	Работа ЭС в режиме решения реальных задач под присмотром инженера по знаниям (редактирует БЗ)	До полугода
6. Сверка с эталоном	Проведение показательных экспериментов и определение степени достижения целей проекта по созданию ЭС	1 неделя
7. Поддержка ЭС	Поиск логических и программных ошибок, выпуск новых версий, адаптация к изменившимся условиям эксплуатации	До 5 лет

предметной области и математическое описание возможных и допустимых типов сущностей и отношений. Например, для БД паспортного стола возможными сущностями будут: «мужчина», «женщина», «дом», «квартира», «улица». Отношениями: «муж», «жена», «дочь»,

«сын», «брат», «зять», «сводный брат» и т. д. Степень детализации как раз и определит возможности ЭС.

3. Проектирование ЭС предполагает создание структуры ЭС. Существенной особенностью проектирования ЭС является зависимость структуры от избранной модели представления знаний. Так, например, выбор в качестве базовой продукционной модели автоматически означает использование редактора правил; выбор фреймовой модели — редактора объектов; выбор вероятностной модели — редактора вероятностей. Вы заметили, что во всех трех случаях необходим редактор базы знаний, что позволяет изобразить некоторую абстрактную структуру ЭС, которой удовлетворяет почти любая реальная система (рис. 77).

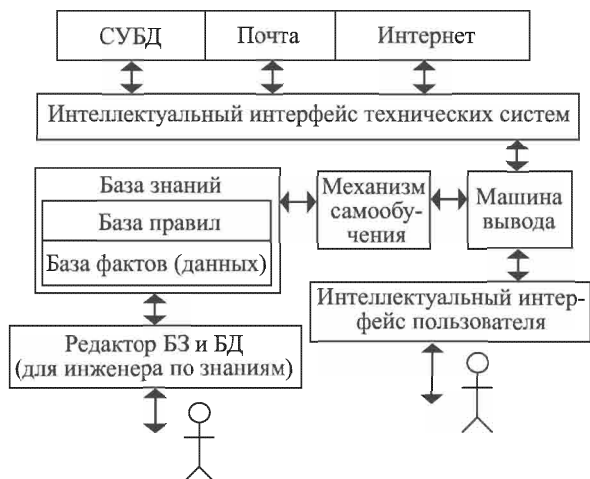


Рис. 77. Структура абстрактной ЭС

Таким образом, механизм работы ЭС можно описать следующими шагами:

1. инженер по знаниям наполнил БЗ (обучил ЭС);
2. пользователь задал системе запрос;
3. интеллектуальный интерфейс уточнил запрос и сформулировал цель для машины вывода;
4. машина вывода попыталась достичь заданную цель на основе знаний, полученных из БЗ;
5. цель не была достигнута из-за недостатка данных;
6. интеллектуальный интерфейс технических систем связался с соответствующими источниками данных;
7. машина вывода вновь попыталась достичь цель;

8. цель не была достигнута из-за получения противоречивых выводов;
9. для разрешения возникших противоречий запустился механизм самообучения, который попытался разрешить возникшие противоречия на основе заложенных в него правил;
10. новые знания, полученные в ходе работы механизма самообучения, поступили в редактор БЗ;
11. инженер по знаниям принял или отверг новые знания и факты;
12. новые знания и факты занесены в БД и БЗ;
13. возврат к п. 4.

Главное отличие ЭС от других программных средств — это наличие базы знаний, в которой знания хранятся в виде совокупности записей на некотором языке представления знаний (ЯПЗ), который позволяет легко изменять и дополнять базу знаний в форме, понятной специалистам — разработчикам экспертной системы. До последнего времени именно различные ЯПЗ были центральной проблемой при разработке ЭС. Сейчас существуют десятки языков или моделей представления знаний. Наибольшее распространение получили следующие модели: продукции, семантические сети, фреймы, исчисление предикатов 1-го порядка, объектно-ориентированные языки и др. Для этих моделей существует соответствующая математическая нотация, разработаны системы программирования, реализующие эти ЯПЗ, и имеется большое число реальных коммерческих ЭС.

Выбор той или иной модели определяется структурой знаний в конкретной предметной области. Выявление структуры предшествует выбору ЯПЗ и разработке базы знаний. Обоснование и конкретизация элементов знаний и их взаимосвязей происходят чаще всего в непосредственном контакте со специалистами предметной области — экспертами. Этот процесс называется извлечением знаний. Разработчики экспертных систем, занимающиеся извлечением и структурированием знаний, называются инженерами по знаниям⁸.

9.5.2. Некоторые практические рекомендации. Для человека, который впервые приступает к разработке экспертной системы, полезными могут оказаться некоторые советы. Прежде всего самостоятельной разработки любых интеллектуальных программ, если только Вы не занимаетесь научной деятельностью в этой области, следует всячески избегать. Разработка экспертной системы на порядок труд-

⁸ В этом месте рекомендуется выполнить практическую (лабораторную) работу «Разработка ЭС на инструментальном языке».

нее, чем разработка традиционной программы. Поэтому необходимо постараться доказать что:

- задача вполне может быть решена традиционными методами;
- необходимость в решении задачи возникает слишком редко.

Опытный инженер сумеет показать истинность этих утверждений применительно к абсолютно любой проблеме. В случае каких-либо непреодолимых обстоятельств следует утверждать, что «никакая интеллектуальная система не может принимать решения лучше нашего уникального эксперта⁹».

Ваши противники (начальники) скорее всего приведут доводы, касающиеся:

- необходимости автоматического распознавания шума и противоречий во входных данных;
- учета большого количества параметров;
- исключительной простоты применяемых экспертом методов принятия решений (эвристик), в силу чего их формализация становится «просто смешной задачей».

Не верьте им. Если автоматическое распознавание шума (для некоторых, отдельных предметных областей) действительно относительно легко реализуемо на практике, то задачи 2 и 3 в общем виде практически не решаемы. К сожалению, понимание этого факта приходит через достаточно длительный промежуток времени, заполненный отчаянными попытками инженера по знаниям «въехать в предметную область». Главная проблема, с которой Вы столкнетесь, будет состоять в том, что Вы неправильно выбрали собственно задачу автоматизации. Вашей главной головной болью станут поиски «доброго эксперта», который согласится проводить с Вами все свое рабочее и свободное время, рассказывая о решаемых им профессиональных проблемах с откровенностью пациента психоаналитика.

Вы не найдете такого эксперта.

Правильный подход состоит в том, чтобы с самого начала выбрать очень узкую задачу, решение которой происходит совершенно прозрачно, понятно и стереотипно несколькими известными, но почему-то еще неформализованными (в данной организации) методами.

Далее следует найти того пользователя, который в конечном счете, возможно, будет пользоваться Вашей программой. Пользователя следует попытаться обучить чему-либо. Квалификация конечных

⁹Ф.И.О. подставить самостоятельно.

пользователей — вот могильный камень большинства подобных проектов¹⁰.

Теперь следует озаботиться поисками эксперта, готового сотрудничать. Основной принцип поиска должен состоять в том, что Вы лично никакого эксперта не ищете. Эксперт должен искать Вас. Когда он Вас найдет (это происходит обычно примерно в середине проекта по созданию ЭС, приблизительно через полгода после того, как Вы полностью разочаровались в своих предыдущих экспертах), Вам придется сильно изменить задачу на автоматизацию, поскольку эксперта всегда интересует то, что интересует его, а не Вас. Эксперт потребует от Вас автоматизировать это, это и еще вот то. На все задания следует дать отрицательный ответ, оставив одну десятую, а лучше двадцатую часть от первоначальной задачи, причем переформулировав ее так, как это нужно Вам. В этот момент можете считать первую, наиболее важную часть проекта (выбор подходящей проблемы для решения) оконченной.

На следующим этапе создания ЭС выгоднее всего приступить к созданию прототипа или прототипной системы. При этом следует придерживаться правила минимальных затрат. Ни в коем случае не следует тратить много времени, сил и средств на разработку и внедрение прототипа. То, что Вы сделаете, это:

- совершенно, абсолютно и ни в коей мере не то, что в конечном счете будет внедрено;
- совершенно, абсолютно и ни в коей мере не то, что на самом деле требуется;
- размерность созданного прототипа, многочисленность поддерживаемых им режимов, полнота базы знаний и все достоинства интерфейса вместе взятые отвлекут пользователя и эксперта от правильной оценки принципиальных ошибок в разработке.

Прототипная система является усеченной версией ЭС, демонстрирующей жизнеспособность выбранного подхода, а также предназначенной для проверки правильности кодирования фактов, связей и стратегий рассуждения эксперта. Объем прототипа — несколько десятков правил, фреймов или примеров.

Прототипных систем будет много. Смиритесь. При создании первых прототипных систем даже не пытайтесь заставить систему вести себя разумно; достаточно того, чтобы она просто работала. В любом случае первые прототипы придется полностью переработать, поэто-

¹⁰ Существует неписанный закон (из которого, конечно, есть исключения), согласно которому человек, впервые начинающий применять возможности современных информационных технологий, в 40-летнем возрасте абсолютно не обучаем.

му выгоднее сразу принять явно ошибочные решения, чем принять другие ошибочные же решения только после долгих и мучительных доработок.

Далее, если приемлемый прототип создан (пользователи, эксперт и сам инженер по знаниям к этому моменту уже уменьшили свои аппетиты еще раз в 10 в «выборе проблемы»), следует более подробно проработать основные режимы работы системы путем внесения дополнительных фактов и правил, т. е. доработать систему до промышленной. Исходные коды в этот момент обычно полностью переписывают. Если на предыдущих этапах использовалась «пустая ЭС», то дописывают некоторые необходимые модули (рис. 78).

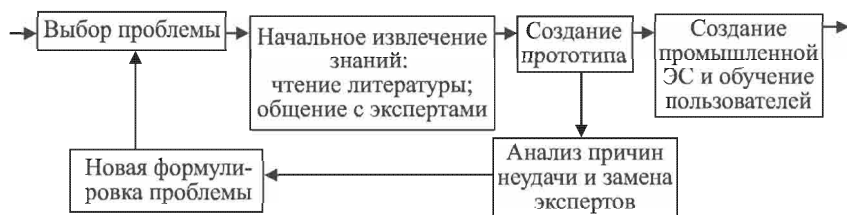


Рис. 78. Примерный цикл разработки ЭС

В ходе выполнения проекта инженеру по знаниям, т. е. Вам, придется обнаружить в себе ряд новых качеств, причем выраженных в крайней степени (либо бросить проект):

- потрясающую, совершенно нечеловеческую волю к победе и полное душевное спокойствие в ситуациях, когда все остальные участники проекта, включая экспертов, проявляют полное беспокойство и недружелюбие по отношению к проекту вообще и Вам лично в частности;
- большой процент лени; только ленивый инженер по знаниям способен «проскочить» практическую реализацию целого ряда совершенно тупиковых идей, кажущихся (эксперту) в настоящий момент потрясающим научным открытием;
- высокую восприимчивость ко всему новому; способность менять стиль собственной мысли, а также «смотреть на проект со стороны», оценивая выгоды и убытки от тех или иных проектных решений;
- интуицию;
- умение с большим удовольствием (для Вас и собеседника) общаться с людьми, которые лично Вам очень неприятны.

ЗАКЛЮЧЕНИЕ

В работе рассмотрены некоторые основные концепции и принципы, лежащие в основе современных методов искусственного интеллекта. Безусловно, работа не полна. ИИ — быстро развивающаяся область знаний. Появляются принципиально новые концепции и задачи.

Основная часть книги посвящена «классическим» методам ИИ, широко используемым на практике, но, увы, не являющимся на сегодняшний день наиболее перспективными. Студентам и аспирантам, желающим продвинуться в применении интеллектуальных систем для решения практически ценных задач, следует ознакомиться с целым рядом новых направлений: теорией искусственных нейронных сетей, методами кластерного анализа и многомерного шкалирования, фракталами и вейвлет-анализом.

К моменту написания книги у автора сложилось впечатление, что наиболее ценными в ближайшие годы будут работы, посвященные математическому или статистическому обоснованию адекватности и надежности вновь создаваемых моделей. В первую очередь это будет относиться к исследованиям по нейроинформатике. Применение нейронных сетей уже сегодня могло бы стать практически повсеместным, но до сих пор не имеется твердых доказательств¹¹ того, какие задачи и с какой надежностью способна решать эта модель.

Важную роль при создании таких доказательств в ближайшие годы начнет играть классическая математика и функциональный анализ, теория чисел и теория множеств.

Методы теории вероятностей, по мнению автора, будут значительно доработаны в применении к реальным задачам, в первую очередь задачам теории игр и систем массового обслуживания. Здесь наибольшее внимание, видимо, вызовет исследование распределений случайных величин и методов управления видом этих распределений.

Вновь создаваемые экспертные системы будут стремиться к принципу «прозрачности работы». Все они будут гибридными, сочетающими в себе как абсолютно predetermined, «детерминистские», так и «интеллектуальные» элементы. Большую роль будут играть методы фиксации данных — построение датчиков самого разного вида. Для военных экспертных систем основой станут спутниковые системы географического позиционирования, дальномеры и системы непрерывного картографирования местности.

¹¹ Кроме результатов практических экспериментов.

Задачи типа «объект–фон», «что изменилось в состоянии объекта», «каково направление движения» потребуют уже не единичных теоретических решения, а создания устройств, практически решающих их в массовом порядке.

Все эти направления и модели очень хороши, но применяться, безусловно, должны только тогда, когда без этого совершенно нельзя обойтись. По мнению автора, глубинное понимание новых моделей и их эффективное применение невозможно без знания некоторых основных, сегодня кажущихся уже довольно примитивными, концепций и методов, изложению которых и была посвящена эта работа.

Автор выражает благодарность своим коллегам и друзьям, оказавшим ценные консультации в ходе написания и публикации работы: проф. Александру Андреевичу Цхаю, доц. Александру Петровичу Яроцкому, доц. Наталье Станиславовне Гарколь, доц. Сергею Владимировичу Бутакову, доц. Константину Борисовичу Кошелеву, доц. Олегу Геннадьевичу Солодкому, доц. Максиму Николаевичу Веревкину.

г. Барнаул

Смолин Д.В.

ПРИЛОЖЕНИЯ

Prolog — это не язык программирования, это способ мышления . . .

