



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информатика и системы управления»

Е.А. КРАЙНОВА, К.В. САДОВА

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Учебное пособие

Самара
Самарский государственный технический университет
2016

УДК 621.1

К 64

Крайнова Е.А., Садова К.В.

К 64 Системы искусственного интеллекта: учеб. пособие / *Е.А. Крайнова, К.В. Садова.* – Самара: Самар. гос. техн. ун-т, 2016. – **81** с.: ил.

Рассмотрена структура исследования в области искусственного интеллекта. Представлены задачи и общие методы решения хорошо определенных задач. Рассмотрены примеры использования интегрированной среды программирования баз знаний на языке Visual Prolog для решения задач искусственного интеллекта, даны рекомендации для построения баз знаний, а также приведены описание и примеры разработки экспертных компьютерных систем.

Учебное пособие предназначено для бакалавров, обучающихся по направлениям 09.03.01 «Информатика и вычислительная техника» и 44.03.04 «Профессиональное обучение» - профилю подготовки «Информатика, вычислительная техника и компьютерные технологии»

УДК 621.1

К 64

Рецензенты: доцент кафедры «Промышленная информатика»
ОАНО ВО «ВУиТ» канд. тех. наук *Н.О. Куралесова,*

преподаватель кафедры «Математика и естественно-
научные дисциплины» филиала ВУНЦ ВВС «ВВА»
в г. Сызрани канд. пед. наук *О.В. Раецкая*

© Е.А. Крайнова, К.В. Садова, 2016

© Самарский государственный
технический университет, 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. СТРУКТУРА ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	7
1.1 Понятие «Искусственный интеллект»	7
1.2 Этапы развития искусственного интеллекта	12
1.3 Классификация искусственного интеллекта	22
ГЛАВА 2. ЗАДАЧИ И МЕТОДЫ ИХ РЕШЕНИЯ	33
2.1 Задачи систем искусственного интеллекта	33
2.2 Общие способы решения задач	36
ГЛАВА 3. ИНТЕГРИРОВАННАЯ СРЕДА ПРОГРАММИРОВАНИЯ БАЗ ЗНАНИЙ VISUAL PROLOG	43
3.1 Описание предметной области с помощью программы на языке Пролог	43
3.2 Факты, правила и целевые утверждения	48
3.3 Структура программы на языке Visual Prolog	56
3.4 Реляционный язык Пролог	60
ГЛАВА 4. ОПИСАНИЕ И ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЭКСПЕРТНЫХ СИСТЕМ	64
4.1 Основные элементы экспертной системы и порядок работы с ней	64
4.2 Примеры использования экспертных компьютерных систем	69
ЗАКЛЮЧЕНИЕ	76
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	78

ВВЕДЕНИЕ

Практически любой созданный, а точнее выпущенный на рынок информационный или технический объект объявляется интеллектуальной системой, правда, в основном в СМИ, рекламных материалах. В действительности это отчасти мода, а отчасти широкое научное и практическое осознание интеллектуальности как одной из важных характеристик окружающего нас мира.

Примерно в 70-е годы прошлого столетия — начале фазы компьютерной революции был совершен концептуальный прорыв в новой области информатики и вычислительной техники, названной искусственным интеллектом. В эти годы была принята новая концепция, которая утверждала, что эффективность программы при решении задачи зависит от знаний, которыми она обладает, а не только от формализмов и методов вывода, которые она использует.

Наиболее значительными работами в области искусственного интеллекта являются разработки мощных компьютерных систем или экспертных систем, т.е. систем основанных на знаниях. Такие программы решения задач с представлением и применением фактических и эвристических знаний, совместной работой экспертов и инженеров по знаниям, разработчиков систем и логическим выводом позволяют переходить к новым информационным технологиям, к новой технологии программирования [5].

В настоящее время идет бурное развитие интеллектуальных систем, интеллектуальных концепций и технологий. Дисциплины, связанные с системами искусственного интеллекта, появились в связи с тенденциями образовательного процесса в сферах практической деятельности, связанных с решением задач интерпретации, диагностики, мониторинга, прогнозирования, планирования, проектирования, обучения, управления для плохо формализуемых проблем и зашумленных данных (знаний) при ограниченных ресурсах. Современный подход к решению таких проблем базируется на методах искусственного интеллекта.

Дисциплина «Системы искусственного интеллекта» играет важную роль в подготовке бакалавров по направлению 09.03.01 «Информатика и вычислительная техника». Различные учебные курсы, относящиеся к проблематике искусственного интеллекта (ИИ), включены в учебные планы многих институтов, университетов по рекомендациям международных ассоциаций ACM, AIS, AITP в 1997 г.

Настоящее учебное пособие «Системы искусственного интеллекта» посвящено вопросам организации, проектирования, разработки и применения систем, предназначенных для обработки информации, базирующихся на применении базовых принципов, подходах, методах и стратегиях разработки систем обычного назначения.

В настоящее время имеется достаточно обширная литература по системам искусственного интеллекта и программам в «демонстрационном варианте», т. е. при относительно небольшой базе знаний. Но для настоящей предметной области требуется весь арсенал средств и методов, накопленных за последние 40 лет. Разработка этого учебного пособия и есть попытка представления методов ИИ для серьезных практических областей: информатики, экономики, машинного перевода, распознавания образов и т. д.

Учебное пособие ставит цель: познакомить читателя с принципами создания и функционирования систем искусственного интеллекта; приобрести студентами компетенции связанные с реализацией систем основанных на логических рассуждениях, механизмах вывода.

Авторы надеются, что предложенное изложение подходов, методов и моделей искусственного интеллекта привлечет множество различных специалистов по многим приложениям, имеющих отношение к знаниям, логическому выводу, технологиям, что вызвано реальной потребностью практики.

Структурно учебное пособие состоит из четырех глав.

В первой главе рассмотрена структура исследования в области искусственного интеллекта: понятие искусственного интеллекта,

этапы развития искусственного интеллекта; классификация нейробионического и информационного направлений; примеры различных классификаций систем искусственного интеллекта.

Во второй главе представлены задачи и общие методы решения хорошо определенных задач.

В третьей главе рассматривается планирование в интеллектуальных системах. Изложены классификация планирования, методы планирования в пространстве состояний, редукции задачи, ключевых состояний и ключевых операторов, анализа средств и целей, а так же примеры планирования систем.

В четвертой главе приводятся описание и примеры разработки экспертных систем медицины и химии, экспертных систем электроники и экспертных компьютерных систем.

В конце каждой главы учебного пособия приведен перечень контрольных вопросов.

Учебное пособие в большей степени основано на материалах отечественных и в меньшей — зарубежных работ, являясь обобщением опыта, накопленного в этой области. Из отечественных работ следует отметить огромный вклад следующих специалистов: Поспелова Д. А., Гавриловой Т. А., Хорошевского В. Ф., Тельнова Ю. Ю., Попова Э. В., Гаскарова Д. В., Андрейчикова А. В., Норенкова И. П., Стефанюка В. Л., Рыбиной Г. В., Вагина В. Н., Аверина В. Н. и др.

Глава 1. СТРУКТУРА ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1.1 Понятие «Искусственный интеллект»

Исследование и создание машин, обнаруживающих поведение, которое у людей называют интеллектуальным, назовем искусственным интеллектом. На сегодняшний день самыми распространенными машинами являются компьютерная техника и средства коммуникации, следовательно, направление искусственного интеллекта относится к области компьютеров и вычислительных систем. Слово «интеллект» происходит от латинского слова «intelligentie», которое, в свою очередь, образовалось от глагола «intelligere», означающего способность понимать, определять смысл.

Строгого определения «искусственного интеллекта» не существует. И. Рич определяет искусственный интеллект как область исследования, направленную на создание компьютеров, которые выполняют такие функции, которые в настоящий момент человек выполняет лучше. К таким функциям, которые проявляются у человека, относят восприятие, анализ, рассуждение, использование знаний, планирование действий, логический вывод и т. д. Очень близкое определение искусственному интеллекту дает Дж. Аллен: «искусственный интеллект – это наука о создании машин, решающих задачи, которые могут решать люди...». Здесь в фокусе искусственного интеллекта оказываются те задачи, которые успешно решаются человеком и плохо – компьютерами. Эти два определения сопоставляют возможности человека и машин. Еще в 1950 году был предложен эмпирический тест Тьюринга А. для определения уровня интеллектуальности машин. В соответствии с тестом эксперт мог вступать в диалог либо с компьютером, либо с человеком. Тьюринг считал поведение компьютера интеллектуальным, если в диалоге участвовал компьютер, а эксперт был не в состоянии определить, с кем он ведет диалог. В дальнейшем стали считать, что машинный

интеллект отличается от человеческого интеллекта и, вероятно, попытка уподобления его естественному интеллекту ошибочна. Важность теста Тьюринга очевидна для оценивания качества современных программ искусственного интеллекта, но он отвлекал научные силы от решения основной задачи искусственного интеллекта – разработки общей теории машинного интеллекта, и использования этой теории для разработки интеллектуальных систем, решающих практические задачи.

Хотя каждый из нас имеет достаточно определенное субъективное представление о том, что следует понимать под человеческим интеллектом, имеется множество определений искусственного интеллекта, предложенных в последнее время.

1. Способность успешно реагировать на любую, особенно новую ситуацию путем надлежащих корректировок поведения.

2. Способность понимать взаимосвязи между фактами действительности для выработки действий, ведущих к достижению поставленной цели.

3. Разработка новых моделей и методов решения задач, традиционно считавшихся интеллектуальными и не поддававшихся ранее формализации и автоматизации.

4. Разработка новых технологий программирования и переход на новые архитектуры ЭВМ.

5. Способность решать прикладные задачи, для которых ранее создаваемые системы были непригодны.

6. Область компьютерной науки, занимающейся автоматизацией разумного поведения.

7. Формализованные методы, позволяющие решать с помощью компьютера задачи управления не хуже, чем естественный интеллект.

8. Новые идеи, подходы решения задач на ЭВМ – новая технология программирования, переход на параллельные машины.

9. Раздел информатики, посвященный моделированию интеллектуальной деятельности человека.

10. Ветвь информатики, которая связана с автоматизацией интеллектуального поведения.

11. Наука о вычислениях, которые делают возможными восприятие, логический вывод и действие.

12. Информационная технология, связанная с процессами логического вывода, обучения и восприятия.

13. Одно из направлений информатики, целью которого является разработка компьютерных систем, способных выполнять функции, традиционно считавшиеся интеллектуальными, – понимание языка, логический вывод, использование накопленных знаний, обучение, планирование действий и т. д.

14. Наука, поставившая своей целью изучение и моделирование атрибута человека. Какова природа мышления? Какие процессы происходят в нашем организме, когда мы думаем, чувствуем, видим, понимаем? Возможно ли в принципе понять, как работает наш мозг, и заставить мыслить неживую природу?

15. Наука, ставящая своей целью создание искусственных систем, достаточно хорошо имитирующих интеллектуальную деятельность человеческой личности, способностям которой мы не перестали удивляться.

Известный британский специалист А. Эндрю особое внимание уделил биологическим и биофизическим проблемам и моделям искусственного интеллекта; Д. Хофштадтер указал на тесную связь искусственного интеллекта с фундаментальной математикой, живописью и классической музыкой; недавно изданная в Нью-Йорке книга Т. Мунаката представляет нейронные сети и генетические алгоритмы, обычно рассматриваемые в основном направлении искусственного интеллекта лишь как вспомогательные технические средства [9].

Итак, на сегодняшний день не существует определения, которое однозначно описывало бы искусственный интеллект. Среди многих точек зрения доминируют следующие три.

1. Фундаментальные исследования, в процессе которых разрабатываются новые модели и методы для решения задач,

считающихся интеллектуальными и не поддававшихся ранее формализации и автоматизации.

2. Исследования, связанные с новыми идеями решения задач на ЭВМ, с разработкой новых технологий программирования и переходом к компьютерам не фон-неймановской архитектуры.

3. Исследования, в процессе которых появляется множество прикладных систем, способных решать задачи, для которых ранее создаваемые системы были не пригодны.

Под интеллектом обычно понимается возможность ставить и достигать цели при изменяющихся обстоятельствах, способность выбирать из множества целей те, которые скорее ведут к желаемому состоянию, адаптация к изменениям в среде и внутренним состояниям путем изменения их изменений [10].

Искусственный интеллект решает сложные задачи: многофункциональные, интегрированные и интеллектуальные, где интеллектуальность напрямую связана с использованием не только синтаксической, но и семантической и прагматической информации.

Интеллектуальность системы (устройства) в «докреативный» период их развития большинство авторов обычно связывают с осуществлением процедур анализа окружающей обстановки, влияющей на эффективность принимаемых решений [1].

В настоящее время интеллектуальность уже связывают с возможностью эффективного действия в условиях, «неожиданных» для системы. Такие системы все чаще стали называть креативными, то есть творческими.

А теперь дадим определение системы искусственного интеллекта. Система искусственного интеллекта – это компьютерная, креативная система (многофункциональная, интегрированная, интеллектуальная) со сложной структурой, использующая накопление и корректировку знаний (синтаксической, семантической, прагматической информации) для постановки и достижения цели (целенаправленного поведения), адаптации к изменениям среды и

внутреннего состояния путем изменения среды или внутреннего состояния.

Сегодня практически любой созданный, а точнее выпущенный на рынок, информационный или технический объект объявляется интеллектуальной системой, правда, в основном в СМИ и рекламных материалах. Иногда в ведущих научных организациях представляются интеллектуальные системы (объекты) с усложнением структуры обычных систем, то есть систем с большими функциональными возможностями. Например: интеллектуальный датчик или измерительный датчик принято считать интеллектуальным, если в месте расположения измерительного преобразователя имеется микропроцессорное устройство, осуществляющее фильтрацию и другие простые операции по предварительной обработке данных непосредственно там, где установлен датчик. То же самое наблюдается в нанотехнологии. Здесь термином интеллектуальный материал называется любой материал, который можно отнести к классу динамических – он может менять свои базовые свойства или структуру, основываясь на анализе внешней обстановки. Такие интеллектуальные материалы применяются, например, в технологии Steals, поскольку они меняют свой цвет, рассеивающие и поглощающие свойства и т. п. в ответ на внешние условия и команды летчика. Гаскаров Д. В. вводит три определения для интеллектуальных систем.

1) Интеллектуальная система – это информационно-вычислительная система с интеллектуальной поддержкой при решении задач без участия оператора (ЛПР).

2) Интеллектуализированная система – это информационно-вычислительная система с интеллектуальной поддержкой при решении задач с участием ЛПР.

3) Система с интеллектуальной поддержкой — система, способная самостоятельно принимать решения [3].

Другими словами: способность системы получать и анализировать информацию, понимать ее и делать новые выводы,

формулировать заключения, то есть «мыслить», помогая естественному интеллекту – человеку, который, корректируя, улучшает принятое интегрированное решение.

Под системой искусственного интеллекта будем понимать аппаратно-программный комплекс, обладающий способностью:

- к накоплению и корректировке знаний на основе активного восприятия информации о мире и обобщенного опыта;
- к целенаправленному поведению на основе логического вывода и алгоритмов различной степени неопределенности представления и управления сложных систем.

Д. В. Гаскаров определяет интеллектуальную систему, как – информационно-вычислительную с интеллектуальной поддержкой при решении задач без участия лица, принимающего решения (ЛПР), а интеллектуализированную систему как информационно-вычислительную систему с интеллектуальной поддержкой при решении задач с участием оператора (ЛПР) [3].

Итак, интеллектуальная поддержка – это способность системы принимать самостоятельное решение на основе имеющихся математического, алгоритмического, программного и интеллектуального обеспечений для сложных целенаправленных, динамических, слабо формализуемых, адаптированных систем с самообучением и самоорганизацией.

1.2 Этапы развития искусственного интеллекта

Еще в 1200-х годах появились попытки создания искусственного человека и его разума. Изобретатель Раймонд Луллий сконструировал машину, состоящую из кругов, размеченных буквами и раскрашенных в разные цвета, которые символизировали различные понятия, элементы стихии, субъекты и объекты знания. Разнообразное их сочетание, приводили с помощью логических операций к выводу «формул знаний».

