

Оглавление

1) Искусственный интеллект – история развития, основные подходы, области исследования.	1
Подходы к пониманию проблемы.....	2
Модели и методы исследований.....	4
2) Три аспекта семиотической системы	6
3) Интеллектуальные системы (Понятия: интеллектуальная задача, проблемная область, блок представления знаний).....	6
4) Общие сведения о моделях представления знаний	6
5) Формы представления знаний.....	7
6) Логические модели представления знаний; понятие формальной системы.....	8
7) Построение системы логического вывода: полный перебор (в глубину, в ширину); эвристический поиск.....	10
8) Построение системы логического вывода: прямая и обратная цепочки рассуждений; И/ИЛИ дерево	11
Прямая цепочка рассуждений и алгоритм ее реализации.....	11
9) Компоненты ЭС, фундаментальные свойства ЭС.....	13
10) Общие вопросы построения ЭС; классы приложений инженерии знаний.....	14
11) Архитектура ЭС	14
Решатель	14
База данных	14
База знаний	15
Компоненты приобретения знаний.....	15
Объяснительный компонент	15
Диалоговый компонент.....	15
12) Пути совершенствования СИИ – общие направления.....	15

1) Искусственный интеллект – история развития, основные подходы, области исследования.

В английском языке словосочетание *artificial intelligence* не имеет антропоморфной окраски, которую оно приобрело в традиционном русском переводе: слово *intelligence* в используемом контексте скорее означает «умение рассуждать разумно», а вовсе не «интеллект» (для которого есть английский аналог *intellect*).

Даются следующие определения искусственного интеллекта:

1. Научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного моделирования тех видов человеческой деятельности, которые традиционно считаются интеллектуальными.

2. Свойство интеллектуальных систем выполнять функции (творческие), которые традиционно считаются прерогативой человека. При этом интеллектуальная система — это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы. Структура интеллектуальной системы включает три основных блока — базу знаний, решатель и интеллектуальный интерфейс, позволяющий вести общение с ЭВМ без специальных программ для ввода данных^[1].
3. Направление в информатике и информационных технологиях, задачей которого является воссоздание с помощью вычислительных систем и иных искусственных устройств разумных рассуждений и действий.
4. Согласно определению Андреаса Каплана и Майкла Хенлейна, искусственный интеллект — это «способность системы правильно интерпретировать внешние данные, извлекать уроки из таких данных и использовать полученные знания для достижения конкретных целей и задач при помощи гибкой адаптации».

Одно из частных определений интеллекта, общее для человека и «машины», можно сформулировать так: «Интеллект — способность системы создавать в ходе самообучения программы (в первую очередь эвристические) для решения задач определённого класса сложности и решать эти задачи».

Подходы к пониманию проблемы

Единого ответа на вопрос, чем занимается искусственный интеллект, не существует. Почти каждый автор, пишущий книгу об ИИ, отталкивается в ней от какого-либо определения, рассматривая в его свете достижения этой науки.

В философии не решён вопрос о природе и статусе человеческого интеллекта. Нет и точного критерия достижения компьютерами «разумности», хотя на заре искусственного интеллекта был предложен ряд гипотез, например, тест Тьюринга или гипотеза Ньюэлла — Саймона. Поэтому, несмотря на наличие множества подходов как к пониманию задач ИИ, так и созданию интеллектуальных информационных систем, можно выделить два основных подхода к разработке ИИ^[14]:

- нисходящий (англ. *Top-Down AI*), семиотический — создание экспертных систем, баз знаний и систем логического вывода, имитирующих высокоуровневые психические процессы: мышление, рассуждение, речь, эмоции, творчество и т. д.;
- восходящий (англ. *Bottom-Up AI*), биологический — изучение нейронных сетей и эволюционных вычислений, моделирующих интеллектуальное поведение на основе биологических элементов, а также создание соответствующих вычислительных систем, таких как нейрокомпьютер или биокомпьютер.

Последний подход, строго говоря, не относится к науке об ИИ в смысле, данном Джоном Маккарти, — их объединяет только общая конечная цель.

Тест Тьюринга и интуитивный подход

Эмпирический тест был предложен Аланом Тьюрингом в статье «Вычислительные машины и разум» (англ. *Computing Machinery and Intelligence*)^[15], опубликованной в 1950 году в философском журнале «*Mind*». Целью данного теста является определение возможности искусственного мышления, близкого к человеческому.

Стандартная интерпретация этого теста звучит следующим образом: «Человек взаимодействует с одним компьютером и одним человеком. На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой. Задача компьютерной программы — ввести человека в заблуждение, заставив сделать неверный выбор». Все участники теста не видят друг друга.

- Самый общий подход предполагает, что ИИ будет способен проявлять поведение, не отличающееся от человеческого, причём в нормальных ситуациях. Эта идея является обобщением подхода теста Тьюринга, который утверждает, что машина станет

разумной тогда, когда будет способна поддерживать разговор с обычным человеком, и тот не сможет понять, что говорит с машиной (разговор идёт по переписке).

- Писатели-фантасты часто предлагают ещё один подход: ИИ возникнет тогда, когда машина будет способна чувствовать и творить. Так, хозяин Эндрю Мартина из «Двухсотлетнего человека» начинает относиться к нему как к человеку, когда тот создаёт игрушку по собственному проекту. А Дейта из «Звёздного пути», будучи способным к коммуникации и обучению, мечтает обрести эмоции и интуицию.