Юрий Городиллов (коллега автора)

1. Инструментальные средства разработки ЭС

1.1. Языки программирования для ИИ. В настоящее время основными языками ИИ считаются LISP, Prolog и C++. Первые два — узкоспециализированные языки, разработанные специально для ИИ — являются языками программирования, последний — инструментальный язык общего назначения, достаточно мощный для того, чтобы создавать, в том числе и программы ИИ; является, пожалуй, самым популярным профессиональным языком.

Применение LISP и Prolog позволяет сосредоточиться на логике решения задачи, а применение C++ — добиться высокой скорости работы программы, реализовать нестандартные процедуры обработки. Современная вычислительная техника обладает очень высокими характеристиками быстродействия и «поддержки пользователя», поэтому долгое и кропотливое кодирование на C++ оправдано только при создании программ-оболочек и интерфейсов.

Пустая экспертная система, она же программа-оболочка или матрица — это программа, содержащая механизм вывода, средства ввода и редактирования фактов и не содержащая фактов конкретной предметной области. Матрицу можно наполнить собственными фактами, отредактировать правила, и она начнет делать выводы по интересующей Вас проблеме.

Другими, менее популярными языками ИИ являются SmallTalk, Forth, LOGO, РЕФАЛ¹², ПЛЭНЕР (Planner). Используются также Pascal, Basic и др.

1.2. Краткие сведения о Prolog. Prolog (программирование в терминах логики, логическое программирование) разработан в 1974 г. в университете Марсея (Франция)¹³ А. Колмаэро (Colmeraurer Alan) на основе фундаментальных работ А. Робинсона (John Alan Robinson) и Р. Ковальского (Robert A. Kowalski). Ряд специалистов считает Prolog учебным языком, отдавая первенство LISP. С точки зрения вычислительной мощности и быстродействия автор поставил бы эти языки рядом. Программа на LISP значительно короче, Prolog-

¹²Разработан В. Ф. Турчиным (МГУ, 1966).

¹³ Ближайший русский аналог называется РЕФАЛ (МГУ).

программа понятнее. Для языка, размер программы на котором сильно ограничен не памятью машины, а способностью программиста понимать, «что же он делает», последнее обстоятельство важно. Существуют несколько различных версий языка: Turbo Prolog, Visual Prolog и др.

Т а б л и ц а 27

Constants	Объявления констант
[GLOBAL] domains	Секция объявления нестандартных и/или составных типов данных. Может отсутствовать
[GLOBAL] database — имя_БД	Необязательная секция объявления предикатов для работы с внутренней базой данных
[GLOBAL] predicates	Секция объявления предикатов
Clauses	Секция объявления правил и фактов
Goal	Секция объявления внутренней цели. Может отсутствовать

Т а б л и ц а 28

Типы данных

symbol	строка, занесенная во внутреннюю таблицу символов системы
string	последовательность символов длиной до 64 Кбайт
char	1-байтовые символы
integer	2-байтовые целые числа со знаком
real	8-байтовые числа с плавающей точкой
ref	ссылочные числа базы данных
reg	регистры микропроцессора AX, BX, CX, DX, SI, DI, DS и ES
file	файл

Т а б л и ц а 29

Операции

Арифметические	+, -, *, /, mod, div
Реляционные	>, <, =, >=, <=, <>, >>
Математические функции	sin, cos, tan, arctan, ln, log, exp, sqrt, round, trunc, abs
Логические	And(«»,»), not, or, ! (отсечение)

Все переменные в Prolog-программе пишут с большой буквы: X, Masha. Если значение переменной нас не интересует, используют «_» — пустая переменная. Свободной называется переменная, значение которой еще не определено.

Т а б л и ц а 30

Ввод-вывод

write	вывод на экран
read	чтение с консоли

Параметры:

%d — обычное десятичное число (chars и integers)
 %u — беззнаковое целое (chars и integers)
 %R — как ссылочное число базы данных (database reference numbers)
 %X — как длинное 16-ричное число (strings, database reference numb)
 %x — как 16-ричное число (chars and integers)
 %s — стоки (symbols and strings)
 %c — как символ (chars and integers)
 %g — действительное в наиболее коротком из возможных форматов (default for reals)
 %e — действительные в экспоненциальной записи
 %f — действительные в формате с фиксированной точкой
 %lf — только для совместимости с языком C (fixed reals)
 \n — новая строка
 \t — табуляция
 \nnn — символ с кодом nnn

Т а б л и ц а 31

Стандартные предикаты Turbo Prolog

Стандартные предикаты охватывают				
Ввод/ вывод	Файловая система	Работа с экраном	Оконная система	Обработка строк

Стандартные предикаты охватывают				
Преобразования типов	Работа с БД	Работа с графикой	Работа с ОС	Низкоуровневые

Более подробно имеет смысл ознакомиться с документацией конкретной версии.

П р и м е р п р о г р а м м ы (Turbo Prolog):

domains

person, activity = symbol

predicates

likes (person, activity)

clauses

likes (ellen, tennis)

likes (john, football)

likes (tom, baseball)

likes (bill, X) if likes (tom, X)

goal

likes (X,Y)

Результат работы (произошла конкретизация значений переменных по имени отношения, см. исчисление предикатов первого

порядка):

X = ellen Y = tennis

X = john Y = football

X = tom Y = baseball

X = bill Y = baseball

1.3. Краткие сведения о Lisp. Создан в 1961 г. (Стэнфорд, США) группой профессора Джона Маккартни. Сокращение LISP (List processing) переводится как «язык обработки списков». В 70–80 гг. широко применялся для решения задач на основе древовидных структур, например, задач лабиринтного поиска и «генетического программирования» (Стэнфорд, проф. Джон Коз). Существует большое количество версий, наиболее известной, по-видимому, является COMMON LISP. Эта версия была поддержана AI Lab M.I.T. при создании LISP — машины в качестве языка системного программирования. К началу 90-х гг. фактически вышел из употребления.

Второе рождение Lisp связано с тем, что Lisp оказался чрезвычайно удобен для описания геометрических фигур. В этом качестве он широко известен пользователям разнообразных САПР — систем автоматизированного проектирования, например, AutoLisp (система AutoCAD).

Lisp-программа не имеет жесткой структуры. Представляет собой последовательность *s*-выражений (т.е. символьных, в заданной грамматике), поступающих последовательно на вход интерпретатора Lisp.

Типы данных

Таблица 32

Целое число	Например, 4	
Вещественное число	Определяется по наличию точки — 4.5	
Символьный атом	Аналог понятия переменной. Значение переменной по умолчанию равно самой переменной	Так называемые s(ymbolic)-выражения
Список	Обозначается (), например (A, B, C)	
Встроенная функция	Список приведен ниже	
Примитив	Только для AutoLisp	

Для некоторых функций, например CAR и CDR, существует сокращенный вариант: C-R, где вместо символа - можно подставить символы D или A. Примеры:

(CAAAR X)<=>(CAR(CAR(CAR X))) или (CADR X)<=>(CAR(CDR X))

Пример программы (μ – LISP).

Вычисление факториала:

(defun factorial (n)

(cond ((= n 1) 1)

(t (* n (factorial (- n 1))))))

Т а б л и ц а 33

Некоторые основные функции

Наименование	Краткое описание
QUOTE A	Функция не оценивает свой аргумент и выдает его в качестве результата
+, -, *, /	Арифметические функции (допустимы только над числами)
SETQ и SET	Функции присваивания
CAR A	Оценивает свой аргумент (A), который должен быть списком, и выдает в качестве значения первый элемент этого списка
CDR A	Оценивает свой аргумент (A), который должен быть списком, и выдает в качестве значения этот список, но без первого элемента
CONS A B	Оценивает аргументы, причем значение первого аргумента может быть произвольным s-выражением, а значением второго должен быть список. Результатом функции является список с головой A и хвостом B
APPEND A B	Пусть значениями A и B являются списки: (A...A) и (B...B), тогда значением функции APPEND будет список: (A...A B...B)
NULL X	Оценивает аргумент и выдает T, если значением T является NIL, и NIL в противном случае. Функция является аналогом NOT
ATOM X	Оценивает аргумент и выдает T, если его значением является атом, и NIL в противном случае
NUMBERP X	Оценивает свой аргумент и выдает значение T, если значением аргумента является число, и NIL в противном случае.
LISTP X	Оценивает свой аргумент и выдает T, если значением аргумента является список, и NIL в противном случае. Является обратной к функции ATOM
EQ A B	Функция сравнения двух атомов
EQUAL A B	Сравнение произвольных s-выражений
COND A A	Функции проверки условия
DEFUN	Функция, определяющая новые функции

2. Некоторые широко известные задачи, системы и методы

В разделе представлен небольшой обзор довольно старой литературы (1960–70 гг.), относящейся, в основном, к вопросам постановки медицинских диагнозов. Интерес состоит в том, чтобы пронаблюдать, во-первых, процесс развития интеллектуальных методов, в том числе тупиковых ветвей, во-вторых, отследить логику постановки задач. Ценность этих постановок в том (ср. с большинством современных предметных специалистов, работающих с интеллекту-

альными системами¹⁴ как с черным ящиком), что тогда делалась попытка приблизить логику работы системы к логике предметной задачи, построить простую, узко направленную систему. Кроме того описаны уже не учебные, а реальные задачи. По источникам можно примерно понять географию центров подобных исследований. Для кого-то эта глава может послужить введением в будущую специальность. Рассмотренные источники приведены в разделе «Литература, полезная для дополнительного ознакомления».

2.1. Задача постановки медицинского диагноза. Безусловно, это классификационная задача. Но что при этом классифицируется? Классифицируются болезненные состояния человеческого организма. Так ли это? Нужно сказать, что здесь не все просто. Медицинский диагноз может представлять собой как наименование болезненного состояния человеческого организма, так и наименование той болезненной причины, которая вызвала это состояние. В первом случае речь идет о классификации состояний человеческого организма, которые затем должны быть диагностированы по некоторому их описанию, во втором — о классификации причин и диагностировании причин, вызывающих изменение состояния организма. В некоторых случаях это различие может быть важным, т.к. для постановки диагноза во втором случае может оказаться, что нужно располагать не только описанием болезненного состояния, но и описанием предшествующего болезни состояния.

В общих чертах логическая структура классификации заболеваний человека может быть представлена в виде разветвленного дерева, на концах ветвей которого стоят диагнозы. Процесс постановки диагноза при этом представляется как движение по стволу и ветвям этого дерева от одного разветвления к следующему (в зависимости от ответов на вопросы, которые ставятся для ветвлений).

Чаще всего задача медицинской диагностики ставится как задача распознавания образов. В рамках этой теории существуют различные постановки задачи. Разделяют задачи:

- выбора;
- прогноза;
- классификации.

П р и м е р: постановки [33, 34]. «Детерминистская постановка задачи распознавания с учителем состоит в следующем: некоторые объекты, описываемые какими-то признаками, разделены учите-

¹⁴ Иногда (в шутку) складывается впечатление, что на современном этапе термины «интеллектуальная система» и «нейронная сеть» эквивалентны.

лем на 2 класса A и B . Нужно по статистическим выборкам A' и B' (обучающим выборкам) объектов классов A и B определить уже без указания учителя, к какому из классов принадлежит некоторый новый объект, не входящий в статистические выборки A' и B' ».

Дифференциальная диагностика [33] — выбор одной болезни из ограниченной группы заболеваний. Все болезни, входящие в группу, рассматриваются как полная система несовместных событий. Альтернативная дифференциальная диагностика — выбор из двух альтернатив.

Постановка [33] — по имеющимся симптомам (наименование и частота данного симптома при данном заболевании) рассчитать вероятность каждого из возможных диагнозов.

Различают, по крайней мере, три класса задач прогноза:

1. прогнозирование состояния здоровья практически здоровых людей в разных условиях и при различных воздействиях;
2. прогнозирование риска появления заболеваний;
3. прогноз течения заболевания и его исхода.

2.2. Некоторые широко применяемые формальные методы.

Существующие методы можно классифицировать по типу решаемых задач и используемой информационной модели (ИМ). Информационная модель в значительной степени определяется спецификой исходных данных. Исходные данные являются специфическими для различных областей медицины. В зависимости от применяемой ИМ различают вероятностные и логические методы, а также методы, основанные на определении расстояний.

Вероятностные алгоритмы различают по использованию априорной вероятности (не использованию) и объему применяемой выборки. Эти алгоритмы решают задачу определения вероятности данного заболевания при наличии данных симптомов. Логические алгоритмы долгое время считали все симптомы одинаково ценными для диагноза. Поэтому возникла идея приписать им веса. Симптомы, характерные для этого заболевания, имели высокий вес, нехарактерные или отсутствующие — низкий, безразличные — промежуточный. Каждый симптом имел один вес как признак заболевания A и другой как признак заболевания B . При вероятностном подходе исходят из соображения, что симптом наблюдается у данного больного с данной частотой (табл. 34).

На основе таблицы частот строят диагностическую таблицу (см. табл. 35).

Здесь по вертикали перечислены симптомы, по горизонтали — заболевания, а на пересечении — вероятность данного симптома при

Таблица 34

Номер наблюдения	100	101	102	103	113	114	115	117	119
Печень пальпируется		X	X	X		X	X	X	

Таблица 35

Заболевания	Априорная вероятность заболевания	Симптомы X_i (вероятность их наличия, %)					
		x_1	x_2	x_3	x_4	x_5	x_6
A1	0,100	0,1	49	50	01	00	01
...	...						
A14	0,014	90	09	01	90	00	

определенном заболевании. Такие данные позволяют решить обратную задачу — по имеющимся симптомам рассчитать вероятность диагнозов. Чаще всего решения этой задачи строятся на формуле Байеса и ее следствиях.

П р и м е р [39]. Дифференциальный диагноз в общем случае выбора между несколькими заболеваниями. Дано: ограниченная группа из d различных заболеваний $(D_1, D_2, \dots, D_d)$. Каждый больной страдает только одним из них. Имеется список из S признаков (симптомов, результатов лабораторных анализов) $S = (S_1, S_2, \dots, S_s)$. За прошлые годы накоплен материал и можно задать $P(S/D_j)$, где S — некоторый симптомокомплекс, можно также определить $P(D_j)$.

