

**ПРИОРИТЕТНЫЙ НАЦИОНАЛЬНЫЙ ПРОЕКТ «ОБРАЗОВАНИЕ»
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

В.О. ЧИНАКАЛ

**ИНТЕЛЛЕКТУАЛЬНЫЕ
СИСТЕМЫ И ТЕХНОЛОГИИ**

Учебное пособие

Москва

2008

*Инновационная образовательная программа
Российского университета дружбы народов*

**«Создание комплекса инновационных образовательных программ
и формирование инновационной образовательной среды,
позволяющих эффективно реализовывать государственные интересы РФ
через систему экспорта образовательных услуг»**

Экспертное заключение –
кандидат технических наук *Ю.М. Цодиков*

Чинакал В.О.

Интеллектуальные системы и технологии: Учеб. пособие. – М.:
РУДН, 2008. – 303 с.: ил.

Рассмотрены основные понятия, подходы, методы и технологии, используемые в современных интеллектуальных системах (ИС) управления сложными техническими системами. Дана классификация основных способов представления в ИС знаний и методов их компьютерной обработки. Приведены типовые модели процессов получения и обработки информации в ИС для формирования баз данных, баз знаний и построения экспертных систем, применяемых в области автоматизации управления техническими системами.

Пособие предназначено для студентов бакалавриата инженерного факультета РУДН, проходящих подготовку в области информатизации и автоматизации по направлениям «Информационные технологии», «Прикладная математика и информатика» и «Автоматизация и управление». Рекомендуется для первоначального ознакомления с основами построения и применения интеллектуальных систем и технологий.

Учебное пособие выполнено в рамках инновационной образовательной программы Российского университета дружбы народов, направление «Комплекс экспортноориентированных инновационных образовательных программ по приоритетным направлениям науки и технологий», и входит в состав учебно-методического комплекса, включающего описание курса, программу и электронный учебник.

ОГЛАВЛЕНИЕ

| | |
|------------------------------------------------------------------|----|
| Предисловие | 5 |
| Введение | 7 |
| Тема 1. Основные понятия и терминология интеллектуальных систем | 27 |
| Тема 2. Различные подходы к построению интеллектуальных систем | 37 |
| 2.1. Логический подход | 37 |
| 2.2. Структурный подход | 38 |
| 2.3. Эволюционный подход | 39 |
| 2.4. Имитационный подход | 41 |
| Тема 3. Системы, основанные на знаниях | 46 |
| 3.1. Системы, основанные на правилах | 46 |
| 3.2. Системы, основанные на автоматическом доказательстве теорем | 49 |
| 3.3. Системы, основанные на автоматическом гипотезировании | 51 |
| 3.4. Системы, основанные на рассуждениях по аналогии | 54 |
| 3.5. Объектно-ориентированные интеллектуальные системы | 55 |
| Тема 4. Основы логического вывода | 58 |
| 4.1. Основные понятия математической логики | 58 |
| 4.2. Аксиомы исчисления высказываний | 61 |
| Тема 5. Методы вывода в исчислении предикатов | 65 |
| 5.1. Основы классического исчисления предикатов | 65 |
| 5.2. Определение языка исчисления предикатов | 65 |
| 5.3. Аксиомы и правила вывода исчисления предикатов | 68 |
| 5.4. Расширение языка исчисления предикатов | 70 |
| 5.5. Разрешимые и неразрешимые задачи | 71 |

| | |
|-------------------------------------------------------------------------------------------|-----|
| Тема 6. Интеллектуальное управление динамическими системами | 73 |
| 6.1. Интеллектуальное управление на основе нечеткой логики | 73 |
| 6.2. Особенности применения систем интеллектуального управления | 82 |
| Тема 7. Системы распознавания образов | 89 |
| Тема 8. Нейронные сети | 107 |
| Тема 9. Краткие сведения об инструментальных средствах разработки интеллектуальных систем | 124 |
| 9.1. Краткие сведения о ПРОЛОГе | 126 |
| 9.2. Краткие сведения о ЛИСПе | 132 |
| 9.3. Язык РЕФАЛ | 134 |
| Тема 10. Основы построения экспертных систем | 140 |
| Тема 11. Разработка экспертных систем реального времени | 151 |
| 11.1. Особенности использования ЭС РВ в системах ИИ | 151 |
| 11.2. Применение ЭС РВ в интеллектуальных системах | 155 |
| Тема 12. Перспективные направления развития ИС и технологий | 159 |
| Темы курсовых работ, лабораторный практикум, практические задания и тесты | 165 |
| Заключение | 178 |
| Литература | 180 |
| Интернет ресурсы | 184 |
| Персоналии | 185 |
| Краткий глоссарий | 188 |
| Приложения | 280 |
| Описание курса и программа | 286 |

ПРЕДИСЛОВИЕ

Современное понятие интеллектуальных систем (ИС) сформировалось в процессе развития теоретических основ кибернетики, современной теории управления, теории алгоритмов, развития современных информационных технологий и обобщения накопленных научных знаний, методов и средств в области искусственного интеллекта (ИИ). Единого общепризнанного определения интеллектуальной системы (как, впрочем, и искусственного интеллекта) не существует. В начальном периоде развития ИИ (в 60-х годах прошлого века) под интеллектуальными системами понимали автоматические системы, способные успешно решать задачи, традиционно относимые к сфере деятельности человека (распознавание визуальных и звуковых образов, игра в шахматы, шашки, доказательство теорем и т.д.). Затем к ИС стали относить системы, моделирующие на компьютере работу клеток мозга и мышления человека. Позже, в 80-х годах прошлого столетия интеллектуальными системам стали считать автоматические компьютерные системы, способные вести осмысленный диалог с человеком и усиливать интеллектуальную деятельность человека в различных сферах деятельности.

В настоящее время искусственный интеллект, интеллектуальные системы и реализующие их технологии отражают высокий уровень развития современных информационных технологий и представляют обширную область проведения теоретических исследований и практического (коммерческого) использования результатов этих исследований в разнообразных технических и гуманитарных приложениях. Имеется огромное количество книг и научных публикаций по различным аспектам теории и практики ИИ и ИС, создана Российская ассоциация искусственного интеллекта (см. [21] <http://www.raai.org>), проводятся конференции по ИИ и ИС, в ВУЗах страны осуществляется подготовка специалистов по ИИ и ИС на основе различных программ, курсов и программно-технических средств.

Данное учебное пособие разработано в рамках создания учебно-методического комплекса (УМК) по курсу «Интеллектуальные системы и технологии», предназначенному для подготовки студентов старших курсов бакалавриата инженерного факультета РУДН, специализирующимся в области автоматизации производства и создания систем управления техническими системами. Предполагается, что студенты уже прослушали курсы по теории автоматического управления, математическим основам кибернетики, моделированию систем, системам автоматизации и управления, базам данных в информационно-управляющих системах, программированию и основам алгоритмизации. С учетом этого, а также в связи с большим объемом информации при ограниченном количестве аудиторных часов в одном семестре основная часть материала дана в виде краткой справочной информации со ссылками на более полные источники, и приведен глоссарий всех основных понятий, использованных в пособии. Отдельные части материала выделены для самостоятельного изучения. В соответствии с этим кроме общего обязательного списка рекомендуемой литературы, дан дополнительный список литературы, рекомендуемый для углубленного изучения рассматриваемых вопросов. При формировании списков литературы учитывалась реальная возможность доступа студентов к рекомендуемым материалам. В связи с этим ограничены ссылки на известные, но труднодоступные классические основополагающие работы, а в первую очередь указаны ссылки на более доступные современные книги, а также различные методические и справочные материалы в Интернете, отдельные фрагменты из которых частично использованы в данном пособии.

Фактическое содержание данного учебного пособия является введением в научные и методические основы создания и применения современных интеллектуальных систем и технологий применительно к решению различных сложных задач управления преимущественно техническими системами.

ВВЕДЕНИЕ

Возникновение понятия интеллектуальных систем тесно связано с понятием искусственного интеллекта (ИИ) и современной теории управления [1-6, 20, 21]. Исследования в области ИИ стали развиваться наиболее активно с 50-х годов прошлого века в связи с развитием средств вычислительной техники, информатики и теории автоматического управления. На разных этапах развития ИИ исследователями различных направлений ИИ предлагались свои определения ИИ [17, 18, 23, 27]. Так в математическом энциклопедическом словаре [27] приводилось следующее определение: «Искусственный интеллект» (от латинского — *intellectus* – познание, понимание, рассудок.) раздел информатики, изучающий методы, способы и приемы моделирования и воспроизведения с помощью ЭВМ разумной деятельности человека, связанной с решением задач». Многие исследователи, особенно зарубежные, предпочитают определение термина «искусственный интеллект» как «*artificial intelligence*» (AI), где слово «*intelligence*» означает «умение рассуждать разумно» (предложено в 1956 г. в Дартмутском колледже в США).

Необходимо отметить, что современная теория ИИ и созданные на ее основе интеллектуальные системы (ИС) и технологии были подготовлены трудами многих исследователей на протяжении многих лет.

Известным греческим ученым Аристотелем (4 век до н.э.) была впервые разработана классическая модель формальной логики, послужившая основой для создания всех современных моделей представления знаний в ИИ. Развитию моделей ИИ способствовали труды Канта по семантическому анализу на основе категорий, труды Буля по усовершенствованию логической системы обозначений и представлению абстрактной логической задачи в виде системы уравнений (Булева алгебра). Средневековый испанский философ, математик и поэт – Раймонд Луллий пытался в XIII веке

создать механическую машину для решения различных задач с использованием разработанной им всеобщей классификации понятий. В XVIII в. большой вклад в разработку идей классификации внесли Лейбниц и Декарт, предложившие независимо друг от друга универсальные языки классификации всех наук. Эти работы считаются первыми теоретическими работами в области искусственного интеллекта.

В 40-х годах XX в. произошло окончательное оформление искусственного интеллекта как научного направления. Были созданы первые ЭВМ, а Норберт Винер создал свои основополагающие работы по кибернетике. В 40-50-х годах прошлого века началось бурное развитие информационных технологий и автоматизации.

В процессе развития ИИ произошло его разделение на два направления науки: «нейрокибернетику» и кибернетику «черного ящика», развивавшихся как в методологии, так и в технологии практически независимо друг от друга. В последние годы эти направления начинают объединяться.

ИИ как область исследований возникла и развивалась наряду с развитием теории автоматического управления. Отдельные элементы интеллектуальных систем начинали применяться примерно с 60-х годов, вначале с основными приложениями в вычислительной технике и информатике, а позже в автоматизации управления.

Применительно к задачам управления в технических системах ИИ понимают как «раздел информатики, изучающий методы, способы и приемы моделирования и воспроизведения с помощью ЭВМ разумной деятельности человека, связанной с решением задач» [1, 2, 5].

В отличие от философии и психологии, искусственный интеллект в области исследований и разработок технических систем ориентирован не столько на понимание, сколько на построение интеллектуальных систем.

В становление новой науки внесли большой вклад многие ученые. Это и Маккарти (автор ЛИСПа – первого языка программирования для задач ИИ), и М. Минский (автор идеи Фрейма и Фреймовой модели представления знаний), А. Ньюэлл, Х. Саймон, Дж. Шоу, Э. Хант и другие.

Хотя ИИ, как область исследований, формально был назван в 1956 году, но активные исследования с применением ЭВМ велись уже около 5 лет. Работы У. МакКаллока и У. Питса в 1943 г. об искусственных нейронных сетях (ИНС) также относятся к ИИ. В процессе своего развития ИИ аккумулировал многие идеи, взгляды и методы разных дисциплин (философии, математики, психологии, лингвистики, информатики, теории управления и др.).

Представители существующих гуманитарных наук – философы, психологи, лингвисты – в 1956-1963 гг. вели интенсивные поиски моделей и алгоритмов человеческого мышления и пытались создавать первые программы на их основе, но реально они не смогли предложить эффективных алгоритмов. Исследователи кибернетики начали создавать собственные модели и разрабатывать различные собственные подходы.

В конце 50-х годов XX в. родилась модель лабиринтного поиска. В этом подходе задача представляется как некоторое пространство состояний в форме графа, и в этом графе проводится поиск оптимального пути от входных данных к результирующим. Была проделана большая работа по разработке этой модели, но в решении практических задач идея большого распространения не получила. В самых первых учебниках по искусственному интеллекту [Э. Хант, 1986; Эндрю, 1985] приведено описание игры таких программ в шашки, шахматы, в игру «15», а также собирание «Ханнойской башни».

Начало 60-х годов – это эпоха эвристического программирования. «Эвристика – правило, теоретически не обоснованное, но позволяющее сократить количество переборov в пространстве поиска. Эвристическое про-

граммирование – разработка стратегии действий на основе известных, заранее заданных, эвристик» [Александров, 1975].

В 1963-1970 гг. к решению задач ИИ стали подключать методы математической логики. Робинсон разработал метод резолюций, который позволяет автоматически доказывать теоремы при наличии набора исходных аксиом [41]. Несколько раньше выдающийся отечественный математик Ю.С. Маслов предложил решение аналогичной задачи другим способом [40], так называемый обратный вывод, впоследствии названный его именем. На основе метода резолюций француз Альбер Кольмероэ в 1973 г. создает язык логического программирования Пролог. Большую известность имела программа «Логик-теоретик», созданная Ньюэлом, Саймоном и Шоу, которая доказывала школьные теоремы. Однако большинство реальных задач не сводится к набору аксиом. Человек, решая производственные задачи, не использует классическую логику, и поэтому применение логических моделей при всех своих преимуществах имеют существенные ограничения по классам решаемых задач.

Значительное развитие в практических приложениях искусственного интеллекта произошло в середине 1970-х годов в США. В это время вместо поисков универсальных алгоритма мышления стала развиваться идея моделирования конкретных знаний специалистов экспертов. В США появились первые коммерческие системы, основанные на знаниях, или экспертные системы (ЭС). Возник новый подход к решению задач искусственного интеллекта – представление знаний. В это время были созданы две первые классические экспертные системы для медицины (MYCIN) и для химии (DENDRAL). Пентагон выделил значительные финансовые ресурсы для создания новой программы министерства обороны США на принципах ИИ – Strategic Computer Initiative (SCI). В начале 80-х годов Европейским Союзом была объявлена аналогичная глобальная программа развития

новых технологий – ESPRIT, в которую была включена проблематика искусственного интеллекта.

Япония включается в развитие работ в области ИИ в конце 70-х годов. Было объявлено о начале разработки проекта машин V поколения, основанных на знаниях. Проект был рассчитан на 10 лет и объединял лучших молодых специалистов крупнейших японских компьютерных корпораций. Был создан специально новый институт ICOT, в котором был разработан достаточно громоздкий и дорогой символьный процессор, программно реализующий ПРОЛОГ-подобный язык, не получивший в последствии широкого признания. Положительным эффектом данного проекта было то, что в Японии появилась значительная группа высококвалифицированных специалистов в области ИИ, получившая существенные результаты в решении различных прикладных задач.

Начиная с середины 80-х годов, повсеместно происходит коммерциализация искусственного интеллекта. Растут ежегодные капиталовложения, создаются промышленные экспертные системы. Растет интерес к самообучающимся системам, издаются десятки научных журналов, ежегодно собираются международные и национальные конференции по различным направлениям ИИ.

Искусственный интеллект становится одной из наиболее перспективных и престижных областей информатики (Computer science).

В Советском Союзе работы по искусственному интеллекту начались даже раньше, чем на западе [31]. В 1954 г. в МГУ, начал свою работу семинар «Автоматы и мышление» под руководством одного из основателей российской кибернетики – академика А.А. Ляпунова В этом семинаре принимали участие физиологи, лингвисты, психологи, математики. Принято считать, что именно в это время родился искусственный интеллект в России. Как и за рубежом, выделились два основных направления: нейрокибернетики и кибернетики «черного ящика».

В 1954-1964 гг. создаются отдельные программы и проводятся исследования в области поиска решения логических задач. В Ленинграде (ЛОМИ – Ленинградское отделение математического института им. В.А. Стеклова) создается программа, автоматически доказывающая теоремы (АЛПЕВ ЛОМИ). Она основана на оригинальном обратном выводе Маслова, аналогичном методу резолюций Робинсона. Среди наиболее значимых результатов, полученных отечественными учеными в 60-е годы, отмечают алгоритм «Кора» М.М. Бонгарда, моделировавший деятельность человеческого мозга при распознавании образов. Большой вклад в становление российской школы ИИ внесли выдающиеся ученые М.Л., Цетлин, В.Н., Пушкин, М.А. Гаврилов и их ученики, продолжившие развитие этого направления науки в России (например, известная Гавриловская школа).

В 1965-1980 гг. происходит рождение нового направления – ситуационного управления (соответствует представлению знаний в западной терминологии). Основателем этой научной школы стал проф. Д.А. Поспелов, разработавший специальные модели представления ситуаций - представления знаний:

В 1974 г. при Комитете по системному анализу при президиуме АН СССР начал работу Научный совет по проблеме «Искусственный интеллект».

По инициативе Совета были организованы комплексные научные проекты, которые были возглавлены ведущими специалистами в данной области. В 1980-1990 гг. проводятся активные исследования в области представления знаний, разрабатываются языки представления знаний, экспертные системы (более 300). В МГУ им. М. В. Ломоносова создается язык РЕФАЛ.

В 1988 г. создается АИИ – Ассоциация Искусственного Интеллекта. Ее членами стали более 300 исследователей. Президентом Ассоциации единогласно избирается выдающийся ученый Д. А. Поспелов. В рамках

Ассоциации проводится большое количество исследований, организуются школы для молодых специалистов, семинары, симпозиумы, раз в два года собираются объединенные конференции, издается научный журнал «Новости ИИ».

Уровень теоретических исследований по искусственному интеллекту в России соответствует мировому. Но в прикладных работах, начиная с конца 80-х гг., все больше сказывается постепенное отставание в технологии. По мнению ведущих ученых на данный момент отставание в области разработки промышленных интеллектуальных систем составляет 4-6 лет.

Основные этапы развития ИИ в [21] характеризуют следующим образом. «Период 1943-1956 – период созревания ИИ. Период с 1952 по 1963 гг. были годами восторженного энтузиазма и ожиданий. Период разочарования 1965-1970 гг., связанный с невозможностью «полного» решения практических задач ИИ из-за типичных ситуаций наступления «комбинаторного взрыва» и провалом попыток решать проблему ИИ путем создания программ типа общего решателя задач – General Problem Solver (GPS). Этот тупик заставил исследователей ИИ перейти к построению более простых специализированных систем, основанных на достаточном объеме экспертных знаний для решения конкретных трудных проблем. Такой подход привел к созданию ряда удачных приложений.

В связи с этими разработками период 1969-1979 гг. явился временем возрождения интереса к ИИ и созданием большого числа различных экспертных систем (ЭС), в том числе многих коммерческих продуктов (см. [14]). ЭС относили к системам ИИ, основанным на знаниях (СОЗ). Такие системы позволяли аккумулировать опыт, знания и умения глубоких профессионалов в различных конкретных видах деятельности и «тиражировать» эти знания. ЭС стали применять менее квалифицированные пользователи или использоваться профессионалами для повышения производительности.

Для «убеждения» клиентов в «правильности» выводов ЭС, в них стали включать подсистемы объяснения полученного решения. Постепенно термином «ЭС» стали обозначать и другие программные системы, решающие более или менее творческие задачи с помощью эффективных специализированных алгоритмов. Некоторые специалисты в области ИИ иногда используют термин «СОЗ» для ЭС.

В начале 90-х годов начинается период активного промышленного применения систем ИИ. Первая коммерческая ЭС компании DEC позволила сэкономить около 40 млн. \$ в год [14]. Начинают более интенсивно развиваться применения искусственных нейронных сетей. Сразу несколько групп исследователей «переоткрыли» алгоритм обучения сетей методом обратного распространения ошибки [1, 2], не зная о предложенном С.Ю. Масловым в России в 1964 году более универсальном алгоритме [40].

Конец 90-х годов характеризуется изменением направлений исследований в ИИ, связанных с применением более строгих математических методов вместо написания компьютерных программ для проверки различных идей [7]. Продолжается развитие в основном уже существующих теорий и обоснование новых утверждений. С использованием строгих теорем и различных экспериментов показывается адекватность известных в ИИ результатов их применения в реальном мире. В это время наблюдается значительное продвижение в развитии работ в компьютерном зрении, машинном обучении, в роботике, в представлении знаний, распознавании речи, системах планирования и методах достоверного и вероятностного рассуждения.

В этот же период начинают более активно использоваться идеи и методы искусственного интеллекта в теории и практике управления, появляются отдельные разработки на базе ИИ, включаемые в состав систем управления (СУ). Такие разработки стали называть интеллектуальными ком-

понентами (ИК) систем управления. На основе классификации ИК, приведенной в работе [7], рассмотрим кратко некоторые из них с учетом особенностей применения этих компонентов в СУ техническими системами.

В качестве наиболее используемого типа ИК для СУ становятся искусственные нейронные сети. Они применяются для решения целого класса задач, где используются не столько уравнения динамики и правила, как в традиционных экспертных системах, сколько опыт, приобретаемый в процессе взаимодействия с реальными объектами.

Искусственные нейронные сети стали активно использоваться в теории и практике управления примерно с середины 80-х годов. В системах управления применяется несколько типов ИНС: многослойный персептрон, сеть Кохонена (как однослойная самоорганизующаяся карта - Kohonen's Self-Organizing Map), сеть Хопфилда, машина Больцмана и другие [1, 3, 12, 39]. В современных условиях ИНС становятся все более часто используемым инструментом, используемым для автоматизации принятия решений. Это связано с тем, что традиционное построение алгоритмов или логических исчислений для решения многих классов задач затруднено. Затруднения вызваны сложностями учета большого числа сочетаний факторов и формализации сложных закономерностей, связывающих условия задачи с результатами.

В сложных системах автоматического управления искусственные нейронные сети хорошо поддерживают уровень быстрого реагирования в задачах оперативного управления объектами. Более мощные интеллектуальные системы могут совмещать и нейронный и логический механизмы принятия решений в сложных ситуациях [12, 13, 39].