Однако последний подход вряд ли выдерживает критику при более детальном рассмотрении. К примеру, несложно создать механизм, который будет оценивать некоторые параметры внешней или внутренней среды и реагировать на их неблагоприятные значения. Про такую систему можно сказать, что у неё есть чувства («боль» — реакция на срабатывание датчика удара, «голод» — реакция на низкий заряд аккумулятора, и т. п.). А кластеры, создаваемые картами Кохонена, и многие другие продукты «интеллектуальных» систем можно рассматривать как вид творчества.

Символьный подход

Исторически символьный подход был первым в эпоху цифровых машин, так как именно после создания Лисп, первого языка символьных вычислений, у его автора возникла уверенность в возможности практически приступить к реализации этими средствами интеллекта. Символьный подход позволяет оперировать слабоформализованными представлениями и их смыслами.

Успешность и эффективность решения новых задач зависит от умения выделять только существенную информацию, что требует гибкости в методах абстрагирования. Тогда как обычная программа устанавливает один свой способ интерпретации данных, из-за чего её работа и выглядит предвзятой и чисто механической. Интеллектуальную задачу в этом случае решает только человек, аналитик или программист, не умея доверить этого машине. В результате создается единственная модель абстрагирования, система конструктивных сущностей и алгоритмов. А гибкость и универсальность выливается в значительные затраты ресурсов для не типичных задач, то есть система от интеллекта возвращается к грубой силе.

Основная особенность символьных вычислений — создание новых правил в процессе выполнения программы. Тогда как возможности не интеллектуальных систем завершаются как раз перед способностью хотя бы обозначать вновь возникающие трудности. Тем более эти трудности не решаются и наконец компьютер не совершенствует такие способности самостоятельно.

Недостатком символьного подхода является то, что такие открытые возможности воспринимаются не подготовленными людьми как отсутствие инструментов. Эту, скорее культурную проблему, отчасти решает логическое программирование.

Логический подход

Логический подход к созданию систем искусственного интеллекта основан на моделировании рассуждений. Теоретической основой служит логика.

Логический подход может быть проиллюстрирован применением для этих целей языка и системы логического программирования Пролог. Программы, записанные на языке Пролог, представляют наборы фактов и правил логического вывода без жесткого задания алгоритма как последовательности действий, приводящих к необходимому результату.

Агентно-ориентированный подход

Последний подход, развиваемый с начала 1990-х годов, называется *агентно-ориентированным подходом*, или *подходом, основанным на использовании интеллектуальных (рациональных) агентов*. Согласно этому подходу, интеллект — это вычислительная часть (грубо говоря, планирование) способности достигать поставленных перед интеллектуальной машиной целей. Сама такая машина будет интеллектуальным агентом, воспринимающим окружающий его мир с помощью датчиков, и способной воздействовать на объекты в окружающей среде с помощью исполнительных механизмов.

Этот подход акцентирует внимание на тех методах и алгоритмах, которые помогут интеллектуальному агенту выживать в окружающей среде при выполнении его задачи. Так, здесь значительно тщательнее изучаются алгоритмы поиска пути и принятия решений.

Гибридный подход

синергичная комбинация нейронных и символьных моделей достигает полного спектра когнитивных и вычислительных возможностей. Например, экспертные правила умозаключений могут генерироваться нейронными сетями, а порождающие правила получают с помощью статистического обучения. Сторонники данного подхода считают, что гибридные информационные системы будут значительно более сильными, чем сумма различных концепций по отдельности.

Модели и методы исследований

Анализируя историю ИИ, можно выделить такое обширное направление как моделирование рассуждений^[16]. Долгие годы развитие этой науки двигалось именно по этому пути, и теперь это одна из самых развитых областей в современном ИИ. Моделирование рассуждений подразумевает создание символьных систем, на входе которых поставлена некая задача, а на выходе требуется её решение. Как правило, предлагаемая задача уже формализована, то есть переведена в математическую форму, но либо не имеет алгоритма решения, либо он слишком сложен, трудоёмок и т. п. В это направление входят: доказательство теорем, принятие решений и теория игр, планирование и диспетчеризация, прогнозирование.

Работа с естественными языками

Немаловажным направлением является обработка естественного языка^[17], в рамках которого проводится анализ возможностей понимания, обработки и генерации текстов на «человеческом» языке. В рамках этого направления ставится цель такой обработки естественного языка, которая была бы в состоянии приобрести знание самостоятельно, читая существующий текст, доступный по Интернету. Некоторые прямые применения обработки естественного языка включают информационный поиск (в том числе, глубокий анализ текста) и машинный перевод^[18].

Представление и использование знаний

Направление инженерия знаний объединяет задачи получения знаний из простой информации, их систематизации и использования. Это направление исторически связано с созданием экспертных систем — программ, использующих специализированные базы знаний для получения достоверных заключений по какой-либо проблеме.

Производство знаний из данных — одна из базовых проблем интеллектуального анализа данных. Существуют различные подходы к решению этой проблемы, в том числе — на основе нейросетевой технологии^[19], использующие процедуры вербализации нейронных сетей.

Машинное обучение

Проблематика машинного обучения^[20] касается процесса самостоятельного получения знаний интеллектуальной системой в процессе её работы. Это направление было центральным с самого начала развития ИИ^[21]. В 1956 году, на Дартмундской летней конференции, Рей Соломонофф написал отчёт о вероятностной машине, обучающейся без учителя, назвав её: «Индуктивная машина вывода»^[22].