Определим $P(D_j/S)$. Апостериорная вероятность $P(D_j/S)$ пропорциональна произведению априорной вероятности $P(D_j)$ на функцию правдоподобия $P(S/D_j)$:

$$P(D_j/S) \sim P(D_j) \times P(S/D_j),$$

где

$$P(S/D_j) = \prod P(S_j/D_j), \quad j = 1, 2, \dots, d \text{ (произведение по } j\text{)}.$$

В случае, если k -ый симптом является непрерывным с плотностью распределения $f_{kj}(x)$ для j -го заболевания, значит вероятность наблюдения величины, лежащей в интервале $(x, x + dx)$, равна $f_{kj}(x)dx$. Если подставить это выражение в предыдущую формулу, то множитель dx сократится.

$$P(D_j/S) = P(D_j) \times P(S_j/D_j) / \sum P(D_j) \times P(S/D_j),$$

$$j = 1, 2, \dots, d.$$

Утверждается, что применение подобной методики позволяет повысить процент правильных диагнозов на 25%–30% по сравнению с врачом¹⁵.

У вероятностного подхода при решении задачи дифференциальной диагностики имеется геометрическая интерпретация (рис. 79), основанная на плотности вероятности кривой распределения значения признака. Многомерная область определения значений симптомов разбивается на подобласти, в которых преобладает, доминирует или всегда отсутствует некоторая болезнь.

Здесь I — зона пространства, где встречается только болезнь A1; II — зона, где преобладает A1; III — зона, где преобладает A2; IV — зона пространства, где встречается только болезнь A2.

Задача дифференциальной диагностики — определить по величине X_1 , X_2 , в какую область попадает точка (состояние больного). В этой области пространства нас будет интересовать вероятность болезни A1 и A2, о которой будем судить по плотности точек. Чем больше областей, тем выше точность диагностики. Каждая область формируется на основе группы обучения.

При этом если априорные вероятности заболеваний примерно равны, то для обучения системы можно взять одинаковое число больных (по каждой болезни), в противном случае группы обучения должны быть пропорциональны априорным вероятностям. Точность принятия решений на основе таких методов напрямую зависит от количества областей, на которые разбито все пространство определения значений признаков, и это количество ограничено. Кроме того, группа обучения должна быть достаточно большой. Отмечается, что проблема изучения вероятностей болезней, необходимых для медицинской диагностики, — это в значительной степени проблема массовых исследований, использования клинических архивов, обобщения материалов многих клиник. В вероятностных моделях все рассматриваемые альтернативы считаются несовместными, поэтому вероятностными методами решается задача выбора или дифферен-

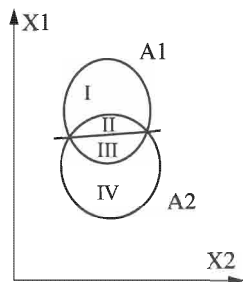


Рис. 79. Геометрическая интерпретация вероятностного подхода

¹⁵Необходимо учесть, что речь идет о ситуации 30-летней давности. Кроме того, среди практикующих специалистов имеется мнение о «европейской» и «американской» моделях диагностирования. Первые более доверяют собственному опыту и интуиции, вторые — показателям очень больших комплексных анализов и исследований. Такое соотношение, видимо, сохраняется и сейчас, потому как самая современная аппаратура значения опыта и интуиции не отменяет.

циальная диагностика (выбор из двух). Для принятия окончательного решения введено понятие «порогового превышения вероятностей», т. е. такого превышения вероятности одной из гипотез, которое соответствует требуемому превышению частоты правильных диагнозов над частотой неправильных. В частности в [39] утверждается, что вероятностный подход позволяет повысить частоту постановки правильных диагнозов на 25% по сравнению с врачом (посмотрите на реальные результаты применения подобной системы, табл. 36). В этом же источнике указано, что использование вероятностного подхода возможно и при небольших выборках, однако в этом случае необходимы некоторые оценки информативности выборки. Результаты принятия решения на основе малых выборок в источнике не описаны. Практические работы по созданию диагностирующих программных систем на базе вероятностных алгоритмов широко велись в 60-е годы XX века.

2.3. Логические алгоритмы. Представления о логических алгоритмах за последние 20 лет претерпели изменения. Изначально считалось, что логический алгоритм — это такой алгоритм, который учитывает лишь симптомы всегда присутствующие или всегда отсутствующие при каждом из диагностируемых заболеваний. Симптомы, которые встречаются при данном заболевании, в 50% случаев не учитываются. Симптомы кодируются в троичной системе счисления, т. е. учитываются наличие, отсутствие, неясность симптома. На основе логического алгоритма происходит формирование множества не доминирующих альтернатив. При этом задача постановки диагноза ставится как выбор одной альтернативы из списка возможных. В 1961 г. Ледли и Ластед модифицировали алгоритм и поставили задачу диагностики иначе. Теперь считалось, что оцениваемая альтернатива (пациент) может быть комбинацией известных альтернатив (иметь несколько заболеваний сразу). Если число симптомов n , а заболеваний m , то число возможных комбинаций симптомов и заболеваний 2^{n+m} , далее исключают комбинации, которые практически не встречаются (исключаются последовательно, путем запроса новой информации). После появления аппарата нечеткой логики логические модели также изменились (см., например, описание MYCIN). Надо сказать, что на сегодняшний день это одна из немногих групп моделей, приводящих к практическим результатам. Модель состоит из нескольких разделов: факты, правила продукций. Факты запоминаются в виде триплета: контекст—параметр—значение (пациент—возраст—значение). Каждому триpletу соответствует коэффициент определенности $[-1, 1]$ ПО, где -1 — отрицание, 1 — утверждение. В начале консультации часть триплетов не укомплектована, неко-

Т а б л и ц а 36

МЭС — диагностика. Сравнительные данные врачебной диагностики и экспертных диагностик ЭВМ за период с 01.01.91 по 31.12.1994, причина вызова — «острый живот»; г. Барнаул

Код нозоформы	Заключительные диагнозы по данной нозологической форме	Совпадения с заключительным диагнозом		
		по предвари- тельному ди- агнозу врача	ЭВМ	
			по 1-му диагнозу	по 2-му диагнозу
1	145	24	88	32
2	39	5	13	4
3	42	14	29	3
4	19	2	10	2
5	7	0	2	2
6	28	3	18	4
7	10	1	3	0
8	75	31	45	4
9	20	3	7	6
10	6	0	1	1
11	1	0	0	0
12	8	2	4	3
13	4	0	0	0
14	3	1	0	0
15	45	5	21	10
16	12	1	4	3
17	13	1	2	1
18	3	1	0	1
19	7	0	0	3
20	3	2	1	0
21	4	2	1	0
22	244	28	119	33

торые триплеты представляют собой цель консультации. Правила продукций: знания эксперта кодируются в виде *если* $\langle \text{предпосылки} \rangle$ *то действие* $\langle \text{ПО} \rangle$, где предпосылка — конъюнкция триплетов, действие включает присвоение значений параметрам триплетов. Правило может быть применено, исходя из известной предпосылки с целью присвоить значения параметрам триплета или исходя из триплета с целью определить, какие триплеты предпосылки необходимо определить. Таким образом, логические модели дополнились очень важной составляющей — деревом диагнозов.

2.4. Ассоциативный и каузальный подходы. После создания MYCIN выявились некоторые недостатки:

- не было попытки моделирования той стратегии рассуждения, которой пользуются эксперты-люди;
- не было попытки воспользоваться знаниями, заложенными в каузальных отношениях.

Система INTERNIST (Питсбург, США). При постановке диагноза была сделана попытка приблизиться к модели человеческого мыслительного процесса. Такой процесс позволяет проследить взаимосвязь между болезнями и их проявлениями. Процесс имеет две стадии:

- 1) ограничение пространства диагностирования (выбор правдоподобных гипотез);
- 2) применение некоторой стратегии (дифференциальная диагностика).

INTERNIST сначала идентифицирует набор заболеваний, характеризующийся некоторыми или всеми симптомами, имеющимися у пациента, затем пытается найти в этом наборе единственное заболевание, наилучшим образом объясняющее ряд выявленных симптомов. Далее производит поиск набора заболеваний, которым соответствуют оставшиеся симптомы, и процесс повторяется до тех пор, пока не останется ни одного необъясненного симптома. Знания в INTERNIST-1 представлены в виде «дерева заболеваний». Заболевания связаны отношением «разновидность». Узлы дерева могут быть терминальными и нетерминальными. Заболевания соотносятся со своими симптомами через два дополнительных отношения. Мощность каждого из этих отношений можно трактовать как вероятность. Отношение «вызывает», мощность [0..5]. Отношение «показывает» (инверсия «вызывает»). Его мощность оценивается через частоту данного симптома при данном заболевании [1..5]. В дополнение к отношениям «вызывает» и «показывает» задано несколько других. В конечном итоге каждому симптому ставятся в соответствие две характеристики: класс и значимость.

Класс — мера стоимости проверки правильности симптома. *Значимость* — насколько важно наличие данного симптома для постановки диагноза конкретного заболевания. Для управления ходом обработки знаний используется пара класс-значимость. Наиболее «дорогостоящий» вопрос задается первым. Далее по набору симптомов и мощностям связей считают «вероятность» каждого узла. При дифференциальной диагностике используются стратегии исключения или дискриминации (см. рис. 80, где П — причина, В — вызывает).

Одной из наиболее красивых моделей этого уровня является модель Вейса–Куликовского (рис. 81), разработанная на примере диагностирования глаукомы. Согласно этой модели, заболевание — это процесс, включающий переходы от одного патофизиологического состояния к другому. Диагноз — идентификация взаимосвязи между

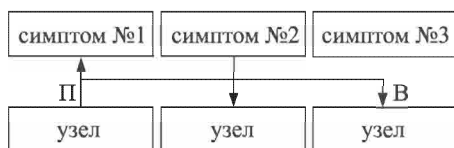


Рис. 80. Взаимосвязь причин и следствий в системе Internist

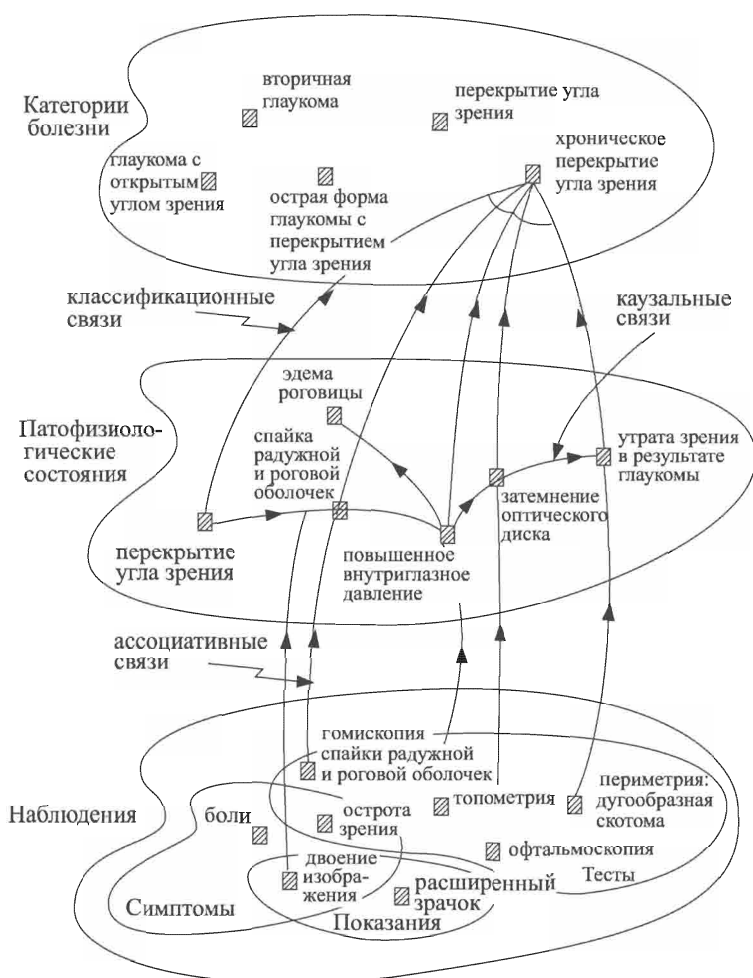


Рис. 81, а. Трехуровневая модель процесса болезни.
Weiss S.M., Kulikowski C.A. и др., 1978

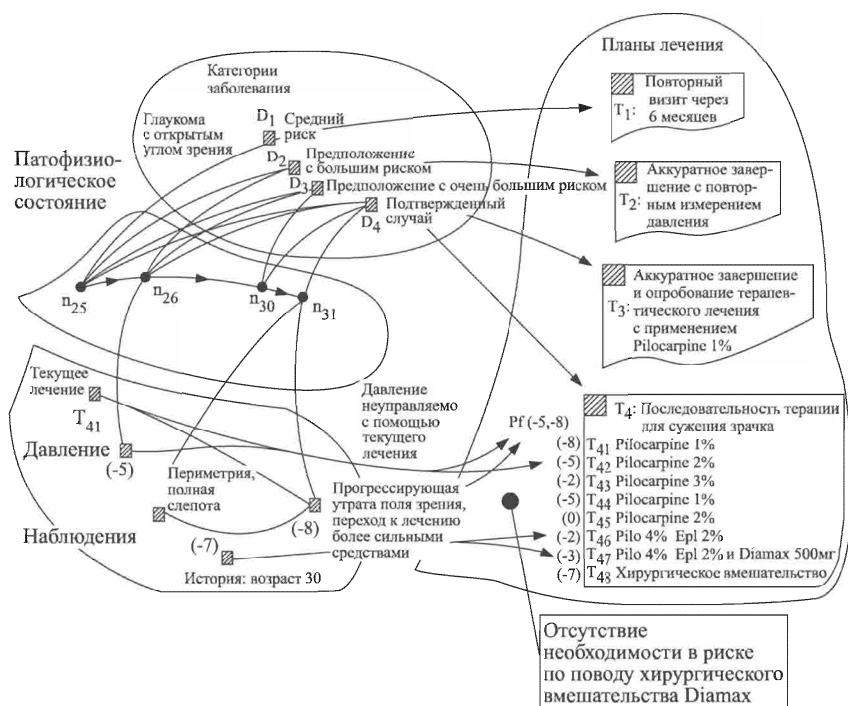


Рис. 81, б. Трехуровневая модель процесса болезни

картой каузальных маршрутов, свойственной пациенту, и категорией заболевания. Модель содержит четыре слоя знаний, три из которых — непосредственно модель заболевания, а четвертый — схемы ее лечения. Модель сетевая. Узлы сети — физиологические состояния, дуги — причинные связи между состояниями. Весь маршрут от начального до конечного состояния обычно отражает полный цикл развития болезни. Продвижение по этому маршруту соответствует нарастанию серьезности заболевания. Ниже слоя патофизиологических состояний находится слой «наблюдений». Узлы — симптомы и проявления, которые, возможно, связаны с различными стадиями болезни. Выше слоя патофизиологических состояний находится слой, ответственный за классификацию заболеваний на уровне более общих понятий. Процесс постановки диагноза сводится к поиску одного или нескольких каузальных маршрутов между патофизиологическими состояниями. Модель реализована в системе CASNET.