В качестве второго типа интеллектуальных компонент систем управления рассматриваются эволюционные (генетические) алгоритмы, которые тоже являются инструментом поиска рациональных решений. Эволюционные алгоритмы (ЭА) понимаются как алгоритмы со специальными структурами

данных [1], оперирующие с популяциями индивидов. Каждый индивид представляет некоторое возможное решение из множества допустимых решений и оценивается некоторой мерой «пригодности». Новая популяция формируется итеративно путем отбора более пригодных индивидов (шаг селекции). Затем с помощью «генетических» операторов некоторые члены этой новой популяции подвергаются преобразованиям (шаг изменений) с целью образования новых решений. После некоторого числа поколений могут возникать решения, близкие к оптимальным.

Эволюционные алгоритмы довольно легко применимы в целях быстрого создания прототипов ИС для апробации решений тех или иных задач. Однако результаты использования ЭА в одних задачах могут быть очень хорошими, но плохими в других. Эволюционные алгоритмы могут комбинироваться с нейронными сетями [22].

Для реализации высокоинтеллектуальных функций систем управления используются интеллектуальные компоненты третьего типа – системы, основанные на «знаниях» (СОЗ, Knowledge-Based Systems). Например, знания могут быть представлены в некотором логическом языке. Обработка знаний с помощью логических средств позволяет получать некоторые предпочтения на множестве допустимых управлений с целью выбора одного из них.

В общем случае СОЗ оперируют с более широкой информацией логическими, объектно-ориентированными и другими моделями, основанными на знаниях экспертов. Вместе с тем СОЗ могут использовать и традиционные алгоритмы, базирующиеся на уравнениях динамики. Поэтому, как и в случае использования нейронных сетей и эволюционных алгоритмов, класс решаемых в ИС задач принципиально расширяется, по сравнению с традиционной проблематикой теории управления.

В английском языке слово «knowledge» означает не только «знание», но еще и «умение». В СОЗ умение представлено процедурной информаци-

ей, для которой прежде всего характерно исполнение. В то время как данные хранятся и пересматриваются, знания преобразовываются и применяются [3]. При этом, в отличие от данных, знания могут иметь не только информационную часть, но и описательную часть для эффективной актуализации знаний. В качестве информационных единиц знаний могут выступать встроенные процедуры, придающие знаниям активность и их первичность по отношению к процедурам [5].

В зарубежной литературе термин «СОЗ» широко используется. К системам, основанным на «знаниях», обычно относят следующие классы:

- 1) системы, основанные на правилах (Rule-Based Reasoning);
- 2) системы, основанные на автоматическом доказательстве теорем (Automatic Theorem-Proving Techniques);
- 3) системы, основанные на автоматическом гипотезировании (Automatic Hypothesizing), т.е. на порождении гипотез;
- 4) системы, основанные на рассуждениях по аналогии (Analogical Reasoning);
- 5) объектно-ориентированные интеллектуальные системы (Object-Oriented Intelligent Systems).

В СОЗ реализуется относительно высокоуровневый процесс умозаключений, а современные искусственные нейронные сети и генетические алгоритмы обычно не включают в понятие «СОЗ», хотя они входят в составе комплексных многоуровневых систем управления. ИНС являются относительно низкоуровневыми средствами интеллектуализации, но одновременно и более быстродействующими, реализуя после соответствующей настройки «рефлекторное» реагирование на входную информацию. Обработывая дополнительно результаты, полученные с помощью логических средств, можно обеспечить более высокоуровневую обработку информации. Характерный пример: автоматическое двухуровневое распознавание образов (например, типов автомашин с борта летательного аппарата).

С точки зрения качества автоматического продуцирования и обработки знаний, системы, основанные на автоматическом доказательстве теорем (АДТ), существенно превосходят другие СОЗ. Но при этом следует помнить, что интеллектуальная сила систем АДТ демонстрируется чаще на примере статических и достаточно хорошо формализуемых задач из разных разделов математики (прежде всего, дискретной математики, алгебры, логики).

Основным механизмом интеллектуализации в СОЗ является тот или иной механизм рассуждений. Схема взаимосвязи основных механизмов рассуждений представлена на рис. 1 (адаптировано из работы [7]). Для повышения эффективности работы СОЗ на практике часто применяют комбинированные механизмы рассуждений, построенные на базе основных. Более подробно работа отдельных механизмов СОЗ рассмотрена в теме 3.

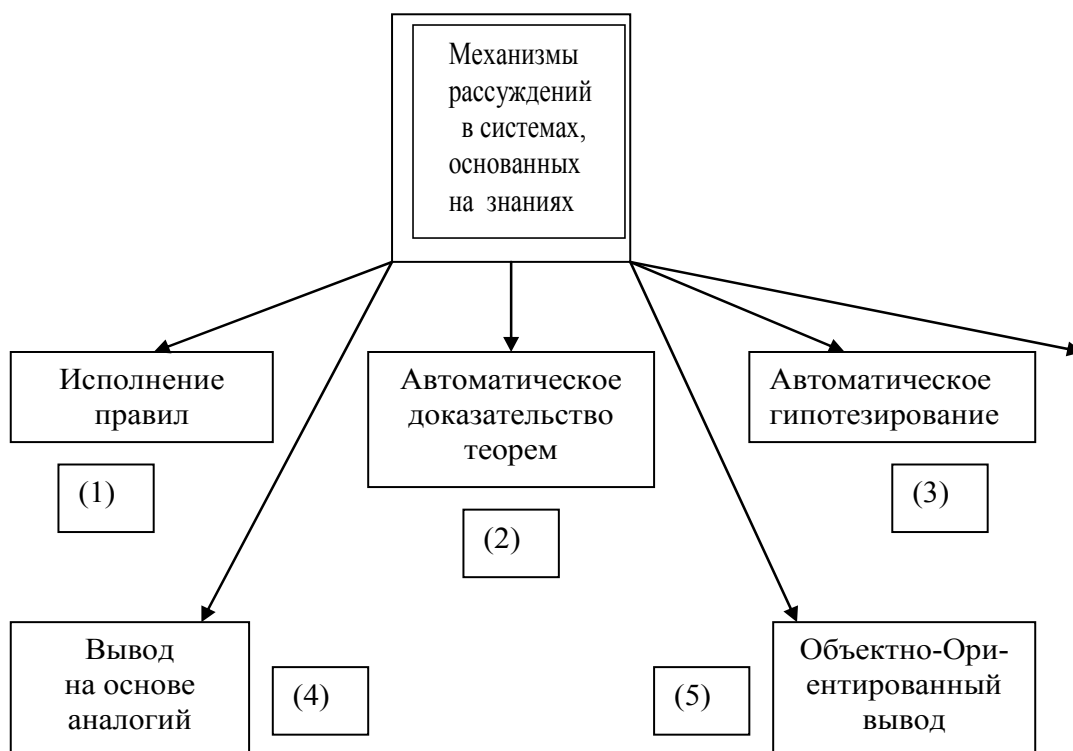


Рис. 1. Некоторые механизмы рассуждений, используемых в системах, основанных на «знаниях»

Какие же особенности применения методов ИИ имеются в задачах управления техническими объектами и системами?

При управлении в сложных технических системах обычно выделяют и реализуют несколько уровней управления, отличающихся решаемыми задачами и целями. Интеллектуальные системы управления включают в себя задачи управления и задачи целеполагания в изменяющихся условиях управления. В системах интеллектуального управления (СИУ) недостающая информация о целях и об изменении целей управления формируется более высоким уровнем общей интеллектуальной системы управления.

При создании большинства систем управления используются, как правило, постановки и методы решения задач управления на базе традиционных математических моделей, в которых динамика управляемого процесса описывается в форме различных уравнений (дифференциальных, конечно-разностных и других). Так как модели и результаты любого моделирования всегда являются приближенными, они не полностью учитывают информацию о реальной динамике систем. В правой части уравнений динамики могут постоянно действовать некоторые неизвестные возмущения. В отношении таких возмущений исследователи часто делают различные смелые априорные предположения об их малости в каждый момент времени, или в среднем, или интегрально. Однако влияние малых возмущений иногда совсем не мало и иногда может потребоваться их учет, например, при исследовании задач устойчивости и др.

Разработка систем управления, построенных на базе понятий адаптивности и робастности также позволяет учитывать влияние недостающей априорной информации по динамике системы, получая ее в режиме реального времени на этапе обучения или управления. В ряде случаев успешное управление в условиях с неполной информацией достигается при использовании нейронных сетей, методов автоматического гипотезирования, обучения, а также применения других методов и средств ИИ.

Однако, существуют классы задач управления, в которых существуют достаточно сложные зависимости, не имеющие обычного аналитического представления, например, в системах, существенно использующих экспертные суждения и знания человека. При управлении в таких системах вместо количественных методов или в дополнение к ним применяются логический и лингвистический подходы. В соответствии с этими подходами допускаются в качестве значений переменных не только числа, но и слова или предложения искусственного или естественного языка [8-16, 29].

В отличие от жесткой предписанной последовательности действий в алгоритмических системах, системы, использующие логические исчисления, допускают различные варианты выбора управления, определяемого некоторыми общими правилами и имеющейся априорной и текущей информацией.

Логический подход может рассматриваться как некоторая альтернатива [24] системному подходу в математическом описании систем, переходу к использованию новых, логических моделей динамики и управления. Конкуренции таких подходов с традиционными численными методами математики, механики, физики способствовало значительное развитие в последние годы символьных аналитических вычислений.

Разработка новых технологий позволяет использовать описанные выше интеллектуальные компоненты, а также другие эффективные средства ИИ в перспективных системах управления. При этом значительно расширяются потенциальные возможности проектирования и управления динамическими системами. Появляются возможности эффективного решения классов задач с неизвестными или изменившимися уравнениями динамики, а также задач, в которых модели в форме традиционных уравнений динамики менее эффективны, чем модели искусственного интеллекта. Подобные задачи возникают, например, в объектах с автоматической рекон-

фигурацией при автоматическом выборе эффективных каналов управления, в задачах планирования действий в изменяющейся среде и др.

Взаимопроникновение методов теории управления и искусственного интеллекта осуществляется в общей области этих дисциплин, все чаще именуемой в англо-язычной литературе интеллектуальным управлением (Intelligent Control). Слово «интеллектуальное» означает, что качество управления должно иметь некоторое сходство с качеством, демонстрируемым человеком.

На рисунке 2 представлены некоторые разделы интеллектуального управления, как пограничной области (3), включающей методы теории управления (1) и методы искусственного интеллекта (2).



Рис. 2. Разделы интеллектуального управления и искусственного интеллекта

На стыке теории управления (1) и нейронных сетей (14) возникло нейроуправление (13). Оно также является разделом интеллектуального управления.

Управление на основе «знаний» образует из разделов (3) и (4) подраздел интеллектуального управления (5), который включает управление на основе правил (7) с подразделом (9) по нечеткому управлению на основе нечетких правил (8) и управление на основе логических моделей (11), включающее управление с применением автоматического доказательства теорем (13). Среди систем управления, основанных на применении АДТ, находятся, к примеру, системы нечеткого вывода резолюционного типа. Правда, из-за ряда нелогических элементов нечеткой «логики» возможности нечеткого варианта метода резолюций сравнительно ограничены. Более эффективен композиционный метод вывода, успешно применяемый в многочисленных работах по нечеткому управлению [22].

Рассмотрим кратко общие цели интеллектуального (и интеллектуально-го) управления. По мнению ряда исследователей [21] эти цели состоят в следующем:

«- полнее использовать доступное знание об объекте и среде, чтобы обеспечить надежное управление с predetermined критерием (например, зная желаемую траекторию, функционал качества управления, целевое множество);

- управлять в творческой (интеллектуальной) манере (подобно человеку), прогнозируя изменения в объекте и среде, сохраняя, например, путем реконфигурации технических и программных средств управления работоспособность объекта даже при больших изменениях, согласовывая и, может быть, пересматривая цели и критерии качества управления;

- улучшать с течением времени способность управлять объектом путем аккумуляции экспериментального знания, т.е. путем обучения на опыте...».

Термин «интеллектуальное управление» становится все более широко употребляемым и в отечественной научно-технической литературе. Однако, как отмечалось в [7], нет единого мнения в вопросе определения этого понятия, а также в выборе более или менее типовой структуры (архитектуры) соответствующих интеллектуальных систем управления.

Довольно распространенным является определение [5, 27]: «система интеллектуального управления должна иметь способность воспринимать (to comprehend) информацию о процессах, возмущениях и условиях функционирования, выводить заключения (to reason) и обучаться».

В [4, 5, 7] содержится ряд интересных результатов по интеллектуальному управлению, полученных в мире в рассматриваемой области с приложениями: в роботике, управлении производством, в космосе, авиации, химическом производстве и вообще в промышленности. В частности, представлена информация о применении интеллектуальных компонентов для улучшения качества управления геометрией и двигателями самолета при отказах и повреждениях.

В упомянутых выше работах отмечалось, что «...предпринимаются многочисленные попытки комбинирования интеллектуальных компонентов разных классов. Так, интересное применение нейронных сетей в сочетании с системами, основанными на знаниях (экспертными системами) предложено в работе [39]. Нейронная сеть функционирует параллельно с экспертной системой, которая поначалу полностью задействована и ответственна за управление. С течением времени, нейронная сеть обучается от экспертной системы и перехватывает управление, обеспечивая более высокое качество. Эта гибридная система предназначена для использования в исследованиях других планет. Экспертная система обеспечивает робастность функционирования, а нейронная сеть – тонкую настройку на специфические условия планеты, которые заранее с достаточной точностью неизвестны. Если произойдут их резкие изменения, то система вновь переда-

ет управление экспертной системе и процесс обучения начинается заново...».

Взаимовлияние теории управления и искусственного интеллекта дано в [7] и схематично показано на рис. 3. Рисунок демонстрирует, как два отдельных источника интеллектуального управления развивались отдельно во времени, а затем соединились и образовали перспективное направление развития теории и практики управления.



Рис. 3. Два основных источника интеллектуального управления

Считается, что, несмотря на уже имевшийся опыт цифрового и логического (автоматного) управления, до 80-х годов было трудно объединить возможности методов теории систем автоматического управления (САУ) и искусственного интеллекта из-за их несовместимости [1].



Рис. 4. К определению интеллектуального управления как пограничной области исследования

С одной стороны, непрерывные модели САУ (часто линейные и с неограниченными переменными), с другой стороны, использование дискретных методов искусственного интеллекта, которые базировались на

двоичных (булевских) переменных, т.е. на пропозициональных языках, близких к автоматным и достаточно традиционным для восприятия специалистами-управленцами. Позднее, системы управления, основанные на нечетких логиках, и нейроуправлении составили проблематику нелинейных систем с непрерывными и ограниченными интервалом $[0,1]$ (или $[-1,+1]$) переменными. Это открыло новый этап, где теория управления и искусственный интеллект становятся значительно более совместимыми.

Другое видение интеллектуального управления представлено в работах G.N. Saridis. Он рассматривал интеллектуальное управление как пограничную область не только теории управления и искусственного интеллекта, но также и исследования операций. На рис. 4, показано стрелками взаимное влияние этих трех базовых областей исследований и разработок.

По оценке различных экспертов [21], наибольшие успехи по прикладному использованию теории интеллектуального управления, особенно в области СОЗ на нечетких правилах [22], достигнуты японскими фирмами. Создается и развивается специальная вычислительная техника для поддержки эффективного функционирования систем, основанных на интеллектуальных компонентах и, прежде всего, на нечетких правилах и нейронных сетях. «Нечеткие» микропроцессоры и нечеткое управление нашли многочисленные приложения в автомобилях, бытовой технике, химическом производстве и т.д. В настоящее время в мире насчитываются тысячи практических разработок подобных систем интеллектуального управления. Ожидается еще больший рост популярности нейроуправления с ростом выпуска средств аппаратной реализации средств нечеткого управления и нейронных сетей на СБИС.

Тема 1. ОСНОВНЫЕ ПОНЯТИЯ И ТЕРМИНОЛОГИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Рассмотрим определения основных понятий и терминологию интеллектуальных систем, введенных в работах различных авторов [1, 2, 5, 6, 17] с использованием соответствующих авторских иллюстраций и комментариев.

Определение 1. «Интеллектуальной называется система, способная целеустремленно¹, в зависимости от состояния информационных входов, изменять не только параметры функционирования, но и сам способ своего поведения, причем способ поведения зависит не только от текущего состояния информационных входов, но также и от предыдущих состояний системы».

По этому определению, интеллектуальная система в общем случае представляется в виде некоего «черного ящика»², на информационные входы которого поступают сигналы о внешней среде, поведение системы наблюдается по измерениям на информационных выходах и зависит от всех состояний системы в настоящем, прошлом и от ее параметров. При этом в общем случае предполагается, что точная структура, параметры системы и алгоритмы изменения структуры и параметров нам неизвестны. Известно только, что эти изменения происходят целенаправленно, так чтобы в системе формировалась или корректировалась конкретная цель, а поведение системы соответствовало достижению сформированной цели.

¹ Различают целенаправленные и целеустремленные системы. Примером системы первого типа может служить артиллерийский выстрел, второй – самонаводящаяся ракета. Суть принципа целеустремленности состоит в корректировании значений входов системы (через обратную связь) на основе оценки получаемых результатов (выходов).

² Введенное У.Р. Эшби наименование объекта исследования, внутреннее устройство которого неизвестно или не принимается во внимание. Модель объекта строится на основе его реакций на воздействия, поступающие на входы извне.

На рис. 1.1 приведен общий вид такой системы [17]. Примером такой интеллектуальной системы может служить любой живой организм. В то же время многие обычные технические системы (ТС) не являются интеллектуальными. При создании ТС стремятся обеспечить повторяющиеся, предсказуемые выходные реакции ТС на одинаковые входные сигналы³.

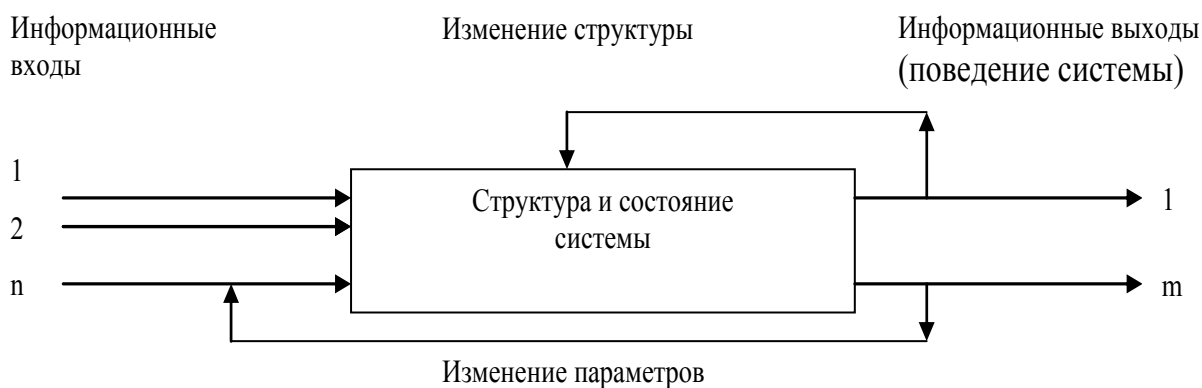


Рис. 1.1. Интеллектуальная система как «черный ящик»

Например, в системе автоматического управления температурой и расходом воды в теплосети регулируется подача топлива в нагреватель и давление на выходе насоса. При аварии на трубопроводе система будет стремиться поддержать давление и температуру в заданных пределах в соответствии с «повышением» расхода воды, воспринимая это как нормальную ситуацию по увеличению потребления горячей воды. Накопление «опыта» с помощью простой регистрации параметров аварии не сможет ничем помочь в аналогичных ситуациях. В то же время в интеллектуальной системе возможен углубленный анализ данных о текущей ситуации, внезапном увеличении расхода воды, несовпадении общего баланса потребления воды на отдельных узлах и участках теплосети и т.д. с целью

³ В технических системах, как правило, о внутреннем устройстве объекта исследования либо известно частично, либо существуют некоторые гипотезы (модель «серого ящика»). В отличие от «черного ящика» эта модель неизвестна или не принимается во внимание. Модель объекта строится на основе его реакций на воздействия, поступающие на входы извне.

проверки гипотезы о наличии аварийной ситуации и принятии соответствующих управляющих решений.

Определение 2. «Интеллектуальной называется система, моделирующая на компьютере мышление человека».

Это определение возникло в 60-х гг. прошлого века, когда усилия многочисленных исследователей были направлены в основном на моделирование работы мозга человека на компьютере. Создавались различные модели клеток мозга – искусственных нейронов, строились компьютерные модели сетей из этих нейронов и изучалось поведение таких систем. На вход моделирующих программ подавались данные реальных измерений, соответствующих некоторым электрическим сигналам, поступающим на вход клетки мозга в живом организме, а на выходе снимались результаты, которые сверялись с данными экспериментальных исследований клеток. В зависимости от рассогласования результатов проводилась адаптация параметров и структуры моделей. Повторение большого количества циклов такого «обучения» приводило к тому, что результаты работы имитационной программы «внешне» постепенно приближались к результатам работы очень маленького элемента мозга человека.

Однако к 80-м гг. XX века идея воспроизведения работы мозга на компьютере потерпела полную неудачу. Тем не менее была разработана теория нейронных сетей, а нейросетевой подход к решению отдельных задач в управлении техническими системами доказал свою полезность на целом ряде практических приложений и, в первую очередь, в задачах распознавания образов и предсказания значений параметров различных моделей.

После того, как исследователи поняли, что создать универсальный искусственный разум невозможно, стали создавать достаточно узкоспециализированные ИС, которые бы дополняли и расширяли возможности человека, а не заменяли его. Появилось новое определение ИС.

Определение 3. «Интеллектуальной называется система, позволяющая усилить интеллектуальную деятельность человека за счет ведения с ним осмысленного диалога».

Таким образом, произошел поворот в постановке задачи на создание и применение ИИ. Вместо требований к машине «мыслить как человек», возникла идея «получения хороших результатов». Небольшая часть исследователей продолжала заниматься моделями, воспроизводящими процессы мышления человека или работу структур головного мозга. Большая часть работ по ИИ стала проводиться по моделям, построенным на базе собственных принципов организации и методов обучения. В рамках этого направления удалось создать ряд систем, демонстрирующих результаты, «похожие на человека».