Обучение без учителя — позволяет распознать образы во входном потоке. Обучение с учителем включает также классификацию и регрессионный анализ. Классификация используется, чтобы определить, к какой категории принадлежит образ. Регрессионный анализ используется, чтобы в рядах числовых примеров входа/выхода и обнаружить непрерывную функцию, на основании которой можно было бы прогнозировать выход. При обучении агент вознаграждается за хорошие ответы и наказывается за плохие. Они могут быть проанализированы с точки зрения теории решений, используя такие понятия

как *полезность*. Математический анализ машинных алгоритмов изучения — это раздел теоретической информатики, известный как вычислительная теория обучения (англ. *Computational learning theory*).

К области машинного обучения относится большой класс задач на распознавание образов. Например, это распознавание символов, рукописного текста, речи, анализ текстов. Многие задачи успешно решаются с помощью биологического моделирования (см. след. пункт). Особо стоит упомянуть компьютерное зрение, которое связано ещё и с робототехникой.

Биологическое моделирование искусственного интеллекта

Отличается от понимания искусственного интеллекта по Джону Маккарти, когда исходят из положения о том, что искусственные системы не обязаны повторять в своей структуре и функционировании структуру и протекающие в ней процессы, присущие биологическим системам. Сторонники данного подхода считают, что феномены человеческого поведения, его способность к обучению и адаптации есть следствие именно биологической структуры и особенностей её функционирования.

Сюда можно отнести несколько направлений. Нейронные сети используются для решения нечётких и сложных проблем, таких как распознавание геометрических фигур или кластеризация объектов. Генетический подход основан на идее, что некий алгоритм может стать более эффективным, если позаимствует лучшие характеристики у других алгоритмов («родителей»). Относительно новый подход, где ставится задача создания автономной программы — агента, взаимодействующей с внешней средой, называется агентным подходом.

Робототехника

Области робототехники^[23] и искусственного интеллекта тесно связаны друг с другом. Интегрирование этих двух наук, создание интеллектуальных роботов составляют ещё одно направление ИИ. Интеллектуальность требуется роботам, чтобы манипулировать объектами^[24], выполнять навигацию с проблемами локализации (определять местонахождение, изучать ближайшие области) и планировать движение (как добраться до цели)^[25]. Примером интеллектуальной робототехники могут служить игрушки-роботы Pleo, AIBO, QRIO.

Машинное творчество

Природа человеческого творчества ещё менее изучена, чем природа интеллекта. Тем не менее, эта область существует, и здесь поставлены проблемы написания компьютером музыки, литературных произведений (часто — стихов или сказок), художественное творчество. Создание реалистичных образов широко используется в кино и индустрии игр.

Отдельно выделяется изучение проблем технического творчества систем искусственного интеллекта. Теория решения изобретательских задач, предложенная в 1946 году Г. С. Альтшуллером, положила начало таким исследованиям.

Добавление данной возможности к любой интеллектуальной системе позволяет весьма наглядно продемонстрировать, что именно система воспринимает и как это понимает. Добавлением шума вместо недостающей информации или фильтрация шума имеющимися в системе знаниями производит из абстрактных знаний конкретные образы, легко воспринимаемые человеком, особенно это полезно для интуитивных и малоценных знаний, проверка которых в формальном виде требует значительных умственных усилий.

Другие области исследований

Наконец, существует масса приложений искусственного интеллекта, каждое из которых образует почти самостоятельное направление. В качестве примеров можно привести программирование интеллекта в компьютерных играх, нелинейное управление, интеллектуальные системы информационной безопасности.

В перспективе предполагается тесная связь развития искусственного интеллекта с разработкой квантового компьютера, так как некоторые свойства искусственного интеллекта имеют схожие принципы действия с квантовыми компьютерами^{[26][27]}.

Можно заметить, что многие области исследований пересекаются. Это свойственно любой науке. Но в искусственном интеллекте взаимосвязь между, казалось бы, различными направлениями выражена особенно сильно, и это связано с философским спором о сильном и слабом ИИ.

2) Три аспекта семиотической системы

Е. И. Ефимов предложил следующую классификацию типов задач [8]. Задачи первого типа, для которых существует формальная схема решений, представленная на некоем формальном языке. Решение задач осуществляется по имеющейся схеме (детерминированной или вероятностной). Алгоритмическое решение таких задач не представляет проблемы. Задачи второго типа, для которых не существует заранее готовой схемы решения, но хорошо известны знания о предметной области. Обычно в этом случае человек сам формирует схему решения на основе трансформации неявного знания в явное знание – решение. Алгоритмическое решение таких задач не представляет проблемы. Задачи третьего типа, для которых не существует заранее готовой схемы решения и не известны знания о предметной области, которые можно трансформировать в решение. Поиск решения таких задач реализуются сложными эвристическими методами. Упрощенно задачи первого и второго типа называют задачами первого рода, а задачи третьего типа задачами второго рода. Для задач первого рода существует возможность семиотического описания, которое и создает возможность семиотического управления.

3) Интеллектуальные системы (Понятия: интеллектуальная задача, проблемная область, блок представления знаний)

Интеллектуальная система (ИС) – автоматизированная система, основанная на знаниях, или комплекс программных, лингвистических и логико-математических средств для реализации основной задачи – осуществления поддержки деятельности человека и поиска информации в режиме продвинутого диалога на естественном языке. Кроме того, информационно-вычислительными системами с интеллектуальной поддержкой для решения сложных задач называют те системы, в которых логическая обработка информации превалирует над вычислительной. Таким образом, любая информационная система, решающая интеллектуальную задачу или использующая методы искусственного интеллекта, относится к интеллектуальным.