В 40-х годах 20 в. с появлением электронно-вычислительных машин искусственный интеллект обрел второе рождение.

Исследования в области искусственного интеллекта имеют такие цели, как:

- выяснение сущности естественного интеллекта (человеческого интеллекта);
- использование машинного интеллекта для преобразования новых знаний и для решения интеллектуальных задач.

Первую цель выполняют психологи. Они контролируют модель поведения человека при решении задач и затем ее корректируют.

Вторую цель выполняет исследователь. Он синтезирует интеллектуальное поведение системы независимо от методов, которыми пользуются люди.

В соответствии с этими целями искусственный интеллект разделился на два основных направления: нейрокибернетику и кибернетику «черного ящика».

Первое направление утверждает, что мыслить может только человеческий мозг и, соответственно, любая машина должна быть выполнена как человеческий мозг, то есть воспроизвести его структуру, принцип действия.

В конце 1950-х годов начали разрабатываться и создаваться первые нейросети и нейрокомпьютеры американскими учеными У. МакКаломом, У. Питтсом, Ф. Розенблаттом, представляющие и в настоящее время нейрокомпьютерное направление систем искусственного интеллекта.

Второе направление говорит не о принципе действия мыслящего устройства, а об адекватном моделировании его функционирования, то есть поисков алгоритмов решения интеллектуальных задач, другими словами, собственных моделей мышления, а не человеческих.

Становление искусственного интеллекта. Работы в области искусственного интеллекта начались с зарождения нейрокибернетики. Так как мозг человека состоит из множества

нервных клеток – нейронов, то исследователи пытались строить разумные машины, имитируя поведение коллектива нейронов. Основную идею нейрокибернетики можно сформулировать следующим образом. Единственный объект, способный мыслить, – это человеческий мозг. Поэтому любое мыслящее устройство должно быть обязательно выполнено по образу и подобию человеческого мозга, воспроизводить его структуру, его принцип действия. Таким образом, нейрокибернетика занимается аппаратным моделированием структуры мозга и его деятельности.

В 1943 году МакКолак У. и Питтс У. предложили модель формального логического нейрона который мог находиться в двух устойчивых состояниях. Хебб Д. в 1949 году разработал простое правило, позволяющее изменять веса связей между нейронами с целью их обучения. В 1951 году Минского М. и Эдмондс Д. разработали нейрокомпьютер, который содержал 40 нейронов.

Термин «искусственный интеллект» был предложен на семинаре Дартмутского колледжа (США) в 1956 году. Первые работы по ИИ проводились в Массачусетском технологическом институте под руководством Минский М. и Дж. Маккарти, в университете Карнеги-Меллона под руководством Саймона Г. и Ньюэлла А.. Они и считаются «отцами» искусственного интеллекта [11].

Эвристический поиск и доказательство теорем (1956–1969). Работы Саймона Г. и Ньюэлла А. по разработке программы «Логик-теоретик» были предназначены для доказательства теорем в исчислении высказываний. С помощью ее были доказаны ряд теорем, но поиск решений был не эффективен.

Ньюэлл А. и Саймон Г. после анализа методов решения приступили к синтезу общих методов поиска решений, разработав следующую программу «Универсальный решатель задач» или GPS (General Problem Solver). Она разрабатывалась с целью имитации процесса решения задач человеком и базировалась на идеях эвристического поиска. GPS основывалась на модели лабиринтного поиска. Согласно этому подходу решение интеллектуальной задачи

выполнялось путем перебора огромного количества вариантов, который представлялся в виде движения по лабиринту. Программа GPS могла настраиваться на предметную область. Для этого необходимо было задать структуру состояний задачи и операторы, преобразующие эти состояния. Решение задачи осуществлялось на основе поисковых алгоритмов в пространстве возможных решений по эвристическим правилам, которые направляли поиск к искомой цели. В настоящее время такая модель признается тупиковой и имеет ограниченное использование.

В этот период развития ИИ основные исследования и разработки были представлены Дж. Маккарти, Дж. Робинсоном, К. Грином, Д. Хеббом, Ф. Розенблаттом, М. Минским:

- разработка языка Лисп (Дж. Маккарти);
- идея представления знаний и логического вывода в системах искусственного интеллекта (Дж. Маккарти);
- разработка метода резолюции (Дж Робинсон);
- разработка вопрос-ответной системы на основе логического представления знаний (К. Грин);
- создание персептрона для распознавания образов (Ф. Розенблат);
- написание книги об ограниченных возможностях персептрона (М. Минский, С. Пейперт).

Таким образом, к концу 60-х годов основное внимание в области ИИ стало уделяться методам представления задач и поиску решений, в частности представлению задач в логической форме и автоматическому доказательству теорем на основе метода резолюций.

Представление знаний (1969–1979). 70-е годы прошлого столетия характеризуются следующими исследованиями.

1. Отказ от поиска универсального алгоритма мышления, а моделирование конкретных знаний экспертов. К концу 60-х годов было обнаружено, что для решения практически важных задач недостаточно одних знаний общего характера. Успешное решение

прикладных задач возможно только при наличии хорошо структурированных специальных знаний.

2. Решаются задачи понимания естественного языка и обработки изображений.

3. Предложено эволюционное программирование (моделирование) Дж. Голландом, то есть процесс моделирования человека заменялся моделированием процесса его эволюции. Впервые предложено решение сложных задач моделированием эволюции с помощью компьютерных алгоритмов (генетических алгоритмов).

Программа DENDRAL, разработанная в 1969 году Э. Фейгенбаумом, Б. Букхененом, Э. Лидербергом, содержала детальные сведения об области органической химии и помогала специалистам определять молекулярную структуру органических соединений по данным, полученным с помощью масс-спектрометра. Масс-спектрометр, разделяя молекулы на фрагменты, измеряет массу и электрический заряд каждого из фрагментов. Чтобы определять множество форм молекул, которые могут состоять из таких фрагментов, в программе использовались эмпирические знания химиков, представленные в форме правил «если-то». Это позволило резко сократить число предлагаемых вариантов решений.

DENDRAL была первой успешно реализованной программой, аккумулирующей знания экспертов. Такие программы получили название «экспертные системы». Они содержат большой объем практических знаний, что позволяет получать ответы (решения).

Далее Э. Фейгенбаум, Б. Букхенен, Э.Шортлифф разрабатывают экспертную систему MYCIN. Она содержит около 450 правил, позволяющих диагностировать инфекционные заболевания крови. MYCIN уже позволяет обрабатывать и получать правдоподобные заключения на основе неопределенных (ненадежных) знаний. С этой целью факты и сами правила в системе характеризовались числовой функцией принадлежности – коэффициентом уверенности (степени достоверности).

После удаления из системы MYCIN базы знаний была представлена оболочка EMYCIN (оставлена только логика управления правилами), которую можно было наполнять знаниями.

С этого момента искусственный интеллект перестал быть наукой и стал приносить практическую пользу. Огромный успех имела экспертная система PROSPECTOR (1979 г.), используемая в геологоразведке месторождений. С появлением экспертных систем бизнес в сфере интеллектуальных информационных технологий впервые становится рентабельным. В системе PROSPECTOR база знаний представлялась в виде семантической сети и система обеспечивала взаимодействие с пользователем на естественном языке.

Семантические сети были предложены в 1967 г. М. Куиллианом. Существенный шаг при решении задачи понимания естественного языка, сделал Виноград, который разработал программу SHRDLU (1968 г.) (перемещение с помощью естественно-языковых команд кубиков и пирамид). Дальнейшее совершенствование систем понимания естественного языка связано с именами Р. Шенка и В. Вудса. Шенк разработал программу, преобразующую входные естественно-языковые высказывания к элементарным концептам, которые можно было представить в памяти ЭВМ. Далее он разработал модель представления знаний: скрипты или сценарии. Наконец, в 1973 году В. Вудс создал систему LUNAR, которая позволяла геологам задавать вопросы на естественном языке относительно образцов пород, доставленных с Луны.

Расширение приложений систем искусственного интеллекта требовало развития моделей представления знаний. В 1973 году А. Колмероэ создает язык логического программирования Пролог, ставший популярным в США, Европе, России. В США создается язык PLANNER, поддерживающий предикатный уровень представления знаний. Теорию фреймов в 1975 году предложил М. Минский. Затем были разработаны языки для работы с фреймовыми моделями: FRL, KRL, GUS и т. д.

Коммерческий успех компьютерной индустрии 1979–1986. Первой интеллектуальной системой, нашедшей применение в промышленности, стала экспертная система K1, разработанная Мак-Дермотом в 1982 году. Система K1 применялась для конфигурации компьютерных систем семейства VAX. Коммерческая версия системы, разработанная корпорацией Digital Equipment совместно с университетом Карнегги-Меллона (США), получила название XCON. К 1986 году эта система позволяла корпорации экономить 70 млн долларов ежегодно. Кроме этого, применение системы сократило число ошибок с 30 % до 1 %.

В 1981 году Япония объявляет о начале проекта машин 5-го поколения, базирующихся на принципах искусственного интеллекта. Этот проект способствовал активизации исследований в области искусственного интеллекта во многих странах.

Начиная с 1985 года, экспертные системы, а затем и системы, воспринимающие естественный язык (ЕЯ-системы), а затем и нейронные сети стали активно использоваться в коммерческих приложениях.

Коммерческие успехи к фирмам-разработчикам систем искусственного интеллекта пришли не сразу. На протяжении 1960 – 1985 гг. успехи искусственного интеллекта касались в основном исследовательских разработок, которые демонстрировали пригодность СИИ для практического использования.

Этап разработки и становления интеллектуальных систем 1-го поколения (1986–1996). В рамках исследований по искусственному интеллекту сформировалось самостоятельное направление – экспертная система, или инженерия знаний. В задачу этого направления входят исследования и разработка программ (устройств), использующих знания и процедуры вывода для решения задач, кажущихся трудными для людей экспертов.

Огромный интерес к экспертным системам со стороны пользователей вызван по крайней мере тремя причинами:

- экспертные системы ориентированы на задачи, для которых отсутствуют или неизвестны алгоритмы их решения;

- экспертные системы позволяют пользователям, не знающим программирования, самостоятельно разрабатывать задачи, используя свой опыт и знания;

- экспертная система позволяет получать результаты, не уступающие, а иногда и превосходящие возможности людей-экспертов.

Этот период охватывает технологии разработки традиционных (простых) экспертных систем и начала разработки интегрированных экспертных систем. И те и другие в настоящее время объединяются в экспертные системы первого поколения.

Работы Дж. Хопфилда, Д. Румельхарте и Г. Хинтона по моделям нейронных сетей послужили толчком к лавинообразному росту применений таких моделей для решения практических задач. Помимо теоретических разработок, этому способствовало и появление мощных аппаратно-программных средств, позволяющих моделировать нейронные сети достаточного уровня сложности.

В этот период источником знаний являются эксперт с эмпирическими знаниями, а базы знаний составляли отдельные формы (модели): продукционные (в основном), фреймовые, семантические сети, решающие деревья. Логический вывод представлялся детерминированным и дедуктивным, а язык общения с пользователем – фразами и терминами жесткой конструкции прикладной области.

Этап разработки интеллектуальных систем II поколения (1996–2000). Этот этап характеризуется интегрированными и гибридными принципами построения систем, где основой являются БЗ с любыми функциями знаний (библиотек) и автоматическим извлечением их; дедуктивным, абдуктивным, индуктивным, нечетким логическим выводом; проблемно-ориентированным языком общения, и обработкой не только статической, но и динамической информации.

Предыдущие этапы развития искусственного интеллекта характеризуются разрозненными подходами, основанными на выделении частного свойства понятия «интеллект». Например:

1) принцип знание центризма обусловил развитие и господство в течение определенного периода когностивистских моделей искусственного интеллекта, в частности логических. В соответствии с такими моделями на первый план выдвигается способность интеллектуальной системы рассуждать, а действия (поведение) рассматриваются как нечто вторичное. Основное внимание при этом уделяется логическому выводу;

2) выделение в определении искусственного интеллекта функции обучения (адаптации) способствовало развитию коннекционистских (нейронные сети) и эволюционных моделей (генетические алгоритмы);

3) трактовка искусственного интеллекта с позиций способности к восприятию и коммуникации привела к развитию моделей понимания изображений и естественного языка.

Начиная с начала 90-х годов в искусственном интеллекте стали преобладать две основные тенденции: интеграция и децентрализация.

Интеграционные процессы проявились в разработке интегрированных и гибридных систем искусственного интеллекта, объединяющих в себе преимущества разнородных моделей, например нечеткие экспертные системы и нейронные сети. В таких интегрированных системах могут поддерживаться различные модели представления знаний, разные типы рассуждений, модели восприятия и распознавания образов.

Процессы децентрализации связаны с рассмотрением искусственного интеллекта с позиций коллективного поведения большого числа взаимодействующих между собой интеллектуальных агентов. При этом интеллект агента рассматривается как подсистема управления деятельностью в процессе взаимодействия с другими агентами.

В основе распределенного (децентрализованного) интеллекта лежит функционально-структурная единица — агент, способная:

- воздействовать на других агентов и самих себя;
- образовывать свои собственные цели;
- общаться с другими агентами;
- функционировать без прямого вмешательства со стороны любых средств и осуществлять самоконтроль (автономность);
- воспринимать часть среды своего функционирования;
- строить локальное представление среды;
- выполнять обязанности и оказывать услуги;
- самовоспроизводиться.

Важно отметить, что при решении конкретной задачи агенты образуют структурное сообщество, в котором наблюдается определенная кооперация между агентами.

Например, можно говорить об агентах, выполняющих простую передачу сообщений, о координирующих агентах, которые организуют взаимодействие в группе агентов, о поисковых агентах и др.

Таким образом, согласно коллективистской модели, основным объектом исследований искусственного интеллекта является сообщество неоднородных, взаимодействующих агентов, а основное содержание разработок связано с созданием интеллектуальных агентов, обладающих заданными свойствами, и вычислительных структур, поддерживающих взаимодействие агентов.

Начало XXI века характеризуется началом исследований и разработкой систем с интеллектуальным интерфейсом: самообучающиеся, адаптивные, а затем и гибридные системы искусственного интеллекта, объединяющие в себе возможности, представленные нейронными сетями и моделями представления знаний. Объединение нейронных сетей и баз знаний может выполняться различными способами. В простейших случаях нейронная сеть выполняет предварительную или завершающую обработку информации в системах, основанных на знаниях. В более

интересных случаях осуществляется встраивание нейронных сетей в базы знаний, и наоборот.

В настоящее время ведутся исследования и разработки систем гибридного интеллекта на основе адаптивного информационного взаимодействия коллективов людей и ЭВМ, т. е. взаимодействие естественных интеллектов в природе, обществе, технике, которые включают в себя системы искусственного интеллекта. Например, процесс совместной интеллектуальной деятельности студентов и ЭВМ как компонентов системы гибридного интеллекта есть процесс формирования абстрактного алгоритма решения управления данным объектом.

Такой процесс может осуществляться как переход коллектива операторов от индивидуальных по языку и субъективных по соотношению с объективной реальностью и между собой (интуитивных) отражений с использованием ассоциативных информационных моделей к психическим моделям, адекватным оперативной структуре задачи, и к возможности построения информационных моделей. Это и есть путь к гибридизации участвующих интеллектов или создания системы человек – машина – среда.

1.3 Классификация искусственного интеллекта

Среди специалистов ИИ нет единой точки зрения на область искусственного интеллекта и даже на цели исследования. В настоящее время различают две точки зрения (направления) к моделированию искусственного интеллекта (AI – artificial intelligence): искусственный разум, направленный на моделирование внутренней структуры системы, и машинный интеллект, заключающийся в строгом задании результата функционирования.

Разделение работ по искусственному интеллекту на два направления связано с существованием двух точек зрения на вопрос, каким образом строить системы искусственного интеллекта.