Подобные системы позволили использовать уникальные способности человека и, в тоже время, освободиться от свойственных человеку недостатков по скорости реакции, возможностям запоминания и т.д.

Компьютер обладает огромной памятью, совершает миллионы операций в секунду и может мгновенно реагировать на возникновение различных событий и ситуаций. В то же время компьютер сам по себе не способен мыслить и отвечать за собственные поступки.

Следовательно, разграничение сторон интеллектуальной деятельности человека и машины состоит в обеспечении с помощью компьютера мощной поддержки принятия решений человеком [1, 14-16, 36-37]. Компьютер должен служить квалифицированным советчиком человеку, обеспечивать быстрый анализ ситуаций, генерировать наиболее эффективные варианты действий и предлагать их человеку. Рассмотрение предложенных вариантов и пояснения целесообразности выбора вариантов оставались бы за человеком. С учетом полученных от человека дополнительных разъяснений или изменения целей компьютер может вновь проанализировать варианты действий и выдать новые варианты. Окончательный выбор подхо-

дящего варианта остается за человеком, и он несет ответственность за его реализацию. Осознание такого понимания функций интеллектуальной системы привело к следующему более развернутому определению ИС [1]:

Определение 4. «Под интеллектуальной системой понимается объединенная информационным процессом совокупность технических средств и программного обеспечения, работающая во взаимодействии с человеком (коллективом людей) или автономно, способная на основании сведений об окружающей среде и собственном состоянии при наличии знаний и мотивации синтезировать цель, принимать решение о действии и находить рациональные способы достижения цели».

В этом определении ИС подчеркивается роль знаний и мотивации для синтеза целей, что принципиально отличает ИС от систем интеллектуального управления (СИУ), где цели заданы.

Поясним эту ситуацию на примере системы автоматического предотвращения столкновений в воздухе. С помощью локаторов система анализа воздушной обстановки обнаруживает в своей зоне наблюдения летящие объекты, определяет их курс, высоту, строит прогнозы траекторий движения всех объектов и выдает информацию человеку-оператору о возможных траекторных угрозах прежде, чем человек успеет оценить опасность столкновения некоторых объектов. Информация об этом появляется на пультах управления диспетчеров, и человек с учетом всей дополнительной информации (наличия топлива, погодных условий и т.д.) принимает решение (подтверждает или нет) рекомендации системы о передаче на летящие объекты изменений в рекомендуемых параметрах движения. В случае, полностью автоматического управления существовала бы реальная опасность отправить, например, на повторный круг самолет, у которого заканчивается запас топлива или возникла какая-либо нештатная ситуация на борту. В случае, если бы анализом ситуаций занимался только человек, то могло быть упущено время для безопасного расхождения самолетов.

На рис. 1.1. показано, что общая модель системы имеет произвольное количество входов и произвольное количество выходов, некоторую структуру и параметры, которые могут изменяться в зависимости от измеренных значений и сравнения текущего и желаемого состояния системы. Результаты сравнения зависят от природы исходных сравниваемых величин и применяемых методов сравнения. Изменения структуры системы могут поддерживаться разными способами, например, методом исключения малозначимых элементов, практически не влияющих на информационные выходы. Чтобы определить и сравнить альтернативные состояния объекта, необходимо измерить характеристики его входов и выходов. Измерение выполняется с помощью измерительных средств и некоторой алгоритмической операции, ставящей в соответствие данному наблюдаемому состоянию объекта, процесса или явления некоторое обозначение, число или символ.

Последующие возможности обработки данных в значительной мере зависят от способов измерения данных и выбора соответствующего типа измерительной шкалы. В порядке возрастания мощности шкал в теории измерений определяют 6 основных типов шкал: номинальная, порядковая, интервальная, циклическая, отношений, абсолютная. Краткие сводные характеристики различных видов шкал, данные в работе [17], приведены в табл. 1.

Номинальная шкала, называемая также шкалой наименований или классификационной шкалой, позволяет рассматривать и сопоставлять любым двум различимым состояниям объекта объекты разные обозначения и применять соответствующие алгоритмы измерения, а для неразличимых состояний приписать одинаковые обозначения.

Номинальную шкалу можно определить соотношениями:

1. Либо $A = B$, либо $A \neq B$;
2. Если $A = B$, то $B = A$;
3. Если $A = B$ и $B = C$, то $A = C$.

Здесь символ $=$ обозначает отношение эквивалентности; а в случае, если A и B – числа, то это знак равенства.

Для обозначения классов на номинальной шкале можно использовать слова, символы, цифры, их различные комбинации (например, на шкале размеров: малый, средний, большой и т.д.). Порядок следования классов не имеет значения, можно производить проверку совпадения классов, но не выполнение арифметических операций над ними.

Следующий тип шкал это порядковые шкалы (сильного, слабого и частичного порядков), часто применяемые в социологических исследованиях и позволяющие указать некий порядок следования классов, но не расстояния между ними.

Интервальные шкалы позволяют уже оперировать с числами. На этих шкалах известны расстояния между любыми двумя делениями (квантификаторами). Длина интервала не зависит от его расположения на шкале. Шкала отношений позволяет выполнять арифметические действия над значениями квантификаторов, как над числами (длина, вес, деньги).

Абсолютная шкала является уникальной, имеет абсолютный нуль и абсолютную единицу, т.е. является «самой сильной» (числовая ось). Применяя более сильную шкалу можно получить больше сведений об изучаемом объекте, но при этом необходимо выбирать тип шкалы, адекватный виду измерений.

Совокупности измерений (элементы данных) образуют базу исходных данных (БД), характеризующих состав и структуру имеющегося состояния системы и позволяющих сравнивать это состояние с целевым состоянием. Обычно сравнивают или вектора, или множества, используя различные меры «близости», такие как меры Минковского, коэффициенты ковариации, корреляции, ранговой корреляции по Кендалу или Спирмену (для числовых рядов), мера Танимото (для сравнения множеств), расстояние Хэмминга (множеств с одинаковым количеством элементов) и другие меры [27].

Типы измерительных шкал

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

Выбор той или иной меры определяется целью сравнения, размерностью измеряемых величин, типом применяемых шкал и степенью зависимости между величинами системы. Например, в качестве меры для сравнения точек в некотором пространстве признаков можно использовать меры Минковского:

$$\|X - Y\| = \sqrt[p]{\sum w_i |X_i - Y_i|^p},$$

где X и Y – две точки в n -мерном ($i=1,2,\dots,n$) пространстве признаков), w_i – веса предпочтений признаков, p – показатель, характеризующий вид меры. При $p=2$, имеем обычную евклидову меру, при $p=1$ – так называемую манхеттенскую метрику и т.п.

Используя различные меры близости и сходства можно сравнивать объекты по наборам признаков, хранящимся в БД, строить определенные зависимости, модели, использовать результаты сравнения данных для образования новых знаний.

Знания – основной термин теории искусственного интеллекта. Часто знания определяют как отношения между элементами данных. Знания получаются в результате применения к исходным данным (декларативной информации) некоторых методов обработки с использованием внешних процедур – присоединенных процедур или процедурной информации. Необходимость установить отношения между сущностями приводит к применению новых подходов и методов, реализуемых с помощью специальных программных средств. Работа программы искусственного интеллекта позволяет делать «вывод на знаниях», или «вывод новых знаний». В ИИ рассматриваются различные виды знаний [2, 6]. Основные виды – это декларативные и процедурные знания (см. Глоссарий). Возникают знания в результате сопоставления информационных единиц (ИЕ), определения и устранения противоречий между ними. Знания характеризуются рядом свойств, таких как: внутренняя интерпретируемость (уникальное и одно-

значно определяемое имя ИЕ); структурированность в виде установления определенных отношений (однозначных и многозначных) между ИЕ, типа «является частью...», «род-вид» и т.д.; принадлежность к некоторому пространству (не обязательно метрическому). Существуют различные методы и модели представления знаний, рассматриваемые ниже.

Следующий термин ИИ – состояние. Отдельные ИЕ и вся система в целом могут находиться в некоторых состояниях. Например, выключатель включен или выключен, система работает или нет и т.д. Некоторые системы могут иметь бесконечное число состояний.

Цель – основная характеристика систем ИИ. Интеллектуальные программы в отличие от обычных вычислительных программ не содержат алгоритмов, а содержат только цель и правила движения к цели. Алгоритм строится по ходу достижения цели с учетом конкретных ситуаций.

Модель представления знаний – основной тип модели ИИ. Реализация конкретных систем, основанных на знаниях, происходит в рамках одной или нескольких моделей представления знаний или языков представления знаний. Реальные системы часто состоят из нескольких классических моделей, ограничений и условностей.

Предметная область – это область человеческой деятельности, в которой решаются различные задачи с помощью подходов и методов ИИ.

Определения других понятий и терминов приведены в глоссарии.

Тема 2. РАЗЛИЧНЫЕ ПОДХОДЫ К ПОСТРОЕНИЮ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В настоящее время при построении систем ИИ и выборе метода представления знаний используются различные подходы или их комбинации. Рассмотрим кратко основные из них.

2.1. Логический подход

Является наиболее распространенным. Возникновение этого подхода связано напрямую со способностями человека к развитому логическому мышлению, отличающему его принципиально от животных. Основы логического подхода начали развиваться от логики Аристотеля, исчисления высказываний и Булевой алгебры. В дальнейшем были введены современные понятия формальных аксиоматических систем и на этом базисе развито классическое исчисление предикатов. Булева алгебра была существенно расширена за счет введения предметных символов, отношений между ними, кванторов существования и всеобщности. Исходные данные стали представляться в базах данных и базах знаний в виде фактов, аксиом и правил логического вывода отношений между ними.

Фактически все построенные на логическом принципе ИС реализуют машины доказательства теорем. Каждая такая машина имеет блок генерации цели и систему вывода, пытающуюся доказать данную цель как теореме. Если цель удастся доказать, то используя трассировку примененных правил можно получить необходимую цепочку действий для реализации сгенерированной цели. «Интеллектуальная мощь» подобной системы определяется возможностями генератора целей, машины доказательства теорем и эффективностью конкретных применяемых методов и алгоритмов.

При этом известно, что использование исчисления предикатов (ИП) позволяет добиться более полноценной реализации задач ИИ, чем применение только исчисления высказываний (ИВ). Однако и в рамках ИВ возможно добиться большей выразительности логического подхода, если использовать нечеткую логику (НЛ). Основным отличием НЛ является то, что правдивость высказываний может принимать в ней не только значения категорических высказываний типа «да», «нет» (1/0), но и ряд промежуточных значений. Например, значения типа: «мало вероятно» – (0.25), «не знаю» – (0.5), «более вероятно» – (0.75) и др. Используя данный подход можно точнее представить реальные ситуации и мышление человека в условиях неполной информации.

В то же время в вычислительном плане классический логический подход для большинства реальных задач характеризуется большой трудоемкостью из-за возможного полного перебора вариантов во время поиска доказательства. Классический вариант данного подхода из-за возможного «комбинаторного взрыва» обеспечивает решение задач только при сравнительно небольшом размере базы данных и требует эффективной реализации вычислительного процесса.

2.2. Структурный подход

Название структурного подхода связано с попытками построения ИИ путем моделирования на ЭВМ структуры человеческого мозга, включающей модели нейронов мозга и нейронных сетей. Одной из первых таких реализаций структурного подхода был простейший персептрон Фрэнка Розенблатта, предложенный им и исследованный в 1957-1959 гг. для распознавания зрительных образов. Позднее это направление развилось в теорию распознавания образов и «искусственные нейронные сети» (ИНС). Разработанные модели ИНС различаются по строению отдельных нейронов, вариантам топологических связей между ними и применяемым алго-

ритмам обучения ИНС. Несколько подробнее эти вопросы рассмотрены в темах 7 и 8 данного пособия.

2.3. Эволюционный подход

Эволюционный подход используется также достаточно широко при построении систем ИИ. В этом подходе основное внимание уделяется построению начальной модели и правилам, по которым она может изменяться (эволюционировать). При этом модель может быть составлена с использованием самых различных методов, включая нейронные сети, наборы логических правил и любые другие модели. Кроме правил в программе определяются также критерии оценки качества каждого варианта.

После запуска программы с начальным вариантом модели начинается итеративное построение и оценка вариантов. На основании проверки и оценки качества моделей, полученных на каждой итерации, отбираются наиболее удачные из них. На основе удачных вариантов и правил генерируются новые модели, из которых опять выбираются самые удачные и т. д.

Некоторые исследователи считают, что существуют не эволюционные модели, а эволюционные алгоритмы обучения. Тем не менее, модели, полученные при эволюционном подходе, выделяют в отдельный класс, как имеющие ряд характерных особенностей. Основной особенностью является замена построения модели на разработку алгоритма ее модификации по формальным критериям. Обычно отмечают также, что полученные модели фактически не добавляют новых знаний для систем ИИ.

В настоящее время это направление получило развитие под названием «генетические алгоритмы». Согласно [7] «...генетический алгоритм – это алгоритм, оперирующий с популяцией индивидов $P(t) = \{x_{it}, \dots, x_{nt}\}$, где $t = 1, 2, \dots$ номер итерации. Каждый индивид представляет некоторое возможное решение из множества допустимых решений S , а каждое решение x_{it} оценивается заданной мерой его «пригодности». На очередной ите-

рации $t+1$ формируется новая популяция путем отбора более пригодных индивидов – выполняется шаг селекции, а за ним следует шаг изменений. На шаге изменений отдельные члены новой популяции подвергаются преобразованиям с помощью «генетических» операторов, и образуются новые решения. При этом используются два типа преобразований: унарные (одноместные) преобразования типа мутаций $m_k: S \rightarrow S$, приводящие к появлению новых индивидов путем малых изменений одного индивида; преобразования перекрестного типа $C_j: S_n \rightarrow S$, порождающие новые индивиды путем комбинирования «составных частей» нескольких индивидов. При отсутствии заикливания после генерации некоторого числа поколений могут возникать рациональные решения, приближающиеся в асимптотике к оптимальным...».

В настоящее время разработано много различных модификаций «генетических алгоритмов», образующих целый класс методов. К этому классу можно отнести и так называемый метод группового учета аргументов (МГУА), который активно пропагандировался в нашей стране академиком А.Г. Ивахненко еще в 80-х гг. прошлого века. Одна из известных первых попыток применения генетических алгоритмов в компьютерах была не слишком удачная попытка организовать с помощью мутаций обучение небольшой программы на ФОРТРАНе. Алгоритм реализации случайного выбора операторов приводил для большинства мутаций программы к неисполняемому коду.

Современные усовершенствованные рабочие варианты генетических алгоритмов позволяют в ряде случаев получать достаточно эффективные результаты при разумном ограничении множества допустимых наборов варьируемых операторов. Рассмотрим небольшой пример.

Предположим, необходимо построить граф, удовлетворяющий некоторым требованиям (синтез структуры системы управления, оптимальный маршрут обхода роботом препятствия, оптимальную топологию размеще-

ния торговой сети и т.п.). Каждый индивид – это граф. В качестве начальных данных эволюционной программы задается: начальная популяция графов $P(I)$, порождаемая случайно или эвристически; оценочная функция, учитывающая требования задачи по оценке пригодности каждого графа и формализующая отношения предпочтения (хуже, лучше) на множестве индивидов; - множество мутационных операторов, преобразующих граф. Перекрестные операторы комбинируют структуры двух или более графов. Например, если искомый граф должен быть связным и ациклическим (т.е. деревом), тогда некоторый мутационный алгоритм, зависящий от специфики задачи, может удалять какие-то ребра и для связывания двух возникающих подграфов добавлять всякий раз некоторое новое ребро. Мутации могут быть и более общими, т.е. нарушающими свойство графа «быть деревом», но тогда это свойство должно учитываться оценочной функцией для отбрасывания на шаге селекции графов «не деревьев». В результате работы такого генетического алгоритма можно получить рациональный вариант искомого графа.

2.4. Имитационный подход.

Имитационный подход используется при построении различных систем ИИ. Подход основан на введенном У.Р. Эшби классическом базовом понятии кибернетики «черного ящика», рассмотренного в теме 1. Модель такого объекта исследования строится на основе его поведения, реакций на воздействия, поступающие извне на его входы, характеризует связи между реакциями и вызвавшими их воздействиями и внешне имитирует способность человека копировать поведение других, не понимая, как это происходит и почему. Основным недостатком имитационного подхода (как и эволюционного) является низкая информационная способность большинства построенных моделей (в смысле понимания структуры и параметров модели «внутреннего» устройства черного ящика).

Следует иметь в виду, что при построении систем искусственного интеллекта для управления в технических системах классическая модель черного ящика используется редко. О реальных технических объектах и системах обычно все же имеется та или иная априорная и текущая информация. Поэтому в применениях ИИ к техническим системам чаще рассматривают так называемые «серые», или даже «белые» ящики.

Согласно [5]: «серый ящик» – объект исследований, о внутреннем устройстве которого либо известно частично, либо существуют некоторые гипотезы. В отличие от черного ящика модели серого ящика учитывают помимо связей между реакциями и внешними воздействиями и все те частичные сведения, которые известны о его строении. Соответственно, «белый ящик» – объект исследований, о внутреннем устройстве которого имеются формальные знания или данные, например, в виде дифференциальных уравнений. Таким образом, фактически речь идет об объектах с различным уровнем неопределенности наших знаний об их структурах и параметрах.

Наличие тех или иных знаний и данных о самом объекте еще не гарантирует нам возможность хорошего управления объектом. Во-первых, любое знание и любая модель всегда лишь приближенно отражают реальный объект. Во-вторых, необходимы знания о возможных действующих на объект возмущениях и, наконец, необходимо иметь или уметь формировать цели интеллектуальной системы для различных уровней управления.

В связи с этим при построении сложных интеллектуальных систем часто используют смешанные (комбинированные) подходы к построению систем ИИ. В таких системах часть работ выполняется на базе одного подхода, а часть – другого, и таким образом очень четкой границы между различными подходами на практике нет. Особенно наглядно это видно при построении систем ИИ, включающих различные уровни управления.

На верхних уровнях управления для принятия решений чаще используются логические подходы. Задачами систем нижних уровней управления является обеспечение взаимодействия ИС с внешней средой – получение и первичная обработка информации и формирование реакций ИС в зависимости от решений, принятых на верхних уровнях. Поэтому в системах нижнего уровня, где происходит обработка первичной информации, чаще используются структурный, эволюционный и имитационный подходы и соответствующие решения, «подсмотренные» человеком в живой природе. Это позволяет в условиях требований реального времени организовать более эффективное распознавание зрительных, звуковых, тактильных и других первичных информационных образов, решать задачи идентификации, моделирования развития ситуаций в реальном и ускоренном времени, организовать программные решения отдельных задач и .д.

Так, например, структуры современных систем технического зрения во многом копируют оптическую зрительную систему глаза животных и человека. С помощью матрицы светочувствительных элементов (аналог светочувствительных клеток глаза – палочек и колбочек) воспринимаются отдельные элементы изображения. Применяя различные графические фильтры можно с помощью искусственной нейронной сети реализовать предварительную обработку видеосигналов, затем выделить отдельные элементы изображения – горизонтальные, вертикальные, диагональные линии, контуры, различные области со светлыми, темными или цветными элементами, моделируя в технической системе работу отдельных функций мозга.

Таким образом, по аналогии с иерархическим принципом управления в живых организмах, в системах ИИ можно также строить и использовать разбиение решения общей задачи системы на отдельные подзадачи, реализуемые в соответствующих подсистемах.

В отличие от живой природы подсистемы ИС могут быть построены на базе различных подходов, обеспечивающих реализацию отдельных функций ИС аналогично имеющимся функциям у человека. При этом можно даже расширить некоторые функциональные возможности видеть или слышать в любом диапазоне волн, поддерживать с высокой точностью и скоростью заданные координаты управления различными исполнительными механизмами, например, управлять движением робота или поддерживать эффективное управление опасными процессами на границе безопасности и т.д. При таком подходе упрощается построение ядра системы ИИ и ее отдельных подсистем.

Общее сложное решение глобальных задач заменяется на более эффективное решение более простых подзадач, распределенных между несколькими уровнями и подсистемами. Высший уровень ИИ (также как и высший уровень нервной системы, связанный с большими полушариями мозга) ставит лишь общую задачу (цель), определяет стратегию решения задачи и определяет промежуточные подцели для подзадач, решаемых на более низких уровнях управления.

В качестве примера рассмотрим схему решения тестовой задачи управления роботом – взятия детали заданного типа из контейнера, содержащего различные типы деталей [4]. Для поставленной глобальной цели на верхнем уровне управления роботом определяются общие сценарии решения всей задачи и отдельных подзадач, определяются тактические подцели и проверяемые условия выполнимости отдельных подзадач. Действия двигательных подсистем робота, реализующих решение поставленной задачи, определяются на более низких уровнях управления (также как и инстинктивные реакции у человека).

Затем начинается поэтапная реализация общего (стратегического) сценария и тактических сценариев отдельных подзадач на среднем и нижних уровнях управления. Подключаются информационные подсистемы

робота, системы технического зрения, распознавания образа заданной детали, определения ее координат в контейнере с учетом пространственной ориентации и т.д. Система навигации и датчики положения робота и его отдельных частей определяют координаты начального положения робота и его отдельных частей, в частности, левого и правого схвата.

После этого этапа решаются задачи определения возможных опорных траекторий перемещения центра масс робота и движения его отдельных частей с учетом возможных траекторных препятствий, определяется выбор рабочего схвата (левый или правый) и траекторий его перемещения. И только после этого начинается решение задач детализации построения движений робота. Таким образом, видно, что решение этих задач осуществляется на существенно более низких уровнях управления, аналогично тому, как происходит детализация построения движений у человека на уровнях более низких, чем командный уровень коры больших полушарий. При этом в интеллектуальной системе все время осуществляется контроль выполнимости каждой задачи сценария, производится корректировка параметров движения по результатам показаний датчиков и, при необходимости, модифицируется сценарий или выбирается другой, либо производится откат к решению задачи более верхнего уровня управления.