Критерий, который мы будем применять, может быть назван прагматическим. Мы объявим программную систему **интеллектуальной**, если она решает **задачи**, для решения которых специалист должен напрячь свой интеллект.

Проблемная область - некая совокупность задач и вопросов, которая решается исследователями.

База знаний - совокупность систематизированных основополагающих сведений, относящихся к определённой области знания, хранящихся в памяти ЭВМ, объём которых необходим и достаточен для решения заданного круга теоретических или практических задач. В системе управления БЗ используются методы искусственного интеллекта, специальные языки описания знаний, интеллектуальный интерфейс.

4) Общие сведения о моделях представления знаний

Знания о предметной области, ее объектах и закономерностях описываются на некотором формальном языке, называемом языком представления знаний (ЯПЗ). При этом ЯПЗ должен обеспечивать не только возможность формальной записи знаний, но и необходимую обработку элементов этой записи. Совокупность знаний, хранящаяся во

внешней памяти ЭВМ, называется системой знаний; над ней производятся различные операции: поиск необходимой информации, ее модификация, интерпретация знаний, вывод новых знаний, на основе имеющихся и т.д. Организация интеллектуальных компьютерных систем работы со знаниями в той или иной прикладной области зависят не только от специфики самих систем, но и в существенной мере от используемых ЯПЗ и метода хранения системы знаний в ЭВМ. В современных ИИ-системах знания хранятся в специальных БД - базах знаний (БЗ). Существует самая непосредственная аналогия между понятиями модель данных, используемая в БД-технологии и понятием способ представления знаний в БЗ-технологии.

Грубо говоря, можно сказать, что если БД содержит собственно данные о некоторой предметной области, то БЗ содержит как сами данные, так и описание их свойств. Однако между БД и БЗ существуют принципиальные различия, рассматриваемые ниже

Процесс построения БЗ на основе информации эксперта состоит из трех этапов: описание предметной области, выбор способа и модели представления знаний и приобретение знаний.

Сам процесс построения БЗ достаточно сложен, как правило, плохо структурирован и носит итеративный характер, заключающийся в циклической модификации БЗ на основе результатов ее тестирования.

На первом шаге построения БЗ четко очерчивается предметная область, на решение задач из которой ориентируется проектируемая ЭС, т.е. инженер знаний определяет область применения будущей системы и класс решаемых ею задач. В перечень работ данного шага входят: (1) определение характера решаемых ЭС задач и основных понятий, объектов предметной области, а также отношений между ними; (2) установление специфических особенностей предметной области и (3) выбор модели представления знаний.

После решения первых двух вопросов инженер знаний формально описывает предметную область на языке представления знаний (ЯПЗ), т.е. создает модель представления знаний. В настоящее время универсальный способ представления знаний отсутствует, поэтому инженер знаний должен максимально учитывать специфику исходной предметной области.

5) Формы представления знаний

Императивные формы представления знаний – это традиционные (процедурные) способы описания процессов решения задач в виде последовательностей операций над данными, совершаемых согласно заданным алгоритмам или формулам (как, например, последовательность операций вычисления выражения $y := x \cdot (z + v)$). В процедурах знания (связи, зависимости, законы) представлены (учтены) неявно: в организации вычислительного процесса, в структуре программы решения задачи, в характере и последовательностях операций. По этой причине императивная форма представления знаний наиболее эффективна с вычислительной точки зрения (по затратам времени и памяти на решение задачи), поскольку в процедурах поиска решения глубоко учитывается специфика конкретной проблемной области и решаемой задачи, что является важным фактором при создании систем, работающих в реальных условиях (в системах реального времени). Главный недостаток этой формы представления знаний – сложность внесения изменений, что делает ее непригодной для использования в слабо изученных и изменяющихся ПО.

Декларативные формы представления знаний разработаны в рамках исследований по искусственному интеллекту. Их отличительная особенность в том, что знания относительно ПО в этом случае описываются в виде совокупности утверждений, характеризующих состав, свойства, законы строения и поведения (например, закон Ома в электротехнике связывает зависимость величины тока, напряжения и сопротивления, позволяющей вычислить каждую из этих величин, если заданы две

другие). Знания в этой форме можно использовать для решения любых задач, связанных с данной ПО. Постановка задачи в этом случае сводится к описанию свойств искомого решения (цели), способ же поиска решений (механизм поиска, «машина» вывода) универсален и не зависит ни от поставленной задачи, ни даже от ПО, что весьма важно при описании слабо изученных и изменяющихся ПО. Главный недостаток этой формы представления знаний – низкая вычислительная эффективность (по затратам времени и памяти), поскольку в процедурах поиска решения не учитывается специфика решаемой задачи и ПО, что делает эту форму непригодной для применения в системах реального времени.

Комбинированные формы описания знаний создаются, чтобы преодолеть недостатки и сохранить достоинства императивной и декларативной форм. Достигается это за счет того, что хорошо обоснованная, устойчивая и формализованная часть знания воплощается в эффективных процедурах, а слабо изученная и изменчивая составляющая знания представляется в декларативной форме. Главный недостаток комбинированных форм представления знаний – трудность их теоретизации из-за их составного характера, что препятствует созданию теоретически обоснованных методов построения баз знаний и методов поиска решений с использованием таких форм представления знаний.