2.5. Практические приложения продукционной модели. Одной из наиболее известных (и практически результативных) на сегодняшний день является продукционная модель. При применении данной

модели к диагностике все же возникают некоторые проблемы:

- поддержание непротиворечивости больших БЗ;
- учет предыстории.

Некоторые авторы указывают на то, что продукционные системы имеют предел по количеству используемых правил, выход за этот предел означает, что непротиворечивость знаний в БЗ обеспечить уже практически невозможно. В частности называется цифра в 1000 правил. Используя продукционную модель, достаточно трудно описать процессы, происходящие в организме человека во времени. Кроме того реальные процессы могут быть циклическими, как, например, в случае хронического заболевания.

Условно-практическое приложение затронуто в [42]. Поставлена задача определения прогноза продолжительности жизни. Построена продукционная модель, исходными данными для которой являются вес, сложение, потребление жиров, количество холестерина в крови, потребление соли, раса, происхождение, потребление алкоголя, пол, возраст, характер, курение. На основе исходных данных определяются ожидаемая продолжительность жизни (Z_n) и дополнительный фактор риска. Принятие решения основано на формализованном знании экспертов (рис. 82).

На основе продукционной модели создана система MYCIN, которая предназначена для постановки диагнозов и лечения инфекционных болезней. Система решает задачу идентификации поражения организма и определение лечения. Факты в MYCIN запоминаются в виде триплета: контекст – параметр – значение (контекст — это пациент, параметр — атрибут контекста (возраст), значение — экземпляр параметра (25 лет)). Каждому триплету соответствует коэффициент определенности (ПО), определенный на интервале $[-1; 1]$, где -1 — полное отрицание, $+1$ — полное утверждение. В начале консультации часть триплетов не укомплектована, некоторые триплеты представляют собой цель консультации. Знания эксперта кодируются в виде:

если $\langle$ предпосылки $\rangle$, *то* действие ПО,

где предпосылка — конъюнкция триплетов, действие включает присвоение значений параметрам триплетов. Правило может быть применено:

1. исходя из известной предпосылки с целью присвоить значения параметрам триплета;
2. исходя из триплета с целью определить, какие триплеты предпосылки необходимо определить.

ПО вычисляется с использованием показателей определенности предпосылки правила и исходного показателя определенности трип-

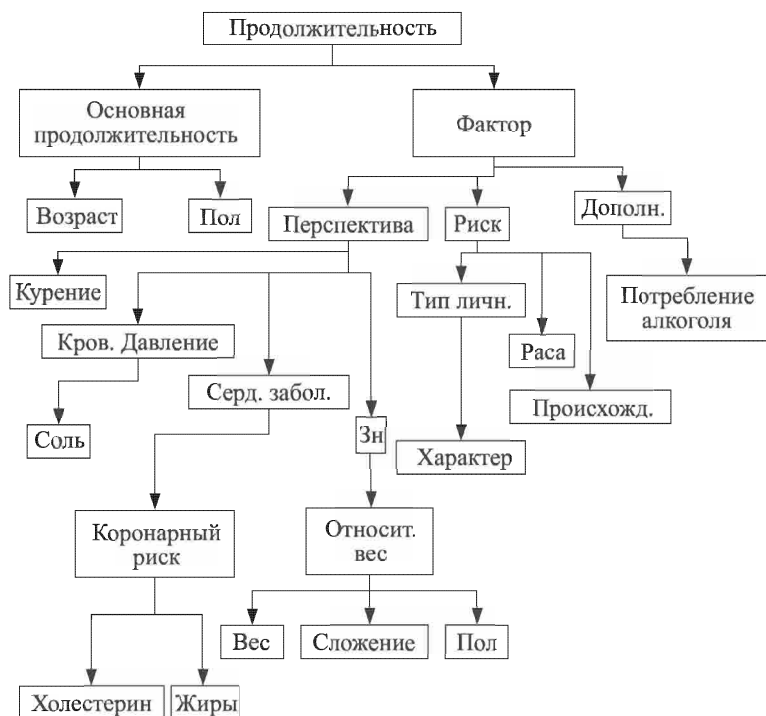


Рис. 82. Модель продолжительности жизни

лета, если он существовал. ПО предпосылки равен минимуму из показателей составляющих ее предложений, после чего это значение умножается на ПО самого правила (для части, ответственной за действие), отсюда имеет результирующий показатель (РП). Если прежде триплет не существовал, то РП — это ПО триплета, иначе если он уже существует с исходным показателем ИП, то:

$$\text{ПО} = \text{ИП} + \text{РП}(1 - \text{ИП}) \quad \text{РП, ИП} > 0$$

$$\text{ПО} = -(|\text{ИП}| + |\text{РП}|(1 - |\text{ИП}|)) \quad \text{РП, ИП} < 0$$

$$\text{ПО} = -(|\text{ИП}| + |\text{РП}|(1 - \min|\text{ИП}|)) \quad \text{РП, ИП} < 0$$

Сочетание 1 и -1 дает в результате 1. MYCIN — одна из широко используемых в мире систем.

2.6. Практические приложения критериальных языков. Методы, основанные на определении расстояний¹⁶. Прежде всего

¹⁶ Более подробно о подобных вещах следует посмотреть в библиотечном разделе «кластерный анализ».

[34] необходимо располагать достаточно полным описанием больного и клинических проявлений имеющегося у него заболевания. Это описание представляет собой ряд данных о больном (возраст, вес, температура, результаты анализа крови и др.). Этих данных должно быть много и они должны быть такими, чтобы их было достаточно для постановки диагноза. Выбор этих признаков производится на основе имеющейся врачебной практики. Число признаков при таком описании может быть довольно значительным: несколько десятков, сотен или даже тысяч. Эти данные можно перенумеровать и каждый из этих перенумерованных признаков представить в виде числа. Представление признаков в виде чисел естественно, когда речь идет о таких признаках, как возраст, вес, рост, кровяное давление. Однако числами можно представить (закодировать) и признаки не числовой природы, такие как пол, наличие или отсутствие головной боли, бессоницы и т. д. Так, например, признак, характеризующий сон больного, можно кодировать следующим образом. Если у больного длительная значительная бессонница, то этот признак кодируется цифрой 0, недостаточность продолжительности сна — цифрой 1, нормальный сон — цифрой 2, наличие некоторой сонливости — 3, значительная сонливость — 4. Таким образом, состояние сна больного характеризуется некоторым числом, которое может принимать значения 0, 1, 2, 3 и 4. После того как все признаки перенумерованы и в отношении каждого признака установлена система числовых обозначений, совокупность признаков больного характеризуется некоторым рядом чисел $X_1, X_2, \dots, X_{n-1}$. Совокупность чисел $X_1, \dots, X_n$, характеризующих состояние больного, можно геометрически интерпретировать как точку в многомерном пространстве признаков. По осям координат этого пространства откладываются численные значения признаков (рис. 83).

Пусть теперь перед нами стоит задача дифференциальной диагностики двух заболеваний (обозначим их A и B) и пусть мы располагаем значительным числом историй болезней больных с заболеваниями A и B , для которых у нас нет сомнения в правильности поставленного диагноза. Найдем численные характеристики $X_1, X_2, \dots, X_n$ признаков каждого из больных и будем каждую из последовательностей $X_i, X_s, \dots, X_n$ изображать точкой. В результате в пространстве признаков появится два множества точек (назовем их A и B), соответ-

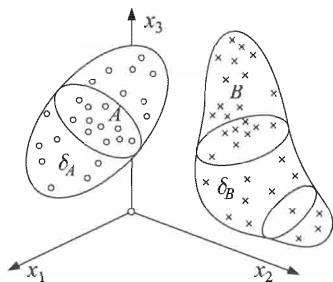


Рис. 83. Геометрическая интерпретация задачи диагностики

ствующим больным, имеющим заболевание A и соответственно B . В случае, когда пространство признаков трехмерно, эти множества можно представить геометрически наглядно, как показано на рис. 83. На этом рисунке точки, соответствующие больным с болезнью A , лежат отдельно от точек, соответствующих больным с болезнью B . Не представляет труда найти замкнутые поверхности s_a и s_b , внутри которых лежат точки первого и соответственно второго множеств A и B . Пусть теперь к нам пришел новый больной и мы хотим узнать, какой из двух болезней, A или B , он болен. Для этого найдем численные характеристики $X_1, X_2, \dots, X_{n-1}$ заболевания и соответствующую этим значениям точку в пространстве признаков. Если эта точка лежит внутри поверхности S_a , в гуще точек множества A , то мы вправе предположить, что у этого больного болезнь A , аналогично, если эта точка лежит внутри поверхности S_b , в гуще точек множества B , естественно предположить болезнь B . Если же эта точка лежит вне поверхностей S_a и S_b , следовательно, и вне множеств точек A и B , то скорее всего у этого больного нет ни заболевания A , ни заболевания B .

Эта группа методов получила за последние 20 лет значительное развитие. Изначально применялись методы, основанные на количественном выражении клинического опыта (балльная диагностика). Симптомы разделяют на основные и вспомогательные. Главный симптом — 2 балла, вспомогательный — 1 балл. Например, для диагноза «ревматизм» (по данным [34]) считается достаточным установить у больного 1 главный и 2 вспомогательных симптома, менее 4-х баллов ревматизм не ставят, но и не отвергают. При обработке 50-ти историй болезни с диагнозом «ревматизм» при использовании такого подхода получили 48% правильных диагнозов (остальные были не определены). При использовании схемы С. А. Холдина (1962) — 78%. Холдин С. А. предложил формулу: $z = x_1 + x_2 + \dots + x_r$, где z — сумма баллов; x_r — оценка величины (прогностического значения) признака в баллах, причем баллы, характеризующие тяжесть состояния организма и проведенное лечение, имели разные знаки. В зависимости от z ставились 4 возможных прогноза (метод решает задачу прогноза). Развитие этого метода привело к появлению метода линейных дискриминантных функций (МЛДФ), метода фазовых расстояний и др. МЛДФ основан на определении различий между значениями признака при двух заболеваниях (с учетом предыстории), определяется временное соотношение появления одинаковых значений признака при разных заболеваниях. Сравнение альтернатив происходит на основе так называемой суммы общего упорядоченного ряда: $z = x_1 + x_2 + \dots$, где x_r — признак аль-

тернативы z . Необходимо применять некоторые дискриминантные коэффициенты, различные для разных признаков (принцип действительно оказался практически результативным, и его модификация заимствована современными нейронными сетями).

В частности, в источнике [33] описана подобная система. В работе проанализирована диагностическая ценность 2000 признаков при 10 заболеваниях: острая, затяжная, хроническая пневмония, бронхоэктатическая болезнь, хронический бронхит, бронхиальная астма, туберкулез легких. Материал получен на 480 больных. На основании показателей информативности, полученных по методике А. Л. Шермана, отобраны 303 признака, которые послужили основой диагностической матрицы. Для автоматизированного распознавания применен метод Байеса, экзаменационную выборку наблюдений составили данные о 200 пациентах. Для автоматической классификации применен метод нелинейного отображения выборочных точек (L -мерного исходного признакового пространства) в пространство меньшей размерности $R^{L'}$ ($L' < L$). Особо важными считаются отображения в 2-х, 3-х мерные пространства (поскольку их можно визуализировать). Пусть исходное множество многомерных наблюдений $X_i\{x_i^1, \dots, x_i^L\}$, $i = 1, \dots, n$, спроектированы в соответствующие n наблюдений $Y_i = \{y_i^1, \dots, y_i^L\}$. При этом попарные расстояния $d_{ij}^* = \{\sum (x_i^{(k)} - x_j^{(k)})^2\}^{1/2}$, $k = 1, \dots, L$, между точками X_i и X_j в исходном L -мерном пространстве преобразуются в расстояния d_{ij} в $R^{L'}$: $d_{ij} = \{\sum (y_i^{(k)} - y_j^{(k)})^2\}^{1/2}$, $k = 1, \dots, L$. В качестве меры искажения конфигурации исходных точек вводится некоторая величина Δ .

Минимизация функции Δ достигается в результате итерационной процедуры: на первом шаге набор $y_1, y_2, \dots, y_n$ задается произвольно (или с помощью какого-либо метода). Пусть Δm — ошибка отображения после m -ой итерации. Следующая $m + 1$ итерация задается с учетом некоторого коэффициента α .

А α определяется эмпирически ($\alpha \approx 0,3; 0,4$). Состояние каждого больного характеризуется 85 симптомами (из анамнеза) объективного и специального обследований. В случае расширения матрицы за счет учета большего числа информативных признаков (385 признаков) результаты значительно лучше. Таким образом, уменьшение размерности вектора признаков ухудшает картину классификации. Отмечено, что клиника протекания бронхита разнообразна. Существует группа больных с хроническим бронхитом, клиника которых характерна, в то же время существуют подгруппы пациентов, клини-

Таблица 37

Результаты диагностирования «методами расстояний»

	Машина		Человек	
	правильно	ошибка	правильно	ошибка
Хронический бронхит	30%	50%	60%	30%
Бронхиальная астма	100%	0%	—	—
Саркоидоз внутригрудных лимфоузлов	50%	50%	—	—
Рак легких	100%	0%	—	—
Острая пневмония	100%	0%	—	—
	85 признаков		385 признаков	

ка которых практически не отличается от хронической пневмонии. Автор считает, что необходимо четче разграничить эти 2 заболевания.