Сторонники нейробионики моделируют искусственным образом процессы, происходящие в мозгу человека. Этот путь искусственного интеллекта исследует понимание механизмов восприятия, выявление способов работы мозга, создание технических средств для моделирования биологических структур и протекающих в них процессах. Эта точка зрения состоит в том, что именно изучение механизмов естественного мышления и анализ данных о способах формирования разумного поведения человека могут создать основу для построения систем искусственного интеллекта, причем построение это должно осуществляться, прежде всего, как моделирование, воспроизведение техническими средствами принципов и конкретных особенностей функционирования биологических объектов [13].

Практически это разработка элементов, подобных нейронам, объединение их в системы – нейросети, нейрокомпьютеры. В настоящее время эти технологии являются очень перспективными и быстро развивающимися.

Сторонники второй точки зрения убеждены, что «важнее всего результат», т.е. хорошее совпадение поведения искусственно созданных и естественных интеллектуальных систем, а что касается внутренних механизмов формирования поведения, то разработчик искусственного интеллекта вовсе не должен копировать или даже учитывать особенности естественных, живых аналогов.

Вторая, но доминирующая точка зрения на искусственный интеллект называется информационной, где основной целью является не построение технического аналога биологической системы, а адекватное моделирование функционирования системы, т. е. создания средств решения задач, традиционно считающихся интеллектуальными.

Первое направление искусственного интеллекта рассматривает данные о нейрофизиологических и психологических механизмах интеллектуальной деятельности и, в более широком плане, разумного поведения человека. Оно стремится воспроизвести эти механизмы с

помощью тех или иных технических устройств, с тем чтобы «поведение» таких устройств хорошо совпадало с поведением человека в определенных, заранее задаваемых пределах. Развитие этого направления тесно связано с успехами наук о человеке. Для него характерно стремление к воспроизведению более широкого, чем в машинном интеллекте, спектра проявлений разумной деятельности человека. Системы искусственного разума базируются на математической интерпретации деятельности нервной системы во главе с мозгом человека и реализуются в виде нейроподобных сетей на базе нейроподобного элемента – аналога нейрона.

Нейроподобные сети в последнее время являются одним из самых перспективных направлений в области искусственного интеллекта и постепенно входят в бытность людей в широком спектре деятельности.

Сети первой группы, такие, как сети обратного распространения ошибки, сети Хопфилда и др., используются для распознавания образов, анализа и синтеза речи, перевода с одного языка на другой и прогнозирования. Это вызвано такими особенностями сетей, как восстановление изображения по его части, устойчивость к зашумлению входного сигнала, прогнозирование изменения входов и параллельность вычислений. Также немаловажной характеристикой является способность функционировать даже при потере некоторой части сети.

Сети второй группы используются как системы управления в реальном времени несложных объектов. Это управление популярными в последнее время интеллектуальными агентами, исполняющими роль виртуальных секретарей. Особенности данной группы является появление некоторых внутренних стимулов, возможность к самообучению и функционированию в реальном времени [12].

И, наконец, сети третьей группы, являющиеся дальнейшим развитием предыдущих, представляют собой уже нейроподобные системы, и нацелены они на создание экзотических в настоящее

время виртуальных личностей, информационных копий человека, средой обитания которых является глобальная сеть Интернет. Данное направление только зарождается, но есть немалый шанс, что мы станем свидетелями ситуации рождения виртуальных людей, подробно описанной фантастами и режиссерами.

Сейчас в Интернете повсеместно можно встретить признаки зарождения подобных проектов, призывы объединиться всем научным потенциалом способного думать человечества в целях очеловечивания Интернета, преобразования его в разумную систему или среду обитания разумных систем. Раз существуют подобные предпосылки, значит, ничто не остановит полет человеческой мысли на пути достижения поставленной цели.

На основании вышеизложенного можно сделать вывод о том, что основные направления искусственного интеллекта связаны с моделированием, но в случае машинного интеллекта мы имеем дело с моделированием феноменологическим, имитационным, а в случае искусственного разума – с моделированием структурным.

Примеры различных классификаций систем искусственного интеллекта

Андрейчиков А. В. и. Андрейчикова О. Н предлагают следующие классы систем:

1. Интеллектуальная информационная система – одно из главных направлений в искусственном интеллекте. Целью такой системы является исследование и применение знаний высококвалифицированных экспертов для решения задач. Это имитация человеческого искусства анализа неструктурированных проблем, т. е. разработка моделей представления, извлечения и структурирования знаний. Частным случаем таких систем являются экспертные системы.

2. Естественнo-языковой интерфейс и машинный перевод. Это исследование методов и разработка систем, обеспечивающих

реализацию процесса общения человека с компьютером на естественном языке (системы Е-Я общения). Системы машинного перевода с одного языка на другой больших потоков информации – это интеллектуальная система, состоящая из базы знаний и сложных моделей, базирующихся на структурно-логическом подходе, включающих анализ и синтез естественно-языковых сообщения.

3. Генерация и распознавание речи. Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии.

4. Обработка визуальной информации. В таких системах решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений – это трансформирование графических образов в новые изображения. Задача анализа – это преобразование исходных изображений в данные другого типа (например, текст). И наконец, синтез изображений – получение графических объектов по алгоритмам построения изображения (машинная графика).

5. Обучение и самообучение. Это разработка моделей, методов и алгоритмов для систем автоматического накопления и формирования знаний с использованием процедур анализа и обобщения данных. Это системы Data-mining, Knowledge, Discovery и другие.

6. Распознавание образов. Система распознавания объектов осуществляет отнесение объектов к классам, а классы описываются совокупностями определенных значений признаков.

7. Игры и машинное творчество. Машинное творчество — это музыкальные программы, машинные программы живописи и графики, стихов, прозаических произведений, изобретение новых объектов, интеллектуальные компьютерные игры.

8. Программное обеспечение систем искусственного интеллекта. Инструментальные средства для разработки систем искусственного интеллекта включают специальные языки программирования, ориентированные на обработку символьной

информации (LISP, SMALLTALK), языки логического программирования (PROLOG), языки представления знаний (OPS-5, KRL, FRL), интегрированные программные среды, содержащие арсенал инструментальных средств (KE, ARTS, GURU, G2), а также оболочки экспертных систем (BUILD, EMYCIN, EXSYS, Professional, ЭКСПЕРТ).

9. Новые архитектуры компьютеров. Это создание компьютеров не фоннеймовской архитектуры, ориентированные на обработку символьной информации. Известны параллельные и векторные компьютеры, но весьма высокой стоимости в настоящее время.

10. Интеллектуальные роботы. В настоящее время используются программируемые манипуляторы с жесткой схемой управления [1].

Гаскаров Д. В. предлагает следующие наиболее распространенные типы классификации интеллектуализированных систем.

1. По степени реализации (стадии существования) рис. 1.6.

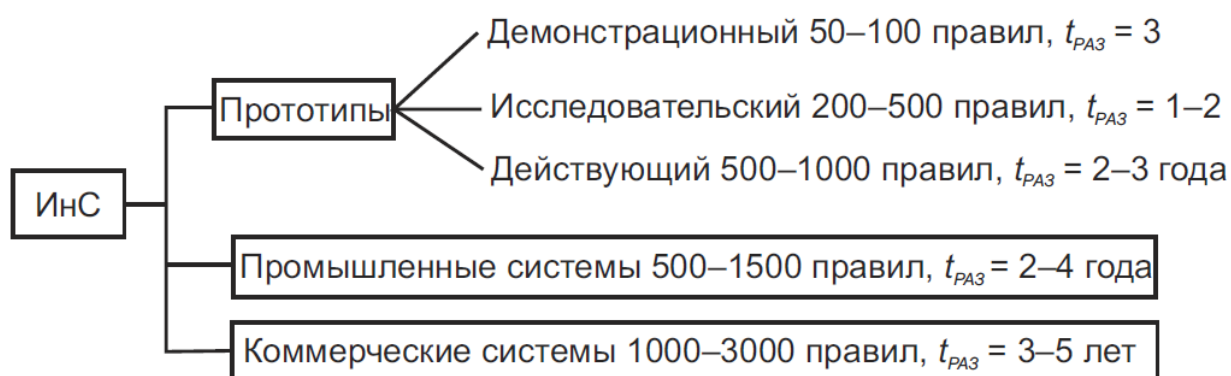


Рис. 1.6 – Классификация ИИС по степени реализации

2. По степени сложности (рис. 1.7).

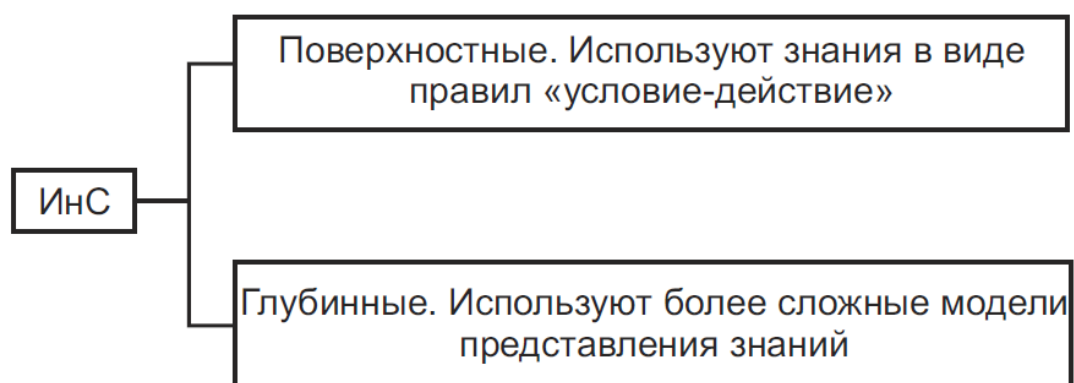


Рис. 1.7 – Классификация ИнС по степени сложности

3. По степени интеграции (рис. 1.8).

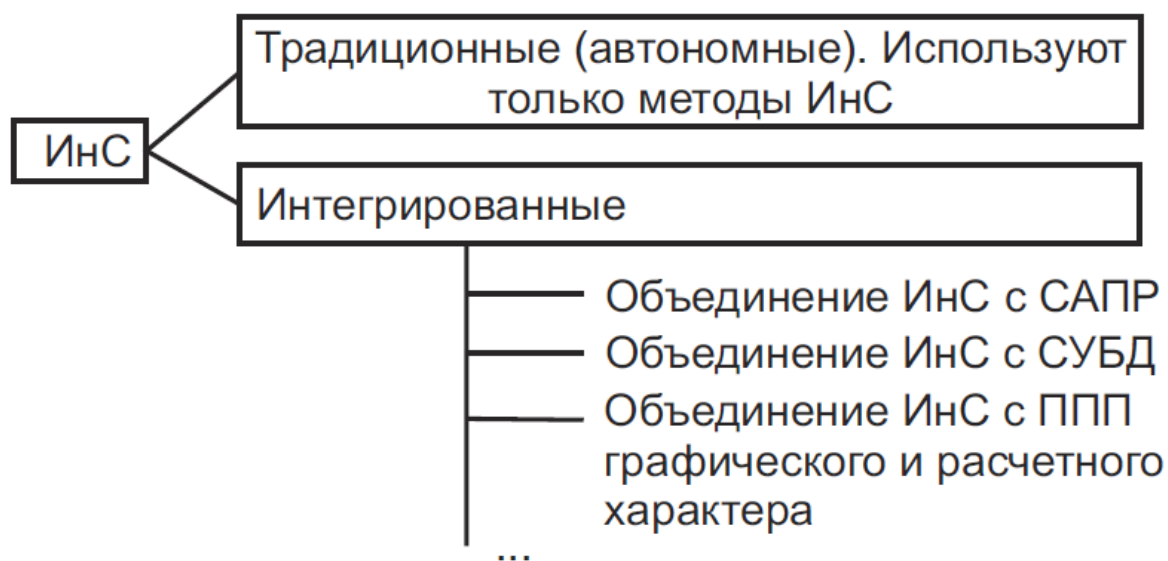


Рис. 1.8 – Классификация ИнС по степени интеграции

4. По сложности и типу ЭВМ (рис. 1.9).

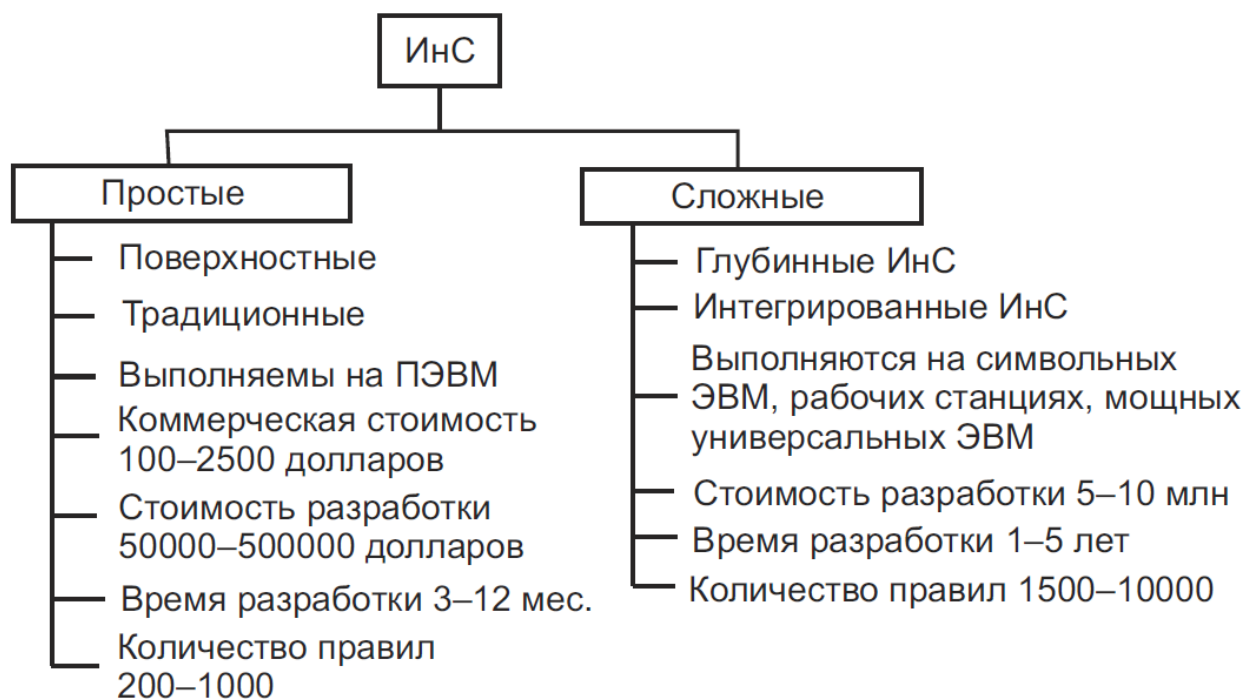


Рис. 1.9 – Классификация ИИС по сложности и типу ЭВМ

5. По типу области экспертизы (типу предметной области, рис. 1.10).

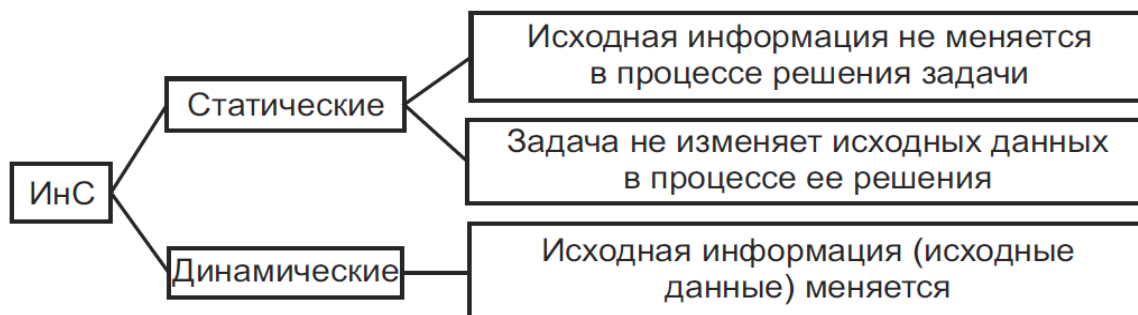


Рис. 1.10 – Классификация ИИС по типу области экспертизы

6. По эволюции развития (рис. 1.11).