6) Логические модели представления знаний; понятие формальной системы

- логические модели;
- продукционные модели;
- сетевые модели;
- фреймовые модели.

Чисто декларативные формы представления знаний реализовать весьма сложно, но наиболее близкими к ним вариантами являются логические и продукционные модели. В сетевых и фреймовых моделях находит воплощение идея комбинированных форм представления знаний.

Логические модели [3, 4, 5, 13, 14, 16, 17]

Все предметы, взаимосвязи, события и процессы, составляющие основу необходимой для решения задачи информации, называют *предметной областью*. Мысленно предметная область представляется состоящей из реальных компонент, называемых *сущностями*. Сущности каждой конкретной предметной области характеризуются свойствами и находятся между собой в определенных *отношениях*. Свойства и отношения между сущностями описывают посредством *суждений*. Суждения в каждом языке (как естественном, так и формальном) выражают в виде *предложений*.

Языки, предназначенные для описания предметных областей, называются *языками представления знаний*. Универсальным языком представления знаний является естественный язык, но применение естественного языка для машинного представления знаний наталкивается на ряд препятствий, главным из которых является отсутствие формальной семантики естественного языка. *Семантика* – это смысловое значение единиц языка. В естественном языке смысл слов и выражений часто зависит от текста, в котором они встречаются. Кроме того, естественные языки – это «живые», постоянно меняющиеся и развивающиеся языки.

Для представления математического знания пользуются формальными логическими языками – исчислением высказываний и исчислением предикатов. Эти языки имеют ясную формальную семантику, и для них разработаны формальные методы логического вывода. Поэтому исчисление предикатов было первым логическим языком, который стали применять для формального описания пригодных для этого предметных областей.

Описания предметных областей, выполненные в логических языках, называются *логическими моделями*. Логические модели, построенные с применением языков логического программирования, довольно широко применяются в базах знаний систем искусственного интеллекта и экспертных систем.

Формальные системы. Многие научные теории (не обязательно математические) строятся по следующему принципу. Сначала предлагаются некоторые основные понятия и некоторые исходные законы (аксиомы), присущие основным понятиям. Далее формулируются производные понятия и по определенным правилам доказываются некоторые утверждения (теоремы), относящиеся к основным и производным понятиям. Совокупность основных и производных понятий, аксиом и теорем, построенная таким способом, называется *аксиоматической системой*.

Часто аксиомы, а значит, и теоремы, аксиоматической системы сохраняют истинность при замене одних основных понятий другими (как, например, в теории колебаний, которая находит применение в механике, электронике, оптике). Это позволяет рассматривать аксиоматические системы с двух позиций: синтаксически (принципы построения правильных и истинных предложений) и семантически (связь смысла правильных и истинных предложений со смыслом основных понятий).

Для исследования синтаксиса аксиоматической системы требуется ее полная формализация, т. е. символическое представление основных и производных понятий, аксиом, правил вывода и теорем. Поэтому *формальная аксиоматическая теория* (*формальная система*) – это синтаксический аспект (сторона) аксиоматической системы. Точное же определение понятия формальной аксиоматической теории включает следующие компоненты.

Во-первых, каждая формальная аксиоматическая теория должна иметь свой *формальный язык*. Формальный язык считается полностью определенным, когда задано (счётное) множество его символов и описаны формулы языка. Любая конечная последовательность символов языка называется выражением этого языка. Среди всех возможных выражений выделяются формулы языка, под которыми подразумеваются правильно построенные, утверждающие нечто осмысленное предложения языка.