2.7. Весовые коэффициенты. Простейший способ найти эти коэффициенты — определить отношение разности средних величин значения признака для 2-х разных заболеваний к общей дисперсии этого признака при обоих заболеваниях. Способам вычисления таких коэффициентов посвящены многочисленные работы. В теории многомерного шкалирования коэффициентам придается геометрический смысл — поворот (рис. 84). Даны два признака. Точки 1, 2, 3 называются изображениями (это 3 больных). Из рисунка видно, что иногда ни X_1 , ни X_2 не позволяют отличить больного заболеванием A_1 от A_2 , так как проекции точек и на X_1 и на X_2 совпадают. В этом

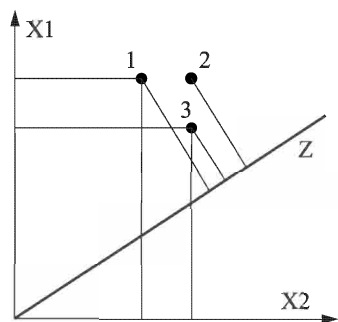


Рис. 84. Поворот осей — один из наиболее «эффективных» методов многомерного шкалирования

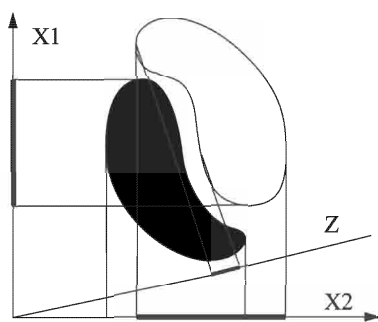


Рис. 85. Линейно неразделимые области

случае осуществляют поворот оси координат. На оси Z проекции уже не совпадают. В пространстве признаков в принципе всегда можно найти две области, в одной из которых сосредоточены все точки, изображающие состояние больного заболеванием A_1 , а в другой —

точки, изображающие состояние больного с заболеванием A_2 , если эти состояния различны. Отыскание оси Z — достаточно сложная задача, хотя само решающее правило относительно простое¹⁷. Данный метод имеет ограничения при сложном характере взаимосвязи между величиной признака и вероятностью заболеваний A_1 и A_2 , при неполном обследовании больного (рис. 85). Области в пространстве соответствуют состояниям больных с заболеванием A_1 и A_2 . Показаны проекции этих осей на оси признаков и дискриминантную ось Z . Представлен случай, когда проекции A_1 и A_2 в значительной мере перекрываются, поэтому ось Z расположена так, чтобы это перекрытие было минимально. Зоны, где проекции перекрывают друг друга, обозначены утолщенной линией.

2.8. Таксономия. В биологии существует раздел *таксономия*, занимающийся систематизацией классов в зоологии. Область применения таксономии также была расширена для классификации других предметов и явлений. Предположения, что переходные области классов являются размытыми и предметы в классе принадлежат к различным множествам, вызвали появление новой теории. Раньше классы полагались дихотомическими, с резко выраженными границами, теперь последние рассматриваются как нечеткие и размытые. В численной таксономии предложены два способа классификации, основанные на «полном подобии объектов, которые могут быть классифицированы», и на «использовании всех имеющихся в нашем распоряжении характеристик без введения каких-либо весовых коэффициентов». Распределение по группам и классам осуществляется исходя не из единственного признака, а из совокупности ряда признаков, в частности на основании сравнительного подобия. Сравнение начинается с определения «единичных» характеристик, которые нельзя разделить на независимые характеристики. Затем «операционные таксономические единицы» (ОТЕ — элементы, события, люди, предметы, которые должны быть классифицированы) объединяются в группы с учетом сходства и различия их свойств. Предмету соответствует точка в многомерном пространстве, значит, можно составить матрицы смежности. Расстояние вводится в произвольной метрике. В частности, в [39] указаны 3 способа таксономической оценки сходства:

- 1) коэффициенты ассоциации;
- 2) коэффициенты корреляции;
- 3) показатели расстояния.

¹⁷ На рис. 85 показана ситуация, когда нельзя найти линейное разделяющее правило.

Простейший коэффициент ассоциативности может быть представлен отношением количества совпадающих признаков, сравниваемых ОТЕ, к количеству несовпадающих. Таким образом, коэффициент стремится к 0 при полном совпадении и к 1 при полном несовпадении. В таксономии выработаны некоторые удобные виды таких коэффициентов.

П р и м е р. Даны k -я и j -я таксономические единицы. Пусть число признаков, имеющих у обоих ОТЕ, равно n_{11} , число отсутствующих признаков — n_{00} . Общее число признаков, имеющих у первой и отсутствующих у второй таксономической единицы, равно n_{10} , обратная комбинация — n_{01} . Числа представлены в табл. 38, где также указаны суммы по рядам и столбцам. Для этих сумм используются принятые в математической статистике обозначения, т. е. $n_{00} + n_{01} = n_0$. Принято также записывать число совпадающих и несовпадающих пар признаков, обозначаемых соответственно m и u , где $m = n_{00} + n_{11}$, $u = n_{01} + n_{10}$, $m + u = n$. Таким образом, существуют m признаков, по которым в обоих ОТЕ имеется совпадение, т. е. у обоих 0 или 1, и U признаков, по которым имеется расхождение. Простейший коэффициент ассоциации $S_1 = m/n$.

Т а б л и ц а 38

Сравнение признаков 2-х ОТЕ

	k -я ОТЕ		
j -я ОТЕ	0	1	Сумма
Признак 0	N_{00}	N_{01}	N_0
Признак 1	n	N_{11}	N_{11}
Сумма	N_{10}	N_{11}	N

Построение пространства признаков играет ключевую роль. В самом деле, нельзя приписывать сходство червям и млекопитающим из-за того, что у них отсутствуют крылья (или наоборот необходимо приписывать сходство по наличию крыльев летучим мышам и насекомым). Это может привести к появлению большой группы признаков, учет которых искусственно приведет к изменению степени сходства. Поэтому следует «исключить совпадения». При исключении по отсутствию признака:

$$S_2 = n_{11}/(n_{11} + u), \text{ обычно (но не всегда) } S \in [0, 1].$$

Приводится также так называемый коэффициент Юла:

$$S_4 = (n_{11}n_{00} - n_{01}n_{10})/(n_{11}n_{00} + n_{01}n_{10}).$$

Он принимает значение 1 при $n_{01} = 0$, -1 при $n_{11} = 0$ или $n_{00} = 0$, равенство 0 указывает на статистическую независимость признаков.

Таксономические коэффициенты корреляции не имеют каких-либо особенностей. Обычно применяются к непрерывным переменным или к дискретным переменным, характеризующимся некоторой упорядоченностью. Описаны различные частные виды коэффициентов. Лучше других известен коэффициент корреляции смешанных моментов, используемый для измерения сходства между двумя ОТЕ. Допустим, все элементы основной матрицы $n \times t$ — непрерывные переменные, тогда для j -й таксономической единицы имеем n измерений, обозначаемых x_{ij} , $i = 1, \dots, n$. Пусть среднее значение этих величин равно $X_{j\text{cp}}$. Аналогично для k -й таксономической единицы имеются измерения x_{ik} , $i = 1, \dots, n$, со средним $X_{k\text{cp}}$.

$$r_{ik} = \sum (x_{ij} - x_{j\text{cp}})(x_{ik} - x_{k\text{cp}}) / [\sum (x_{ij} - x_{j\text{cp}})^2 \sum (x_{ik} - x_{k\text{cp}})^2]^{1/2},$$

сумма во всех случаях $1, \dots, n$, $r = -1$ — полная отрицательная корреляция. Определить точный смысл этого критерия часто бывает трудно, однако ясно, что это определенный числовой показатель ассоциативности. Если допустить, что рассматриваемая случайная величина характеризуется двухмерным распределением, то можно применить статистический критерий значимости и установить, например, является ли значимым отличие данного коэффициента от 0 или различие 2-х коэффициентов друг от друга.

2.9. Показатели расстояния. При использовании показателей расстояния возникают определенные проблемы при нормировании, поскольку относительные веса признаков сильно зависят от типа используемой шкалы. Один из возможных способов решения — кодировать каждую случайную величину в интервале $[0, 1]$ и оперировать нормированными значениями.

Другая проблема состоит в выборе «решающего правила» или, что тоже самое, математического вида «разделяющей функции». На рис. 85 показана ситуация, когда задачу нельзя решить при помощи функции линейного вида. Необходима либо функция высшего порядка, либо ее кусочная аппроксимация. В самом деле, Вы не сможете провести прямую линию так, чтобы все точки незаштрихованной области и заштрихованной остались по разные стороны от нее. *Гибридные* методы сочетают в себе элементы различных подходов. В частности, к таким методикам относится методика определения прогноза течения заболевания и его исхода. Для определения прогноза учитываются независимые функциональные признаки, критические границы изменения этих признаков, веса. Для коррекции состояния больного вводится функция лечебных воздействий, зависящая от времени, время считается неоднородным (различают

латентный период, время реакции на воздействие, время накопления воздействий); вводится функция управления (от лечебных воздействий), и каждое воздействие характеризуется дозой. Отмечается, что медико-биологические признаки коррелируют между собой, и после лечебных воздействий необходимо их корректировать.

Последние 10 лет активно ведутся разработки по применению элементов теории функционального анализа в искусственном интеллекте. В частности, разрабатываются сетевые модели представления знаний. В публикации [43] рассмотрены некоторые результаты применения программной системы, основанной на нейронной сети, для диагностирования инфаркта миокарда. Использовались 20 параметров. Показатель правильных диагнозов составил 92% и 4% ложных. В работе указано, что основной проблемой при распознавании инфаркта миокарда врачами является большое количество ложных тревог (до 29%), альтернативные методы рассматриваются через следующие критерии: чувствительность и специфичность метода. Идеальный метод диагностики должен, во-первых, не пропускать ни одного действительно больного человека, а во-вторых, не пугать здоровых людей. Рассмотрены также применения сетей к диагностике рака и расшифровке ЭКГ.

Применяемый формальный метод в значительной мере зависит от имеющихся исходных данных. Исходные данные определяются конкретной предметной областью. Признаки объектов и явлений подразделяются на детерминированные, вероятностные, логические и структурные [36]. Исходные данные: графики, измерения, выполненные на различных шкалах, булевы результаты анализов и др. Источник [38] обращает внимание на то, что исходные данные измерены на различных шкалах. Если не учитывать этот факт при построении решающего правила, то результат принятия решения может оказаться бессмысленным. Задачи формирования пространства признаков и сбора исходных данных сами по себе являются очень сложными, едва ли не более сложными, чем построение решающего правила. Поэтому существуют работы, целиком посвященные некоторым аспектам этих задач. В частности, [40] указывает на то, что после того, как определены основные требования к пространству признаков, необходимо создать так называемые стандартизованные истории болезней (СИБ) (табл. 39). Такие СИБ будут различны для кардиологического, неврологического и других профилей. Такая СИБ должна быть интерактивной, поэтому существует задача построения сценария диалога. Создание диалоговых систем, ориентированных на конечного пользователя, — важная предпосылка для использования ВТ непосредственно специалистами. Суть подхода,

предлагаемого [40], заключается в том, чтобы объединить средства управления диалогом и средства управления данными на основе:

- разработки языка описания сценария диалога (включая язык манипулирования БД);
- унификация описания сценария диалога и ЯМД — независимость аппарата ведения данных от СУБД.

Т а б л и ц а 39

Пример диалога с медицинской ЭС

№	Вопросы	Ответы	Переход
1	Жалобы обожженного	1. Есть 2. Нет	На 2 На 15
2	Общая слабость	1. Есть 2. Нет	На 3
3	Жжение в области ожогов	1. Есть 2. Нет	На 4 На 0
...	...		
15	Обстоятельства и причины травмы	1. На воздухе 2. В закрытом помещении	На 16 На 27

3. Сводная таблица некоторых широко известных интеллектуальных систем

Дата, место разработки, коллектив разработчиков	Наименование	Назначение	Принципы функционирования	Разновидности	Особенности
Стэнфордский университет, 1965–83	DENDRAL	Распознавание химических структур. Пользователь дает системе некоторую информацию о веществе, а также данные спектрометрии (инфракрасной, ядерного магнитного резонанса и масс-спектрометрии), и та в свою очередь выдает диагноз в виде соответствующей химической структуры	Правила	GENOA, META-DENDRAL (1970–76)	Современная разновидность называется GENOA
Стэнфордский университет, 1972–80	MYCIN	Рекомендация в лечении заболеваний крови	Правила продукции (около 500), обратная цепочка с коэффициентами уверенности и эвристиками. Запоминает сеансы работы в БД. Процент правильных диагнозов достигает 70	EMYCIN, PUFF, NEOMYCIN	Язык LISP. Последовательно задает все более узкие уточняющие ситуации вопросы. Временами вопросы получают группы

Стэнфордский университет, 1974–77	TEIRESIAS	Программа для извлечения знаний для дальнейшего занесения в БЗ MYCIN			Одна из целого комплекса программ, разработанных для поддержки MYCIN. Позднее многие из них вошли в EMYCIN
	EMYCIN	Самая первая оболочка ЭС. Создана на основе MYCIN			
	PUFF	Анализ нарушений дыхания. Создана на основе EMYCIN			
1977–80	CENTAUR	Решает те же задачи, что и PUFF	Гибридная модель: правила продукций и фреймы		Создана, чтобы сравнить эффективность фреймового и продукционного подходов и оба с гибридным. Стоит из двух частей (потому так и называется)
Стэнфордский университет	NEOMYCIN	Обновленная MYCIN. Предназначена для обучения молодых докторов	Решена проблема «смешения понятий при большом числе правил», введена процедура контроля на «глупые вопросы», которые временно задавала MYCIN; порядком вопросов, задаваемых системой, стал оптимизироваться		Использует специально разработанный классификатор болезней, организованный в виде фреймов

1975–84	MOLGEN	Поддержка исследований в молекулярной биологии	Фреймовая модель	Более современная модификация — SPEX; самая последняя MOLGEN-II	Одна из наиболее сложных систем
1978–81	QUIST	Поддержка «почти естественного» языка при запросах к БД. Оптимизация запросов			
Корпорация DEC, 1985	XCON	Помогает пользователям выбрать конфигурацию компьютера	Несколько тысяч правил. Прямой вывод	RuleWorks	До сих пор используется компанией Compaq, поскольку число комплекующих, из которых может быть собрана машина, для клиента очень велико
Университет Райт (Огайо, США)	AIDS prevention	Сокращение риска заразиться СПИДом	Основана на модели прецедентов (БЗ на 75 штук). Прецедент состоит из истории больного и внешних обстоятельств. Специальная функция подыскивает прецедент, похожий на пациента, и советует, что ему делать		
Рэджан корпорейшн, США	WILLARD	Предказание погоды (бури в центральной части США)	Продукционная модель. Правила обкатаны на многолетних данных о погоде		Создана на основе оболочки RULEMASTER