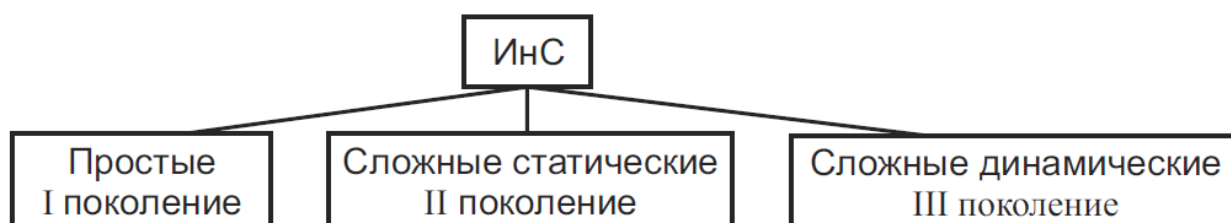


Рис. 1.11 – Классификация ИИС по эволюции развития

7. По типам решаемых задач (рис. 1.12).

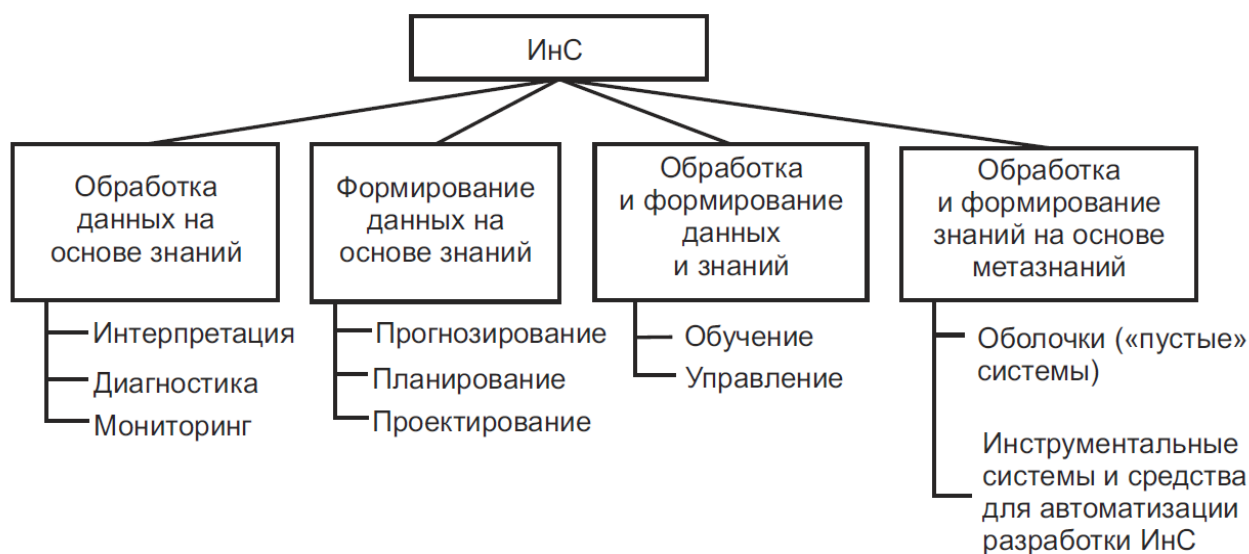


Рис. 1.12 – Классификация ИнС по типам решаемых задач

Ясницкий Л. Н. классифицирует системы искусственного интеллекта по следующим направлениям:

1. Системы, основанные на знаниях.
2. Нейросетевые и нейрокомпьютерные технологии.
3. Распознавание образов.
4. Игры и творчество.
5. Компьютерная лингвистика.
6. Интеллектуальные роботы.
7. Компьютерные вирусы.
8. Интеллектуальное компьютерное моделирование [16].

В. Н. Бондарев, Ф. Г. Аде считают основными направлениями исследований в искусственном интеллекте следующие:

1. Представление задач и поиск решений.
2. Доказательство теорем.
3. Представление знаний.
4. Экспертные системы.
5. Обучение и выявление закономерностей.
6. Общение на естественном языке.
7. Распознавание образов.

8. Компьютерное зрение.
9. Языки программирования систем искусственного интеллекта [2].

Таким образом, анализ систем (направлений) искусственного интеллекта показал, что в основном авторы вышепредставленных классификаций повторяют направления Д. А. Поспелова с небольшими изменениями, указывающими на приоритеты авторов и на развитие систем искусственного интеллекта в последнее время.

Проанализировав вышеприведенные классификации, мы можем предложить следующие направления искусственного интеллекта.

1. Экспертные системы I поколения (простые, сложные) и II поколения (интегрированные, многофункциональные, интеллектуальные, креативные (творческие) и гибридные).

2. Интеллектуальные производственные системы: вопрос-ответные системы, расчетно-логические системы, интеллектуальные САПР, САНИ, АСУ, СППР.

3. Нейросети и нейрокомпьютеры (нейросетевые и нейрокомпьютерные технологии).

4. Построение и автоматизация построения БЗ, анализ, обработка и выявления знания.

5. Обучение и самообучение (консультационные системы, интеллектуальные тренажеры, системы школьного и вузовского образования).

6. Эволюционное моделирование (генетические алгоритмы, классифицирующие системы, генетическое программирование, эволюционное программирование, эволюционные стратегии)

7. Системы машинного перевода.

8. Системы Е-Я общения (ведение диалога, понимание-преобразование высказываний с Е-Я на внутренний язык, обработка высказываний – формулирование выходных высказываний на Е-Я).

9. Системы речевого общения (синтез (текст-смысл), анализ и распознавание речи (смысл-текст)).

10. Системы обработки визуальной информации (обработка, анализ и синтез изображений).

11. Системы распознавания образов.

Иногда в литературе системы машинного перевода и системы Е-Я общения объединяются в одно направление – компьютерную лингвистику. И последнее – авторы считают целесообразным выделить интеллектуальные системы управления, наиболее актуальную часть систем, основанных на знаниях (статические, динамические, реального времени).

Можно выделить и менее значимые направления систем искусственного интеллекта:

1. Игры и машинное творчество.
2. Архитектуры компьютеров.
3. Компьютерные вирусы.
4. Интеллектуальное математическое моделирование.
5. Интеллектуальные роботы.
6. Системы с интеллектуальной обратной связью и интеллектуальными интерфейсами.
7. Многоагентные системы.

Контрольные вопросы и задания к главе 1

1. Проведите анализ представленных определений искусственного интеллекта.
2. Сформулируйте определение искусственного интеллекта, данное Д. А. Поспеловым.
3. Какие сложные задачи решает искусственный интеллект?
4. Проведите сравнение интеллектуальных систем в докреативный и креативный периоды их развития.
5. Представьте определение СИИ.
6. Приведите примеры интеллектуальных систем.
7. Расскажите о трех определениях для интеллектуальных систем, представленных Гаскаровым Д. Б.

8. Дайте характеристику двух целей искусственного интеллекта.
9. Назовите два основных направления искусственного интеллекта.
10. Расскажите о становлении искусственного интеллекта.
11. Проведите анализ эвристического поиска и доказательства теорем при решении задач.
12. Опишите представление знаний в интеллектуальных системах.
13. Расскажите об этапе разработки и становления интеллектуальных систем I поколения.
14. Расскажите об этапе разработки и становления интеллектуальных систем II поколения.
15. Сравните две точки зрения на область искусственного интеллекта.
16. Опишите нейросети трех групп.
17. Представьте классификацию систем нейронной обработки.
18. Представьте классификацию нейронных систем по типу входных и выходных сигналов.
19. Представьте классификацию интеллектуальных нейронных систем по параметрам управления.
20. Классифицируйте нейросети по типу связей и типу обучения.
21. Какие предметные области используют нейросетевые задачи?
22. На какие три части делится информационное направление (вторая точка зрения на искусственный интеллект)?
23. Опишите программы решения интеллектуальных задач в информационном направлении.
24. Опишите системы, основанные на знаниях в информационном направлении.
25. Опишите интеллектуальное программирование в информационном направлении.

26. Что вы понимаете под когнитивной графикой?
27. Представьте функциональное содержание интерактивной компьютерной графики.
28. Приведите примеры различных классов систем искусственного интеллекта.
29. Дайте характеристику классификации Д. В. Гаскарова систем искусственного интеллекта.
30. Сформулируйте классификацию Л. Н. Ясницкого, В. П. Бондарева, Е. В. Луценко.

Глава 2. ЗАДАЧИ И МЕТОДЫ ИХ РЕШЕНИЯ

2.1 Задачи систем искусственного интеллекта

Задачи систем искусственного интеллекта охватывают самые разные предметные области, среди которых лидируют бизнес, производство, медицина, проектирование и системы управления. Все задачи можно классифицировать по следующим общим основаниям.

1. Задачи анализа и синтеза. В задаче анализа задана модель сущности и требуется определить неизвестные характеристики модели. В задаче синтеза задаются условия, которым должны удовлетворять характеристики «неизвестной» модели сущности, и требуется построить модель этой сущности.

2. Статические и динамические. В статических задачах явно не учитывают фактор времени и/или не изменяют знания об окружающем мире в процессе своих решений.

3. Использование общих утверждений для представления знаний, не содержащих явных ссылок на конкретные сущности, которые необходимо определить.

4. Использование частных ссылок, содержащие ссылки на конкретные сущности (объекты).

По типу решаемой задачи различают следующие задачи:

- интерпретация: процесс определения смысла данных (построение описаний по наблюдаемым данным);
- диагностика: процесс соотнесения объекта с некоторым классом объектов и/или обнаружение неисправностей в системе (отклонение параметров системы от нормативных в технике и в живых организмах);
- мониторинг: непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы;

– прогнозирование: построение планов действий объектов, будущих событий на основе моделей прошлого и настоящего. В прогнозирующих системах часто используют динамические модели, в которых значения параметров «подгоняются» под заданную ситуацию. Выводимые из этих моделей следствия составляют основу для прогнозов с вероятностными оценками;

– планирование: конструирование плана действий объектов способных выполнять некоторые функции, т. е. программы действий. Оно основано на моделях поведения реальных объектов, которые позволяют проводить логический вывод последствий планируемой деятельности;

– проектирование: разработка ранее не существовавшего объекта и подготовка спецификаций на создание объектов с заранее определенными свойствами. Степень новизны может быть разной и определяется видом знаний и методами их обработки;

– обучение: диагностика, интерпретация, планирование, проектирование. Системы обучения выполняют такие функции, как диагностика ошибок, подсказывание правильных решений, аккумуляция знаний о гипотетическом «ученике» и его характерных ошибках, диагностирование слабости в познаниях обучаемых и нахождение соответствующих средств для их ликвидации. Системы обучения способны планировать акт общения с учеником;

– управление: интерпретация, прогноз, планирование, моделирование, оптимизация выработанных решений, мониторинг, т.е. функция системы, поддерживающая определенный режим ее функционирования или управления поведением сложной системы в соответствии с заданными спецификациями;

– отладка, ремонт: выработка рекомендаций по устранению неисправностей;

– поддержка принятия решений – совокупность процедур, обеспечивающих ЛПР необходимой информацией и рекомендациями

для процесса принятия решений (выбор и/или, генерация альтернатив).

Задачи интерпретации данных, диагностики, поддержки принятия решений относятся к задачам анализа, задачи проектирования, планирования и управления – к задачам синтеза. К комбинированному типу задач относятся обучение, мониторинг и прогнозирование.

Термин «решение задач» (problem solving) употребляется в искусственном интеллекте в ограниченном смысле. Речь идет о хорошо определенных задачах, решаемых на основе поисковых алгоритмов.

Задача считается хорошо определенной, если для нее имеется возможность задать пространство возможных решений (состояний), а также способ просмотра этого пространства с целью поиска конечного (целевого) состояния, соответствующего решаемой задаче. Поиск конечного состояния задачи заключается в применении к каждому состоянию алгоритмической процедуры с целью проверки, не является ли это состояние решением задачи. Данная процедура продолжается до тех пор, пока не будет найдено решение.

Примерами хорошо определенных задач являются: доказательство теорем, поиск маршрута на графе, планирование робота в среде с препятствиями и т. д.

Человек обычно не решает задачу в той форме, в которой она изначально формулируется. Он стремится представить задачу таким образом, чтобы ему удобно было ее решать. Для этого он выполняет преобразование исходного представления задачи с целью сокращения пространства, в котором необходимо выполнять поиск решения задачи. Этап выбора подходящей формы представления задачи настолько обыден, что мы часто не осознаем его важности. В то же время форма или способ представления задачи в значительной мере определяет успех ее решения.

При выборе способа представления задачи обычно учитывают два обстоятельства: представление задачи должно достаточно точно

моделировать реальность; способы представления должны быть такими, чтобы решателю задач было удобно с ним работать.

Поскольку под решателем задач в искусственном интеллекте понимается компьютер, то мы и рассмотрим способы представления задач, удобные для их решения на ЭВМ. К ним относятся следующие наиболее часто используемые способы (формы):

- представление задач в пространстве состояний;
- представление, сводящее задачу к подзадачам;
- представление задач в виде теорем.

Данные представления и соответствующие универсальные методы поиска решений разрабатывались преимущественно на начальных этапах развития искусственного интеллекта. Позже было замечено, что для решения многих практических задач одних универсальных стратегий недостаточно. Необходим также большой объем знаний и наличие практического опыта. Исследование в области искусственного интеллекта сосредоточились на представлении и приобретении знаний. Однако это не снизило значимости разработанных стратегий поиска решений, так как они представляют некоторые общие схемы управления механизмом вывода систем, основанных на знаниях. Рассматриваемые ниже способы представления задач и методы поиска их решений играют важную роль во многих системах искусственного интеллекта, включая экспертные системы, понимание естественного языка, доказательство теорем и обучение.

2.2 Общие способы решения задач

Процесс решения задачи, как правило, включает два этапа: представление задачи и поиск (перебор). Успех решения задачи в значительной мере определяется формой ее представления. Формы представления задачи могут быть различными и зависят как от природы самой задачи, так и от ее решателя.

Например, при вычислении интеграла человек стремится путем преобразований представить его так, чтобы воспользоваться его табличными интегралами.

Выбирая кратчайший маршрут движения, человек может воспользоваться схемным представлением возможных путей перемещения и оценками расстояний между промежуточными пунктами.

Характер человеческого мышления таков, что этап и форма представления задачи, которую он использует, не всегда им осознаются. Так, шахматист не может четко объяснить форму представления шахматной ситуации, позволяющую ему не перебирать все возможные варианты продолжения при поиске очередного хода. Поэтому этап представления задачи часто выпадает из поля зрения человека. Тем не менее важность этого этапа осознается сразу же при попытке построить программно реализуемый алгоритм решения задачи.

Поиск формы представления задачи, удобный для ее машинного решения, является трудно формализуемым творческим процессом. Можно выделить следующие употребительные формы: представление в пространстве состояний, представление путем сведения задачи к подзадачам, представление в виде теоремы, комбинированное представление.

Полное представление задачи в пространстве состояний включает описание всех состояний или только начальных, задание операторов, отображающих одни состояния в другие, и задание целевого состояния.

Возможны различные формы описания состояний задачи. В частности, могут быть использованы строки, векторы, матрицы и графы. Выбирая ту или иную форму описания состояний, следует позаботиться о том, чтобы применение оператора, преобразующего одно состояние в другое, оказалось достаточно простым.

Операторы могут быть заданы с помощью таблицы, связывающей каждое входное состояние с некоторым выходным. Для

больших задач такое задание оператора практически затруднительно. При описании состояния в форме строки (вектора) удобно задать оператор в виде правила переписывания.

Процедура поиска решения в пространстве состояний состоит в том, чтобы найти последовательность операторов, которая преобразует начальное состояние в целевое. Решением задачи будет указанная последовательность операторов. Деревом называется ориентированный граф, в каждую вершину которого входит только одна дуга, за исключением одной вершины, называемой корнем дерева. Таким образом, в дереве каждая вершина, за исключением корня, является концом ровно одной дуги и началом одной или нескольких дуг.

Вершины V_i порождаются вершиной V . V – родительская вершина, а V_i – дочерние вершины.