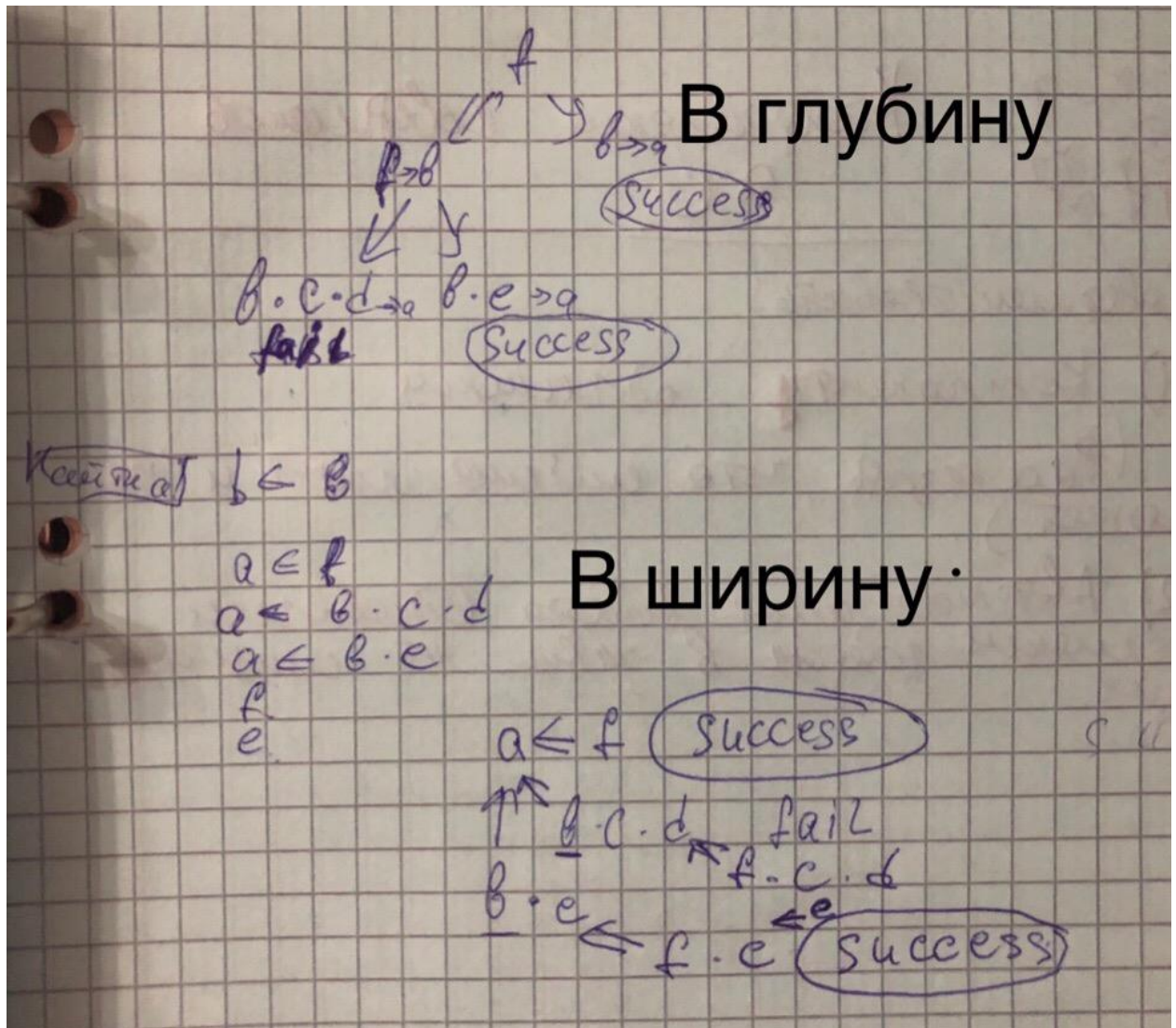
Во-вторых, каждая формальная аксиоматическая теория должна иметь свою *систему аксиом* – подмножество заведомо истинных формул, из которых по правилам теории могут быть выведены все истинные предложения этой теории (обычно к системе аксиом предъявляются требования непротиворечивости, независимости и полноты, среди которых обязательным является лишь требование непротиворечивости).

В-третьих, каждая формальная аксиоматическая теория должна располагать конечным множеством *правил вывода*. Каждое правило вывода содержит формулы-посылки и формулу-заключение, выводимую при определенных этим правилом условиях из формул-посылок. Формула-заключение называется непосредственным следствием формул-посылок по данному правилу вывода.

Доказательством в формальной аксиоматической теории называется всякая последовательность формул, в которой каждая формула есть либо аксиома теории, либо непосредственное следствие каких-либо предыдущих формул этой последовательности по одному из правил вывода данной теории. Формула A называется *теоремой* теории (обозначается $\vdash A$) тогда и только тогда, когда существует доказательство, в котором A является заключительной формулой.

Выводом из множества гипотез (посылок) Γ в формальной аксиоматической теории называется всякая последовательность формул, в которой каждая формула есть либо аксиома теории, либо гипотеза из множества Γ , либо непосредственное следствие каких-либо предыдущих формул этой последовательности по одному из правил вывода данной теории. Формула A называется *следствием множества гипотез* Γ (обозначается $\Gamma \vdash A$) тогда и только тогда, когда существует вывод из множества гипотез Γ , в котором A является заключительной формулой.

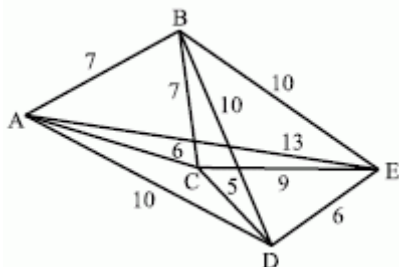
7) Построение системы логического вывода: полный перебор (в глубину, в ширину); эвристический поиск



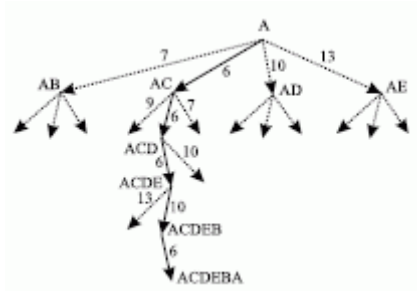
Однако при значительном числе состояний время поиска возрастает экспоненциально, и в этом случае могут помочь *алгоритмы эвристического поиска*, которые обладают высокой вероятностью правильного выбора решения. Рассмотрим некоторые из этих алгоритмов.

Алгоритм наискорейшего спуска по дереву решений

Пример построения более узкого дерева рассмотрим на примере задачи о коммивояжере. Торговец должен побывать в каждом из 5 городов, обозначенных на карте.



Задача состоит в том, чтобы, начиная с города А, найти минимальный путь, проходящий через все остальные города только один раз и приводящий обратно в А. Идея метода исключительно проста - из каждого города идем в ближайший, где мы еще не были. Решение задачи показано на рис .



Такой алгоритм поиска решения получил название *алгоритма наискорейшего спуска* (в некоторых случаях - наискорейшего подъема).