Intelligent Applications Ltd.	AMETHYST	Диагностика неисправностей в машинах с вращающимися частями	Продукционная модель	Создана на основе оболочки Crystal. Интерфейс с датчиками вибрации
	EXSOPS	Предлагает вариант устройства фундамента на конкретной площадке с конкретным типом почв	Продукционная модель	Связь с Dbase IV. Умеет отвечать на вопросы «а что если?»
	POMME	Управление яблонным садом: болезни, насекомые, замораживание, сорняки, засуха	Продукционная модель	Создана на основе Crystal-подобной оболочки
Compaq	SMART	Помогает инженерам, работающим на «горячей линии», отвечать на звонки неопытных клиентов	Основана на модели прецедентов	Prolog. Состоит из двух подсистем, одна из которых моделирует развитие инфекции в саду
Португалия, Гидроэлектростанция в Iberduero-Labein	CONTROL OF PLANT (Alarm system)	Анализирует и интерпретирует информацию от автоматической системы оповещения (более 100 различных опасностей)	Продукционная модель + специальная мат. модель	Имеет специальную систему систематизации случаев (прецедентов)
				Система реального времени. Позволяет определить опасность, локализовать неисправность, отследить развитие ситуации, рекомендовать действия и собирать статистику

GTE Labs inc.	COMPASS	Определение неисправностей в телефонном оборудовании	Продукционная модель + прецеденты		Создана на основе оболочки КЕЕ. Интегрирована с оборудованием телефонной станции
	Prospector	Содействие поиску коммерчески оправданных месторождений полезных ископаемых			
Каун О.П., Силич В.А., Силич М.П., г. Томск, 2000	ESISIP	Продукционная система с нечеткой логикой. Применяется в основном для иллюстрации модели на занятиях ИИ			
Из источника [42]	EXPOB	Учебная система, довольно распространенная в ВУЗах России			

4. Примерные планы практических работ

4.1. «Правильная постановка целей».

Цель работы: выработать у обучаемого навыки правильного выделения целей системы, ее декомпозиции и поиска возможных (как можно больше) путей ее достижения.

Содержание работы: цель, дерево целей, декомпозиция целей; факты и правила; прямая и обратная цепочка рассуждений.

Постановка задачи примером. Одно из возможных определений интеллектуальной программы — это компьютерная программа, управляемая целями, а не данными. При разработке такой программы в первую очередь следует думать не об алгоритме, а о целях, которые данная программа должна достигать.

Цель системы можно понять неправильно, либо правильно, но неточно. Умение правильно и точно определять цели, достигаемые при применении интеллектуальной программы — одно из базовых умений специалиста по ИИ.

П р и м е р. Вам поставлена задача: «Создать компьютерную программу, которая оценит, умеет ли ребенок считать».

1. Студенты предлагают свои варианты алгоритма. Все эти варианты страдают неполнотой, поскольку сама цель, фактически, не сформулирована. Что значит «уметь считать»? Следует ли, проверять, например, знание дифференциального исчисления? А формул сокращенного умножения?

2. Определение цели. В ходе обсуждения вы приходите к выводу, что в действительности следует проверить, знает ли ребенок арифметику.

3. Студенты предлагают свои варианты тестовых заданий на знание арифметики. На этот раз решение кажется им очевидным и почти единственным. Обратите их внимание на вопрос об оценке знаний в случае, если ребенок часть тестов выполнит правильно, а часть ошибочно. Как оценить общий результат? Следует ли считать, что цель «знаю арифметику» не достигнута, если из 10 тестовых заданий 2 выполнены с ошибкой?

4. Уточнение цели. Возможны два уточнения цели: «определить, насколько хорошо ребенок знает арифметику» и «определить, знает ли ребенок арифметику безошибочно».

5. Декомпозиция уточненной цели, построение дерева целей системы с указанием степени достижения цели. Возможны несколько правильных вариантов декомпозиции, различающихся количеством и порядком следования подцелей, т. е. экономичностью модели.

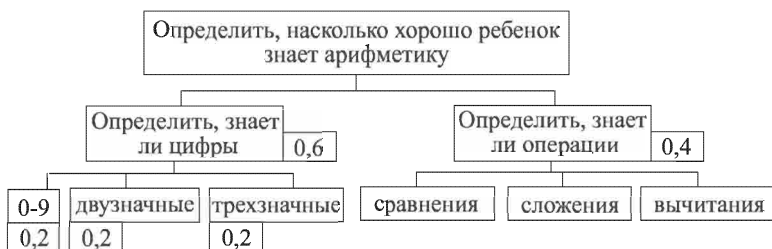


Рис. 86. Фрагмент возможного дерева целей

6. Студентам предлагается самостоятельно решить ряд аналогичных задач.

7. Студентам предлагается написать программу на одном из языков программирования, реализующую решенную задачу.

Задания.

1. Как добраться на работу вовремя.
2. Как сделать салат.
3. Как узнать, хочет ли кошка есть.
4. Тест: знаете ли вы английскую грамматику?
5. Загадка: «угадай животное» (птица, зверь или рыба).
6. Как получить диплом специалиста.

Обсуждение работы.

При построении дерева целей можно использовать, по крайней мере, две различные стратегии, основанные на рассуждениях вида:

- 1) что нужно сделать, чтобы получить заданный результат;
- 2) что получится, если я поступлю «вот так».

Первая хороша тем, что позволяет относительно быстро наметить план действий, приводящий к достижению поставленной цели. Например, «для того, чтобы достать высоко висящий предмет, следует подпрыгнуть и схватить, а при неудаче — подставить ящик, подпрыгнуть и схватить». При таком подходе мы как бы «обслуживаем заданную цель», поясняем, какими путями она точно могла бы быть достигнута, идем от цели к фактам. Варианты, которые мы генерируем, обладают различной экономичностью. Некоторые из них приводят к цели за малое число «сложных» шагов, другие требуют много «простых» шагов. Например, получить диплом можно «экстерном» или обучаясь на протяжении 5 лет.

Преимущество второй стратегии в том, что, проанализировав каждое из возможных действий, можно выработать оптимальную последовательность действий (например, самую короткую). Пример: «если я подпрыгну, то, возможно, схвачу высоко висящий предмет», «если я встану на ящик, то точно смогу достать выше, подпрыгнув».

Перебирая такие последовательности, мы рано или поздно найдем ту, которая приводит к поставленной цели. Здесь мы идем от фактов к цели.

Первую стратегию называют нисходящей или, что тоже самое, «обратной цепочкой рассуждений», вторую — восходящей или прямой.

Прямая цепочка не обязательно приводит к достижению поставленной цели. В общем случае два дерева целей, про которые известно, что они оба приводят к достижению поставленной цели, но получены на основе разных стратегий, различны.

Дерево целей может быть получено методом «встречной волны», когда анализируются противоречия между результатами обратной и прямой волны. Такое дерево обеспечивает оптимальный путь достижения цели.

Итоги работы. Разработка программы состояла из:

- 1) определения целей;
- 2) определения фактов, имеющих отношение к этим целям (важно, чтобы факты исчерпывающе соответствовали своей цели, например при проверке «знает ли ребенок правила сложения» следует проверить знание сложения с «0», с «1», без переноса разряда, с переносом разряда);
- 3) получение данных, соответствующих фактам, характеристик для заданной ситуации или объекта;
- 4) оценки данных, используя правила и механизм вывода.

4.2. «Элементы формальной логики для управления техническими устройствами».

Цель практики: выработать у обучаемых навыки записи задач в виде уравнений математической логики, продемонстрировать достоинства и недостатки такого метода управления техническими устройствами.

Содержание практики: анализ задачи (выделение сущностей и отношений), преобразования формул к удобному виду (ДНФ и др.). Оценка вычислительных затрат на всю систему уравнений и на каждое уравнение в частности и установление оптимального в некотором смысле порядка следования переменных в уравнении и уравнений в системе.

Задания. Написать программу на заданную тему.

1. Интеллектуальный лифт. При примерно одновременном нажатии кнопок вызова с разных этажей направляется сначала к ближайшему этажу.
2. Диагностика заболеваний у собаки (на основе приведенного фрагмента текста).

3. Интеллектуальный выключатель света. Посетитель гостиницы идет по этажам. Свет включается перед ним и выключается после того, как посетитель прошел этаж.
4. Классификации (животные, книги, произвольные и т. п.)

Приблизительный внешний вид результатов работы:

$$Y_1 = X_1 + X_2 \times X_3$$

$$Y_2 = X_4 + X_1 \times X_3$$

...,

где Y — имена заболеваний, а X — названия симптомов.

Некоторые типичные трудности. 1. Построение формул, образующих циклы в определении. 2. Неспособность выделять сущности, заданные неявно. Так, например, в тексте о болезнях собак такой сущностью является слово «воспаление». В тексте его нет, но оно подразумевается. Если эту сущность не выразить явно, то формальная модель не получится связной, нельзя будет от симптомов последовательно дойти до имени болезни.

Материалы к заданиям.

Дерматомикозы: стригущий лишай, парша. Вызываются патогенными низшими грибами. Характеризуются очаговыми выпадениями волос и воспалительными процессами в коже. Болеют собаки всех возрастов. Возбудители, попадая на кожу, при соответствующих условиях (микротрещины, снижение сопротивляемости) разрастаются в эпидермисе, дерме, волосяных фолликулах, шерсти. Организм собаки реагирует воспалительным процессом. Нарушение питания и воспаление фолликулов приводит к выпадению шерсти.

Признаки: появление на коже головы, шеи, реже других мест, ограниченных пятен округлой формы. В месте поражения обламывается и выпадает шерсть, образуются бугорки и корочки. Кожа покрасневшая, припухшая, под корочками иногда сочится гной. Зуд не выражен или выражен слабо, собака обычно не расчесывает пораженные места.

Парша проявляется образованием на коже толстых серовато-желтых корок. Возможно самовыздоровление животных.

Диагностика. Стригущий лишай отличается от других заболеваний кожи сочетанием клинических признаков: ограниченное выпадение волос, корочки и отсутствие зуда. При облучении невидимым ультрафиолетом (лампой Вуда) пораженные грибом *Microsporum* участки люминесцируют желто-зеленым светом. Парша у собак встречается редко. Окончательный диагноз ставят с помощью лабораторных исследований.

Демодекоз. Заболевание, вызываемое паразитирующими в коже микроскопическими клещами *Demodex canis*. Клещ проникает в кожу собаки при непосредственном контакте, а также при контакте с предметами ухода за больной собакой. Поселяется в волосяных луковицах и сальных железах. Механически, а также своими выделениями раздражает кожу, вызывая воспалительный процесс.

Признаки. Чаще болеют молодые собаки. Первичные очаги демодекоза обнаруживают, как правило, на голове: нижней челюсти, спинке носа, веках, губах. Часто поражается кожа в области локтевого и скакательного суставов. В месте поражения кожа утолщается, краснеет, становится складчатой, волосы выпадают полностью или частично. Может образовываться толстый слой ороговевшего эпидермиса, особенно на суставных сгибах. У основания волос образуются мелкие пузырьки (пустулы), заполненные гноем. Они лучше видны на подбородке. Зуд практически отсутствует.

Эктопаразитозы: вши, блохи, власоеды. Паразитирующие на коже нелетающие насекомые. Вши — черно-коричневые, бескрылые, уплощенные сверху вниз насекомые величиной 1,6–2,4 мм. Жизненный цикл полностью протекает на собаке. Питаются кровью. Блохи — черно-коричневые, уплощенные с боков, прыгающие насекомые величиной 2–4 мм. Способны долгое время выживать вне собаки на почве, траве, подстилке, в щелях пола, перебираясь, при возможности, на нового хозяина. Питаются кровью. Нападают на людей.

Власоеды — бледно-желтые уплощенные насекомые величиной 1–2 мм. Питаются чешуйками эпидермиса, кровью и лимфой из ссадин и ранок. Передвигаясь и питаясь, насекомые-эктопаразиты сильно беспокоят собаку. Особенно болезненны укусы блох. Выделения паразитов обладают аллергенными свойствами. Собаки расчесывают зудящие места, травмируют кожу. Блохи и вши скапливаются в малодоступных для собаки местах — на голове, шее, вдоль позвоночника, на крестце.

При большом количестве эктопаразитов собаки худеют, постоянно чешутся, стонут, взвизгивают. У них может развиваться анемия. Паразиты обнаруживаются при осмотре. Собаки делают характерные движения («клацают зубами») при ловле блох. Самый постоянный признак — расчесы вследствие кожного зуда.

4.3. «Программирование на языке Prolog».

Цель практики. Выработать у обучаемого навыки графического описания задачи в виде дерева. Отработать практические вопросы поиска на графе. Получить навыки написания рекурсивных про-

грамм. Изучить основные синтаксические конструкции языка. Получить практическое представление о работе конечных автоматов.

Содержание практики. Анализ задачи, построение дерева и выбор оптимальной стратегии перебора. Прямая и обратная цепочки рассуждений.

Задания: написать программу.

1. Найти выход из квадратного лабиринта (начальная точка задана).
2. Проверить, является ли некоторое выражение допустимым с точки зрения некоторой грамматики.
3. Классификация (дихотомия).
4. Электронный консультант «Какую модель ЭВМ купить?»
5. Не зная о принципах организации Eliza, попытаться написать аналог Eliza¹⁸.

Далее задачи, не очевидные человеку, ввод–вывод на Prolog.

1. Есть ли дорога между городами X и Y ? Дана карта–схема автодорог.
2. Какова длина пути из города X в город Y ?
3. Диагностика заболеваний у собак (описание заболеваний приведено выше).
4. Знаете ли Вы английскую грамматику?
5. Написать базу данных, описывающую животных черного цвета, а затем составить запрос на вывод всех черных животных, но не медведя.
6. Бутси — коричневая кошка. Корни — черная кошка. Мэктевити — рыжая кошка. Флэш, Ровер, Спот — собаки. Ровер — рыжая. Спот — белая. Все животные, которыми владеют Том и Кейт, имеют родословные. Том владеет всеми черными и коричневыми животными. Кейт владеет всеми собаками не белого цвета, которые не являются собственностью Тома. Алан владеет Мактевити, если Кейт не владеет Бутси и если Спот не имеет родословной. Флэш — пятнистая собака. Определить, какие животные не имеют хозяев. Ответ — Спот.
7. Разбить введенное слово на слоги.