Примем, что корень находится на 0 уровне. Вершины, порожденные корнем, – 1 уровень и т. д.

Существуют два типа структур взаимосвязи подзадач: И-структуры и И-ИЛИ-структуры. В структурах типа И для решения основной задачи требуется решить все подзадачи. В структурах И-ИЛИ подзадачи разбиваются на группы, внутри которых они связаны отношением И, а между группами – отношением ИЛИ.

В этом случае для решения исходной задачи достаточно решить все подзадачи только какой-либо одной группы.

Для описания представления сведения задач к подзадачам можно использовать граф, называемый графом редукции задачи (рис. 2.1).

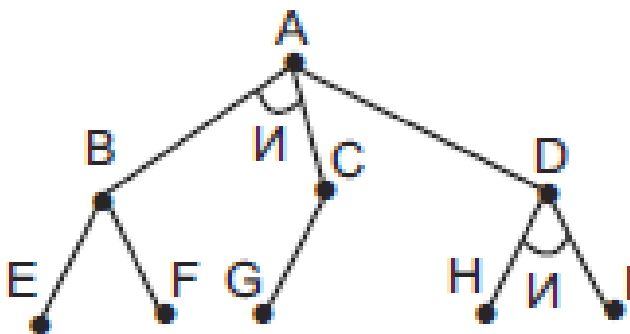


Рис. 2.1 – Дерево редукции задачи

При этом вершинам будут соответствовать задачи, а дугам – операторы редукции задач. Корню соответствует исходная задача, вершинам 1-го уровня – задачи, порожденные исходной задачей. Задача А может быть решена, если будут решены задачи В и С или задача D. Задача В будет решена, если будут решены задачи Е или F. Задача С – если решена G. Задача D – если решены задачи H и I.

Для указания связности вершин используется специальная кривая. Если имеются связанные вершины (дуги), то обычно, вводя при необходимости дополнительные вершины, дерево редукции задачи преобразуют так, чтобы каждая группа связанных вершин имела отдельную родительскую вершину (рис. 2.2).

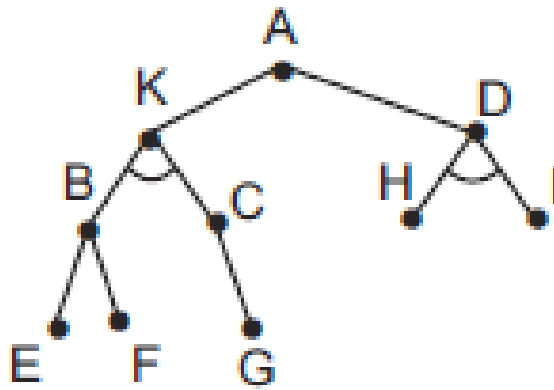


Рис. 2.2 – Преобразованное дерево редукции

Будем рассматривать только такие деревья редукции. Рассмотрим задачу о выборе маршрута, или задача о коммивояжере.

Транспортный робот должен построить маршрут так, чтобы побывать в каждом из n заданных пунктов в точности по разу и возвратиться в исходный пункт. При этом если таких маршрутов несколько, желательно выбрать такой, который имеет минимальную протяженность.

Состояния в этой задаче можно задавать строкой, обозначающей список пунктов, пройденных к текущему моменту.

Операторы – это отображения, соответствующие решению робота направиться из данного пункта в следующий. Целевым состоянием, если не требуется минимальная протяженность

маршрута, является любое состояние, описание которого начинается и кончается A и содержит названия всех других пунктов. Например, состояние ABCDA (рис. 2.3).

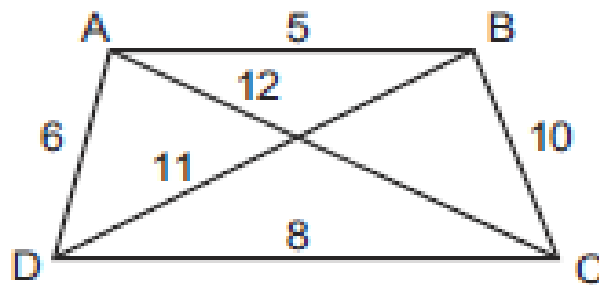


Рис. 2.3 – Состояние ABCDA

Представление задачи, сводящее задачу к подзадачам, предусматривает разбиение исходной задачи на множество подзадач, раздельное решение которых дает решение исходной задачи. Каждая из подзадач может, в свою очередь, быть также разбита на подзадачи. Число уровней разбиения теоретически не ограничено.

На практике разбиение продолжается до получения на нижнем уровне множества задач (подзадач), способ решения которых известен. Такие задачи условимся называть элементарными.

Введем понятия теории графов. Графом (или геометрическим графом) называется геометрическая конфигурация, состоящая из множества V точек, взаимосвязанных с множеством E непрерывных, самонепересекающихся кривых. Точки из множества V называются вершинами, кривые из множества E – ребрами. Если на всех ребрах задано направление, то граф называют ориентированным графом, а его ребра – дугами.

Если заданы два графа G_1 и G_2 , причем множества вершин и кривых графа G_2 являются подмножествами множеств вершин и кривых G_1 , то G_2 называют подграфом графа G_1 , G_1 – над графом графа G_2 . Структурно граф (дерево) редукции задачи, отличается от графа (дерева) состояния тем, что в нем имеются связанные дуги. Связанные вершины называют И-вершинами, несвязанные – ИЛИ-вершинами, а граф называется графом типа И-ИЛИ.

Вершины, соответствующие элементарным задачам называются заключительными. Вершины, не имеющие дочерних вершин и не являющиеся заключительными, называются тупиковыми. Тупиковым вершинам соответствуют задачи, которые в рамках данного представления неразрешимы.

Таким образом, заключительные вершины являются разрешимыми, а тупиковые – неразрешимыми.

Вершина, не являющаяся ни заключительной, ни тупиковой, будет разрешимой тогда и только тогда, когда все ее дочерние вершины разрешимы, если они являются связанными, или хотя бы одна из дочерних вершин разрешима, если они являются несвязанными.

Очевидно, задача А является разрешимой в том и только в том случае, если вершины Н и I являются заключительными или заключительными являются вершины G и F или G и E.

Граф редукции задачи может быть задан в явном виде. Но чаще он, как и граф состояния, задается в неявном виде посредством описания исходной задачи и операторов редукции.

Контрольные вопросы и задания к главе 2

1. Опишите классификацию задач искусственного интеллекта по общим признакам.
2. Представьте классификацию задач искусственного интеллекта по типу решаемой задачи.
3. Перечислите способы представления задач.
4. Какие два этапа необходимы для процесса решения задач?
5. Расскажите о формах описания состояний и операторов при поиске решения задачи.
6. Какие два типа структур взаимосвязи задач вы знаете?
7. Приведите примеры деревьев и преобразованных деревьев редукции задачи.

8. Какие особенности предметной области и требования позволяют правильно выбрать метод решения задачи?
9. Перечислите методы решения задач в одном пространстве.
10. Опишите поиск решения в пространстве состояний и приведите пример.
11. Опишите алгоритмы полного перебора (в ширину), в глубину, упорядоченного перебора решения задач в пространстве состояний.
12. Приведите примеры решения задач по разным алгоритмам в пространстве состояний.
13. Опишите поиск решения задач при редукции задачи на подзадачи.
14. Чем отличается поиск решений на графе редукции задачи от поиска на графе состояний?
15. Сформулируйте алгоритмы поиска решений при редукции задачи.
16. Представьте поисковое дерево редукции задачи для выбранного вами примера.
17. Как определяется эвристическая функция при поиске решения в пространстве состояний?
18. Как определяется эвристическая функция при поиске решения при редукции задачи на подзадачи?
19. Определите суммарную и максимальную стоимость, т. е. стоимость дерева решения в И-ИЛИ графе.
20. Рассмотрите пример дерева решения редукционного графа и определите оптимальное дерево решения для начальной вершины.
21. Перечислите методы поиска решения задачи в иерархических пространствах.
22. Дайте характеристику факторизованного пространства.
23. Опишите поиск решения задачи в фиксированном множестве пространств.
24. На чем базируется поиск решений задачи в изменяющемся множестве иерархических пространств.

25. Опишите поиск решения задачи в альтернативных пространствах.

26. Проанализируйте преимущества поиска решения задачи с использованием нескольких моделей.

Глава 3. ИНТЕГРИРОВАННАЯ СРЕДА ПРОГРАММИРОВАНИЯ БАЗ ЗНАНИЙ VISUAL PROLOG

3.1. Описание предметной области с помощью программы на языке Пролог

Одним из первых языков логического программирования для разработки СИИ является язык Пролог, который раскрывается от первых букв слов: ПРОграммирование на языке ЛОГики.

Пролог отличается от алгоритмических языков тем, что он не раскрывается в виде математических процедур за конечное число шагов. Наибольший интерес Пролог нашел в инженерии знаний и искусственного интеллекта и был базовым языком в японской программе ЭВМ 5-го поколения, ориентированы на исследование логики и искусственного интеллекта и разработку нового поколения компьютеров [7].

В Прологе программист непосредственно занимается задачей, а не способом поиска решений.

Основные отличия Пролога.

1. Наличие символьной логики, называемой исчислением предикатов. При этом мощными средствами в области логического программирования являются следующие механизмы:

- механизм вывода с поиском и возвратом;
- встроенный механизм сопоставления с образцом;
- выразительная структура данных с возможностями ее изменения.

2. Наличие единообразия программы данных. При этом данная программа - лишь две различных точки зрения на объекты Пролога. Кроме того, в единой базе данных можно свободно создавать и уничтожать отдельные элементы, так как не существует различия между программами и данными. Саму программу можно менять во время ее работы. При этом естественным методом программирования является рекурсия.

3. Многие интересные полезные свойства языка непосредственно вытекают из его декларативности, которая позволяет понимать программу, не отслеживая динамику ее выполнения.

4. Истинная мощь языка базируется на сочетании всех вышеуказанных свойств (отличий).

Результатом завершения работы на Прологе являются 2 состояния:

- 1) доказано;
- 2) не доказано.

Причем доказанное утверждение является истинным, но необязательным, так как доказательство зависит от известных фактов и сделанных на их основе выводов.

Факт - некоторое утверждение, причем все факты определяются как доказанные.

Вывод - определение свойств из фактов. Причем, имея множество фактов, мы можем определять новые свойства описанных в них объектов.

Пример:

«Рекс-это собака»: это факт.

«Голди - родитель Рекса»: это факт.

«Джон является родителем Рекса»: это факт.

Вывод. Объект является собакой при условии, что он является родителем собаки. Это формализованный вывод, называемый правилом. Если спросить, какие объекты являются собаками, то получим ответ, что Рекс, Голди и Джон - собаки, при этом Рекс - собака по определению, а Голди и Джон - вследствие вывода.

В последние годы значительно возрос интерес к работам по искусственному интеллекту, особенно к практическому воплощению результатов этих работ в экспертных системах. Одной из наиболее важных задач, связанных с проектированием таких систем, является проблема представления знаний о предметной области. Исследования развиваются в двух направлениях: первое связано с

выбором формализованных моделей представления знаний; второе - с разработкой и внедрением языков представления знаний, обладающих необходимыми изобразительными средствами.

В настоящее время большую популярность приобрел язык логического программирования Пролог, основанный на самой старой модели представления знаний - исчислении предикатов первого порядка.

Пролог предоставляет средства для описания знаний о предметной области в виде фактов и правил вывода. Он понятен экспертам-непрограммистам и даже с успехом может использоваться для обучения детей программированию. Пролог способствует формированию стиля программирования, основанного на идеях модульного программирования и программирования «сверху вниз». В то время как традиционные языки программирования являются процедурно-ориентированными, Пролог основан на описательной или декларативной парадигме в программировании. Это свойство коренным образом меняет программистское мышление и делает обучение программированию на Прологе увлекательным занятием, требующим определенных интеллектуальных усилий.

Пролог применяется при создании приложений в следующих областях:

- разработка быстрых прототипов прикладных программ;
- разработка приложений, связанных с защитой информации;
- управление производственными процессами;
- создание динамических реляционных баз данных;
- перевод с одного языка на другой;
- реализация экспертных систем и оболочек экспертных систем.

Системы программирования на Прологе эксплуатируются на ЭВМ самых разных типов и приобрели широкую известность благодаря наличию таких развитых версий на персональных компьютерах как WinProlog, Visual Prolog, Strawberry Prolog и Arity Prolog [7].

Пролог - это язык программирования, предназначенный для представления и использования знаний о некоторой предметной области. Под предметной областью будем понимать множество рассматриваемых объектов и совокупность знаний о них. Любую взаимосвязь между объектами и (или) их свойствами назовем отношением. Так, для предметной области «Студенческая группа» характерными объектами будут студент, группа, факультет. Свойства объектов - это фамилия, имя, отчество студента, номер группы, наименование факультета. Связи между конкретным студентом и номером группы, между его оценками и размером стипендии, между номером группы и факультетом, специальностью, курсом обучения будут являться отношениями. Таким образом, предметная область - это объекты и отношения.

Объекты могут быть объединены в классы, обладающие определенными свойствами. Элементы этих классов являются конкретными объектами предметной области. Для представления конкретного объекта в программе на Прологе используется константа. Константа - это число или символическое имя объекта. *Например: «учебник», 345, 6.89, «Петров».*

Имена конкретных объектов, отношений, свойств образуются по определенным правилам, зависящим от версии (реализации) языка Пролог, и называются символическими именами. Далее будут рассмотрены правила, характерные для большинства версий Пролога.

Символическое имя (атом) - это неразрывная цепочка букв (в Visual Prolog латинских и русских), цифр и символа подчеркивания, начинающаяся со строчной латинской (русской) буквы.

Для именования объектов и их свойств могут использоваться строки - последовательности любых символов, заключенные в двойные кавычки. В дальнейшем для именования конкретных объектов будем чаще всего использовать строки.

Например: table 1, стол, stock_5, r5z, студент, машина, меню, «Закон», «список аргументов», «группа 343».

Для описания в программе некоторого объекта, принадлежащего определенному классу, используется переменная. Переменная в программе представляется своим именем. Имя переменной в Прологе - это цепочка латинских букв или цифр, начинающихся с прописной латинской буквы или символа подчеркивания. *Примеры имен переменных: D, Leda, X, Y, JP, Lokon, V2 и т. д.*

Чтобы описать отношение, необходимо указать его имя и перечислить либо классы объектов, либо конкретные объекты, связываемые этим отношением:

<имя отношения>(<имя объекта 1>,<имя объекта 2>,...,<имя объекта n>)

Отношение характеризуется именем и числом аргументов. Число аргументов равно числу классов объектов, связанных этим отношением. Для описания отношений в программе на Прологе используются предикаты. Предикат - это логическая функция от n аргументов, имеющая только 2 значения: «истина» и «ложь».

Синтаксис предиката:

<имя предиката>(<аргумент 1>,<аргумент 2>,...,<аргумент n>)

При описании отношения имя предиката совпадает с именем отношения, а аргументы предиката - это связываемые отношением объекты. Если описываемое предикатом отношение имеет место в предметной области, то предикат принимает значение «истина», если оно несправедливо для данной предметной области, то значение предиката «ложь».

В качестве имени предиката (отношения) в Visual Prolog допустимо использовать только символические имена.

Знания о предметной области выражаются на языке Пролог в виде предложений, называемых утверждениями (CLAUSES). Каждое утверждение заканчивается точкой и описывает какое-либо отношение, свойство, объект или закономерность. Структура утверждения проста и имеет одну из форм:

<заголовок>. / факт */*

или

<заголовок> :- <тело>. / правило */*,

где заголовок является предикатом и полностью характеризует описываемое отношение.

Тело утверждения состоит либо из одного предиката, либо из списка предикатов, разделенных знаками «,», «;», «not», соответствующими логическим операциям «и», «или», «не». Таким образом, тело утверждения является логическим выражением. Каждый входящий в это выражение предикат описывает какое-либо отношение. Знак «:-» соответствует слову «если». *Утверждение читается так: «Отношение, стоящее в заголовке, будет истинным, если истинно логическое выражение, находящееся в теле утверждения».*

Утверждения образуют программу. На языке Visual Prolog для них отводится специальный раздел, называемый CLAUSES.