В целом можно сделать общий вывод о том, что существующие в настоящее время различные подходы и реализующие их методы, алгоритмы и устройства позволяют достаточно успешно моделировать работу нижних уровней человеческого интеллекта. Создание систем с полноценным верхним уровнем управления требует пока еще приложения достаточно больших усилий в развитии и применении логического подхода для создания перспективных систем ИИ, основанных на «знаниях».

Тема 3. СИСТЕМЫ, ОСНОВАННЫЕ НА ЗНАНИЯХ

Рассмотрим кратко классификацию систем, основанных на «знаниях» (СОЗ). Как отмечалось в предыдущей теме, эти средства ИИ являются перспективными для реализации верхних (высокоинтеллектуальных) уровней управления. Классификация СОЗ приводится по материалам работы [7].

3.1. Системы, основанные на правилах.

Примерами систем, основанных на правилах, являются следующие достаточно распространенные в области ИИ системы:

3.1.1. Системы на базе продукционных правил (типа «если выполняются условия .А.. , то делай .В.. , иначе делай..С..»).

3.1.2. Системы на нечетких правилах, формулируемых в удобных для человека качественных терминах, а именно в терминах нечетких понятий: много, мало, хорошо, плохо, неизвестно, и т.п.).

3.1.3. Системы логического программирования.

Нечеткие правила обычно имеют семантику «условие – действие» и поэтому являются частным случаем продукционных правил (продукций), которые могут и не использовать нечетких понятий. На основе нечетких правил могут делаться нечеткие логические заключения. Использование нечеткой логики и нечетких понятий позволяет в ряде задач упростить процесс представления знаний.

Системы, основанные на продукционных правилах, используются обычно в режиме интерпретации. В соответствии с некоторой стратегией правила просматриваются одно за другим и, если рассматриваемое на оче-

редном шаге правило применимо (т.е. его условия выполняются), то сразу исполняется и предписываемое этим правилом действие.

Системы 3.1.1, 3.1.2 хорошо применять при моделировании действий квалифицированного оператора, хорошо знакомого со всеми особенностями объекта управления (ОУ) и успешно справляющегося с управлением им «вручную».

Важным свойством, расширяющим возможности систем 3.1.1, 3.1.2, является то, что действия могут включать вставки и удаления из базы знаний новых и устаревших знаний или, соответственно, данных из базы данных. Это позволяет охватить ряд задач реального мира с изменяющимися закономерностями – задачи с динамической моделью. В некоторых системах предусматривается механизм выбора действия, если могут быть применены несколько правил одновременно. Этот механизм может основываться на априорном ранжировании правил по приоритетам или основываться на различных эвристиках. Например, можно потребовать не применять одно и то же правило повторно при тех же значениях аргументов либо предпочитать правила, применимые к более свежим (позднее выведенным) фактам, либо выбирать какие-то специфические правила типа «выбирай второе из двух правил» и т.д., например, в такой ситуации получим:

«Если Мотоцикл (x), то имеются Колеса (x,2)»,

«Если Мотоцикл (x) и Мотоцикл с коляской (x), то имеются Колеса (x,3)».

В системах логического программирования правила тоже имеют вид импликаций, но более широко используется классический логический формализм. Примером систем типа 3.1.3 является Пролог (Prolog) [19]. В системах 3.1.3 правила обычно имеют вид «конъюнкция атомов $\& A_i$ влечет атом В », где атомы могут содержать не только константы, но и перемен-

ные (по смыслу, связанные кванторами всеобщности). Соответствующий язык, включающий также и просто атомы (в частности, факты), называется хорновским языком [2, 5].

В соответствии с процедурной семантикой хорновских дизъюнктов «условие – действие» импликация $\bigwedge_i A_i \rightarrow B$ понимается как процедура, тело которой $\bigwedge_i A_i$ состоит из процедурных вызовов A_i . Тогда решение задачи B можно свести к решению подзадач A_i . И, наконец, процедуры с пустым телом необходимы для окончательного прямого решения задачи. Это соответствует тому, что в системах 3.1.3 используется в отличие от 3.1.1 и 3.1.2 не прямой вывод, а обратный. В этом случае на основе атомов, входящих в запрос, отыскивается правило, условие которого (после подходящих подстановок вместо переменных) следует из этого множества атомов. По этому правилу выводится его заключительный атом, добавляемый в использованное множество атомов, и т.д.

Таким образом создается возможность использования так называемого декларативного стиля программирования, когда предметная область описывается в форме хорновских формул (описывая, что дано и что требуется, без указания алгоритма решения), а затем задается вопрос, на который интерпретатор Пролога находит ответ.

На практике для повышения эффективности поиска выводов используются еще различные нелогические элементы. В системах логического программирования с ограничениями (ЛПО – Constraint Programming) используется дополнительный механизм удовлетворения ограничениям (в частности, решениям логических уравнений) [5-7]. Это позволяет оперировать данными в декларативном стиле не только с задачами логического, но и численного характера. Все это расширяет область приложения логиче-

ского программирования и обеспечивает более совершенное управление логической программой. В работе [8] описан интересный проект использования ЛПО в качестве основы управления в реальном времени для создания системы полностью автоматического пилотирования самолетов.

С точки зрения эффективности вывода, в системах 3.1.3 довольно продвинутым является проект параллельных вычислений – PIM (Parallel Inference Machine) [20]. Этот проект являлся частью японского проекта компьютерных систем 5-го поколения и показал скорость около 64 млн. логических операций в секунду (LIPS).

3.2. Системы, основанные на автоматическом доказательстве теорем

На возможность и перспективность использования в теории и практике управления автоматического доказательства теорем (АДТ) указывал еще в 1960 г. А.А. Фельдбаум [43]. Среди систем, основанных на АДТ, наиболее популярными в области ИИ являются следующие:

3.2.1) системы резольюционного типа, реализующие в разных модификациях метод резолюций. Этот метод является машинно-ориентированным и составляет теоретическую базу языков логического программирования типа Пролог;

3.2.2) системы генценовского типа, представляемые либо в форме так называемого естественного (натурального) вывода, либо в форме секвенциального вывода. В сравнении с резольюционным естественный вывод больше напоминает манеру рассуждений человека. Секвенции играют роль задач и выводы имеют вид некоторых конечных многошаговых процессов последовательной декомпозиции задач на подзадачи до получения тривиальных подзадач.

Метод резолюций и аналогичный ему так называемый обратный метод, были предложены независимо С.Ю. Масловым [40] в 1960 г. и Дж. Робинсоном в 1965 г. [41]. Так называемый «естественный вывод» был

введен в 1934 году Г. Генценом и С. Яськовским и являлся одним из самых ранних исчислений после работ Т. Фреге. В нашей стране одной из первых и наиболее известных систем АДТ была машинная реализация естественного вывода, рассмотренная в [42].

В [7] отмечается, что «...в отличие от систем логического программирования системы АДТ обычно работают в полной первопорядковой логике и существенно превосходят сегодня другие средства ИИ с точки зрения сложности доказываемых ими теорем. С его использованием были решены некоторые открытые задачи, поставленные известными математиками, причем уровень сложности неуклонно повышается. Однако, эти достижения в области систем АДТ относятся в основном к сфере оф-лайн – задач, не имеющих характерных ресурсных ограничений для он-лайн задач реального времени. Таким образом, это в основном сфера решения математических задач в области программирования и автоматизации строгих рассуждений, допускающих формализацию в некотором логическом языке, но не учитывающих ресурсов, отпущенных на проведение полных автоматических расчетов. Последнее не означает, что методы АДТ принципиально не применимы к задачам реального времени. Например, так же как и в логическом программировании [8], известны попытки применить АДТ в управлении движущимися объектами для построения полностью автономных систем...».

Известны примеры реализации такого типа систем [7]. Например, одна из ранних (1970 г.) известных систем управления мобильным интегральным роботом STRIPS. Этот самоходный аппарат совершал передвижение в упрощенной среде по формируемым в устройстве управления (УУ) командам. Типичной задачей, решаемой STRIPS, являлась задача «убрать деталь в контейнер», т.е. задача перемещения детали из некоторой точки рабочего пространства с помощью схвата робота в контейнер. STRIPS умел перемещать схват из одного места в другое, схватывать де-

таль (когда схват и деталь находятся в одном месте) и переносить схваченную деталь в контейнер.

В данном случае АДТ применялся на интеллектуальном уровне управления, обеспечивающего в режиме реального времени планирование действий и формирование последовательности команд для достижения поставленной цели. Такой уровень интеллектуального управления более гибко поддерживается моделями в форме логических исчислений, чем жесткими алгоритмами. Тем не менее, даже в этом простом примере для повышения эффективности работы авторам пришлось применять нелогические элементы: вставки новых знаний и удаления устаревших. Для погружения всего описания в полное исчисление предикатов потребовалось ввести для исчисления ситуаций дополнительную переменную времени в качестве дополнительного аргумента к предикатам, истинность которых меняется в процессе функционирования робота. При этом система АДТ должна быть достаточно мощной.

3.3. Системы, основанные на автоматическом гипотезировании

Автоматизация гипотезирования (выдвижения гипотез) в системах искусственного интеллекта может достигаться по-разному. В частности, автоматическое гипотезирование (АГ) используется в литературе в контексте реализации такой фундаментальной функции, как обучение [1, 7]. Применение обучения оправдано при принятии решений в условиях полностью или частично неизвестной изменяющейся среды.

Системы с элементами АГ обладают, как правило, априорными знаниями, которые можно пополнять различными способами. Весьма важным представляется накопительный (кумулятивный) вид обучения, при котором накопление знаний происходит с использованием примеров и подкреплений и улучшает способность к обучению (с «учителем» или «без учителя»). В настоящее время разработано достаточно много различных под-

ходов и методов обучения, и в обобщенной постановке обучение может быть сведено к достаточно точному представлению некоторой функции.

Индуктивным обучением называется обучение на основе некоторого множества предъявленных «учителем» примеров, включающих пары аргументов и значений. Выдвигаемые конкретные предположения о виде этой неизвестной функции называются гипотезами. На рис. 3.1. приведен процесс кумулятивного обучения с использованием и дополнением запаса априорных знаний и проверкой гипотез на исходных примерах и данных прогнозирования с учетом всей информации.

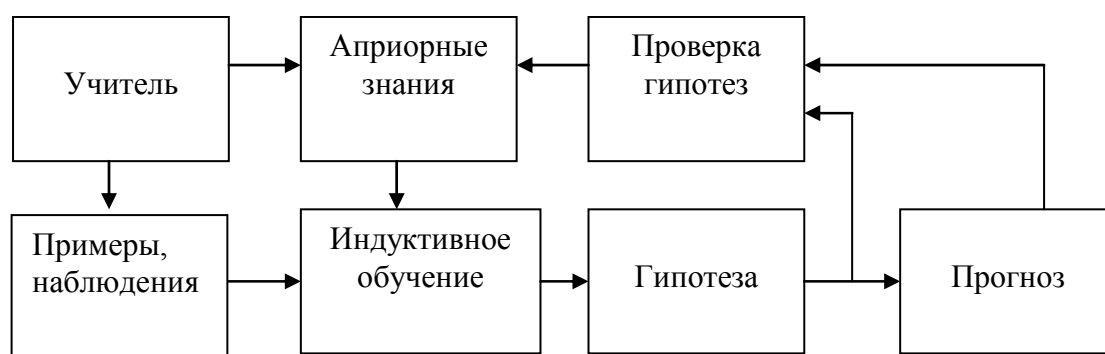


Рис. 3.1. Процесс кумулятивного обучения

Трудности обучения в значительной мере зависят от вида «фактической» функции и от выбранного вида представления (полиномы, логические формулы, булевские функции, вероятностные или нейронные сети и др.). Описание соответствующих методов (обучающихся деревьев решений, адаптивных вероятностных сетей и др.) можно найти в [44].

Другой тип задач обучения возникает, когда среда еще беднее, т.е. обучаемый не получает примеров и, начиная действовать, может временно в порядке обратной связи получать подкрепления (поощрения). Соответствующая проблематика давно занимала специалистов в области коллективного поведения автоматов и адаптивного управления.

Примером кумулятивного обучения является обучение на основе обобщающего анализа («объяснения примеров»). Этот метод извлечения общих правил из индивидуальных наблюдений и некоторой общей теории рассматривал С. Samuelsson. В [7] приводится анализ примера взятия производной по x от x^2 в логическом программировании. Для этого требует 136 шагов доказательства, в которых 99 являются тупиковыми. В интеллектуальной системе было бы естественно, выполнив однажды все эти шаги, впредь сделать это быстрее, а не просто запоминанием в базе данных пары вход – выход вида $(x^2, 2x)$. В противном случае при вычислении производной по y от y^2 пришлось бы снова повторить заново все 136 шагов. Система должна быть способна сформулировать итоговое правило, общее для всех выражений u^2 с неизвестными арифметическими выражениями u , а еще лучше – для u^n .

Одной из новых областей в ИИ является индуктивное логическое программирование (ИЛП). Оно комбинирует индуктивные методы с мощностью первопорядковых представлений, концентрируя их, в частности, на представлении теорий как логических программ. С начала 90-х годов ИЛП стало главной частью исследований в области машинного обучения, благодаря строгости и предложенным полным (универсальным) алгоритмам порождения первопорядковых теорий из примеров [41, 44]. Одним из методов ИЛП является обратная резолюция, позволяющая, например, находить одну из двух посылок (дизъюнктов) по известной второй посылке и их резольвенте (т.е. логическому следствию посылок по правилу резолюции Дж. Робинсона [41]). При этом без потери полноты применяется ряд ограничений для сокращения комбинаторики. Замечательной особенностью систем обратной резолюции является то, что они порождают новые предикаты. По мере того, как новый предикат входит в состав все новых гипотез, он становится все более и более семантически осмысливаемым и

может получить от человека отождествление с некоторыми из известных понятий или новое наименование, если потребуется [7].

Известен и ряд некоторых других методов АГ. В работе [44] описаны логические исчисления с обобщенными кванторами. Эти исчисления применены для формализации так называемых рациональных индуктивных выводов, для построения основ вычислительной статистики и для разработки на стыке математической логики и математической статистики соответствующего метода автоматического образования гипотез GUHA-метода.

Метод АГ на основе логических уравнений первого порядка, предложенный С.Н. Васильевым [7], также относится к одному из перспективных направлений в обеспечении кумулятивного обучения. В той же работе [7] описан метод последовательного порождения гипотез (ПП метод), представляющий интересное сочетание методов решения логических уравнений и автоматического доказательства теорем, а также дано описание систем, основанных на рассуждениях по аналогии.

3.4. Системы, основанные на рассуждениях по аналогии

Вместо использования примеров для их обобщения (п. 3.3) с последующим применением полученного гипотетического знания для решения задачи нередко можно использовать эти примеры более или менее напрямую, способом рассуждения по аналогии.

Эта форма рассуждений варьируется в литературе от формы правдоподобных рассуждений, основанных на степенях правдоподобия или «похожести» (Similarity) [41] до «ленивого» обучения путем объяснения примеров («Lazy» Explanation-Based Learning). Последняя форма рассуждений по аналогии состоит как бы в приспособлении направления обобщения старого примера под потребности новой задачи и наиболее выражена в

рассуждениях по прецеденту (Case-Based Reasoning) и подход, основанный на производной аналогии [5].

Механизм рассуждений в системах, основанных на прецедентах, базируется на коллекционировании и использовании решений старых задач рассматриваемой предметной области для построения решений новых задач [5], что позволяет порой избежать повторной трудоемкой обработки информации. Адаптируемое к новой задаче старое решение выбирается как решение такой ранее решенной задачи, которая в определенном смысле достаточно близка к новой задаче. При этом понятие «близости» к прецеденту может быть формализовано по-разному. Часто принятие решений по прецедентам комбинируется с другими механизмами, например, с логическим программированием [24].

3.5. Объектно-ориентированные интеллектуальные системы

Достоинством декларативного стиля представления знаний и программирования (см. п. 3.1) является, как отмечалось, то, что создателю системы не надо заботиться о потоке управления в программе. По существу, описание задачи представляется слабо структурированной совокупностью отношений. При большом количестве таких отношений понимание Пролог – программы становится практически невозможным, а встраивание нелогических элементов в стратегию вывода – неизбежным [5,6].

Вместе с тем тенденция все более широкого использования ЭВМ в процессах управления приводит на практике к созданию весьма сложных систем управления. С этой точки зрения объектно-ориентированного подхода к представлению и обработке знаний (с его возможностями высокоэффективно поддерживать отношения наследования, использовать значения «по умолчанию» и т.д.) является привлекательным, по крайней мере, в некотором симбиозе с логическим программированием.

Объектно-ориентированные интеллектуальные системы используют декларативно-процедурные (описательно-алгоритмические, дескриптивно-

конструктивные) формы представления знаний и быстрые, хотя и ограниченные по возможностям вывода нового знания, алгоритмы вывода свойств объектов на основе иерархических, сетевых, фреймовых и некоторых другие представления знаний. Для этого используются также языки KRL, KLONE и некоторые другие. Для них характерно двухуровневое представление данных [23] (абстрактная модель предметной области в виде иерархии множеств понятий и конкретная модель ситуации как совокупность взаимосвязанных экземпляров этих понятий), представление связей между понятиями и закономерностей в виде присоединенных процедур, семантический подход к сравнению образцов и поиску по образцу. Инструментом, предназначенным специально для создания экспертных систем, явились языки представления знаний (и соответствующие программные среды) типа ART, OPS5 [14]. Они используют как продукции, так и фреймы.

Фрейм – это структура данных, предназначенная для представления стереотипной ситуации. Он представляет собой совокупность вопросов, которые можно задать о соответствующей воображаемой ситуации. Эта совокупность вопросов должна быть минимальной с точки зрения еще сохранения сущности описываемой ситуации. Другого типа фреймы (ролевые) описывают некоторые процессы, т.е. служат для представления процедурного знания (умений). С ролевыми фреймами ассоциируется, например, информация о том, чего ожидать в следующий момент, что сделать, если ожидания не подтвердятся и т.п.

Следует заметить, что рассмотренная классификация СОЗ 1-5 не является жестким разбиением. Конкретные системы, основанные на «знаниях», могут принадлежать одновременно нескольким классам. В частности, весьма перспективные средства ИИ имеются в пересечении классов 1, 2, 5. Внутри него находятся СОЗ, использующие объектно-логические языки, фреймовые логики (F-logics), логики транзакций (Transaction Logics) [5] и

т.д. Последовательное сочетание объектно-ориентированного подхода с логическим программированием позволяет повысить эффективность последнего с сохранением свойств универсальности (полноты) и корректности обработки знаний [20]. На рис. 3.2, воспроизведенным из работы [7], отображено пересечение классов СОЗ, использующих первые 3 механизма рассуждения (классы СОЗ 4 и 5 на рисунке не представлены). Области пересечения А принадлежат, в частности, системы логического программирования (Пролог и другие), области Б – GUHA-метод, области В – ПП-метод, области Г – некоторые системы ЛПО.



Рис. 3.2. Пересечение классов СОЗ

На рис. 3.2 иллюстрируется, что пересечения А-Г классов 1-3 не пусты. Например, принадлежащая области А система типа Пролог имеет одновременно черты систем, основанных на правилах, и систем доказательства теорем (на множестве хорновских формул исчисления предикатов). Системы ЛПО являются их дальнейшим развитием, включая, например, механизм решения простейших логических уравнений. Так как отыскание решений логических уравнений может иметь характер обзора гипотез, то некоторые системы ЛПО по своим возможностям попадают в наиболее общую область Г.

Тема 4. ОСНОВЫ ЛОГИЧЕСКОГО ВЫВОДА

4.1. Основные понятия математической логики

Основы логического вывода и особенности их применения в различных исчислениях рассматриваются с различной степенью детализации в работах ряда авторов. Ниже дано краткое изложение основ логического вывода, следуя терминологии и обозначениям работ [1, 7, 44].

Для определения формальных аксиоматических систем необходимо задать [44]:

1) Язык системы, в котором определяется алфавит языка, как некоторое множество символов и правил построения из символов этого алфавита элементов языка, называемых формулами определяемого языка.

2) Аксиомы системы. Некоторые из формул языка формальной системы выделяются, как ее аксиомы.

3) Правила вывода системы. Определяется некоторое конечное множество правил $\{R_i\}$ – отношений между формулами, причем, если для формул F_1, \dots, F_T, F_{T+1} выполнено отношение $R_i(F_1, \dots, F_T, F_{T+1})$, то формула F_{T+1} называется непосредственным следствием из F_1, \dots, F_T по правилу R_i .

Кроме того, определение формальных систем должно также удовлетворять требованию, чтобы существовал метод, с помощью которого можно определить, является ли любая данная последовательность символов формулой языка системы, любая данная формула – аксиомой системы и находятся ли формулы F_1, \dots, F_T, F_{T+1} в отношении R_i . Определив язык, алфавит, правила вывода, синтаксис можно задать исчисление высказываний как некоторую логическую теорию. Следуя [17] напомним основы исчисления высказываний.

«...Высказывание (утверждение) – грамматически правильное предложение, взятое вместе с выражаемым им смыслом и являющееся истин-

ным или ложным. Атомарное высказывание – простое, неразложимое, не включает других высказываний в качестве своих частей. Сложное высказывание получается из простых с помощью логических связок: отрицание, конъюнкция (И), дизъюнкция (или), исключаящая дизъюнкция.

Одну из разновидностей простых высказываний – категорические высказывания (суждения) – рассматривал Аристотель. В этих высказываниях утверждается или отрицается наличие какого-то признака у всех или у некоторых предметов рассматриваемого класса. Особенностью именно категорических высказываний в том, что не только устанавливается связь предмета и признака, но и дается количественная характеристика субъекта, т.е. «все», «некоторые». Ниже в табл. 4.1 приведены основные формы и примеры классических категорических высказываний.

Таблица 4.1

| Функторы | Обозначения | Пример интерпретации |
|------------------------|-------------|----------------------------------------|
| «все...есть...» | a | SaP – все собаки лают |
| «некоторые...есть» | i | SiP – некоторые собаки есть кусачие |
| «все...не есть...» | e | SeP – все собаки не есть глупые |
| «некоторые ...не есть» | o | SoP – некоторые собаки не есть игрушки |

Каждое из этих четырех выражений (a, i, e, o) Аристотель рассматривал как логические постоянные, не имеющие самостоятельного содержания и позволяющие получить содержательные истинные или ложные простые высказывания из двух обладающих содержанием имен.