«Синтаксис и Рекурсия на Prolog».

1. Вывести на экран целые числа в порядке убывания от заданного.
2. Вычислить факториал заданного числа.

¹⁸ Eliza — программа, имитирующая собеседника в диалоге, своеобразный «говорун».

3. Вывести на экран целые числа в порядке убывания от заданного до заданного.
4. Возвести число в степень на Prolog.
5. Вычислить арифметическую прогрессию.
6. Задача повышенной трудности «Волк, Коза, Капуста, Фермер».

4.4. «Построение СС на основе фрагмента текста».

Цель практики: активизировать ассоциативное мышление обучающихся, научить строить простейшие семантические сети, показать проблемы, возникающие при определении типа отношений в сети.

Содержание практики: анализ задачи, построение сети, применение методов вывода на сетях.

Ход работы. Начнем с разминки: возьмите лист бумаги и нарисуйте в центре слово «Замок». Подумайте об ассоциациях, скажем, «король», «стены» и «враги». Соедините каждое слово со словом «Замок» линией и надпишите над линиями глаголы, выражающие отношения: «враги за стеной», «король в замке» и т. п.

Задания. Построить граф с указанием типов отношений и вершин:

- 1) институт;
- 2) семья;
- 3) государство;
- 4) автомобиль;
- 5) пароход;
- 6) покупка в магазине;
- 7) самолет;
- 8) сравнение двух текстовых строк;
- 9) разобрать введенное предложение по членам предложения.

Возможны 2 формы без применения ЭВМ. Форма сдачи — отчет. Или написание программы на Prolog или Pascal.

4.5. «Создание фрейма — визуального образа».

Цель практики. Показать глубинный смысл понятия «фрейм» и его схожесть с понятием «объект». Показать область применения фреймов для моделирования событий реального мира. Получить опыт работы с механизмом обмена сообщениями (объектной модели).

Содержание практики: анализ задачи, построение фрейма, реализация на алгоритмическом языке.

Задания. Написать программу «Фрейм — визуальный образ».

1. Игра: кошки-мышки. На экране случайным образом порождаются кошки и мыши. Если кошка видит мышь, она гонится за ней.

Мышь убегает. Если кошка поймала мышь, то мышь умирает. Если одна кошка бросится сразу на 3 мыши, то загрызут саму кошку.

2. Игра: кошка — мышка — собака. Правила аналогичны.
3. Заставка экрана: мячики, отскакивающие друг от друга. По экрану летают мячики, при ударе друг о друга или стенки экрана отскакивают.
4. Стадион (фреймы «люди», фрейм «страх»): смоделировать развитие беспорядков.
5. Автомобильный перекресток.

4.6. «Создание не визуального фрейма».

Цель практики: показать глубинный смысл понятия «фрейм», показать обучаемым, что объект — это совсем не обязательно визуальный образ.

Задания. Написать программу «Фрейм как не визуальный образ»:

- 1) матрица результатов тестирования;
- 2) матрица уравнений (с методами решить, ввести данные и т. п.)

4.7. «Работа с оболочкой ЭС».

Задания. Написать программу на основе формулы Байеса.

1. Медицинская диагностика желтухи у человека (по материалам соответствующей лекции).
2. Ветеринарная диагностика на основе формулы Байеса, на основе отрывка текста. Вероятностные оценки симптомов, при необходимости оценить самостоятельно.

На основе критериальной свертки.

3. Медицинская диагностика желтухи у человека (по материалам соответствующей лекции).
4. Ветеринарная диагностика.

4.8. «Разработка простейшей ЭС на инструментальном языке».

Цель работы. Закрепить весь комплекс полученных знаний, умений и навыков. Получить навык самостоятельной постановки задач.

Требования к результатам: решение достаточно размерной задачи, например, с участием до 10 разных сущностей.

Задания.

1. Прогнозирование курса доллара на основе данных с Интернет сайтов (например, разных валютных бирж).
2. Медицинская диагностика до 10 заболеваний.

3. Оценка успеваемости группы студентов и ранжирование студентов в пространстве Осгуда.

5. Некоторые тесты

Приводимые тесты могут оказаться полезными для оперативной самопроверки. Большое количество подобных тестов в электронном виде можно найти на сайте оболочки информационно-образовательной среды «Chopin» <http://altnet.ru/~mcsmall/index.htm> или при помощи любой поисковой машины по запросу «FLTS», «light», оболочка «Chopin» и т. п.

5.1. Некоторые основные термины и области применения ИИ

№	Вопрос	Варианты ответа
1.	Укажите наиболее точное, по Вашему мнению, определение интеллектуальной системы	1. Система, способная целенаправленно менять параметры и сам способ своего поведения 2. Система, способная менять параметры своего поведения 3. Система, способная обучаться
2.	По Вашему мнению, знания всегда задаются через	1. интенционал понятия 2. любым образом 3. экстенционал понятия
3.	По Вашему мнению, термины «декларативная информация» и «присоединенная процедура»	1. зависят от контекста 2. совершенно разные вещи 3. одно и то же
4.	По Вашему мнению, какова мощность пространства состояний системы «пешеходный светофор» («стойте–идите»)?	1. четыре 2. бесконечна 3. два
5.	По Вашему мнению, система «выстрел из пушки» является	1. целенаправленной 2. зависит от контекста 3. целенаправленной
6.	По Вашему мнению, термины «модель представления знаний» и «язык представления знаний»	1. означают пересекающиеся, но разные вещи 2. означают совершенно разные вещи 3. означают одно и то же
7.	По Вашему мнению, медицина является	1. подобные определения к медицине не применяются 2. хорошо структурированной областью знаний 3. слабо структурированной областью знаний

8.	По Вашему мнению, что такое «биомашина»?	1. машина, в точности соответствующая структуре живого существа 2. живое существо, управляемое некоторой машиной 3. машина, имеющая своей частью живое существо
9.	По Вашему мнению, работы в области создания человекообразных роботов	1. выполняются скорее в рекламных целях 2. имеют большое практическое значение 3. широко велись в основном в 60-х гг.
10.	По Вашему мнению, устойчивость (надежность) некоторой модели представления знаний (применительно к некоторой предметной области) следует рассматривать	1. как безусловное преимущество 2. скорее как недостаток 3. говорить о надежности МПЗ не имеет смысла
11.	По Вашему мнению, является ли характеристика «экономичности» важной для моделей ИИ?	1. в большинстве реальных ситуаций ею можно пренебречь 2. нет, не является 3. да, является

5.2 Основные понятия логической модели представления знаний

№	Вопрос	Варианты ответа
1.	По Вашему мнению, имена «стол» и «стул» следует отнести к	1. пустым именам 2. именам общим 3. именам единичным
2.	По Вашему мнению, описание «комната с маленьким окном и очень большой температурой внутри»	1. задает экстенционал имени «баня» 2. задает объем имени «баня» 3. задает содержание имени «баня»
3.	По Вашему мнению, логический класс «ученики 10-го класса» является	1. сложным 2. простым
4.	По Вашему мнению, если мы выберем из некоторого текста ключевые слова и приведем их к единой морфологической форме и написанию, то получим простейший	1. тезаурус 2. нормативный словарь 3. дескрипторный словарь
5.	По Вашему мнению, между понятиями «исчисление высказываний» и «исчисление предикатов»	1. имеются четко выражаемые в булевой алгебре отличия 2. никакой разницы нет 3. имеется незначительная разница
6.	По Вашему мнению, сколько всего существует правильных модусов силлогизма?	1. 256 2. 24 3. 19

7.	По Вашему мнению, конъюнкция — это связка вида	1. «И» 2. «ИЛИ» 3. исключающее «ИЛИ»
8.	По Вашему мнению, рассматриваются ли в современных системах ИИ противоречивые исчисления?	1. В широко известных нет 2. Чаще всего исчисление получается противоречивым из-за ошибки программиста 3. Да, конечно
9.	По Вашему мнению, применим ли принцип резолюций к исчислению второго порядка?	1. Не применим ни к ИП первого, ни второго порядков 2. Да, конечно 3. Нет, только первого порядка
10.	По Вашему мнению, имеется ли разница между понятиями «вероятности события» и «степени уверенности (по Л. Заде)»?	1. Это идентичные понятия 2. Это совершенно разные понятия 3. Имеется некоторое пересечение, но понятия разные

5.3. Основы языка Prolog

№	Вопрос	Варианты ответа
1.	По Вашему мнению, имеет ли список заданную длину, равную 4? длина([10,[11,11],12], 4)	1. Да 2. Нет
2.	Установите, сопоставляется ли данная пара: [ЖАК, ЖИЛЬ] и [А,В С]	1. Нет 2. Да
3.	Установите, сопоставляется ли данная пара: [дом, осел, лошадь] и [Н,Т]	1. Нет 2. Да
4.	По Вашему мнению, какие из приведенных цепочек являются атомами?	1. человек(георгий) 2. весьма_длинная_последовательность_знаков 3. б 4. В 5. 'человек(георгий)'
5.	Определите размерность структуры квартет (скрипка первая, скрипка вторая, виолончель, бас)	1. 16 2. 4 3. 6 4. данное выражение не является структурой
6.	По Вашему мнению, раздел программы на Prolog, предназначенный для задания внутренних баз данных объявляется как:	1. domains 2. clauses 3. goals; 4. predicates 5. databases

7.	По Вашему мнению, операция отсечения в Prolog обозначается	1. символом «%» 2. оператором «cut» 3. оператором «!» 4. такой операции в Prolog нет
8.	Укажите допустимые в Prolog реляционные операторы	1. «>» 2. «<» 3. «=» 4. «=>» 5. «<=» 6. «<>» 7. «><» 8. «<=>»
9.	По Вашему мнению, какие из фрагментов программы на Prolog позволяют вычислить выражение и вывести результат?	Фрагмент А) GOAL a=4+2, write(A) Фрагмент В) GOAL A=4+2 Фрагмент С) GOAL _:=4+2 Фрагмент D) Ни один из фрагментов
10.	Эквивалентны ли с точки зрения Prolog приведенные фрагменты? Фрагмент А) road(«Cansas»,«Tampa») road(«Tampa»,«Cansas») road(«Tampa»,«Houston») Фрагмент В) road(«Cansas»,«Tampa») road(«Tampa»,«Houston»)	1. фрагменты эквивалентны 2. это зависит от версии Prolog 3. фрагменты не эквивалентны 4. зависит от стратегии вывода

5.4. Основы объектной модели

№	Вопрос	Варианты ответа
1.	По Вашему мнению, инкапсуляция — это	1. механизм обеспечения наследования в объектах 2. совершенно бессмысленное слово 3. способ объединения кода и данных в объектах
2.	По Вашему мнению, может ли один и тот же терминал входить в 2 разных фрейма одной системы ?	1. зависит от контекста 2. не может 3. может
3.	По Вашему мнению, для обмена данными в ООП используется	1. глобальная переменная 2. локальная переменная 3. механизм «сообщений»

4.	По Вашему мнению, представление знаний фреймами эффективно при	1. анализе пространственных сцен 2. автоматическом переводе 3. проектировании микросхем 4. распознавании текста
5.	По Вашему мнению, к основным свойствам объектов относится	1. полиморфизм 2. наследование 3. инкапсуляция
6.	По Вашему мнению, нижние уровни фрейма-экземпляра	1. называются «маркерами» 2. заполнены характерными примерами или данными 3. пусты
7.	По Вашему мнению, объекты-экземпляры, которые во время выполнения программы могут принимать различные формы представления от объекта своего типа до любого из потомков, называют	1. полиморфными 2. виртуальными 3. динамическими
8.	По Вашему мнению, представление знаний фреймами значительно более эффективно, чем при помощи	1. эффективность зависит от решаемой задачи 2. семантических сетей 3. правил продукций 4. нечеткой логики
9.	По Вашему мнению, фрейм может быть описан при помощи правил продукций?	1. зависит от контекста 2. нет 3. да
10.	По Вашему мнению, терминалы фрейма-образца заполнены	1. совершенно пусты 2. так называемыми «заданиями отсутствия» 3. терминальными фактами 4. переменными

Ответы к тесту 5.1.

№	Правильный ответ
1.	1
2.	1
3.	2
4.	3
5.	1
6.	3
7.	3
8.	3
9.	2
10.	2
11.	3

Ответы к тесту 5.2.

№	Правильный ответ
1.	2
2.	3
3.	2
4.	2
5.	1
6.	3
7.	1
8.	2
9.	2
10.	2

Ответы к тесту 5.3.

№	Правильный ответ
1.	2
2.	2
3.	1
4.	2, 3, 5
5.	2
6.	2
7.	3
8.	1, 2, 3, 4, 5, 6, 7
9.	D
10.	3

Ответы к тесту 5.4.

№	Правильный ответ
1.	3
2.	3
3.	3
4.	1, 2, 3, 4
5.	1, 2, 3
6.	2
7.	1
8.	1
9.	3
10.	2

СПИСОК ЛИТЕРАТУРЫ

Классические работы, рекомендуемые к ознакомлению

1. *Quillian M. R.* Semantic memory // In Semantic Information Processing. — M.I.T. Press, Cambridge, Mass., 1968. — P. 227–268.
2. *Osgood C., Suci C. J., Tannenbaum P. H.* The Measurement of Meaning. — Urbana: University of Illinois Press, Chicago, 1957. — 300 p.
3. *Shannon C.* Communication theory of secrecy system // Bell System Technical Journal. — 1949. — Vol. 28 (4). — P. 656–715.
4. *Shannon C.* A Mathematical Theory of Communication // Bell System Technical Journal. — 1948.
5. *Аристотель.* Собрание сочинений.
6. *Винер Н.* Кибернетика или управление и связь в животном и машине. — М.: Наука, 1981.
7. *Дэйвисон М.* Многомерное шкалирование: методы наглядного представления данных. — М.: Финансы и статистика, 1988.
8. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976.
9. Искусственный интеллект. Под ред. Д. А. Поспелова. — М.: Радио и связь, 1990.
10. *Кант И.* О воде. О Земле. Об огне. Критика чистого разума.
11. *Кини Р. Л., Райфа Х.* Принятие решений при многих критериях: предпочтения и замещения. Под ред. Шахова И. Ф. — М.: Радио и связь, 1981.
12. *Ковальски Р.* Логика в решении проблем. — М.: Наука, 1990.
13. *Кохонен Т.* Ассоциативная память. — М.: Мир, 1980.
14. *Мински М.* Фреймы для представления знаний. — М.: Энергия, 1979.
15. *Мински М., Пейперт С.* Перцептроны. — М.: Мир, 1971.
16. *Нейман Дж.* Теория самовоспроизводящихся автоматов. — М.: Мир, 1971.
17. *Нильсон Н.* Принципы искусственного интеллекта. — М.: Радио и связь, 1985.
18. *Поспелов Д. А.* Большие системы (ситуационное моделирование). — М.: Знание, 1975.
19. *Розенблатт Ф.* Принципы нейродинамики (перцептрон и теория механизмов мозга). — М.: Мир, 1965.
20. *Шенк Р.* Обработка концептуальной информации. — М.: Энергия, 1980.