Рассмотрим более подробно различные виды утверждений.

3.2 Факты, правила и целевые утверждения

Все утверждения программы на Прологе делятся на факты, правила и вопросы (или целевые утверждения).

Факты отражают текущее состояние предметной области, содержат конкретную информацию и являются истинными предикатами. Факты соответствуют простым безусловным высказываниям.

Рассмотрим пример описания меню в ресторане. Объекты предметной области - это блюда, которые можно съесть в ресторане, а одним из возможных видов отношений является классификация всех блюд на закуски, вторые мясные или рыбные блюда и десерты. Меню представляет собой небольшую базу знаний (БЗ), которая записывается в виде последовательности фактов следующим образом:

/ МЕНЮ */*

DOMAINS */* описание типов данных */*

name = string

FACTS */* описание динамической базы данных */*
 закуска(name)
 мясо(name)
 рыба(name)
 десерт (name) *
 CLAUSES */* утверждения (факты и правила) БЗ */*
/ Определение отношения закуска в виде фактов */*
 закуска("артишоки_в_белом_соусе").
 закуска("трюфели_в_шампанском").
 закуска ("салат_с_яйцом") .
/ Определение отношения мясо в виде фактов */*
 мясо("говяжье_жаркое") .
 мясо("цыпленок_в_липовом_цвете").
/ Определение отношения рыба в виде фактов */*
 рыба("окунь_во_фритюре").
 рыба("фаршированный_судак").
/ Определение отношения десерт в виде фактов */*
 десерт("грушевое_мороженое").
 десерт (,,"земляника_со_взбитыми_сливками") .
 десерт("дыня_сюрприз").

Эти факты вводят одновременно объекты и их классификацию (отношения). Например, факт *закуска("салат_с_яйцом")* показывает, что салат с яйцом является закуской [14].

Синтаксически правильно записанный факт имеет следующую структуру предиката:

<имя отношения> (<список аргументов>),

где в списке аргументов перечисляются имена объектов (не более 255), связанных данным отношением. Аргументы в списке отделяются друг от друга запятыми, в некоторых случаях могут отсутствовать. Если аргумент представляет собой имя конкретного объекта (свойства) или число, то он является константой Пролога.

<имя отношения> (<список аргументов>).

Структура предиката факт

Определяя с помощью фактов отношения между объектами, необходимо учитывать порядок, в котором перечисляются их имена внутри круглых скобок. Выбрав один раз какой-либо порядок, вы должны везде следовать ему и далее. Например, факт является *_отцом("Петр", "Иван")* означает, что Петр является отцом Ивана, а факт является *_отцом("Иван", "Петр")* говорит уже совсем о другом, а именно, что Иван является отцом Петра. Одно и то же утверждение, записанное в виде факта, может по-разному интерпретироваться. Только автор программы определяет истинную интерпретацию имен объектов и порядок следования аргументов, и им он должен следовать в процессе написания всей программы, отражая в комментариях смысл записанных им высказываний. Комментарий - это текст, заключенный между символами */*... */*, или строка, начинающаяся со знака *%*, например:

*/*Андрею нравится Ольга*/нравится ("Андрей", "Ольга").*

*/*Спица является частью колеса*/ часть объекта("спица", "колесо").*

Язык Visual Prolog позволяет передать смысл аргументов предикатов с помощью специальных разделов описаний *DOMAINS*, *PREDICATES*, *FACTS*, *DATABASE*, которые будут рассмотрены далее в п. «Структура программы на языке Visual Prolog».

Чтобы выполнить простейшую программу на Прологе, откроем файл с программой «Меню» в среде Visual Prolog.

Программа начинает выполняться, если в нее ввести те вопросы, ответы на которые хочет получить пользователь. Для этого предназначен раздел *GOAL* (Цель). В нем записываются необходимые вопросы - третий тип утверждений в программе на Прологе.

Введите такие вопросы: «Есть ли в ресторане рыбное блюдо "окунь во фритюре"?»

GOAL

рыба("окунь_во_фритюре").

В диалоговом окне получаем ответ

yes

или «Является ли мясным блюдом блюдо "артишоки в белом соусе"?»

GOAL

мясо("артишоки_в_белом_соусе") .

no

При получении вопроса (целевого утверждения) программа начинает исполняться. Суть выполнения состоит в поиске в базе знаний заголовка утверждения, соответствующего вопросу. Если такое утверждение (факт) есть, то на экране появляется ответ «*yes*» (да). Если нет, как в случае вопроса *закуска("салат из_помидоров")*, то на экране появляется отрицательный ответ «*no*» (нет).

Вопрос, или целевое утверждение, с помощью предиката описывает отношение (гипотезу), истинность которого для данной предметной области неизвестна и должна быть определена в процессе выполнения программы, в процессе согласования вопроса с утверждениями базы знаний. Синтаксис вопроса совпадает с синтаксисом предиката.

Продолжим работу с «Меню». Предположим, что мы хотим получить информацию обо всех закусках. В этом случае мы, естественно, задали бы вопрос: «Какие блюда являются закусками?» или более формализовано: «Существуют ли те блюда (объекты) *X*, которые являются закусками?». Здесь имя *X* обозначает не какой-то конкретный объект, а любой, принадлежащий некоторому множеству (может быть, пустому) объектов, обладающих свойством быть закуской и которые нужно найти в меню. В этом случае говорят, что *X* есть переменная.

На Прологе вопрос о закусках записывается так:

GOAL

закуска(X).

Будет получен ответ:

X=артишоки_в_белом_соусе X=трюфели_в_шампанском

X=салат_с_яйцом

3 Solutions

Все эти ответы содержат множество тех значений переменной X , при которых утверждение *закуска*(X) истинно.

С помощью отношений, которые составляют начальную базу знаний, можно конструировать более сложные и более общие отношения. Например, с помощью отношений *мясо* и *рыба*, выражающих то, что их аргумент является вторым мясным или рыбным блюдом, можно определить отношение *блюдо*: «Блюдо - это второе мясное или рыбное блюдо», что записывается на Прологе в виде двух правил следующим образом:

```
PREDICATES /* Описание предиката блюдо */
```

```
блюдо(name)
```

```
CLAUSES
```

```
/* Правила, задающие определение понятия блюдо */
```

```
блюдо(Y):-мясо(Y).
```

```
блюдо (Y):-рыба(Y).
```

и читается так: « Y является блюдом, если Y - второе мясное блюдо, или Y является блюдом, если Y - второе рыбное блюдо». Последовательность двух правил означает их дизъюнкцию (операцию «или»: первое правило или второе). Область действия переменной ограничена правилом, в котором она определена. Поэтому переменная из первого правила никак не связана с переменной Y из второго. Вопрос «Что является блюдом?», выраженный в виде:

```
GOAL
```

```
блюдо(Y),
```

вызовет следующие ответы:

```
Y=говяжье_жаркое
```

```
Y=цыпленок_в__липовом_цвете
```

```
Y=окунь_во_фритюре
```

```
Y=фаршированный_судак
```

4 Solutions

Можно построить и более сложные правила. Займемся теперь составлением обеда, в который входят закуска, второе блюдо (мясное

или рыбное) и десерт. Обед является, следовательно, тройкой (X, Y, Z), где X - закуска, Y - блюдо, Z - десерт.

В Прологе это выражается очень естественно в виде следующего правила:

/ Определение отношения "обед" */*

обед(X,Y,Z):-закуска(X), блюдо(Y), десерт(Z).

Оно читается так: «X, Y, Z удовлетворяют отношению обед, если X удовлетворяет отношению закуска, Y удовлетворяет отношению блюдо и Z удовлетворяет отношению десерт». Формально говоря, мы определили новое отношение как конъюнкцию (операцию И) трех других отношений, как конъюнкцию предикатов.

На вопрос «Что является обедом?»:

GOAL обед(X,Y,Z)

вы получите ответ:

X=артишоки_в^белом_соусе, Y=говяжье_жаркое,

г=грушевое_мороженое

X=артишоки_в_белом_соусе, Y=говяжье_жаркое,

г=земляника со взбитыми сливками

36 Solutions

т. е. список всех 36 возможных комбинаций из трех блюд. Попробуйте это выполнить [15].

Правила описывают зависимость некоторого отношения от группы других отношений (зависимость предиката от группы других предикатов), называемых условиями (или целевыми утверждениями). Правило соответствует условному высказыванию (импликации).

Правило состоит из заголовка и тела правила. Заголовок и тело соединяются с помощью символа «:-», соответствующего в русском языке слову «если». Правила также заканчиваются точкой. Заголовок правила описывает отношение, для определения которого предназначено правило. Тело правила в большинстве случаев представляет собой конъюнкцию предикатов (целевых утверждений), которые должны быть последовательно согласованы с базой знаний для того, чтобы заголовок правила был истинным. Предикаты (цели)

в теле правила разделяются запятыми. Правило - это некоторое общее утверждение. Оно описывает закономерность, свойственную не какому-то конкретному объекту, а целому классу объектов. Поэтому в аргументы правил входят переменные. Правило можно рассматривать и как разбиение сложной задачи (цели) на более простые подзадачи (подцели).

Примеры правил:

любит(X, "баскетбол"):-любит(X, "бег").

/ X любит баскетбол, если X любит бегать */*

можно_купить(X) :-есть_в_магазине(X, V) , V<3000.

/ можно купить X, если X есть в магазине и стоит меньше 3000 р. */*

старше (P1, P2) : - возраст (P1,V1) ,

возраст (P2,V2), V1>V2.

/ субъект P1 старше субъекта P2, если его возраст больше возраста P2 */*

Вернемся к программе «Меню». Уточним вопрос, сохранив то же множество отношений: нас интересуют обеды с главным блюдом из рыбы. Этот вопрос имеет вид:

GOAL

Обед(X,Y,z), рыба(Y).

и на естественном языке формулируется следующим образом: «Какие блюда X, Y, Z составляют обед и в этом обеде второе блюдо - рыба?». Вопрос состоит из конъюнкции двух целевых утверждений (предикатов), истинность которых должна быть подтверждена в процессе выполнения. Сначала будут найдены такие значения переменных X, Y, Z, которые составляют обед, т. е. при которых первый предикат станет истинным. Такими первыми значениями являются:

X=артишоки_в_белом_соусе,

Y=говяжье_жаркое,

Z=грушевое_мороженое

После сопоставления с базой знаний первой цели программа перейдет ко второму отношению рыба(Y) со значением, которое

приняла переменная $Y = \text{говяжье_жаркое}$. При этом будет проверяться предикат $\text{рыба}(\text{"говяжье_жаркое"})$. Поскольку база знаний не содержит такого утверждения, то предложенный выбор значений не удовлетворяет запросу - выбор неудачен, и нужно анализировать следующие варианты обедов.

Выполните программу и посмотрите, какие ответы будут получены на вопрос:

GOAL

обед(X,Y,Z), рыба(Z).

Вы убедитесь, что на этот раз получено всего 18 решений, в каждом из которых второе блюдо рыбное.

Пополним базу знаний новым классом отношений, введя значение калорийности для каждого блюда.

DOMAINS

kol_vo = integer

FACTS

калории(name, kol_vo)

CLAUSES

калории("артишоки_в_белом_соусе",

калории("трюфели_в_шампанском"),

калории("салат_с_яйцом",202)

калории("говяжье_жаркое"),

калории("цыпленок_в_липовом_цвете", 400).

калории("окунь_во_фритюре", 270).

калории("фаршированный_судак", 254) .

калории("грушевое_мороженое", 223) .

калории("земляника_со_взбитыми_сливками", 289) .

калории("дыня_сюрприз", 122).

Утверждение $\text{калории}(\text{"салат_с_яйцом"}\wedge 202)$ означает, что одна порция салата содержит 202 калории.

Тогда, чтобы узнать калорийность всех закусок, зададим вопрос:

GOAL

закуска(X), калории(X,Y).

Для тех, кто беспокоится о своем здоровье или о стройности своей фигуры, важно определить такой объект, как сбалансированный обед, г. е. обед, калорийность которого не превышает, например, 800 калорий:

PREDICATES

значение(name, name, name, kol_vo)

сбалансированный_обед(name, name, name)

CLAUSES

/ Определение отношения "калорийность обеда" */*

значение(X,Y,Z,V) :- калории(X,E), калории(Y,P), калории(Z, D)
V=E+P+D.

/ Определение отношения "сбалансированный обед" */*

сбалансированный_обед(X, Y, Z):- obeft(X,Y,Z),
значение(X,Y,Z;V), V<800.

Здесь «=» - встроенный предикат, при выполнении которого вычисляется значение арифметического выражения, стоящего справа от «=», и это значение присваивается переменной, стоящей слева;«+» - операция сложения; «<» - операция сравнения.

Посмотрите, какие ответы будут получены на запрос:

GOAL

сбалансированный_обед(X,Y,Z).

3.3 Структура программы на языке Visual Prolog

Рассмотрим еще один пример программы на языке Visual Prolog.

/ .программа "Студенты" */*

DOMAINS

student = string

facultet = string

gruppa = integer

FACTS

Студент(student, gruppa)

наим_фак(integer, facultet)

группа(gruppa)

CLAUSES

/ определение отношения "группа" с помощью фактов */*

группа(943).

группа(843) .

группа(232).

/ определение отношения "студент" с помощью фактов */*

студент("Орлова Л.И.", 943).

студент("Семенова М.П.", 232).

студент("Цуканова В.В.", 943).

/ определение отношения "наименование факультета"*

*с помощью фактов */*

наим_фак(1,"РТФ").

наим_фак(2,"ФЭ").

наим_фак(3,"ФАИТУ").

наим_фак(4, "ФВТ") .

наим_фак(7,"ИЭФ").

PREDICATES

nonterm факультет(gruppa, facultet)

вторая_цифра(gruppa, integer)

CLAUSES

/ определение отношения "факультет" с помощью правила */*

факультет(NG,F):- вторая_цифра(NG, G2), наим_фак(G2,F).

/ определение отношения "вторая цифра" с помощью правила */*

вторая_цифра(NG, G2) :- G1 = NG mod 100,

G2 = G1 div 10.

GOAL

группа(Gruppa), студент(Student, Gruppa);

студент(Student1,Gruppal), факультет(Gruppal, Facultet1).

Программа на языке Visual Prolog имеет структуру, состоящую из следующих

разделов:

DOMAINS */* описание типов данных */*

DATABASE (FACTS) / описание предикатов динамической
базы данных */*

PREDICATES / описание предикатов */*

CLAUSES / утверждения: факты и правила */*

GOAL / вопросы пользователя */*

В разделе DOMAINS программист определяет классы объектов, задавая их символические имена, и с помощью стандартных типов данных описывает их свойства.

Например, классы объектов типа «студент», «факультет», «группа» задаются следующим образом:

DOMAINS

student = string

facultet = string

gruppa = integer

Таким образом, в этом разделе программист создает свои типы (области - DOMAINS) данных.

В разделах *PREDICATES*, *DATABASE (FACTS)* описываются образцы предикатов, а именно: задаются имена предикатов, число их аргументов, смысл аргументов с помощью имен классов. Например:

PREDICATES

nondeterm факультет(gruppa, facultet)

вторая_цифра(gruppa, integer)

Такое описание показывает, что в предикате факультет первым аргументом является номер группы, а вторым - наименование факультета. Ключевые слова *determ* и *nondeterm* в описании предиката означают, что предикат может быть согласован с базой знаний соответственно один раз или много раз.

Различие *PREDICATES* и *DATABASE* в том, что в *DATABASE* описываются предикаты динамической базы данных, состоящей только из фактов, которые можно добавлять, удалять и изменять в процессе выполнения программы. Раздел *PREDICATES* содержит описание фактов и заголовков правил, однако соответствующие этим описаниям утверждения нельзя динамически изменять.

Раздел *CLAUSES* содержит собственно программу, т. е. все факты и правила, описывающие предметную область и составляющие базу знаний.