При составлении сложных высказываний, осуществлении логического вывода (получения следствия из двух посылок) необходимо заботиться не только об истинности заключения, но и о его осмысленности. Для формального анализа осмысленности можно использовать так называемый «логический квадрат» (рис. 4.1). На логическом квадрате можно выделить

особые фигуры (логического вывода), следуя которым получается гарантированно осмысленный результат.

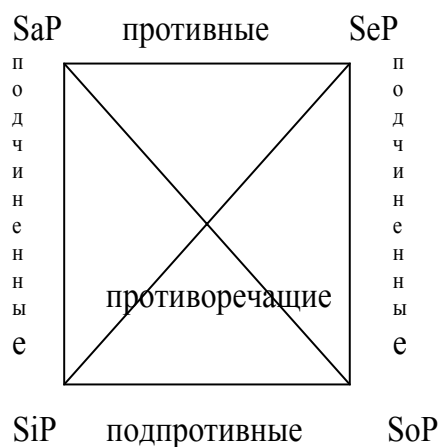


Рис. 4.1

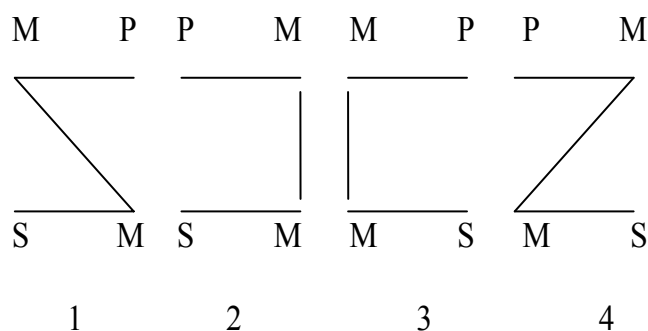


Рис. 4.2

Логический квадрат задает правила вывода (отношения между формулами). Отличительной особенностью исчисления высказываний является то, что в качестве терминов (S,P) могут выступать только логические переменные. В исчислении предикатов на месте термина может быть целый предикат. Правильный вывод от истинных посылок всегда ведет к истинному заключению. Если хотя бы одна из посылок является ложной, правильное рассуждение может давать в итоге как истину, так и ложь. Неправильный вывод от истинных посылок может вести как к истинным, так и ложным заключениям.

Аристотель рассматривал силлогизмы – дедуктивные умозаключения, в которых из двух категорических высказываний выводится новое категорическое высказывание. В каждом силлогизме должно быть 3 термина: меньший (субъект S), больший (предикат P) и средний (термин M, присутствующий в посылках, но отсутствующий в заключении). В зависимости от положения среднего термина различают 4 фигуры силлогизма (рис. 4.2). Разновидности фигур, отличающиеся характером посылок и заключений, называют модусами. Всего возможно 256 модусов, из которых только 24

правильные, включающие в себя 5 ослабленных⁴. Правильные модусы имеют названия⁵: фигура 1 – *barbara, celarent, darii, ferio, barbari¹, celaront¹*; фигура 2 – *cesare, camestres, festino, baroco, cesaro¹, camestros¹*; фигура 3 – *darapti, disamis, datisi, felapton, bocardo, ferison*; фигура 4 – *bramantip, camenes, dimaris, fesapo, fresison, cameno¹*. Классический пример (фигура 1, модус *barbara*): Все люди (человеки) смертны (все М есть Р), Сократ – человек (все S есть М), Сократ смертен (все М есть Р).

4.2. Аксиомы исчисления высказываний

Рассмотрим основные законы и наиболее важные аксиомы в исчислении высказываний, следуя [17].

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

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

Ассоциативность: $\alpha_1 \& (\alpha_2 \& \alpha_3) \equiv (\alpha_1 \& \alpha_2) \& \alpha_3$

$$\alpha_1 \vee (\alpha_2 \vee \alpha_3) \equiv (\alpha_1 \vee \alpha_2) \vee \alpha_3.$$

Законы Де Моргана:

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

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

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

Аксиомы классического исчисления высказываний (ИВ):

$$\alpha \rightarrow (\beta \rightarrow \alpha),$$

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

$$(\alpha \& \beta) \rightarrow \alpha,$$

$$(\alpha \& \beta) \rightarrow \beta,$$

⁴ В ослабленных модусах силлогизма заключения являются в некоторых модусах частноутвердительными или частноотрицательными высказываниями, а в других модусах – общеупотребительными или общеотрицательными.

⁵ Выделенные жирным шрифтом буквы соответствуют обозначениям и соответствующим функторам в табл. 4.1.

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

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

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

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

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

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

Правила вывода.

1. Из истинности условия импликации и истинности самой импликации следует истинность следствия импликации (модус поненс⁶):
 $\alpha, \alpha \rightarrow \beta \vdash \beta$.

2. Из формулы $\alpha(p)$ выводима формула $\alpha(P)$, получающаяся постановкой P вместо p (правило подстановки): $\alpha(p) \vdash \alpha(P)$.

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

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

⁶ Modys ponens (лат.) – «правило отделения».

- из истинности дизъюнкции и отрицания одного из ее дизъюнктов следует истинность формулы, получающейся из дизъюнкции удалением этого дизъюнкта (простая резолюция или удаление дизъюнкта):

$$\alpha \vee \beta, \neg \beta \vdash \alpha;$$

- из истинности двух дизъюнкций, одна из которых содержит дизъюнкт, а другая ее отрицание, следует формула, являющейся дизъюнкцией исходных формул без упомянутого дизъюнкта его отрицания (резолюция):

$$\alpha \vee \beta, \neg \beta \vee \gamma \vdash \alpha \vee \gamma \text{ или, эквивалентно, } \neg \alpha \rightarrow \beta, \beta \rightarrow \gamma \vdash \neg \alpha \rightarrow \gamma.$$

Для оценки истинности формул в современных ИС используются следующие методы:

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

Для определения тождеств часто используют алгоритм редукции, основанный на доказательстве путем приведения к абсурду. Алгоритм удобен, если формула содержит много импликаций. Вопрос о выводимости некоторой формулы сводится тогда, согласно принципу дедукции, к анализу невыполнимости множества ее дизъюнктов. Таким образом, невыполнимость формулы можно проверить, порождая логические следствия из нее до тех пор, пока не получится пустой дизъюнкт. Данный метод основан на применении принципа резолюций. Рассмотрим простые примеры

ПРИМЕР 1. Даны формулы A, B, C. Будем считать истинными дизъюнкты (1) $(A \vee C)$ и (2) $(B \vee \bar{C})$.

1. Предположим, что $C = \langle \text{И} \rangle$ (Истина), тогда первый дизъюнкт $(A \vee C) = \text{И}$ при любом A . Подставляя $C = \text{И}$ во второй дизъюнкт, получим $B \vee L$ и, следовательно, $B = \text{И}$, (т.к. весь дизъюнкт истинен).

2. Предположим, что $C = \langle \text{Л} \rangle$, тогда второй дизъюнкт истинен при любом B , а первый принимает вид $A \vee L$ и, следовательно, $A = \text{И}$, (т.к. весь дизъюнкт истинен).

Таким образом, не зависимо от C $(A \vee B) = \text{И}$, и новый дизъюнкт (резольвенту) можно добавить в базу знаний, исключая контрарные формулы C и \bar{C} .

ПРИМЕР 2. Даны дизъюнкты $(p \vee q \vee \bar{r})$ и $(\tau \vee r)$, получаем резольвенту $(p \vee q \vee \tau)$.

ПРИМЕР 3. Дизъюнкты (p) и (\bar{p}) дадут в сумме пустой дизъюнкт.

Данный алгоритм легко формализуется и используется на ЭВМ. Для исключения возможности закливания машины из-за порождения одного и того же дизъюнкта неограниченное число раз в специальные языки вводят операцию «отсечение».

Тема 5. МЕТОДЫ ВЫВОДА В ИСЧИСЛЕНИИ ПРЕДИКАТОВ

5.1. Основы классического исчисления предикатов

Понятие классического исчисления предикатов (ИП) сформировалось на основе развития логики как науки о методах рассуждений и получило современное оформление к концу XIX века после работ Дж. Буля, де Моргана, Ч. Пирса, Пеано, Г. Фреге и программы Гильберта [27] по обоснованию непротиворечивости математики. Появление компьютеров привело к возрастанию роли логики, как средства представления знаний и решения задач, активизации исследований в области формальных выводов, поиска «универсального языка» для механизации заключений. Теоретические исследования по теории алгоритмов и сложности вычислений показали принципиальные и практические ограничения для таких глобальных проектов, как Универсальный Решатель Задач (GPS) [14]. Появилась потребность в новых логических подходах, развитии неклассической логики, построении новых языков и исчислений для решения новых задач.

Основой построения новых подходов и исчислений является классическое исчисление предикатов, краткое представление о котором дано ниже в минимально необходимом объеме [7, 44].

5.2 Определение языка исчисления предикатов

Рассмотрим определение языка исчисления предикатов L_0 .

Символами алфавита языка исчисления предикатов являются:

- 1) логические связки \rightarrow («влечет»), \neg («не») и вспомогательные символы (скобки, запяты);
- 2) счетное множество предметных переменных X_1, X_2, \dots для обозначения произвольных объектов некоторого мира (универсума);
- 3) множество (возможно, пустое) предметных констант a_i для обозначения фиксированных объектов;

4) счетное множество функциональных букв f_j^n ($n, j \geq 1$) для порождения на базе первичных объектов t_1, \dots, t_n типа x_i, a_i некоторых первичных производных объектов $f_j^n(t_1, \dots, t_n)$, а также производных объектов на базе первичных и ранее порожденных производных объектов; все первичные и производные объекты именуется далее термами;

5) непустое множество предикатных букв P_j^n ($n, j \geq 1$) для обозначения отношений между n термами, как элементарных утверждений о свойствах объектов универсума;

6) символ \forall («для любого»), для построения утверждений вида «для любого x имеет место свойство $F(x)$, где $F(x)$ – некоторое более простое утверждение), с помощью которого образуются выражения $\forall x$;

7) символ \exists («для некоторых»), для построения утверждений вида «для некоторых x имеет место свойство $F(x)$, где $F(x)$ – некоторое более простое утверждение), с помощью которого образуются выражения $\exists x$.

Выражения $\forall x$ и $\exists x$ называются, соответственно, кванторами общности и существования.

Функциональные буквы, примененные к переменным и константам, порождают термы:

- 1) всякая предметная переменная или предметная константа есть терм;
- 2) если f_j^n – функциональная буква и t_1, \dots, t_n – термы, то $f_j^n(t_1, \dots, t_n)$ есть терм;
- 3) других термов нет.

Формулы исчисления предикатов определяются следующим образом:

- 1) всякая элементарная формула $P_j^n(t_1, \dots, t_n)$, где P_j^n – предикатная буква, t_i – термы, есть формула;
- 2) если A, B – формулы, x – предметная переменная, то выражения $A \rightarrow B, \neg A, \forall x A$ есть формулы;
- 3) других формул нет.

Множество формул, определенное выше, назовем языком L_0 исчисления предикатов первого порядка.

V есть собственная подформула формулы A и пишется $V \subset A$, если формула V использовалась при построении A .

V есть подформула формулы A и пишется $(V \subseteq A)$, если она либо совпадает с A , либо является ее собственной подформулой.

Скобки применяются для указания способа, которым построена формула. Например, скобки позволяют отличить формулу $A \rightarrow (B \rightarrow C)$, построенную из подформул A и $B \rightarrow C$, от формулы $(A \rightarrow B) \rightarrow C$, построенной из $A \rightarrow B$ и C .

Областью действия квантора $\forall x$ называется подформула A в формуле $\forall x A$, иногда говорят, что квантор $\forall x$ управляет формулой A или ее подформулами.

Чтобы указать факт, что A не содержит вхождений переменной x , пишут $A]x[$ (« A не содержит x »). В общем случае, через $A]M[$, где M – множество переменных, обозначается отсутствие в A вхождений всех переменных из M .

Связанным называется вхождение переменной x в данную формулу, если оно является частью входящего в эту формулу квантора $\forall x$ или находится в области действия входящего в эту формулу квантора $\forall x$ (управляется этим квантором).

Свободным называется вхождение переменной x в данную формулу в противном случае.

Например, в формуле

$$P_1^2(x_1, x_2) \rightarrow \forall x_1 P_1^1(x_1) \quad (4.1)$$

первое вхождение переменной x_1 является свободным, а второе и третье – связанными.

Свободной переменной называется переменная в данной формуле, если существуют свободные ее вхождения в эту формулу.

Связанной переменной называется переменная в данной формуле, если существуют связанные ее вхождения в эту формулу.

Переменная может быть одновременно свободной и связанной в одной и той же формуле, как, например, переменная x_1 в (4.1).

Замкнутой называется формула, если в ней нет свободных переменных.

Пусть X_{i_1}, \dots, X_{i_k} – переменные, $A \in L_0$. Независимо, являются ли эти переменные свободными в A и существуют ли в A другие свободные переменные, будем употреблять запись $A(X_{i_1}, \dots, X_{i_k})$ для обозначения формулы A . Тогда через $A(t_1, \dots, t_k)$ будем обозначать последующий результат подстановки термов t_1, \dots, t_k соответственно вместо всех свободных вхождений в A переменных X_{i_1}, \dots, X_{i_k} , либо использовать более подробную запись результата подстановки в виде: $A(t_1/X_{i_1}, \dots, t_k/X_{i_k})$.

Терм t называется свободным для переменной x в формуле A , если никакое свободное вхождение X в A не лежит в области действия никакого квантора $\forall y$, где y входит в t .

Примеры.

1) Терм X_j свободен для X_i в $P_1^1(X_i)$ (нет вообще кванторов в данной формуле), но не свободен для X_i в $\forall X_j P_1^1(X_i)$.

2) Терм $f_1^2(X_1, X_3)$ свободен для X_1 в $\forall X_2 P_1^2(X_1, X_2) \rightarrow P_1^1(X_1)$,

так как нет кванторов с X_1 и X_3 , управляющих вхождениями переменной X_1 , но не свободен для X_1 в $\forall X_3 \forall X_2 P_1^2(X_1, X_2) \rightarrow P_1^1(X_1)$ (квантор с X_3 управляет первым свободным вхождением X_1).

5.3 Аксиомы и правила вывода исчисления предикатов

Из языка L_0 выделяются следующие аксиомы исчисления предикатов (A.n):

$$(A.1) A \rightarrow (B \rightarrow A)$$

$$(A.2) (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C));$$

$$(A.3) (\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B);$$

$$(A.4) \forall x A(x) \rightarrow A(t),$$

где t – произвольный терм, свободный для x в A ;

$$(A.5) \forall x (A \rightarrow B) \rightarrow (A \rightarrow \forall x B),$$

здесь A не содержит x . A, B, C – произвольные формулы.

Аксиомы А.1–А.5 называют логическими для отличия их от собственных аксиом, возникающих при построении на базе исчисления предикатов конкретных теорий.

Например, частным случаем аксиомы (А.4) является $\forall x \forall y A \rightarrow \forall y A$.

Правилами вывода в исчисления предикатов являются:

1) правило МР («модус поненс»): из A и $A \rightarrow B$ непосредственно следует B ;

2) правило обобщения: Gen («generalization»): из A непосредственно следует $\forall x A$.

В исчислении предикатов выводом из множества гипотез Γ называется последовательность A_1, \dots, A_n формул, такая, что для любого i формула A_i есть либо частный случай какой-либо аксиомы, либо элемент множества формул Γ , либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода.

Вывод с пустым множеством гипотез Γ называется просто выводом.

Теоремой называется формула A , если существует вывод, в котором последней формулой является A ; при этом формула A называется выводимой, и этот факт обозначается через $t \vdash A$. Аналогично выводимость A из множества гипотез Γ обозначается через $\Gamma \vdash A$.

Пример. Формула $F = \forall y \forall x A$ выводима из гипотезы $\forall x \forall y A$, поскольку можно указать следующий вывод F из этой гипотезы:

1. $\forall x \forall y A$ (гипотеза),

2. $\forall x \forall y A \rightarrow \forall y A$ (частный случай аксиомы (А.4)),
3. $\forall y A$ (из первых двух формул по правилу MP),
4. $\forall y A \rightarrow A$ (частный случай аксиомы (А.4)),
5. A (из третьей и четвертой формул по правилу MP),
6. $\forall x A$ (из предыдущей формулы по правилу Gen),
7. $\forall y \forall x A$ (из предыдущей формулы по правилу Gen).

5.4. Расширение языка исчисления предикатов

Для удобства пользования языком можно допускать в нем избыточность выразительных средств. Для этого в язык исчисления предикатов включают дополнительные связки \vee , $\&$, \leftrightarrow и символ \exists . При этом учитывается, что: $A \vee B$ есть сокращение для $\neg A \rightarrow B$; $A \& B$ есть сокращение для $\neg (A \rightarrow \neg B)$; $A \leftrightarrow B$ есть сокращение для $(A \rightarrow B) \& (B \rightarrow A)$; $\exists x A$ есть сокращение для $\neg \forall x \neg A$.

Кроме того употребляются пропозициональные константы «Л» («ложь») или «противоречие») и «И» («истина»), где: Л есть сокращение для формулы $G \& \neg G$, G некоторая замкнутая формула; И есть сокращение для $\neg Л$.

По определению в расширенном языке исчисления предикатов формула F есть теорема исчисления предикатов тогда и только тогда, когда теоремой является формула, полученная из F замещением определяемых связок \vee , $\&$, \leftrightarrow , кванторных символов \exists и пропозициональных констант И, Л на их формульные определения.

Пример.

Формула $A \rightarrow (B \vee A)$ есть теорема исчисления предикатов в расширенном языке, так как формула $A \rightarrow (\neg B \rightarrow A)$ есть частный случай аксиомы А1.

Так как расширение множества теорем исчисления предикатов оставляет неизменным множество всех теорем, сформулированных в исходном языке с помощью связок \neg , \rightarrow и символа \forall , то оно является консервативным.

В настоящее время многие исследователи проводят работы по развитию и исследованию различных вариантов расширения языка исчисления предикатов. Например, в [7] рассматривается расширенный язык L позитивно-образованных формул (по-формул), исчисление J по-формул, стратегии поиска вывода в исчислении J с использованием различных эвристик, а также приводятся примеры успешного решения ряда задач. В связи с ограниченным объемом данного курса рассмотрение различных расширений языка ИП рекомендуется для самостоятельного изучения.

5.5. Разрешимые и неразрешимые задачи

В соответствии с требованием к формальным системам синтаксические понятия определения термина, формулы, аксиомы, вывода в исчислении предикатов даны таким образом, что для любых символов всегда можно ответить, является ли эта последовательность термом, формулой или выводом, а для любой формулы определить, является ли она аксиомой.

Разрешимой называется задача, если существует некая формальная процедура, решающая в принципе эту задачу, независимо от реальных возможностей и ограничений по времени и пространству. Это понятие применяется для решения массовых проблем, к которым относится бесконечный класс однотипных задач. Проблема разрешима, если существует единая процедура, решающая каждую задачу этой массовой проблемы. Примеры массовых проблем: «решить линейное уравнение», «определить четное или нечетное число» и т.п.

Теоретически было показано:

- массовые проблемы определения свойств последовательностей символов быть термом, формулой, аксиомой или выводом являются разрешимыми;

- свойство формул ИП быть теоремой является неразрешимым.

Таким образом, нет процедур, которые бы определяли за сколь угодно большое, конечное число шагов, является ли формула теоремой. Это свойство называется неразрешимостью ИП.

Однако есть процедуры, которые порождают по очереди все выводы, начиная с простейших, и перечисляют таким образом все теоремы («алгоритм Британского музея»).

Говорят, что множество всех теорем ИП перечислимо, а множество формул ИП, не являющихся теоремами, неперечислимо, и следовательно, не существует процедуры, которая порождала бы все формулы ИП, не являющиеся теоремами (иначе имелось бы две процедуры, перечисляющие соответственно теоремы и не теоремы). Свойство перечислимости всех теорем ИП иногда называют его полуразрешимостью.

Тема 6. ИНТЕЛЛЕКТНОЕ УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

В данной теме рассматривается применение интеллектуального управления на примере систем управления, основанных на правилах и описанных в работах.

6.1. Интеллектуальное управление на основе нечеткой логики

Разработка систем интеллектуального управления (СИУ), основанных на правилах (п.3.1), предполагает выбор переменных состояния и выхода, в терминах которых и формулируются правила [7]. В самом простейшем случае – интеллектуальных аналогов широко распространенных в теории автоматического управления способов формирования управления объектом управления (ОУ). Этот выбор можно реализовать в соответствии с позиционным управлением по положению и скорости ОУ, а именно в соответствии с принципами пропорционального (П) и пропорционально-дифференциального (ПД) управления. Разумеется, могут использоваться аналоги и других типовых способов формирования управления, например, пропорционально-интегрального (ПИ) или пропорционально-интегрально-дифференциального (ПИД) управления. Для нелинейных систем управления способы формирования управления могут получаться и более сложными.

Рассмотрим структурную схему простейшей следящей системы (рис. 6.1).

Обычная функция регулятора – поддерживать достаточно близкие значения сигнала y на выходе ОУ к заданному значению уставки (входного сигнала $y_{вх}$) для любого момента времени в условиях действия возмущения f . Другими словами – необходимо обеспечивать достаточно малость ошибки (рассогласования) $e = y_{вх} - y$. Показатели ошибки e , используемые в

посылках правил, аналогичных ПИД-управлению, обычно выбираются из числа следующих трех видов [7]:

- ошибка e ,
- изменение ошибки, обозначаемое Δe или \dot{e} ,
- сумма ошибок δe .