8) Построение системы логического вывода: прямая и обратная цепочки рассуждений; И/ИЛИ дерево

Прямая цепочка рассуждений и алгоритм ее реализации.

В простейшем случае применение правил сводится к попыткам поочерёдной инициализации правил в том порядке, в котором они записаны. Обычно вводится целевая переменная (GOAL, G). Считается, что применение продукций может быть закончено, если достигнуто значение целевой переменной. Идея прямой цепочки рассуждений заключается в последовательном применении правил по порядку, с возвратом после просмотра всех правил снова в начало, до тех пор, пока не будет получено значение целевой переменной. Мы можем определить для данного значения правила несколько переменных или переопределить. В любой продукционной системе можно получить значение переменной тремя способами:

1. Начальное присваивание, т.е. перед применением правил
2. Из правил – переменная стоит в следствии правила
3. Запрос у пользователя и ввод значения в диалоге

Недостатки: избыточность, попытка инициализировать ненужные правила

Достоинства: простота алгоритма, простота реализации.

Переменная 1	<u>Реализация алгоритма:</u>
	1. Завести массивы
	Очередь вывода
	Таблица переменных

Перем	№ правила	№ условия	Текущ.значение	Следствие

Текущее состояние системы

Переменная	Правило	№ условия	Назначение	Следствие

Алгоритм ПЦР:

1. Фиксируется исходное состояние системы, в очередь переменных вывода заносится целевая переменная, формируется таблица переменных с текущими значениями переменных.
2. Организуется цикл просмотра правил в том порядке, в котором они записаны, а внутри проверка условий в той последовательности, в которой они стоят в конъюнкциях. Осуществляется проверка первого правила, в нём первого условия, а в условии первой переменной. Если этой переменной нет в таблице переменных, то выдаётся сообщение об отсутствии решения. Если она имеется в таблице и у неё есть значение, то осуществляется переход к следующей переменной первой конъюнкции. Если значение переменной равно NIL, то проверяется можно ли её

запросить у пользователя, если Да, то значение запрашивается, если Нет, то переменная помещается в очередь переменных вывода и осуществляется переход к проверке следующего правила. Если первая конъюнкция истинна, то переход к проверке второй конъюнкции, и она проводится аналогично. Если конъюнкция ложна, то осуществляется переход к проверке следующего правила.

3. Если посылка правила истинна, то реализуется действия части “То” правила (т.е. действия, стоящего в следствии). В результате этих действий состояние системы меняется, что отображается в таблице текущего состояния системы (TSS), таблица переменных обновляется, а те переменные, которые получили значение, вычеркиваются из таблицы очереди вывода. Если значение получила целевая переменная, то логический вывод завершается и осуществляется выход из цикла проверки правил (т.е. конец – цель достигнута), иначе переход к проверке следующего правила.
4. Цикл проверки правил продолжается до последнего правил, если цель не достигнута, то осуществляется переход снова к 1 правилу.

В процессе реализации цикла проверки, закладывается условие отсутствия изменения очереди переменных вывода. Если в течение нескольких циклов очередь вывода не изменяется, то делается вывод о том, что данная система правил не достаточна для получения решений и осуществляется выход из цикла проверки правил.

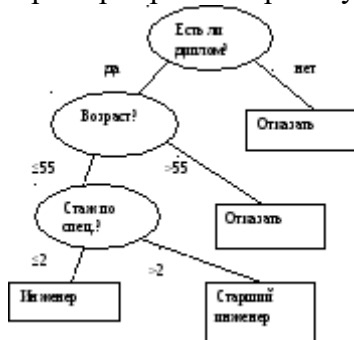
Обратная цепочка рассуждений.

Основная идея ОЦР – это выделение тех правил, которые непосредственно влияют на процесс логического вывода.

Определяется:

1. Стек логического вывода
2. TSS (Переменная, Правило, № условия, Следствие, Значение)
3. Таблица переменных

Пример: прием на работу



Наличие диплома – d

Возраст – v

Стаж – s

Должность – dolgn

R1: IF d AND (v ≤ 55) AND (s ≤ 2) THEN dolgn = “Инженер”

R2: IF d AND (v ≤ 55) AND (s > 2) THEN dolgn = “Старший инженер”

R3: IF d AND (v > 55) THEN dolgn = “Отказать”

R4: IF NOT d THEN dolgn = “Отказать”

Имя переменной	Правило	№ условия	Текущее значение	Следствие
D	R1	1	True	-
V	R1	2	32	-
S	R1	3	5	-
Dolgn	R1	-	Nil	+
D	R2	1	True	-
V	R2	2	32	-

Запрос: На какую должность принять дипломированного специалиста со стажем работы 5 лет в возрасте 32 года?