Прикладные работы, рекомендуемые к ознакомлению

21. *Братко И.* Программирование на языке Prolog для искусственного интеллекта. — М.: Мир, 1990.
22. *Элти Дж., Кумбе М.* Экспертные системы: концепции и примеры. — М.: Финансы и статистика, 1987. — 192 с.
23. *Марселлус Д.* Программирование экспертных систем на turbo Prolog. — М.: Финансы и статистика, 1994.
24. *Уотерман Д.* Руководство по экспертным системам. — М.: Мир, 1989.

Современные работы,
условно-обязательные для изучения

25. Нечеткие множества в моделях управления и ИИ/АН СССР. Под ред. Д. А. Поспелова. — М.: Наука, 1986.
26. *Тейз А., Грибомон П., Луи Ж.* Логический подход к искусственному интеллекту: От классической логики к логическому программированию. — М.: Мир, 1990. — 432 с.
27. *Девятков В. В.* Системы искусственного интеллекта. — М.: Изд-во МГТУ им. Баумана, 2001.
28. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. — СПб.: Питер, 2000. — 384 с.
29. *Горбань А. Н.* Обучение нейронных сетей. — М.: СП Параграф, 1990.
30. *Горбань А. Н., Россиев Д. А.* Нейронные сети на персональном компьютере. — Новосибирск: Наука, 1996.
31. *Ивин А. А.* Элементарная логика. — М.: Дидакт, 1994.
32. *Трахтенгерц Э. А.* Компьютерная поддержка принятия решений. — М.: СИНТЕГ, 1998.

Литература, полезная для дополнительного
ознакомления

33. *Молотков В. Н.* Диагностика и автоматическая классификация неспецифических заболеваний легких с помощью ЭВМ // Медицинская и физиологическая кибернетика. — Киев: Изд. Института кибернетики им. Глушкова В. М. — 1981.
34. *Гублер Е. В.* Вычислительные методы распознавания патологических процессов. — Л.: Медицина, 1970.
35. *Ван Гиг Дэс.* Прикладная общая теория систем. — М.: Мир, 1981.
36. *Горелик А. Л.* Современное состояние проблемы распознавания. — М.: Радио и связь, 1985.
37. *Загоруйко Н. Г.* Методы распознавания и их применение. — М.: Советское радио, 1972.
38. *Перебудов Ф. И., Тарасенко Ф. П.* Введение в системный анализ. — М.: Высшая школа, 1989.
39. *Бейли Н.* Математика в биологии и медицине. — М.: _Мир_, 1970.
40. *Пачин С. Ф.* Средства интерактивного сбора и анализа медицинской информации // Биологическая и медицинская кибернетика. — Киев: Изд. Института кибернетики им. Глушкова В. М. — 1982.
41. *Бабко С. А.* Определение информативных показателей методом факторного анализа // Биологическая и медицинская кибернетика. — Киев: Изд. Института кибернетики им. Глушкова В. М. — 1982.
42. *Сойер Б., Фостер Д. Л.* Программирование экспертных систем на Паскале. — М.: Финансы и статистика, 1990.
43. *Ежов А., Чечеткин В.* Нейронные сети в медицине // Открытые системы. — 1997. — № 4.

Предметный указатель

Lisp, 160

Prolog, 157

Аристотель, 42

Байес Т., 128

Бонч-Бруевич М. А., 66

Буль Дж., 61

Винер Н., 13

Голлерит Г., 13

Заде Л., 67

Кант И., 44

Ковальски Р., 13

Коллинз А., 97

Куиллиан Р., 97

Локк Дж., 44

Мински М., 84

Нейманн Дж., 13

Паскаль Б., 13

Понтрягин Л.С., 13

Поспелов Д.А., 32

Симмонс Р., 98

Шапиро С., 98

Шелли П., 12

Шенк Р., 98

Шеннон К., 13, 76

алгоритм

— редукции, 65

бэктрекинг, 8

вероятность события, 69

вершина

— начальная, 76

— порождение, 75

— разрешимая, 80

— терминальная, 75

— целевая, 75

вывод

— на знаниях, 19

— обратный, 74

— прямой, 74

высказывание, 53

— простое, 54

— атомарное, 63

— подчиненное, 55

— противоречащее, 55

геометрия вычислительная, 131

граф

— И-ИЛИ, 80

— решающий, 80

данные, 18

деление, 48

дескриптор, 53

джорджтаунский эксперимент, 17

диаграммы Эйлера, 56

доказательство, 59

задача

— абстрактная, 31

— выбора, 30, 105

— выбора критериальная, 106

— генерации альтернатив, 30

— классификации, 29

— лабиринтного поиска, 76

— логики, 42

— медицинской диагностики, 169

— параметры решения, 31

— понимания речи, 90

— правила формулировки, 30

— путь решения, 75

— редукция, 80

— слушания речи, 90

— составления плана, 30

— традиционная, 29

— уточнение, 31

знания, 18

— продукционная модель, 72

— база, 73

— извлечение, 22

- логическая модель, 40
- модель представления, 39
- особенность, 19
- поверхностные, 143
- свойства, 19
- язык представления, 21
- модель представления, 21

имя, 47

- объем, 48
- однозначные и многозначные, 48
- равнозначные, 48
- содержание, 47
- точное и неточное, 48

индексирование

- координатное, 52
- инженер по знаниям, 22
- инженерия знаний, 20

инкапсуляция, 92**информация**

- декларативная, 18
- процедурная, 18

исчисление

- предикатов, 61
- предикатов второго порядка, 63
- предикатов первого порядка, 63
- противоречивое, 62

квантор, 63**класс, 48, 49**

- абстрактный, 84
- простой, 49
- классификация, 49
- алфавитно-предметная, 52
- емкость и глубина, 50
- иерархическая, 50
- компьютер
- параллельный, 130
- конституента, 63
- концепт, 98
- коэффициент
- (степень) уверенности, 29
- коэффициент уверенности, 73

логика

- пороговая, 131
- силлогизм, 56
- элементарная, 67

машины

- вывода, 73
- множество
- Парето, 122
- альтернатив, 106
- возможных путей, 92
- дизъюнктов, 65
- категорий, 47
- явно доминирующих альтернатив, 111
- явно плохих альтернатив, 122
- модель
- адекватность, 32
- междисциплинарный перенос, 28
- надежность, 32
- обучаемого, 37
- представления знаний, 32
- формулы, 62
- мышление
- параллельное, 84

наследование, 92

- в семантической сети, 101
- нечеткость, 34
- неясность, 34

область

- исследования, 23
- предметная, 21
- применения методов ИИ, 34
- слабо структурированная, 34
- хорошо структурированная, 34
- обучающая среда, 38
- опровержение, 59
- отношение, 18, 80, 97
- видимости, 99
- каузальное, 91, 168
- между именами, 48
- семантическое, 99
- транзитивность, 100
- унарное, бинарное, тернарное, 43
- отсечение, 66

перебор (поиск)

- в ширину, глубину, 75
- переменная
- свободная, 158
- поиск
- эвристические методы, 77
- мягкий, 38
- полиморфизм, 92

- постусловие, 72
- правила
 - продукций, 72
- правило
 - подстановки, 64
 - эвристическое, 78
- предикат, 61
- принцип
 - Р. Беллмана, 78
 - резолюций, 65
- проблема
 - ИИ теоретическая, 24
 - анализа мерности пространства, 88
 - замыкания, 90
 - заслонений, 89
 - искажения перспективы, 88
 - объект-фон, 85
- программирование
 - визуальное, 79
 - с обратным слежением, 77
- пространство
 - неметрическое, 100
- процедура
 - внешняя, 18
 - присоединенная, 18, 75
- процесс
 - воображения, 84, 85
 - детализации, 87
 - изменения фреймов, 87
 - мышления, 85
 - обновления фреймов, 87
 - обучения, 87
 - оправдания, 87
 - повторного распознавания, 85
 - распознавания, 84, 85
 - советования, 87
 - согласования, 86
 - сопоставления, 87
 - составления резюме, 87
 - управления, 88
- разделяющее пространство/ гипер-
плоскость/функция, 113
- рассуждение
 - неточное, 48
- режим реального времени, 36
- резолюция, 65
- решение
 - адекватность, 32
 - надежность, 32
 - стоимость, 32
 - точность, 31
 - экономичность, 32
 - методики принятия, 122
- робот
 - биомашина, 25
 - двуногий, 25
- семантика, 97
- сеть
 - концепт, 102
 - нейронная, 130
 - семантическая, 97
 - сценарий, 99
- силлогизм
 - модус, 56
- система
 - имитирующая творческие процессы, 23
 - интеллектуальная, 15
 - интеллектуальная информационная, 23
 - поиска информации, 38
 - сложная, 34
 - состояние, 67
 - фреймов, 86, 87
 - целенаправленная и целеустремленная, 15
 - экспертная, 23
- систематика, 51
- словарная статья, 53
- состояние, 20
 - начальное, 21
- софизмы и паралогизмы, 60
- степень (коэффициент)
уверенности, 67
- сущность, 43
- таблицы переходов, 20
- таксономия, 53
- тезаурус
 - информационно-поисковый, 52
- точка зрения, 32
- транспьютер, 130
- триггер, 66
- умозаключение
 - дедуктивное, 55
 - индуктивное, 55

устойчивость, 31

формула

— выполняемая, 63

— общезначимая, 63

— семантика, 62

фрейм, 84

— вершина-терминал, 86

— главный, 85

— глобальная система, 90

— задания отсутствия, 85

— маркеры, 86

— образец, прототип, 92

— ожидания, 85

— сценарий, 91

— трансформация, 86, 88

— центральный, 88

цель, 20

— дерево, 37

шкала

— измерительная, 114

эвристика, 21

эвристический

— поиск, 75

экономичность, 32

экспертная система, 136

энтимема, 57

эффективность

— критерий, 21

язык (общения людей), 43

СОДЕРЖАНИЕ

От автора	3
Программа курса	5
 Глава 1. Искусственный интеллект как вершина развития информационных технологий	 12
1.1. Предыстория теории искусственного интеллекта	12
1.2. Этапы развития программных средств	14
1.3. Понятие «искусственный интеллект»	15
1.3.1. Термины и определения	17
1.3.2. Современные области исследований в искусственном интеллекте	23
1.3.3. Современные теоретические проблемы искусственного интеллекта	24
 Глава 2. Основные теоретические задачи искусственного интеллекта	 28
2.1. Искусственный интеллект — междисциплинарная область исследований	28
2.2. Перечень традиционных задач искусственного интеллекта	29
2.2.1. Более строгие формулировки задач	30
2.2.2. Правила формулировки задач искусственного интеллекта	30
2.2.3. Выбор параметров решения задачи	31
2.2.4. Выбор модели решения (представления знаний)	32
 Глава 3. Области применения методов искусственного интеллекта	 34
3.1. Хорошо и плохо структурированные предметные области	34
3.2. Модели представления знаний	39
 Глава 4. Логическая модель для представления знаний	 40
4.1. Виды логических моделей, общие термины и определения	40
4.2. Формальная (Аристотелева) логика	42
4.2.1. Имена	47
4.2.2. Высказывания	53
4.2.3. Процедуры доказательства и опровержения	57
4.3. Математическая реализация формальной логики	61
4.3.1. Методы автоматического доказательства теорем (исчисление предикатов)	64
4.3.2. Элементы теории нечетких множеств Л. Заде	67
 Глава 5. Продукционная модель для представления знаний	 72
5.1.1. Описание предметной области правилами и фактами	72
5.1.2. Метод полного перебора в ширину	76
5.1.3. Метод полного перебора в глубину	76
5.1.4. Эвристические методы поиска в пространстве состояний	77
5.1.5. Решение задач методом разбивания на подзадачи	79
5.1.6. Представление задачи в виде И-ИЛИ графа	80
5.1.7. Управление системой продукции	81
5.1.8. Достоинства и недостатки продукционной модели	81

Глава 6. Фреймы для представления знаний	83
6.1.1. Анализ пространственных сцен	88
6.1.2. Понимание смысла предложения	90
6.1.3. Практическая реализация фреймовой модели	92
Глава 7. Семантические сети для представления знаний	97
Глава 8. Новые модели представления знаний	104
8.1. Критериальные методы	104
8.1.1. Задача выбора	105
8.1.2. Измерительные шкалы	114
8.1.3. Некоторые методики принятия решения	122
8.1.4. Учебный пример многокритериального принятия решений	123
8.2. Вероятностные методы	128
8.3. Нейронные сети	130
Глава 9. Экспертные системы	136
9.1. Определение термина «экспертная система»	136
9.2. Технология работы с экспертными системами	137
9.3. Структура экспертных систем	138
9.4. Классификация экспертных систем	143
9.4.1. Классификация по решаемой задаче	145
9.5. Технология разработки экспертных систем	148
9.5.1. Основные этапы разработки	148
9.5.2. Некоторые практические рекомендации	151
Заключение	155
Приложения	157
1. Инструментальные средства разработки ЭС	157
2. Некоторые широко известные задачи, системы и методы	161
3. Сводная таблица некоторых широко известных интеллектуальных систем	182
4. Примерные планы практических работ	187
5. Некоторые тесты	195
6. Ответы к тестам	200
7. Список литературы	201
8. Предметный указатель	203