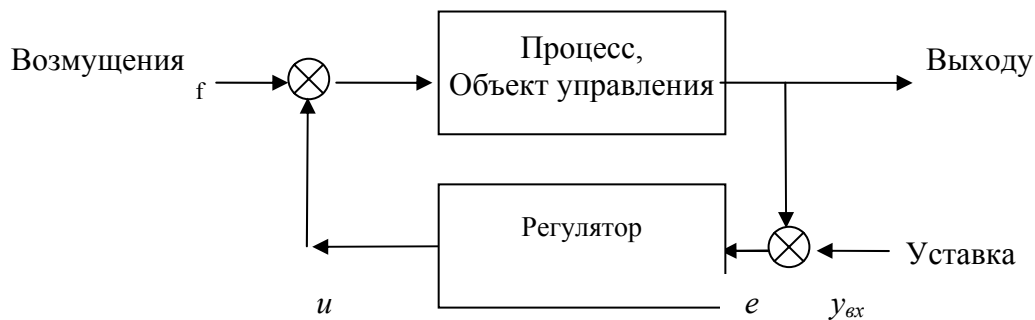


Рис. 6.1. Структурная схема простейшей следящей системы

Показатели управления u на выходе регулятора, т.е. на входе в процесс, используемые в заключениях правил, обычно выбираются из числа следующих:

- изменение управления, обозначаемое Δu или \dot{u} ,
- управление u .

Кроме того, в случае дискретного времени $k = 1, 2, \dots$, по аналогии с обычным регулятором, имеем

$$e(k) = y_{\text{вх}} - y(k),$$

$$\Delta e(k) = e(k) - e(k-1),$$

$$\Delta u(k) = u(k) - u(k-1),$$

$$\delta e(k) = \sum_{i=1}^{k-1} e(i),$$

где $u_{\text{вх}}$ – желаемый выход процесса, u – текущий выход.

Уравнение обычного ПД-регулятора имеет вид

$$u = KP e + KD \dot{e},$$

где KP , KD – коэффициенты усиления ПД-регулятора (пропорциональной и дифференциальной составляющих в законе управления). Дискретному аналогу отвечают правила, имеющие форму [7]:

ЕСЛИ $e(k)$ обладает свойством «имя свойства»

И $\Delta e(k)$ обладает свойством «имя свойства»,

ТО $u(k)$ обладает свойством «имя свойства»,

где <имя свойства> – предикатный символ, например, РАВНО-НУЛЮ, ПОЛОЖИТЕЛЬНОЕ-МАЛОЕ, ПОЛОЖИТЕЛЬНОЕ-СРЕДНЕЕ, ПОЛОЖИТЕЛЬНОЕ-БОЛЬШОЕ, ОТРИЦАТЕЛЬНОЕ-МАЛОЕ,... Смысл этих правил очевиден.

Если уравнение обычного ПИ-регулятора

$$u = KP e + KI \int e dt,$$

где KI – коэффициент усиления интегрального члена в законе управления, преобразовать в эквивалентную форму

$$\dot{u} = KP \dot{e} + KI e,$$

то предыдущая форма правил (дискретного аналога) сохраняется. В этом случае для получения значения управления $u(k)$ изменение управления $\Delta u(k)$ суммируется с $u(k-1)$.

Форма представления правил дискретного П-регулятора имеет вид:

ЕСЛИ $e(k)$ обладает свойством «имя свойства», ТО $u(k)$ обладает свойством «имя свойства». Уравнению обычного ПИД-регулятора

$$u = KP e + KV \dot{e} + KI \int e dt$$

отвечают правила следующей формы:

ЕСЛИ e обладает свойством «имя свойства»

И Δe обладает свойством «имя свойства»,

И δe обладает свойством «имя свойства»,

ТО u обладает свойством «имя свойства».

В некоторых случаях, когда доступны знания не только в форме показателей ошибки e (e , Δe , δe), но и, например, показатели выхода у процесса, то в посылки правил могут включаться свойства переменной u и других показателей выхода.

Примером такого правила является выражение:

ЕСЛИ давление обладает свойством <БОЛЬШОЕ>

И изменение-давления обладает свойством <ПОЛОЖИТЕЛЬНОЕ-МАЛОЕ>,

ТО подвод-отвод-энергии обладает свойством <ОТРИЦАТЕЛЬНОЕ-СРЕДНЕЕ>,

где давление – выход у процесса, а подвод-отвод-энергии – управление.

Если имена свойств в правилах рассматривать как предикатные символы, то возможна формализация этих правил в предикатной форме с выходом на методы СОЗ. Такое чисто символическое представление правил достаточно также для проведения некоторого качественного анализа вопросов устойчивости системы управления. При этом в составе регулятора должны быть преобразователи входов регулятора (аналоговых или цифровых) в двоичные, т.е. истинностные значения предикатов (истина – ложь), и обратные преобразователи – двоичных выходов в обычные для ОУ сигналы.

Если же эти имена свойств рассматривать как нечеткие значения лингвистических переменных в смысле [9, 22], то соответствующие правила являются нечеткими. В составе нечеткого регулятора должен быть фа-

зификатор, преобразующий обычные (четкие, аналоговые или цифровые) значения входов регулятора в нечеткие (лингвистические) значения, и дефазификатор для обратного преобразования нечетких значений выходных переменных в обычные величины. Помимо качественного анализа, можно обеспечить и количественное описание поведения системы управления путем количественной интерпретации содержательного смысла нечетких значений лингвистических переменных в терминах нечетких множеств или функций принадлежности [22].

Основными задачами, решаемыми при создании СИУ с нечеткой логикой, являются: 1) сопоставление описаний состояний ОУ с условиями истинности продукционных правил; 2) определение стратегии использования правил.

В зависимости от того, как решается вторая задача, в [7] предлагается различать два типа формирования управления: «ситуация-действие» (С-Д) и «ситуация-стратегия управления-действие» (С-СУ-Д).

В случае «С-Д» правила задаются в явной форме, образуя базу знаний. Стратегия просмотра неизменна и на очередном шаге используется правило, условия истинности которого наиболее соответствуют в смысле некоторой меры близости состоянию ОУ.

В случае «С-СУ-Д» явно заданных правил нет. Они выводятся на основе некоторой нечеткой ситуационной сети, как на нечетком взвешенном графе переходов внутри множества эталонных описаний состояний ОУ. Набор правил, необходимых для вывода управляющего действия, а также порядок их применения определяются стратегией управления нечетким маршрутом в сети от исходной ситуации к целевой. В нестандартных условиях управления это обеспечивает большую гибкость и устойчивость управления ОУ.

Значения переменных, входящих в правила, динамически меняются, а значение δe зависит от всей предыстории. Так как правила в нормальном режиме эксплуатации СИУ не изменяются, то такую СИУ можно считать статической.

Вначале нечеткое управление использовалось в виде статических правил, а затем были созданы адаптивные нечеткие регуляторы, построенные на базе двух основных схем. Первый тип – нечеткие регуляторы с неявной адаптацией [22], использующие для синтеза регулятора в реальном времени промежуточную модель процесса (рис. 6.2).

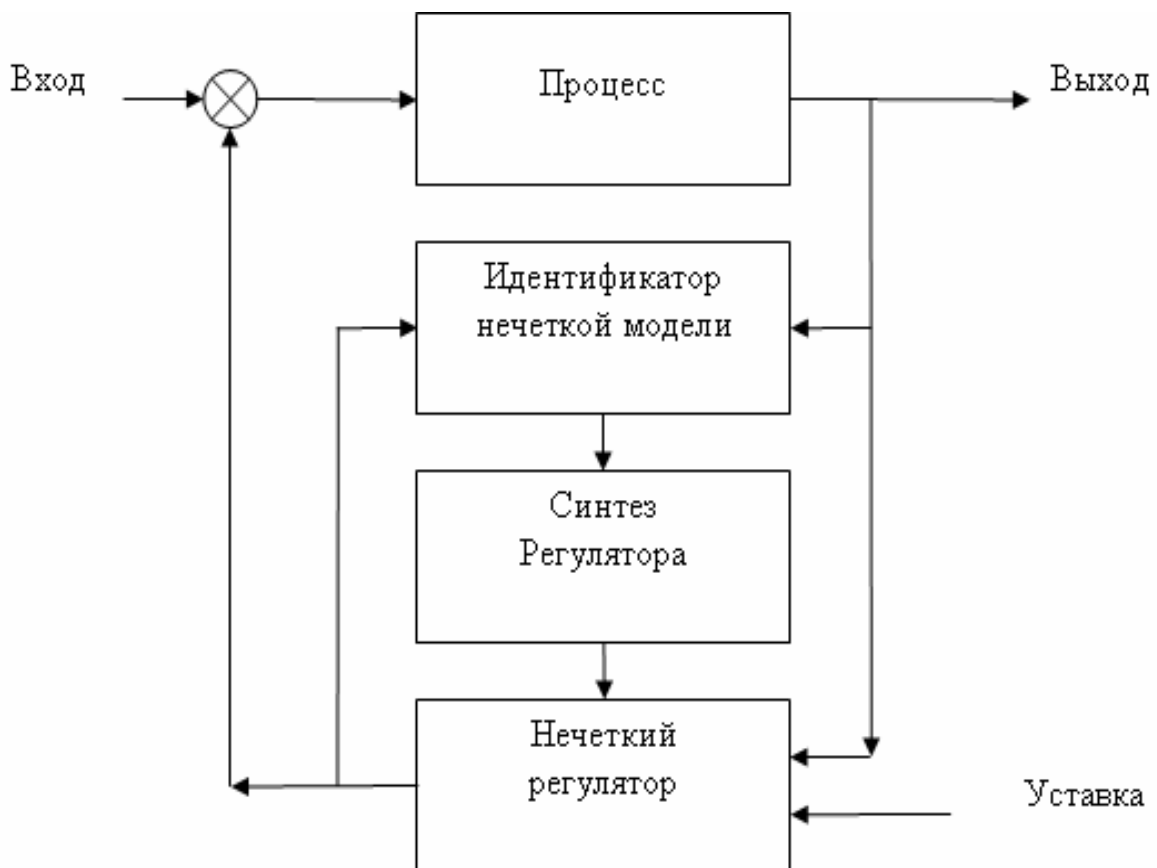


Рис. 6.2. Нечеткий регулятор с неявной адаптацией

Второй тип – регулятор в форме прямой адаптации с модификацией базы правил по результатам наблюдения функционирования контура управления (рис. 6.3). Обычно эти нечеткие регуляторы называют типовыми самоорганизующимися, основанными на нечеткой логике [22].

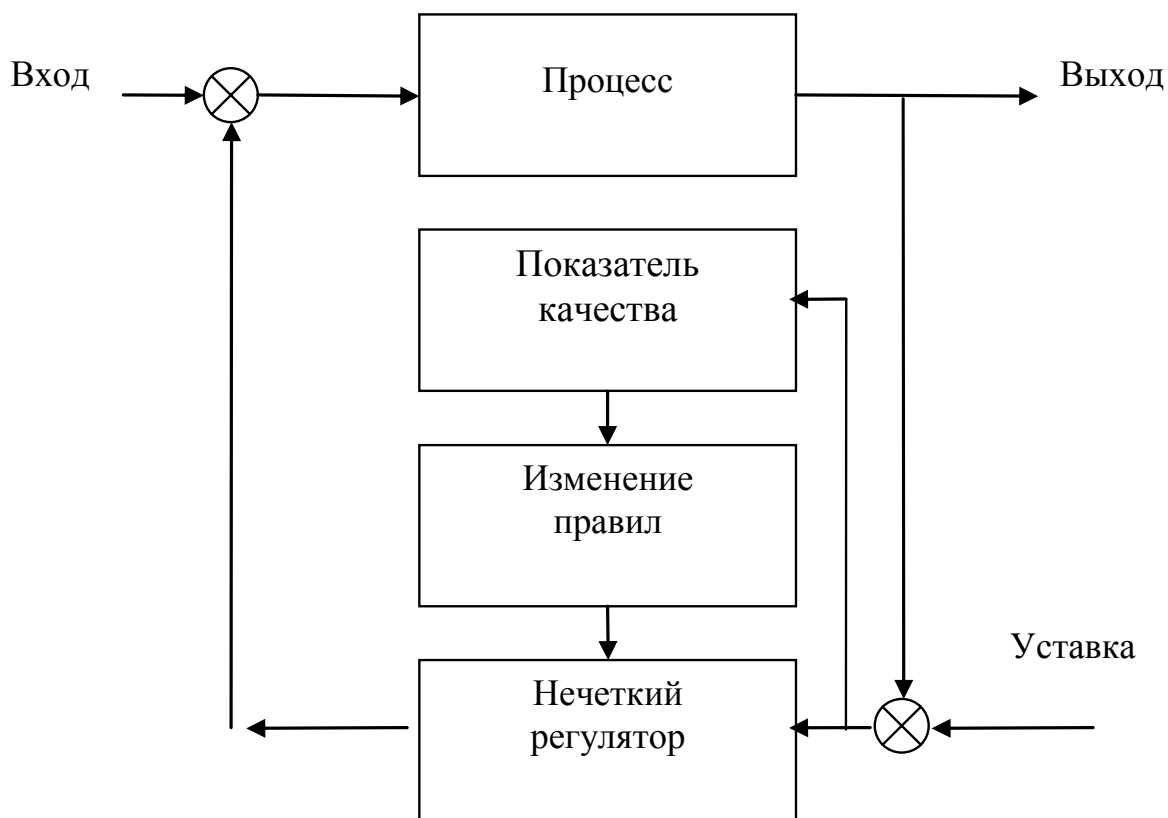


Рис. 6.3. Типовой самоорганизующийся нечеткий регулятор

В качестве примера системы интеллектуального управления на базе экспертно-нечеткого регулятора приведем СИУ для проводки грузового судна между островами без вмешательства человека [7]. Входами в процесс (ОУ) являются u (скорость) и δ (угол поворота руля), выходами – курс судна Ψ (относительно осей фиксированной системы координат) и положение в плоскости (x, y) . Двухуровневая структура системы управления представлена на рис. 6.4.

Нечеткий регулятор нижнего уровня обеспечивает для любого момента времени достаточную близость курса Ψ к значению уставки Ψ_r , вырабатываемой экспертным регулятором верхнего уровня. Этот регулятор является экспертной системой реального времени, формирующей на основе некоторой совокупности экспертных правил уставку Ψ_r и значение управления скоростью u , подаваемое на задатчик скорости ОУ.

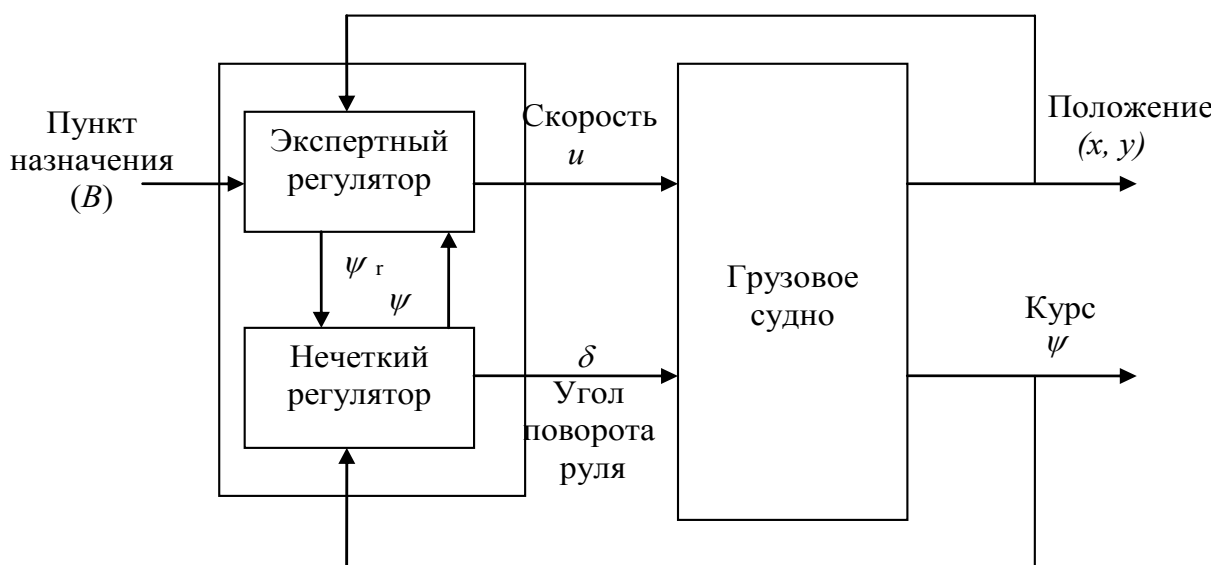


Рис. 6.4. Двухуровневый экспертно-нечеткий регулятор

Нечеткий регулятор реализует ПД-регулятор, используя ошибку $\delta = \Psi - \Psi_r$ и ее производную для выбора значения управления углом поворота рулей δ . Экспертный регулятор использует курс Ψ , текущую позицию (x, y) и знание пункта назначения (B) для определения, с какой скоростью двигаться и какой курс Ψ_r задать нечеткому регулятору.

Экспертный регулятор использует некоторую систему приоритетов для правил при управлении процессом вывода. Регулятор отыскивает курс и скорость, основываясь на положении островов, обеспечивая подходящее

маневрирование между ними подобно опытному капитану. При этом для задачи перевода судна из точки А в точку В, представленной на рис. 6.5, оказывается достаточно всего лишь 10-ти правил для представления опыта капитана. Правила характеризуют требования замедления судна на поворотах, ускорения на прямых участках, формирование траектории, изображенной на рис. 6.5 пунктиром. Регуляторы обеспечивают управление судном по отслеживанию заданной траектории.

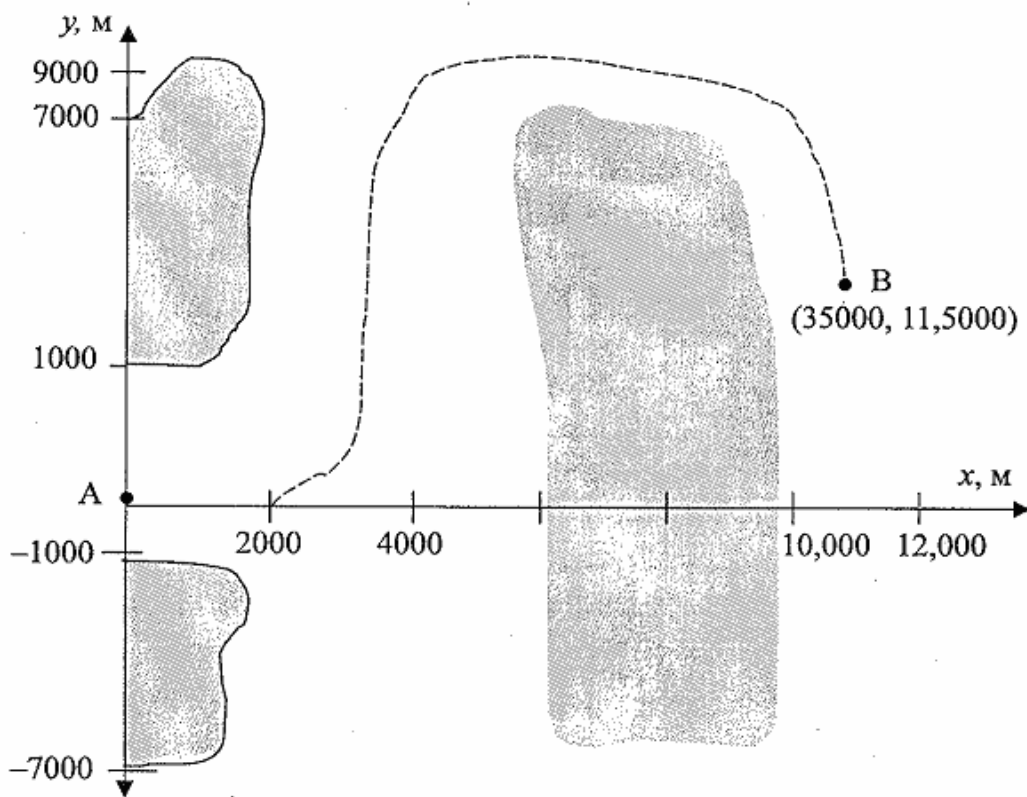


Рис.6.5. Карта островов для проводки судна

Относительно этой системы в [7] отмечается, что

а) «...Степень интеллектуальности СИУ ... характеризуется качеством функционирования и степенью автономности от вмешательств человека. При этом верхний (экспертный) регулятор имеет дело с более медленными аспектами движения, так он подправляет скорость значительно реже, чем нижний (нечеткий) регулятор обновляет угол поворота руля...».

б) «...что возможно добавление еще одного – третьего (более высокого) уровня для встраивания других функций, таких как:

- дружественный пользовательский интерфейс для капитана и судовой команды;

- интерфейс с метеоинформацией, используемой для изменения проводки судна, основываясь на условиях моря;

- высокоуровневое планирование маршрута доставки грузов;

- способность к обучению, позволяющая улучшать качество функционирования с течением времени, используя оценки функционирования на предыдущих рейсах;

- управление или координация другими, более сложными подсистемами, такими как обнаружение и идентификация отказов, минимизация затрат топлива и времени...».

Рассмотренные примеры показывают достаточно большие потенциальные возможности применения СИУ, основанных на нечетком управлении. Однако существуют и крайние мнения в оценке полезности и особенностей применения нечеткого интеллектуального управления [1, 7, 22].

6.2. Особенности применения систем интеллектуального управления

С одной стороны, горячие сторонники этой технологии уверены, что нечеткое управление обещает прорывы в решении сложных технических проблем с помощью малых затрат и будет революционизировать технику управления. С другой стороны, разработчики и исследователи традиционных систем управления прогнозируют в будущем спад нечеткого управления в теории и приложениях, так как считают что «...все, что может быть

сделано в нечетком управлении, может быть сделано также и традиционными средствами теории управления...».

Однако эти позиции не учитывают того факта, что использование экспертных знаний об объекте и возможностях систем управления позволяет значительно повысить степень автоматизации управления во многих производственных процессах и ситуациях. В этом случае нечеткое управление предлагает метод представления и использования экспертных знаний оператора, технолога или инженера-разработчика, представляемые в конкретной и понятной форме правил типа: «...если ситуация такая, то следует сделать следующее...». Особенно это помогает при управлении в нештатных ситуациях или при пуско-наладочных операциях, когда приходится выбирать различные виды технических средств и алгоритмов управления. Кроме того, опыт и знания высококвалифицированных операторов, технологов и автоматчиков могут легко тиражироваться на аналогичные процессы, установки и эффективно использоваться и опытными пользователями, и пользователями с недостаточно высоким уровнем подготовки. Для подтверждения этого можно привести из работ [4, 7-9, 21-23] целый ряд примеров применения нечеткого управления.