В программе могут присутствовать несколько утверждений, описывающих одно и то же отношение. Все они являются возможными различными вариантами описания этого отношения. Таким образом, совокупность утверждений с одинаковыми заголовками, т. е. с одинаковым именем и тем же числом аргументов, называется определением отношения или процедурой. Утверждения одного определения отношения должны быть сгруппированы вместе. Все утверждения о предметной области (программа), находящиеся в области действия программы Пролога, называются базой знаний.

Раздел *GOAL* содержит вопросы пользователя.

Классы объектов предметной области, введенные программистом, описываются с помощью следующих стандартных типов данных:

- *symbol* - символическое имя (person,y,a,mas);
- *string* - строка - любая последовательность символов в двойных кавычках («Катрин», «Домодедово»);
- *char* - отдельный символ, заключенный в апострофы ('R7 *');
- *integer* - целое число в диапазоне от -32 768 до 32 767;
- *byte* - целое число в диапазоне от 0 до 255;
- *word* - целое число в диапазоне от 0 до 65 535;
- *real* - любое число, может быть представлено в экспоненциальной форме.

Данные типа *symbol* в отличие от данных типа *string* запоминаются в таблице символов. Таблица символов размещается в оперативной памяти, поэтому ее использование обеспечивает наиболее быстрый поиск. Однако для построения таблицы символов требуется дополнительное время.

3.4 Реляционный язык Пролог

Логические программы можно рассматривать как мощное расширение модели реляционной базы данных (БД). Факты программы образуют отношения БД. Правила реализуют запросы. Основные операции реляционной алгебры легко выражаются в логическом программировании. Пять основных операций определяют реляционную алгебру: объединение, симметрическая разность, декартово произведение, проекция и выборка. Покажем, как каждая из них выражается в логической программе.

Операция объединения строит одно n -арное отношение из двух n -арных отношений r и s . Новое отношение, которое обозначим r_union_s , является объединением r и s . Это отношение задается логической программой, состоящей из двух правил:

$$r_union_s(X1, X2, \dots, Xn) :- r(X1, X2, \dots, Xn).$$
$$r_union_s(X1, X2, \dots, Xn) :- s(X1, X2, \dots, Xn).$$

Пример:

$$блюдо(Y) :- мясо(Y).$$
$$блюдо(Y) :- рыба(Y).$$

Из примера следует, что объединение отношений можно использовать для обобщения знаний о предметной области.

Определение симметрической разности использует понятие отрицания:

$$r_diff_s(X1, X2, \dots, Xn) :- r(X1, X2, \dots, Xn), not s(X1, X2, \dots, Xn).$$
$$r_diff_s(X1, X2 \# \dots, Xn) :- s(X1, X2, \dots, Xn), not r(X1, X2, \dots, Xn).$$

Пример:

$$футболист_либо_волейболист(X) :- футболист(X), not волейболист(X).$$
$$футболист_либо_волейболист(X) :- волейболист(X), not футболист(X).$$

Декартово произведение r_xjs может быть определено одним правилом:

$$r_x s(X1, X2, \dots, Xm, Y1, Y2, \dots, Yn) :- r(X1, X2, \dots, Xm),$$
$$s(Y1, Y2, \dots, Yn).$$

Пример:

/ Определение отношения "обед" */*

обед(X, Y, Z) :-закуска(X), блюдо(Y), десерт(Z).

Проекция состоит в построении отношения, использующего лишь некоторые аргументы исходного отношения:

rl3(X1, X3) :- r(X1, X2, X3).

Пример:

аптека(б, "Циолковского, 7", "44-67-57").

телефон_аптеки(NomApt, TelApt):-

аптека(NomApt, _, TelApt).

Чтобы получить выборку, достаточно в теле правила записать условия отбора:

rv(X1, X3):- r(X1, X2, X3), X2>X1.

Пример:

аптека(б, "Циолковского, 7", "44-67-57").

телефон_аптеки(NomApt, TelApt):-

аптека(NomApt, _, TelApt), NomApt < 8

*/*Определение отношения сбалансированный обед*/*

сбалансированный_обед(X, Y, Z) :-обед(X, Y, Z),

значение(X,Y, Z,V), V<800.

Операция соединения двух или нескольких отношений определяется следующим образом:

/ определение отношения "факультет" с помощью правила */*

факультет(NG, F) :-вторая_цифра(NG, G2),

наим_фак(62,K).

Здесь 2 отношения связаны общей переменной G2.

В заключение этой главы выделим основные понятия среды программирования Visual Prolog. Итак, программа на Прологе представляет собой базу знаний о некоторой предметной области и имеет структуру. Программа состоит из утверждений (фактов и правил) и их описаний. Совокупность утверждений, имеющих одинаковые имена отношений с одним и тем же числом аргументов (т. е. одинаковые заголовки), называется определением отношения или процедурой. Утверждения с одинаковыми именами, но с разным

числом аргументов образуют разные уникальные определения в программе.

Элементарными составляющими утверждений являются предикаты со своими списками аргументов, описывающие отношения предметной области. Предикаты представляют собой логические функции, приобретающие значение (истина или ложь) в процессе выполнения программы. Их структура имеет вид:

<имя отношения>(<список аргументов>).

Элементами списка аргументов могут быть только термы, отделяющиеся друг от друга запятой.

Терм - это синтаксически правильная конструкция Пролога. Различают следующие типы термов: константы, переменные, списки, строки, структуры. Программа начинает выполняться после ввода вопроса или целевого утверждения, которое записывается в форме предиката в разделе

GOAL

<имя отношения>(<список аргументов>)

и формулирует цель решаемой задачи. В процессе выполнения устанавливается истинность или ложность поставленной цели.

Программирование на языке Пролог включает следующие этапы:

- объявление фактов, задающих отношения между конкретными объектами или объектами и их свойствами;
- определение правил о зависимости одних отношений или объектов от других;
- формулировка цели решаемой задачи в виде вопроса (гипотезы, целевого утверждения).
- Чтобы программу, написанную на базовом Прологе, адаптировать к языку Visual Prolog, необходимо выполнить следующее:
 - утверждения программы записать в разделе *CLAUSES* в соответствии с правилами: имена предикатов должны быть

символическими именами, имена объектов предметной области - строками;

- затем в разделе *DOMAINS* создать и описать свои типы данных, используя названия классов объектов;

- в разделе *PREDICATES* описать предикаты программы, указав их имена и классы объектов, являющиеся аргументами предикатов;

- в разделе *FACTS* или *DATABASE* описать только те факты, которые могут в процессе выполнения программы изменяться, удаляться и добавляться;

- в разделе *GOAL* задать вопрос пользователя.

Контрольные вопросы и задания к главе 3

1. Как появилось название языка логического программирования – «Пролог» ?

2. В чем заключаются основные отличия Пролога от других языков?

3. Что такое факт? Приведите пример определения новых свойств описанных в фактах объекта.

4. Перечислите основные области применения Пролога.

5. Дайте определение «предметной области».

6. Что такое строки и для чего они используются?

7. Что такое утверждение и какова его структура?

8. Что необходимо учитывать при определении отношения между объектами?

9. Из каких элементов состоит правило? Приведите пример правил.

10. В каких разделах описываются образцы предикатов – имена предикатов, число их аргументов?

11. В чем отличие *PREDICATES* и *DATABASE* ?

12. Может ли в программе присутствовать несколько утверждений, описывающих одно и то же отношение?

13. Какие типы данных используются в программе?
14. В чем отличие данных типа *symbol* от данных типа *string*?
15. Перечислите основные операции, определяющие реляционную алгебру.
16. Приведите пример операции объединения.
17. Что является элементарной составляющей утверждения?
18. Какие виды термов вы знаете?
19. Перечислите этапы программирования на языке Пролог?
20. Каким образом адаптировать программу, написанную на базовом Прологе, к языку Visual Prolog?

Глава 4. ОПИСАНИЕ И ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ЭКСПЕРТНЫХ СИСТЕМ

4.1 Основные элементы экспертной системы и порядок работы с ней

В основе экспертной системы лежит база знаний. Создание и использование экспертной системы для реализации баз знаний дали возможность интегрировать методы логического вывода, моделирующие разумные заключения, характерные для человека. Это связано с тем, что практика требует создания автоматизированных систем, принятия решения и создания систем искусственного интеллекта, заменяющих человека. Это стало возможным на базе разработки формальных моделей принятия решений во многих предметных областях.

Любая экспертная система включает в себя три основных компонента:

- база знаний и интегрированные базы знаний;
- машина вывода с механизмом логического вывода реализацией функции, а также синтаксис логических выводов;
- интерфейс разумных умозаключений.

Графические экспертные системы включают такие элементы как объекты, факты, правила, связи, атрибуты, значения, механизм логического вывода, реализацию различных функций, синтаксис методов логического вывода. Эти элементы связаны интерфейсом, вводом базы знаний, корректировкой базы знаний, вводом методов логического вывода, выводом результатов [4].

В процессе проектирования экспертной системы необходимо реализовать следующие свойства.

1. Ограниченность определенной сферой экспертизы, т.е. проектируемая экспертная система должна быть сугубо профессиональной.

2. Способность рассуждать и получать результаты при сомнительных данных и дать объяснения своим рассуждениям.

Для того чтобы экспертная система не приводила в тупик при решении задач, она должна быть гибкой и иметь расширенный диапазон вопросов.

1. Четкое разделение объектов или фактов и связанных с ними механизмов вывода. Для реализации данной характеристики экспертная система должна иметь гибкую систему логического вывода, основанную на различных методах.

2. Проектируемая экспертная система должна быть наращиваемой системой и должна расширяться для конкретной профессиональной деятельности.

3. Использование четных и нечетных правил механизма логического вывода и синтеза методов.

4. Получение на выходе пользователем лаконичных ответов, советов с комментариями и предоставлением в некоторых случаях графической информации [8].

Основу любой экспертной системы предоставляет база данных, которая содержит объекты и факты, утверждения в виде правил, связывающие объекты. Файлы и объекты описывают то, что известно о данной конкретной области в данный момент. Эти данные представляют краткосрочную информацию, т. е. они могут быть изменены или дополнены в процессе работы. Правила устанавливают ситуационные концептуальные и причинные взаимосвязи между объектами. Правила являются более долговременной информацией. Они порождают или используют факты из, того, что известно. В процессе проектирования для конкретных фактов устанавливается или описываются атрибуты или свойства. При этом, под объектами будут пониматься как физические, так и символические представления. Каждый атрибут объекта или факта может принимать свое значение. В общем случае в базе данных записываются триплеты (*объект+атрибут+значение*). Для упрощения экспертная система без потери и эффективности разрабатывает пары: *объект+значение*. Это зависит от модели. Причем для однозначности экспертная система объекты и значения описывает через отношение атрибут. Введенные

объекты или факты, их атрибуты и значения вводятся в базу знаний посредством соответствующих механизмов, которые должны присутствовать в интерфейсе. При проектировании экспертной системы триплеты или пары должны учитываться в механизмах логического вывода. Для реализации гибкости экспертных систем должны быть предусмотрены выражения неопределенности достоверности фактов. В процессе эксплуатации базы знаний постоянно пополняется недостающими фактами, поэтому является активной. В поисках источника для наполнения базы данных предпочтение отдается жизненному опыту, а не высокому интеллекту. Считается, что эксперт исходит из продолжительных наблюдений за событиями в некоторой конкретной области и создает более полезную базу знаний, чем гениальный аналитик.

Экспертная система представляет собой сцепленный список структурированных данных. Он может иметь нулевые объекты. Каждый узел в списке имеет двойной указатель, который определяет начало списка значений и связанных с именем объекта. Внутренний список называется списком объектов значений, который индивидуален для каждого узла. Порядок связи между узлами такой, что каждое предыдущее значение указывает на последующий узел до тех пор, пока не будет исчерпан весь список. Данная процедура представляет собой рекурсию.

Для создания и манипулирования списком объектов создаются указатели и указатель значения согласно атрибутам. Эти указатели определяют конкретные записи с ранее определенными значениями или именем объекта посредством индексного указателя. В экспертной системе предусмотрено наличие указателя на анонимную переменную или пустую NIL [8].

Машина вывода

Одним из главных элементов экспертной системы является машина вывода, которая осуществляет поиск в базе знаний соответствующих триплетов или пар по правилам рациональной логики, согласно введенным правилам для получения решения.

Машина вывода приводится в действие при получении запроса пользователя и выполняет следующие задачи:

- сравнивает информацию, содержащуюся в запросе пользователя с информацией базы знаний (поисковые системы);
- осуществляет поиск определенных целей или причинных связей;
- оценивает, относит определенность фактов, основываясь на соответствии коэффициентов доверия, связанных с каждым из фактов.

Если база знаний однозначная, то достаточно использовать понятие множеств.

Если существует неопределенность неоднозначность в системе, тогда необходимо использовать понятия нечетных множеств.

Таким образом, машина вывода предназначена для построения умозаключений и ее действия аналогичны рассуждениям эксперта. В процессе машина вывода предполагает гипотетические решения путем формирования и формирования пробных гипотез и в дальнейшем проверки этих гипотез на соответствие указанной цели. Данная процедура оценивания продолжается до тех пор, пока не будет набран или определен наиболее вероятный путь к получению приемлемого результата.

Интерфейс

В задачу интерфейса входит организация обмена информации между пользователями и машиной вывода, т.е. осуществление диалога. При этом интерфейс решает следующие задачи:

- ввод базы знаний, т. е. ввод триплетов или пар, а также правил взаимосвязи объектов;
- корректировка базы знаний и правил связи между объектами;
- удаление не нужных пар, триплетов, правил;
- ввод или получение результата решений.

В процессе работы интерфейс должен создать видимость произвольной беседы, т.е. диалога, применяя повседневные выражения в правильно построенные предложения и понятия базы

данных. Интерфейс представляет собой естественный язык и РС. В процессе описания базы знаний целесообразно использовать теоретико-множественные понятия анализируемых триплетов, правил и понятий. При этом абстрактный уровень позволяет осуществлять общение механизма работы экспертной системы.

На основании введенных триплетов и правил разрабатываются правила и их взаимосвязи, которые являются более общими, чем сами объекты.

С учетом теоретико-множественных подходов построения базы знаний алгоритмизируются механизмы логического вывода и синтеза методов логического вывода.

Причем введенные механизмы должны связывать воедино триплеты и правила для сознания лаконических ответов на сформулированные вопросы.

При описании элементов логического вывода для повышения интеллектуальности проектируемой системы целесообразно использовать аппарат нечетных множеств, так как это позволяет увеличить глубину поиска и тем самым получить более глубокий ответ. Кроме того, использование нечетных множеств позволяет исключить спорные вопросы решения задач.

На основании используемой базы знаний, а также механизмов логического вывода тщательно разрабатывается интерфейс пользователя, который должен обеспечить профессиональный инструментарий для проектирования систем искусственного интеллекта.

Осуществляется ввод экспертной системы и ее отладка. В процессе отладки осуществляется проверка работы на конкретные вопросы согласно базе знаний. Кроме того, анализируются те вопросы, которые могут поставить экспертную систему в критические точки, т.е. решение сомнительных задач.

Исследование должно продолжаться до тех пор, пока экспертная система не будет отвечать за все возможные вопросы. В процессе отладки производится уточнение объектов и фактов, триплетов и правил.

Система считается завершенной, если она отвечает на все поставленные вопросы в заданных границах применения экспертной системы.

4.2 Примеры использования экспертных компьютерных систем

Тема искусственного интеллекта всегда была в информатике страной, населенной массой проблем, не поддающихся решению традиционными способами. Эта область привлекла внимание, прежде всего разносторонних специалистов, которых не испугало ее открытое, лишенное всякой организации пространство, - людей, которых влечет задача узнать, как мы мыслим. Такие исследователи, как Марвин Минский, Джон Мак-Карти, Герберт Саймон, Пат Хейес, Дональд Мичи и Бернард Мельтцер, стали первопроходцами для тех, кто следовал за ними по пути, пролегающему через информатику, психологию и математическую логику.

А зачем при построении экспертных систем нам вообще нужны технологии, используемые в задачах искусственного интеллекта? Почему нас не устраивают традиционные информационные технологии, такие как математическое моделирование? Или, другими словами, тот факт, что экспертные системы начали развиваться в русле задач искусственного интеллекта, является случайностью либо в этом была своя логика?