S	R2	3	5	-
Dolgn	R2	-	Nil	+
D	R3	1	True	-
V	R3	2	32	-
Dolgn	R3	-	True	+
D	R4	1	Nil	-
Dolgn	R4	-	True	+

Алгоритм ОЦР:

1. Фиксируем начальное состояние системы и осуществляем начальные присваивания, формируем стек, в который заносим должность и табличные переменные.
2. Определяется первое по порядку правило, в следствии которого осуществляется присваивание нового значения целевой переменной. Просматриваем данное правило начиная с первого условия конъюнкции. Если первая переменная в условии отсутствует в табличных переменных, то выдаётся сообщение об ошибке. Если она есть в таблице, и имеет значение, то переходим к проверке второй переменной в первом условии. Если значение равно nil, то определяется возможность запросить переменную у пользователя. Если нельзя её запросить у пользователя, то она помещается в стек логического вывода и становится очередной целевой переменной.
3. Проверяется, есть ли правила, в следствии которых осуществляется присваивание значения новой целевой переменной. Если есть, то берётся первое по порядку и проверяется аналогично пункту 2. Если правило отсутствует, то процесс завершается и выдаётся сообщение об ошибке.
4. Если проверка всех условий проходит в конъюнкции правила, то инициализируется часть ТО правила, состояние системы меняется: из стека удаляется имя промежуточной целевой переменной, обновляется TSS и табличные переменные.
5. Процесс повторяется по пунктам 2-4 до тех пор, пока или получится значение основной целевой переменной (стек будет пустым), или будет выявлено, что состояние системы в течение длительного времени не меняется и стек не опустошается.

Достоинства ОЦР: экономичность, рассматриваются правила, которые непосредственно участвуют в логическом выводе.

Недостатки ОЦР: большая сложность

Замечания:

1. При построении дерева надо стремиться к тому, чтобы число проверок в одном правиле было минимальным, тогда правило легче объяснить и инициализировать.
2. Чтобы избежать большого количества проверок, в дерево можно ввести дополнительные переменные, которые называются индикатором пути. Индикатор пути увеличивает число правил, но сами они при этом становятся проще. Индикатор пути также должен быть отражён в табличных переменных.

9) Компоненты ЭС, фундаментальные свойства ЭС

Стандартная статическая экспертная система состоит из следующих основных компонентов: 1) рабочей памяти, называемой также базой данных; 2) базы знаний; 3) решателя, называемого также интерпретатором; 4) компонентов приобретения знаний; 5)

объяснительного компонента; 6) диалогового компонента. Рассмотрим теперь каждый компонент более подробно. Рабочая память (по абсолютной аналогии с рабочей, т. е. оперативной памятью компьютера) предназначена для получения и хранения исходных и промежуточных данных решаемой в текущий момент задачи. База знаний предназначена для хранения долгосрочных данных, описывающих конкретную предметную область, и правил, описывающих рациональное преобразование данных этой области решаемой задачи. Решатель, называемый также интерпретатором, функционирует следующим образом: используя исходные данные из рабочей памяти и долгосрочные данные из базы знаний, он формирует правила, применение которых к исходным данным приводит к решению задачи. Одним словом, он действительно «решает» поставленную перед ним задачу; Компонент приобретения знаний автоматизирует процесс заполнения экспертной системы знаниями эксперта, т. е. именно этот компонент обеспечивает базу знаний всей необходимой информацией из данной конкретной предметной области. Компонент объяснений разъясняет, как система получила решение данной задачи, или почему она это решение не получила и какие знания она при этом использовала. Иначе говоря, компонент объяснений создает отчет о проделанной работе. Данный компонент является очень важным во всей экспертной системе, поскольку он значительно облегчает тестирование системы экспертом, а также повышает доверие пользователя к полученному результату и, следовательно, ускоряет процесс разработок. Диалоговый компонент служит для обеспечения дружественного интерфейса пользователя как в ходе решения задачи, так и в процессе приобретения знаний и объявления результатов работы.

Свойства

1. качество работы (поведения)
2. скорость достижения принятия решений
3. уровень, на котором находятся решения
4. Специализация

10) Общие вопросы построения ЭС; классы приложений инженерии знаний

В настоящее время известно несколько **принципов построения экспертных систем**: системы, основанные на правилах, и системы, использующие глубинные представления. Первые для своей работы требуют набора формальных логических правил типа приведенных нами, вторые, использующие глубинные представления, основаны на представлении информации в виде так называемых фреймов и значительно более сложны, чем системы, основанные на правилах. Однако эти системы пока еще только начинают выходить за стены лабораторий.

11) Архитектура ЭС

Решатель

Решатель предназначен для того, чтобы, используя исходные данные из рабочей памяти и знания из БЗ, формировать такую последовательность правил, которые, применяясь к исходным данным, позволяют решать необходимую задачу.

База данных

В базе данных (рабочей памяти) хранятся исходные и промежуточные данные решаемой задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах и системах управления базами данных (СУБД) для обозначения всех данных, хранимых в системе.

База знаний

База знаний необходима для хранения долгосрочных данных, описывающих рассматриваемую область, и правил, описывающих необходимые действия над данными этой области.

Компоненты приобретения знаний

Компоненты приобретения знаний автоматизируют процессы наполнения ЭС знаниями, осуществляемый пользователем-экспертом. Эксперт высокого уровня вносит свои знания в экспертную систему, точнее, в компонент приобретения знаний. На этом компоненте в последующем и основывается весь эвристический метод решения задач.

Объяснительный компонент

Объяснительный компонент интерпретирует в доступную для пользователя форму методы решения задачи или принятия системой определенного решения. Кроме того, он выполняет функции объяснения порядка использования данных, необходимых для принятия решения. Это облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Диалоговый компонент

Диалоговый компонент необходим для организации дружественного общения с пользователем, как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы. Название этого компонента говорит само за себя – он выполняет функции диалогового интерфейса. С его помощью в систему вносятся вопросы и поправки, а также визуализируются ответы.

12) Пути совершенствования СИИ – общие направления

1. Компонент объяснения (рассказать, что система может и не может)
2. Автоматизация выбора стратегии решения задач в зависимости от существующих решенных задач