Например, в [7] приведены сведения как одна португальская компания реализовала в целлюлозно-бумажной промышленности нечеткое управление автоклавами на верхнем уровне. Для управления технологическим процессом были использованы 25 нечетких правил, реализующих стратегии управления, осуществляемые обычно оператором. Основным достижением от повышения степени автоматизации и круглосуточной поддержки рациональной стратегии управления явилось существенное уменьшение (до 60%) вариации качества продукции. Кроме того, последовавшая за этим оптимизация программного обеспечения системы управления привела к значительному уменьшению затрат энергии и потребления сырья. Общий возврат инвестиций в разработку пакета программ по нечет-

кому управлению и базы знаний реальной системы управления был достигнут через несколько месяцев.

В той же работе рассматривается пример решения задачи перемещения объектов разной массы многозвенным манипулятором робота вдоль заданной траектории. Если имеются хорошие и точные модели, то используя известные методы можно реализовать ПИД-регулятор, хорошо работающий с известными массами внутри узкого диапазона их изменения. Однако при больших внешних возмущениях или в случае существенных параметрических изменений наблюдается резкое ухудшение качества функционирования обычного ПИД-регулятора, обеспечивающего или быструю реакцию со значительной ошибкой или более точную, но медленную реакцию, или даже возникновение проблемы стабилизации в целом.

Нечеткое управление позволяет в этом случае реализовать простые, робастные решения, перекрывающие широкий диапазон возможного изменения параметров и больших возмущений. Для этого частного случая была реализована СИУ с нечетким скользящим режимом [22]. Система демонстрировала характеристики, аналогичные ПИД-регулятору для заданной массы с небольшими вариациями, но перекрывала по качеству ПИД-решение при вводе больших изменений массы. Решение было использовано в СИУ длинным многозвенным манипулятором (около 10 м), установленным на космической станции [7]. Система устраняла негативное влияние люфтов в соединениях звеньев на конечную ошибку позиционирования и обеспечивала требуемую точность в широком диапазоне нагрузок.

В работе [4] описываются примеры нечеткого управления запуском-выпуском изделий на технологической операции «металлизация» производства прецизионных резисторов и модели управления роботом-манипулятором в системе «глаз-рука». Подробно описываются используемые нечеткие модели типа «С-СУ-Д» и «С-Д». Известны описания успешного использования нечеткого управления в проекте самолета с высоко-

технологичными крыльями улучшенной аэродинамики (Rockwell's experimental advanced technology wing).

Для перехода к методам интеллектуального управления имеются и другие существенные обстоятельства, способствующие устранению часто встречающегося коммуникационного барьера между конструкторами изделия и разработчиками систем управления. В интеллектуальном управлении применяются языки, понятные соответствующим коллективам разработчиков. При этом проблемы коммуникации и разделения сфер ответственности значительно упрощаются использованием простых и корректных трансляторов из языка нечетких правил в язык элементарных объектов и алгоритмов (табличные функции, интерполяции, компараторы и т.д.).

В [4] описан пример, решения задачи в сфере автомобильной электроники по управлению холостым режимом работы двигателя на базе 8-разрядных микропроцессоров. Нечеткое управление обеспечило поддержание постоянной скорости холостого режима двигателя при больших возможных параметрических изменениях в системе (включая разные дорожные условия и дополнительное энергопотребление рулевым механизмом, кондиционером и т.д.). Время разработки СИУ составило примерно 6 человеко-месяцев в отличие от 2 человеко-лет в рамках традиционного подхода.

Показательно, что в Японии слово «нечеткий» стало одним из самых популярных и приобрело самостоятельное маркетинговое значение. Большинство вновь вводимых бытовых приборов маркируются «нечеткий» или «нейро-нечеткий», и в быту оно ассоциируется с терминами «современный», «высококачественный», «дружественный». В новых бытовых приборах, стиральных машинах «кнопки одного касания» заменяют большие старые управляющие панели. Продажи японскими производителями бытовой нечетко управляемой техники составляют ежегодно несколько миллиардов американских долларов.

В тоже время этот маркетинговый эффект в гораздо меньшей степени наблюдается в области промышленных применений. В некоторых случаях нечеткое управление видимо будет использоваться в промышленности и без особой необходимости, в основном для демонстрации в технологической конкуренции компаний использования модных методов интеллектуального управления. Кроме того, компании используют нечеткое управление даже не столько для улучшения характеристик системы управления, сколько для уменьшения затрат в сфере патентной конкуренции. Применение нечеткого управления при качественно эквивалентных решениях помогает обходить существующие патенты.

Для улучшения существующих систем управления имеют некоторые перспективы применения гибридных систем, сочетающих интеллектуальные и традиционные технологии. Но в тоже время сейчас пока нет ясности, где проходит граница рационального применения нечеткого интеллектуального управления. Процитируем некоторые комментарии [4, 7, 22] к основным утверждениям, публикуемым в рекламных и научных целях.

«...Нечеткие регуляторы – более робастны, чем обычные. Имеются многие приложения, где использование нечеткого управления в чистом виде или в комбинации, например, с ПИД-управлением обеспечило высокую робастность систем управления. Но известны и другие случаи, например, попытки управления системами воздушного кондиционирования на основе нечетких логик, построенных с двумя небольшими отличиями в структуре и базе знаний. Одна попытка оказалась успешной, обеспечив высокую робастность даже при больших возмущениях, другая привела к неустойчивости. По-видимому, еще не до конца методологически ясно, для какого типа технических задач управления нечеткое управление ведет к улучшению робастности и устойчивости и как проектные решения влияют на эти свойства. Вместе с тем, для нечетких систем управления есть первые приемлемые критерии устойчивости. Такие критерии обсуждаются и

можно утверждать, что нечеткое управление и в этом смысле начинает конкурировать с обычным нелинейным управлением...».

«...Нечеткое управление сокращает время разработки. Пример из автомобильной электроники, упомянутый ранее, как и ряд других, подтверждает это, хотя на начальных этапах освоения технологии нечеткого управления возможен, разумеется, и обратный эффект».

«...Продукция, использующая нечеткое управление, легче продается. Это характерно для бытовой техники в Японии. Вместе с тем, современный маркетинг должен фокусироваться не на модном слове «нечеткий», а на потребительских свойствах: повышение уровня интеллектуальности и дружелюбности интерфейса, дополнительные функции и ресурсные аспекты, такие, как экономия энергии, уменьшение расхода свежей воды (в стиральных машинах) и т.д.

По оценке настоящего и будущего рынка приборов нечеткого управления, данной в [22], примерно 10-15% всей электрической и электронной техники выиграет от использования нечеткого управления в форме «усиливающей» или «безальтернативной» технологии.

С другой стороны, в областях с чисто информационными технологиями управления (базы данных, экспертные системы в финансовом секторе и т.д.) имеется все еще малое, хотя и возрастающее число приложений нечеткой логики...».

«...Нечеткое управление ведет к более высокой степени автоматизации для сложных, плохо структурированных процессов. Это справедливо, но только если имеется подходящее знание о процессе, которое может быть хорошо представлено в терминах нечетких правил. Существуют различные процессы, для которых такого знания нет вовсе или его нет в необходимой степени. Например, если в системе управления возможны ситуации таких отказов элементов (подсистем), которые заранее не предусмотрены и не обеспечены средствами их обнаружения и локализации, равно

как и автоматическим выходом в соответствующие априорно предусмотренные аварийные режимы, то система управления должна над имеющейся базой знаний принять наиболее рациональное решение (осуществить реконфигурацию) с predetermined заранее целью (сохранение работоспособности и т.п.) или, еще лучше, с возможным пересмотром цели и критериев качества управления. При этом может понадобиться заранее не предусмотренным (совокупностью экспертных правил «если..., то ...») способом спланировать целую последовательность действий по выводу системы в работоспособное состояние. Для этого нужны другие методы интеллектуализации, с другими формами представления и обработки знаний, вообще говоря, обеспечивающие более высокий уровень интеллекта. В частности, могут понадобиться более мощные методы «обдумывания» ситуации, обрабатывающие знания более общего вида, нежели просто использующие правила, априорно заготовленные или несколько модифицируемые в процессе функционирования. К таким методам относятся, например, методы АДТ и вообще методы логического вывода...».

Приведенные комментарии показывают, что в настоящее время область интеллектуального управления развивается по различным направлениям, имеет свою специфику и особенности применения. Так же как и в традиционных системах управления для построения конкретных СИУ необходим детальный анализ соответствующих целей, задач и особенностей управления объектом для обоснования выбора рационального варианта структуры и методов интеллектуального управления. Такой выбор, особенно при разработке комплексных систем управления, должен учитывать и постоянное развитие других перспективных подходов и методов управления, включая нейронные сети и различные системы распознавания образов.

Тема 7. СИСТЕМЫ РАСПОЗНАВАНИЯ ОБРАЗОВ

Понятие «Образа» определяется в [6] как «изображение типичного или обобщенного представителя некоторого класса объектов». В Большой советской энциклопедии (БСЭ) «Образ» (или распознаваемый класс) определяется как совокупность входных сигналов, имеющих некоторые характерные свойства. Воспринимающая их система (она называется распознающей) должна реагировать на все сигналы этой совокупности одинаковыми ответами. Это понятие играет очень важную роль в системах ИИ, огромное количество работ посвящено исследованию проблем распознавания образов, созданию различных методов, алгоритмов и систем для их решения, а также практического применения [13, 21, 43, 45-47]. В этой теме на основе материалов [21] рассмотрим кратко основные особенности решения проблем распознавания образов.

Согласно БСЭ «Распознавание образов» – это научное направление, связанное с разработкой принципов и построением систем, предназначенных для определения принадлежности данного объекта к одному из заранее выделенных классов объектов. Под объектами в распознавании образов понимают различные предметы, явления, процессы, ситуации, сигналы. Каждый объект описывается совокупностью основных характеристик (признаков, свойств) $X = (x_1, \dots, x_i, \dots, x_n)$, где i -я координата вектора X определяет значения i -й характеристики, и дополнительной характеристикой S , которая указывает на принадлежность объекта к некоторому классу (образу).

Набор заранее расклассифицированных объектов, т.е. таких, у которых известны характеристики X и S , используется для обнаружения закономерных связей между значениями этих характеристик и называется обучающей выборкой. Те объекты, у которых характеристика S неизвестна,

образуют контрольную выборку. Отдельные объекты обучающей и контрольной выборок называются реализациями.

Одна из основных задач распознавания образов это выбор правила (решающей функции) D , которое устанавливает по значению контрольной реализации X ее принадлежность к одному из образов. Правила указывают «наиболее правдоподобные» значения характеристики S для данного X . Выбор решающей функции D требуется произвести так, чтобы стоимость самого распознающего устройства, его эксплуатации и потерь, связанных с ошибками распознавания, была минимальной.

Задачи распознавания образов часто применяются в различных экспертных системах (ЭС), например, в ЭС для геологии. Примером задач такого типа может служить задача различения нефтеносных и водоносных пластов по косвенным геофизическим данным, позволяющим обнаружить пласты, насыщенные жидкостью. Значительно сложнее определить, наполнены они нефтью или водой. Требуется найти правила использования информации, содержащейся в геофизических характеристиках, для отнесения каждого насыщенного жидкостью пласта к одному из двух классов — водоносному или нефтеносному. При решении этой задачи в обучающую выборку включают геофизические данные вскрытых пластов.

Успех в решении задачи распознавании образов зависит в значительной мере от удачного выбора признаков X . Исходный набор характеристик часто бывает очень большим. В то же время приемлемое правило должно быть основано на использовании небольшого числа признаков, наиболее важных для различения одного образа от другого. Так, в ЭС для задач медицинской диагностики важно определить, какие симптомы и их сочетания (синдромы) следует использовать при постановке диагноза данного заболевания. Поэтому проблема выбора информативных признаков — важная составная часть проблемы распознавания образов.

В основной задаче распознавания образов при построении решающих функций D используются закономерные связи между характеристиками X и S , обнаруживаемые на обучающей выборке, и некоторые дополнительные априорные предположения. Например, принимаются следующие гипотезы: характеристики X для реализаций образов представляют собой случайные выборки из генеральных совокупностей с нормальным распределением; реализации одного образа расположены «компактно» (в некотором смысле); признаки в наборе X независимы и т.д.

Разработка теории распознавания образов вначале проводилась с целью понимания процессов восприятия образов человеком. При восприятии человеком внешнего мира, мозг всегда производит разделение воспринимаемых ощущений, выделяя в отдельные группы похожие явления. Все объекты, имеющие некоторые общие признаки, относятся к одной группе, несмотря на определенные существенные различия. Классическим примером являются задачи распознавания текстов и звуков. Одни и те же буквы алфавита, написанные различными почерками, воспринимаются нами как один и тот же образ, так же как и звуки, соответствующие одной и той же ноте, взятой в любой октаве и на любом инструменте. На множество состояний технического объекта реакция оператора поста управления объектом также может быть одной и той же.

Проведенные многочисленные исследования показали, что для выработки представления о группе восприятий определенного класса человеку часто достаточно ознакомиться с небольшим числом ее представителей, т.е. происходит процесс обучения человека. Ребенку достаточно показать всего один раз какую-либо картинку или букву, и он может потом найти их среди других, даже если они значительно отличаются от ранее показанных. Это свойство мозга подтверждает, что существует реальное понятие образа, проявляющееся в том, что ознакомление с конечным числом явлений

из одного и того же множества дает возможность узнавать большое число его представителей.

В технических приложениях в качестве образа можно рассматривать и некоторые совокупности состояний объекта управления, характеризующиеся тем, что для достижения заданной цели требуется одинаковые воздействия на объект. Характерные объективные свойства образов проявляются в том, что обучение разных людей можно проводить на различном материале наблюдений, а потом они могут правильно классифицировать одни и те же объекты независимо друг от друга. Кроме того, восприятие образов позволяет с определенной достоверностью узнавать большое число объектов одного класса на основании ознакомления с конечным их числом, а также способность человеческого мозга отвечать на большое число воздействий внешней среды конечным числом ответных реакций. Эти объективные свойства образов позволяет моделировать процесс их распознавания и эффективно решать проблемы обучения распознаванию образов (ОРО).

Общая проблема распознавания образов включает этапы обучения и распознавания. Этап обучения может происходить двумя основными способами – «с учителем» и «без учителя». Обучение с учителем осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного образа и различными – на все объекты различных образов.

Обучение без учителя также осуществляется путем предъявления системе ОРО отдельных объектов, но без указания их принадлежности какому либо образу. В этом случае система ОРО должна «сама» выделить и распознать образы объектов, относящиеся к одинаковым и к разным классам.

Завершается процесс обучения после предъявления системе ОРО конечного числа объектов без каких-либо дополнительных подсказок.

В качестве объектов обучения могут быть параметры состояния технического объекта и систем управления, визуальные и звуковые изображения (буквы, записи шумов вибрации), либо другие характеристики каких-то явлений внешней среды или состояния распознаваемых объектов. При обучении с учителем указываются только сами объекты и их принадлежность образу.

За этапом обучения следует этап распознавания новых объектов, который позволяет оценить действия уже обученной системы на контрольной выборке.

Известный пример задачи из области ОРО [21] приведен на рис.7.1, он дает представление о сложности решения подобных задач. На рисунке представлены 12 задач, для которых следует выделить признаки, характеризующие отличия левых триад картинок от правых. Для формализованного решения подобных задач требуется полномасштабное моделирование логического мышления.

Для решения задач распознавания образов создаются специальные распознающие системы. Эти системы могут с помощью различных методов и алгоритмов распознавать не только зрительные и слуховые образы, но и решать задачи распознавания сложных процессов и явлений, возникающих, например, при различных заболеваниях, при управлении производственными процессами, предприятиями, экономикой, оценке социальных процессов или осуществлении транспортных или военных операций. В каждой из таких задач анализируются некоторые явления, процессы, состояния внешнего мира, называемые далее объектами наблюдения.

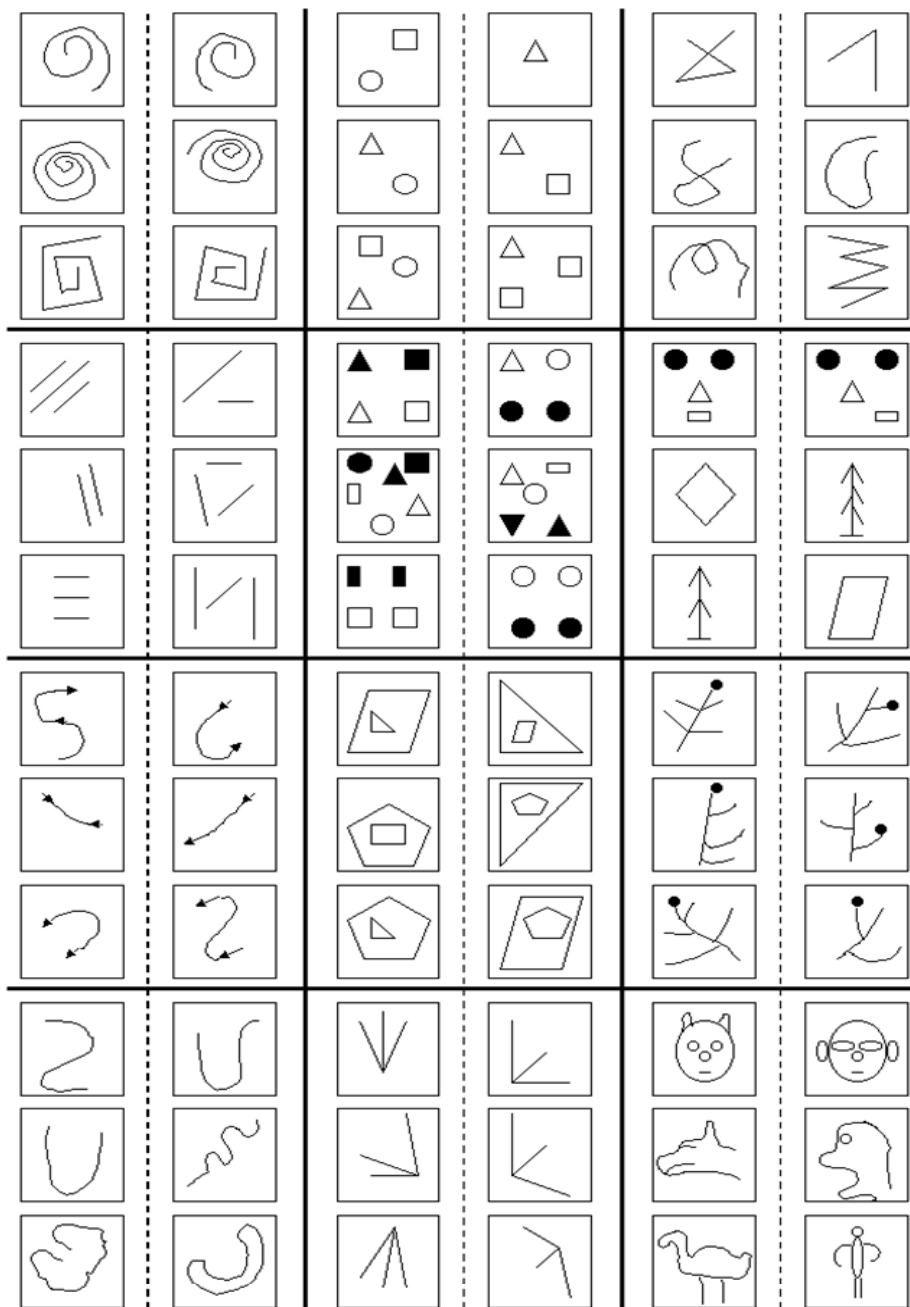


Рис. 7.1.

Прежде чем начать анализ какого-либо явления или объекта, нужно получить необходимую исходную информацию о характеристиках объектов. Получение этой информации происходит путем измерения некоторых сигналов, поступающих на воспринимающие элементы распознающей системы (множество различных сенсоров, датчиков). Затем производятся преобразования измеренных сигналов в те или иные показатели, характери-

зующие различные свойства исследуемых объектов (получение «изображения»), и уже потом производится окончательная обработка множества всех измеренных параметров изображения для решения задач распознавания образов.

При использовании методов распознавания образов для решения задач управления часто вместо термина «изображение» применяют термин «состояние». Состояние — это некоторая характеристика в определенной форме подмножества измеряемых текущих характеристик наблюдаемого объекта. Ситуацией принято называть некоторую совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же наборами характеристик объекта. Например, если в качестве объекта наблюдения рассматривается некоторый технический объект управления, то ситуация объединяет такие состояния этого объекта, в которых следует принимать одни и те же управляющие решения.

Для иллюстрации процесса обучения распознаванию образов рассмотрим применяемые в теории распознавания образов геометрический и структурный подходы [13, 22, 45-47].

Так как состояния объекта определяются вектором характеристик, то каждый вектор можно представить в виде точки в некотором пространстве признаков. Если утверждается, что при показе состояний возможно однозначно отнести их к одному из нескольких образов, то тем самым утверждается, что в некотором пространстве существуют соответствующие области, не имеющих общих точек, и что изображения состояний — это точки из этих областей. Каждой такой области можно приписать наименование, т.е. дать название, соответствующее образу. Тогда процесс обучения распознаванию образов можно проиллюстрировать, используя термины геометрического подхода и ограничившись для простоты случаем распознавания двух различных образов (А и В).

Согласно геометрическому подходу необходимо разделить две заранее не определенные области в некотором пространстве, где отображаются точки из этих областей. О самих областях нет каких-либо сведений о их расположении, границах или принадлежности каждой точки к какой-либо области.