Не вдаваясь в длительные рассуждения, можно ответить, что нет ничего плохого в использовании для построения экспертных систем подходящих традиционных технологий, если это приводит к желаемому результату.

Экспертная система ABEL

Использование экспертной медицинской системы: ABEL помогает клиницистам диагностировать нарушения кислотно-щелочного и водно-солевого равновесия у пациентов, применяя знания о заболеваниях и вызываемых ими симптомах. Система использует причинно-следственную модель возможных заболеваний

пациента для упорядочения вопросов к клиницисту и направления процесса диагностических рассуждений. Эта модель содержит данные о пациенте и знания о взаимосвязях между различными патологическими состояниями. Система проверяет, нет ли в данных ошибок, сравнивая ответы клинициста на свои вопросы с предсказаниями модели. Знания представлены в виде причинно-следственной сети, разновидности семантических сетей, задающей причинно-следственные связи между заболеваниями и симптомами. Система была разработана в Массачусетском технологическом институте и доведена до уровня исследовательского прототипа.

Экспертная система: PTRANS

Сфера применения: Промышленность. Описание экспертной системы: PTRANS помогает управлять производством и распределением компьютерных систем компании DEC. Она использует описание заказа клиента и информацию о работе завода, чтобы разработать план сборки и тестирования заказанной компьютерной системы, в том числе определяет сроки ее изготовления. PTRANS наблюдает за тем, как персонал справляется с выполнением этого плана, диагностирует трудности, предлагает способы их преодоления и прогнозирует возможные тормозящие нехватки материалов или их излишки. PTRANS рассчитан на работу совместно с XSEL, помощником продавца, так что как только сделан заказ, можно гарантировать согласованную дату поставки. PTRANS - это основанная на правилах система с прямой цепочкой рассуждений, реализованная на языке OPS5. Она была разработана совместно компанией и Университетом Карнеги-Меллон и доведена до уровня исследовательского прототипа.

Экспертная система: DART

Описание экспертной системы: DART оказывает помощь при диагностировании неисправностей компьютерных систем, используя информацию о конструкции диагностируемого устройства. Система работает непосредственно с информацией о структуре и предполагаемом поведении устройства и оказывает

помощь при поиске проектных недочетов в заново создаваемых устройствах. Система применяется для простых вычислительных схем и телепроцессорных устройств IBM 4331. DART использует независимую от конструкции устройства процедуру вывода, которая аналогична процессу доказательства теорем, когда система пытается получить доказательство относительно причин неисправностей устройств. Эта система реализована на языке MRS и была разработана в Станфордском университете. Она доведена до уровня исследовательского прототипа (Diagnostic Assistance Reference Tool).

Экспертная система: IDT

Описание экспертной системы: IDT помогает техническому персоналу найти заменяемые блоки, которые следует заменить для устранения неисправностей в ЭВМ PDP 11/03. Система использует знания о тестируемых блоках, например, функции его компонент и их связи друг с другом, чтобы выбрать и выполнить диагностические тесты, а также интерпретировать их результаты. Система основана на правилах, использует прямую цепочку рассуждений, реализована на FRANZ LISP и OPS5. IDT была разработана компанией DEC и доведена до уровня исследовательского прототипа.

Экспертная система: XCON

Описание экспертной системы: XCON составляет конфигурацию вычислительной системы VAX 11/780. Исходя из заявок пользователя, она принимает решение о том, какие компоненты должны быть добавлены для получения полностью работоспособной системы и каковы должны быть пространственные взаимосвязи между всеми компонентами. XCON выводит набор схем, указывающих эти пространственные взаимосвязи, сообщая его техническому персоналу, который затем собирает систему VAX. XCON составляет конфигурацию, применяя знания об ограничениях на взаимосвязи компонент к стандартным процедурам составления конфигураций ЭВМ. Система не

интерактивная, основана на правилах, использует прямую цепочку рассуждений в схеме логического вывода. XCON реализована на языке OPS5 и разработана в Университете Карнеги-Меллон совместно с DEC. Эта экспертная система коммерческого уровня повседневно составляет конфигурации ЭВМ серии VAX для DEC и является самой крупной и наиболее зрелой среди работающих систем (eX-pert CONfigurer of VAX 11/780 computer system).

Экспертная система: TIMM/TUNER

Описание экспертной системы: TIMM/TUNER оказывает помощь в настройке вычислительных систем VAX/VMS для того, чтобы уменьшить проблемы, возникающие в постоянно изменяющейся вычислительной среде. Она взаимодействует с администратором системы, задавая ему ряд вопросов, чтобы рекомендовать действия, такие, как подстройка параметров системы или значений, выбираемых пользователем, перераспределение или снижение требований пользователя, изменение программой разработки пользователя или приобретение новой аппаратуры. Система использует схему представления знаний, основанную на правилах, созданную в рамках TIMM, системы коммерческого уровня для автоматизации приобретения знаний. TIMM/TUNER была разработана в General Research Corporation и доведена до коммерческого уровня.

Экспертная система: YES/MVS

Описание экспертной системы: YES/MVS оказывает помощь дежурным операторам в слежении и управлении операционной системой MVS (множественная виртуальная память) на больших ЭВМ компании IBM. YES/MVS ориентирована на 6 основных классов задач: сохранение достаточного объема памяти для YES (системы ввода заданий), управление обменом в сети ЭВМ, находящихся в одном здании или помещении, планирование выполнения длительных расчетов вне первой смены, реагирование на ошибки аппаратуры, слежение за работой подсистем программного обеспечения и наблюдение за работой системы в целом. YES/MVS

работает в режиме реального времени, непосредственно интерпретируя сообщения MVS и отправляя или команды операционной системе, или рекомендации дежурному оператору. Экспертная система YES/MVS основана на правилах, со схемой управления с помощью прямой цепочки рассуждений. Она реализована на расширенной версии языка OPS5 группой разработчиков экспертных систем фирмы IBM. Разработчики планируют постоянно использовать систему YES/MVS на коммерческой основе.

Экспертная система: BDS

Описание экспертной системы: BDS оказывает помощь, при выявлении неисправных модулей в большой сети коммутации сигналов, которая называется baseband distribution subsystem. Эта система использует показания испытательной аппаратуры, чтобы обнаружить неисправные печатные платы или другие детали, смонтированные на шасси, из-за которых могла возникнуть неисправность. BDS основывает свой диагноз как на стратегиях эксперта-диагноста, так и на знаниях о структуре, функциях и причинных связях компонентов электронного устройства. BDS реализована на языке LES и использует представление знаний, основанное на правилах, и управление посредством обратной цепочки рассуждений. Она была разработана в Lockheed Palo Alto Research Laboratory и доведена до стадии исследовательского прототипа.

Экспертная система: TALIB

Описание экспертной системы: TALIB автоматически синтезирует топологию интегральных схем типа n-МОП. В качестве входной информации система получает описание элементов схемы, которые нужно разместить на кремниевой подложке, их соединений друг с другом, а также топологических и геометрических требований относительно краевой зоны вдоль периметра схемы. По этим спецификациям система строит правильную и компактную топологию структуры. TALIB создает и уточняет планы размещения элементов и их соединений и затем реализует их,

применяя знания о характеристиках соединений подсхем и о распространении ограничений между подсхемами. TALIB - это основанная на правилах система с прямой цепочкой рассуждений, реализованная на языке OPS5. Она была разработана в Университете Карнеги-Меллон и доведена до уровня исследовательского прототипа.

Экспертная система: ACE

Экспертная система ACE определяет неисправности в телефонной сети и дает рекомендации по необходимому ремонту и восстановительным мероприятиям. Система работает без человеческого вмешательства, анализируя сводки-отчеты о состоянии, получаемые ежедневно с помощью CRAS, программы, следящей за ходом ремонтных работ в кабельной сети. ACE обнаруживает неисправные телефонные кабели и затем решает, нуждаются ли они в планово-предупредительном ремонте и выбирает, какой тип ремонтных работ вероятнее всего будет эффективным. Затем ACE запоминает свои рекомендации в специальной базе данных, к которой у пользователя есть доступ. Система принимает решения, применяя знания относительно телефонных станций, сообщения системы CRAS и стратегии анализа сетей. Представление знаний в системе основано на правилах; используется схема управления посредством прямой цепочки рассуждений. ACE реализована на OPS4 и FRANZ LISP и работает на микропроцессорах серии AT&T 3B-2, размещенных в подстанциях наблюдения состояния кабеля. Она разработана в Bell Laboratories. ACE прошла опытную эксплуатацию и доведена до уровня коммерческой экспертной системы.

ЗАКЛЮЧЕНИЕ

В настоящее время в сферу применений искусственного интеллекта входят практически все направления информатики и коммуникаций.

В четырех разделах учебного пособия рассматриваются основные вопросы искусственного интеллекта: структура исследований, задачи и методы их исследования, примеры использования интегрированной среды программирования баз знаний на языке Visual Prolog для решения задач искусственного интеллекта, даны рекомендации для построения баз знаний, а также приведены описание и примеры разработки экспертных компьютерных систем.

В пособии описаны традиционные модели и технологии создания интеллектуальных систем, а также новые перспективные подходы к решению проблем, возникающих в области искусственного интеллекта.

В последние годы в различных предметных областях получили широкое применение разнообразные интеллектуальные системы, относящиеся к разделительному процессу (Knowledge Discovery, Data Mining, многоагентные системы и т. д.) и интеграционному процессу (обучающие, поисковые, автоматизированного проектирования, баз данных и знаний, поддержки принятия решений, реинжиниринга бизнес-процессов, производственные, геоинформационные и т. д.). Они являются системами обработки знаний, ориентированными, прежде всего, на решение плохоформализуемых, слабоструктурированных задач управления, свойственных человеку и характеризующихся неопределенностью (нечеткостью, противоречивостью, немонотонностью и т. д.) знаний [6].

С точки зрения информационных технологий следует выделить следующие интеллектуальные системы: экспертные, нейросетевые, системы с параллельной технологией вычислений, распределенные интеллектуальные.

Отметим проблемы, существующие в настоящее время в системах искусственного интеллекта:

- разработка и автоматизация технологий извлечения знаний и формирования баз знаний;
- разработка интеллектуальных человеко-машинных интерфейсов;
- вложение семантической информации в формализмы представления знаний, накопление и использование их;
- организация текущих знаний в самостоятельные, специализированные фрагменты для практического использования;
- структурирование статических и динамических знаний;
- адаптация и эффективность решения задач, в основанных на знаниях системах (планирование, понимание).

Итак, искусственный интеллект охватывает широкий спектр проблем, представляющий сферу исследований следующих поколений ученых.

Однако настоящие исследования и практика в искусственном интеллекте позволяют смотреть в будущее вполне оптимистически, но и наивно полагать, что без ответов на многие вопросы человечество может построить действительно «разумные машины».

Контрольные вопросы и задания к главе 4

1. Какие компоненты включает в себя любая экспертная система?
2. Перечислите основные элементы графических экспертных систем.
3. Какие свойства необходимо реализовать в процессе проектирования экспертной системы?
4. Что такое машина вывода?
5. Какие задачи выполняет машина вывода?

6. Что необходимо использовать при описании элементов логического вывода для повышения интеллектуальности проектируемой системы?

7. Какие задачи решает интерфейс?

8. В каком случае система считается завершённой?

9. Для чего предназначена экспертная система ABEL?

10. Какая модель лежит в основе системы ABEL?

11. Для решения каких задач используется экспертная система PTRANS?

12. На каком языке реализована система PTRANS?

13. Дайте описание экспертной системе DART.

14. С какой информацией работает система DART?

15. Для чего предназначена экспертная система IDT?

16. Какие знания использует система IDT?

17. Дайте описание экспертной системе XCON.

18. Для решения каких задач используется экспертная система TIMM/TUNER?

19. Для чего предназначена экспертная система YES/MVS?

20. На каком языке реализована система YES/MVS?

21. Как расшифровывается аббревиатура MVS?

22. Дайте описание экспертной системе BDS.

23. На каком языке реализована система BDS?

24. Для чего предназначена экспертная система TALIB?

25. Что выступает в качестве входных данных в системе TALIB?

26. Дайте описание экспертной системе ACE.

27. Может ли система ACE работать без человеческого вмешательства?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Андрейчиков А. В. Интеллектуальные информационные системы: учебник / А. В. Андрейчиков, О. Н. Андречикова. – М.: Финансы и статистика, 2006. – 424 с. – №12.
2. Бондарев В. Н. Искусственный интеллект : учеб. пособие для вузов / В. Н. Бондарев, Ф. Г. Аде – Севастополь : Изд-во СевНТУ, 2002. – 615 с.
3. Гаскаров Д. В. Интеллектуальные информационные системы: учебник для вузов / Д. В. Гаскаров. – М.: Высш. шк., 2003. – 431 с.
4. Джарратано Д./Экспертные системы: принципы разработки и программирование /Д. Джарратано, Г. Райли.-4-е изд.- М.:Вильямс,2007.-1152с.:ил.
5. Ларичев О. И. Системы основанные на экспертных знаниях: история, совершенное состояние и некоторые перспективы // Труды Седьмой национальной конференции по искусственному интеллекту с международным участием. – М.: Изд-во физико-математической литературы, 2000.
6. Луценко Е. В. Интеллектуальные информационные системы / Е. В.Луценко. – Краснодар: КубГАУ, 2006. – 615 с.
7. М. Тим Джонс Программирование искусственного интеллекта в приложениях / М. Тим Джонс— Электрон. текстовые данные.— М.: ДМК Пресс, 2011.— 312 с.
8. Малышева Е.Н. Экспертные системы: учебное пособие по специальности 080801 «Прикладная информатика»/ Малышева Е.Н.— Электрон. текстовые данные.— Кемерово: Кемеровский государственный институт культуры, 2010.— 86 с.
9. Оссовский С. Нейронные сети для обработки информации: пер. с польского Н. Д. Руданского / С. Оссовский. – М.: Финансы и статистика, 2002. – 344 с.
10. Павлов С. Н. Интеллектуальные информационные системы : учеб. пособие / С. Н. Павлов. – Томск: Томский межвузовский центр дистанционного образования, 2004. – 328 с.

11. Павлов С.Н. Системы искусственного интеллекта. Часть 1: учебное пособие/ Павлов С.Н.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011.— 176 с.

12. Ручкин В.Н./Универсальный искусственный интеллект и экспертные системы /В.Н. Ручкин, В.А. Фулин.- СПб.: БХВ-Петербург, 2009.-240с.: ил.

13. Сысоев Д.В. Введение в теорию искусственного интеллекта: учебное пособие/ Сысоев Д.В., Курипта О.В., Проскурин Д.К.— Электрон. текстовые данные.— Воронеж: Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2014.— 171 с.

14. Шрайнер П.А./Основы программирования на языке Пролог: курс лекций: учеб. пособие для вузов. - М.:ИНТУИТ,2005.-176с.

15. Хабаров, С.П. Интеллектуальные информационные системы. PROLOG – язык разработки интеллектуальных и экспертных систем: учебное пособие для бакалавров и магистров направлений подготовки 230400 Информационные системы и технологии и 230200 Информационные системы. — Электрон. дан. — СПб. : СПбГЛТУ, 2013. — 140 с.

16. Ясницкий Л.Н./Введение в искусственный интеллект: учеб. пособие для вузов.-3-е изд., стер.-М.:Академия,2010.-176с.

Учебное пособие

*КРАЙНОВА Екатерина Анатольевна
САДОВА Кристина Владимировна*

Системы искусственного интеллекта

Редакторы:
*Е.С. Захарова
И.А. Назарова*

Подписано в печать 30.11.16г.
Формат 60х84 1/16. Бумага офсетная
Усл. п. л. 5 Уч.-изд. л. 3,9
Тираж 100 экз. Рег. № 12/16sf

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Самарский государственный технический университет»
443100, г. Самара, ул. Молодогвардейская, 244. Главный корпус

Отпечатано в типографии
Самарского государственного технического университета
Филиал в г. Сызрани, 446001, г. Сызрань, ул. Советская 45