В процессе обучения системе предъявляются случайно выбранные точки, и сообщается информация только о том, к какой области (А или В) принадлежат предъявляемые точки. В качестве цели обучения можно задать, например, построение разделяющей поверхности для всех точек, предъявляемых в процессе обучения, а также и для всех остальных точек, принадлежащих этим областям. Иногда в качестве цели задается построение поверхностей, ограничивающих искомые области, в которых находятся точки каждого образа. С математической точки зрения задача обучения состоит в построении некоторых функций от векторов-изображений. При этом значения функций должны быть, например, положительны на всех точках одного образа и отрицательны на всех точках других образов. В связи с предположением, что области не имеют общих точек, можно построить некоторое множество (или хотя бы одна) разделяющих функций.

В случае, если необходимо выделить не два, а n образов, то задача состоит в построении поверхности, разделяющей все n областей, соответствующие этим образам, друг от друга. Задача эта может быть решена, например, путем построения общей функции, принимающей над точками каждой из n областей определенные, но разные значения. Можно решать данную задачу и путем построения n различных функций, каждая из которых принимает одно значение над точками одной из областей, и другое — над точками остальных.

Выбор того или иного подхода к выбору и построению функции, разделяющей области точек существенно зависит от свойств этих областей. Можно указать бесчисленное количество различных областей, кото-

рые содержат заданные точки, и при любом построении по ним разделяющей поверхности, всегда найдется другая область, пересекающая эту поверхность и содержащая данные точки.

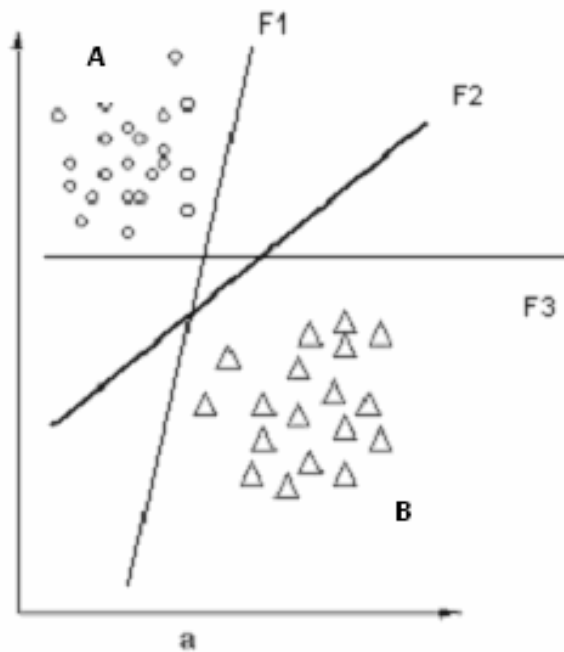


Рис. 7.2а. Линейная разделяющая поверхность

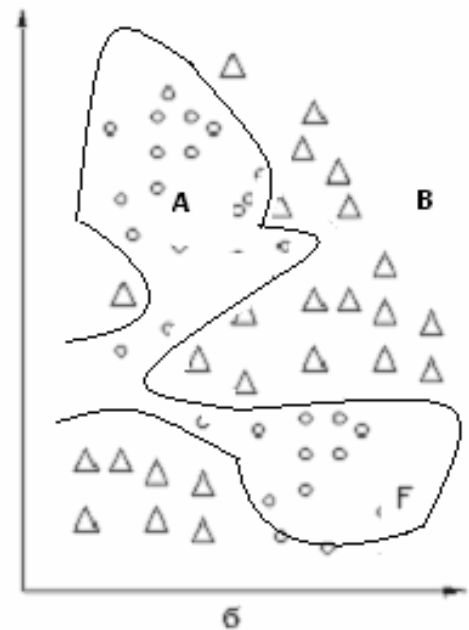


Рис. 7.2б. Нелинейная разделяющая поверхность

Для решения математической задачи об аппроксимации функции по информации о ней в ограниченном множестве точек необходимо ввести определенные ограничения на классе рассматриваемых функций с учетом характера информации, добавляемой учителем в процессе обучения. Обычно используется гипотеза о компактности образов.

Понятно, что аппроксимация разделяющей функции будет более легкой задачей, если области, подлежащие разделению, компактны и разнесены в пространстве. На рис. 7.2а, показан случай простого разделения компактных образов А и В, а на рис. 7.2б заведомо более сложного. Действительно, в случае, изображенном на рис. 7.2а, области могут быть разделены некоторыми линейными плоскостями F1, F2, F3 и т.д., и даже при

больших погрешностях в определении разделяющей функции они все же будут продолжать разделять области. В случае же на рис. 7.2б, разделение осуществляется сложной поверхностью F , и при незначительных отклонениях ее формы возможны ошибки разделения. Используя подобные представления о различном расположении областей в пространстве признаков была выдвинута гипотеза компактности - «образам соответствуют компактные множества в пространстве признаков».

Выдвигая подобную гипотезу исходили из предположения, что удачное задание соответствующего пространства признаков определяет и само свойство делимости образов в этом пространстве согласно некоторой неизвестной, но существующей классификации. При этом под компактными множествами образов понимаются некие области более плотного расположения точек в пространстве изображений, между которыми существуют пустые разделяющие области.

Были предприняты многочисленные попытки проверки этой гипотезы на основе реальных экспериментальных данных. Однако эксперименты не всегда подтверждали эту гипотезу и показали, что она хорошо выполняется в задачах, где удается найти простое решение (задачи типа рис. 7.2а). Но гипотеза не подтверждалась для сложных задач типа рис. 7.2б, в которых или не удавалось найти решение, или они решались с большим трудом с привлечением нечеткого подхода.

Тем не менее формулировка гипотезы компактности способствовала формированию понятия абстрактных образов в случайных пространствах, образующихся при случайном выборе координат пространства признаков. Соответствующие изображения в них также будут распределены случайно, а их распределения в некоторых частях пространства могут располагаться различным образом. Считая некоторое случайно выбранное пространство абстрактным изображением, можно предположить существование в этом абстрактном пространстве компактных множеств точек. Тогда в соответст-

вии с гипотезой компактности, можно назвать абстрактными образами данного пространства те множества объектов, которым в абстрактном пространстве соответствуют компактные множества точек.

Следует отметить, что при рассмотрении проблем обучения распознаванию образов используют также структурный (лингвистический) подход, например, при распознавании зрительных изображений в системах технического зрения роботов [4]. Перечислим основные этапы этого подхода.

Сначала строится словарь исходных понятий, включающий типичные наборы. Для этого:

- выделяется набор исходных понятий — типичных фрагментов, встречающихся на изображениях («точка», «линия», «угол прямой» и т.п.);
- выделяется набор характеристик взаимного расположения фрагментов («слева», «снизу», «внутри» и т.д.).

Затем с использованием понятий словаря строятся различные логические высказывания, называемые иногда предположениями.

Из большого количества высказываний отбираются те, которые являются наиболее существенные для данного конкретного случая.

Решается задача построения достаточно полных описаний образов на основе сегментации изображений объектов на составные части.

Построенные полные описания используются для решения вопроса о принадлежности заданного объекта определенному образу.

При лингвистической интерпретации устанавливается аналогия между структурой изображений и синтаксисом языка для использования аппарата математической лингвистики.

Кратко рассмотрим другие разновидности задач, решаемых в различных типах систем распознавания образов. Изображения, представленные на рис. 7.1 характеризуют так называемую задачу обучения с учителем. Для каждой из этих задач представлены обучающие последовательно-

сти правильно решенных задач и можно использовать информацию о принадлежности каждого объекта из обучающей последовательности тому или иному образу.

Следуя материалам [21] кратко изложим постановку решения другой классификационной задачи – задачи самообучения (обучения без учителя).

На описательном уровне такие задачи формулируются следующим образом: системе последовательно предъявляются объекты без каких-либо указаний об их принадлежности к образам. На входном устройстве системы множество объектов отображается на множество изображений. Система ОРО с использованием некоторого заложенного в нее заранее свойства делимости образов производит самостоятельную классификацию предъявляемых объектов из обучающей последовательности.

После некоторого процесса самообучения система должна научиться распознавать и уже знакомых объекты, и те, которые ранее ей не предъявлялись, но относятся к изученным классам..

Таким образом] «...процессом самообучения некоторой системы называется такой процесс, в результате которого эта система без подсказки учителя приобретает способность к выработке одинаковых реакций на изображения объектов одного и того же образа и различных реакций на изображения различных образов. Роль учителя при этом состоит лишь в подсказке системе некоторого объективного свойства, одинакового для всех образов и определяющего способность к разделению множества объектов на образы...» [21].

Как уже указывалось выше, в качестве такого объективного свойства задается свойство компактности образов о взаимном расположении точек в выбранном пространстве признаков. Оно и содержит информацию о том, как следует разделить множество точек. Эта информация и определяет то свойство делимости образов, которое оказывается достаточным для самообучения системы распознаванию образов.

В [21] отмечалось, что «...большинство известных алгоритмов самообучения способны выделять только абстрактные образы, т.е. компактные множества в заданных пространствах. Различие между ними состоит, по-видимому, в формализации понятия компактности... иногда это повышает ценность алгоритмов самообучения, так как часто сами образы заранее никем не определены, а задача состоит в том, чтобы определить, какие подмножества изображений в заданном пространстве представляют собой образы... В таком понимании задачи алгоритмы самообучения генерируют заранее не известную информацию о существовании в заданном пространстве образов, о которых ранее никто не имел никакого представления.

Кроме того, результат самообучения характеризует пригодность выбранного пространства для конкретной задачи обучения распознаванию. Если абстрактные образы, выделяемые в процессе самообучения, совпадают с реальными, то пространство выбрано удачно. Чем сильнее абстрактные образы отличаются от реальных, тем «неудобнее» выбранное пространство для конкретной задачи.

В процессе обучения в некоторой системе производится выработка той или иной реакции на группы внешних идентичных сигналов путем многократного воздействия на систему сигналов обратной связи – внешней корректировки. Такую внешнюю корректировку в обучении принято называть «поощрениями» и «наказаниями». Механизм генерации этой корректировки практически полностью определяет тот или иной алгоритм обучения. При самообучении системе не сообщается дополнительная информация о верности реакции. В случае изменяющихся условиях работы системы для поддержания требуемого качества обучения используют методы адаптации.

Адаптация — это процесс изменения параметров и структуры системы, а возможно, и управляющих воздействий на основе текущей информа-

ции с целью достижения определенного состояния системы при начальной неопределенности и изменяющихся условиях работы.

Обучение — это процесс, в результате которого система постепенно приобретает способность отвечать нужными реакциями на определенные совокупности внешних воздействий, а адаптация — это подстройка параметров и структуры системы с целью достижения требуемого качества управления в условиях непрерывных изменений внешних условий...»

Для практической реализации решения задач обучения распознаванию образов в настоящее время разработано большое число различных методов и алгоритмов [13, 21, 45, 46, 47]. Рассмотрим самый простой подход, связанный с построением кусочно-линейных моделей, обучающихся распознаванию образов, и реализованный одним из первых в классе устройств типа персептрона.

Персептроны.

Персептроны реализуют один из методов решения задач обучения распознаванию образов, основанный на моделировании работы гипотетического механизма человеческого мозга. Структура модели заранее постулируется на уровне биологических знаний или гипотез о биологических механизмах работы мозга. Первый персептрон был предложен в 1962 году известным американским нейрофизиологом Френком Розенблаттом. Описание и структура простейшего персептрона воспроизведены из работы [21]. Структура представлена на рис. 7.3.

Структура простого персептрона включает совокупность чувствительных (сенсорных) элементов (S-элементов), на которые поступают входные сигналы. S-элементы связаны случайным образом с совокупностью ассоциативных элементов (А-элементов). Выходы А-элементов отли-

чаются от нуля только тогда, когда возбуждено достаточно большое число S-элементов, воздействующих на один A-элемент.

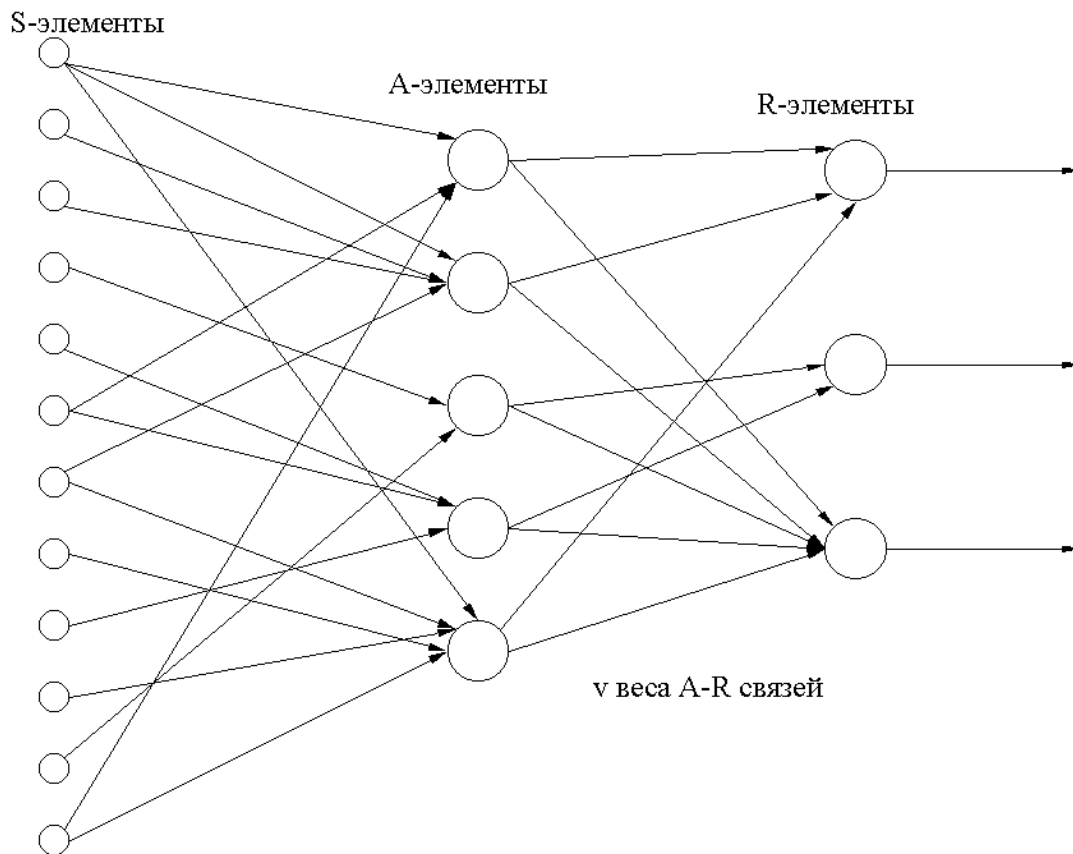


Рис. 7.3. Структура простейшего персептрона

Выходы A-элементов соединены с реагирующими элементами (R-элементами) связями, коэффициенты усиления (v) которых переменны и изменяются в процессе обучения. Взвешенные комбинации выходов R-элементов составляют реакцию системы, которая указывает на принадлежность распознаваемого объекта определенному образу. Если распознаются только два образа, то в персептроне устанавливается только один R-элемент, который обладает двумя реакциями — положительной и отрицательной. Если образов больше двух, то для каждого образа устанавливают свой R-элемент, а выход каждого такого элемента представляет линейную комбинацию выходов A-элементов:

$$R_j = \Theta_j + \sum_{i=1}^n v_{ij} x_i, \quad (7.1)$$

где R_j — реакция j -го R-элемента; x_i — реакция i -го A-элемента; v_{ij} — вес связи от i -го A-элемента к j -му R элементу; Θ_j - порог j -го R-элемента.

Аналогично записывается уравнение i -го A-элемента:

$$x_i = \Theta_i + \sum_{k=1}^s y_k \quad (7.2).$$

Здесь сигнал y_k может быть непрерывным, но чаще всего он принимает только два значения: 0 или 1. Сигналы от S-элементов подаются на входы A-элементов с постоянными весами равными единице, но каждый A-элемент связан только с группой случайно выбранных S-элементов. Предположим, что требуется обучить перцептрон различать два образа V_1 и V_2 . Будем считать, что в перцептроне существует два R-элемента, один из которых предназначен образу V_1 , а другой — образу V_2 . Перцептрон будет обучен правильно, если выход R_1 превышает R_2 , когда распознаваемый объект принадлежит образу V_1 , и наоборот. Разделение объектов на два образа можно провести и с помощью только одного R-элемента. Тогда объекту образа V_1 должна соответствовать положительная реакция R-элемента, а объектам образа V_2 — отрицательная.

Перцептрон обучается путем предъявления обучающей последовательности изображений объектов, принадлежащих образам V_1 и V_2 . В процессе обучения изменяются веса v_i A-элементов. В частности, если применяется система подкрепления с коррекцией ошибок, прежде всего учитывается правильность решения, принимаемого перцептроном. Если

решение правильно, то веса связей всех сработавших А-элементов, ведущих к R-элементу, выдавшему правильное решение, увеличиваются, а веса несработавших А-элементов остаются неизменными. Можно оставлять неизменными веса сработавших А-элементов, но уменьшать веса несработавших. В некоторых случаях веса сработавших связей увеличивают, а несработавших — уменьшают. После процесса обучения перцептрон сам, без учителя, начинает классифицировать новые объекты.

Если перцептрон действует по описанной схеме и в нем допускаются лишь связи, идущие от бинарных S-элементов к А-элементам и от А-элементов к единственному R-элементу, то такой перцептрон принято называть элементарным α -перцептроном. Обычно классификация $C(W)$ задается учителем. Перцептрон должен выработать в процессе обучения классификацию, задуманную учителем.

Френком Розенблаттом было сформулировано и доказано несколько основополагающих теорем о работе перцептрона, две из которых определяют основные свойства перцептрона, они приведены ниже.

Теорема 7.1. Класс элементарных α -перцептронов, для которых существует решение для любой задуманной классификации, не является пустым.

Эта теорема утверждает, что для любой классификации обучающей последовательности можно подобрать такой набор (из бесконечного набора) А-элементов, в котором будет осуществлено задуманное разделение обучающей последовательности при помощи линейного решающего пра-

вила

$$R_j = \Theta_j + \sum_{i=1}^n v_{ij} x_i. \quad (7.3)$$

Теорема 7.2. Если для некоторой классификации $S(W)$ решение существует, то в процессе обучения α -персептрона с коррекцией ошибок, начинающегося с произвольного исходного состояния, это решение будет достигнуто в течение конечного промежутка времени.

Смысл этой теоремы состоит в том, что если относительно задуманной классификации можно найти набор A -элементов, в котором существует решение, то в рамках этого набора оно всегда будет достигнуто в конечный промежуток времени.

Однако, необходимо иметь в виду, что при формулировке подобных теорем обычно обсуждают свойства так называемого бесконечного персептрона, т.е. персептрона с бесконечным числом A -элементов со всевозможными связями с S -элементами (полный набор A -элементов). В таких персептронах решение действительно всегда существует, а раз оно существует, то оно и достижимо в α -персептронах с коррекцией ошибок.

Очень интересную область исследований представляют собой многослойные персептроны и персептроны с перекрестными связями, но теория этих систем практически мало разработана.

Отметим, что предложенная и исследованная Розенблаттом модель персептрона фактически стала первой машинной реализацией простейшей нейронной сети. В развитие этих исследований в 70-е годы прошлого века было предложено много и других интересных разработок. Например, была разработана модель когнитрона, способного хорошо распознавать достаточно сложные образы независимо от поворота и изменения масштаба изображения. Все эти исследования стали той базой, на основе которой сформировалось современное перспективное направление нейрокибернетики – нейронные сети. Краткие понятия об основных видах классических нейронных сетей приведены в теме 8.

Тема 8. НЕЙРОННЫЕ СЕТИ

История исследований в области нейронных сетей началась в 50-х годах прошлого века. Впервые способность нейронной сети к обучению была исследована Дж. Маккалоком и У. Питом, в 1943 году вышла их работа «Логическое исчисление идей, относящихся к нервной деятельности», в которой была построена модель нейрона и сформулированы принципы построения искусственных нейронных сетей (ИНС). В 1982 году американский биофизик Дж. Хопфилд предложил оригинальную модель нейронной сети, названную его именем. В последующие несколько лет было найдено множество эффективных алгоритмов: сеть встречного потока, двунаправленная ассоциативная память и др. В 1986 году Дж. Хинтон и его коллеги опубликовали статью с описанием модели нейронной сети и алгоритмом ее обучения с обратным распространением ошибки (back propagation). Это дало новый толчок исследованиям в области искусственных нейронных сетей.

В 70-х годах прошлого века в Киевском институте Кибернетики велись работы над стохастическими нейронными сетями. В настоящее время в России существует несколько сильных школ нейроинформатики. Информацию о проводимых в них современных исследованиях, о разработках нейронных сетей и их применениях можно найти в [13, 21-23, 39]. Тематика современных исследований в области ИНС очень обширна, поэтому ограничимся только кратким тезисным описанием основных понятий ИНС, принципов построения и применения.

Искусственная нейронная сеть, как и ее биологический прототип, состоит из множества одинаковых элементов — нейронов. С рассмотрения модели отдельного нейрона и начнем знакомство с ИНС, следуя в основном материалам [21].

Биологический нейрон моделируется в ИНС как устройство, имеющее несколько входов (дендритов) и один выход (аксон). Каждому входу ставится в соответствие некоторый весовой коэффициент (w_i), характеризующий пропускную способность i -го канала и оценивающий степень влияния сигнала с этого i -го входа на сигнал « u » на выходе.

Построение современных ИНС значительно разнообразнее своего биологического прообраза, однако внешняя аналогия функционирования сохраняется. Под действием сигналов, поступающих в данный нейрон от других нейронов, происходит возбуждение этого нейрона и генерируется собственный сигнал, передаваемый окружающим нейронам и т.д. По аналогии с биологическими нейронами в искусственном нейроне через некоторое время происходит процесс торможения, и активность возбужденных нейронов на короткое время прекращается. Затем происходит повторение процессов возбуждения и торможения.

Математическая модель нейрона представляет по существу нелинейный функциональный преобразователь. В зависимости от конкретной реализации обрабатываемые нейроном сигналы могут быть аналоговыми или цифровыми (1 или 0). В теле нейрона происходит взвешенное суммирование входных возбуждений, и далее это суммарное значение « d » является аргументом активационной функции нейрона. Таким образом типовым элементом ИНС является искусственный нейрон, состоящий из сумматора и активационного элемента, реализующего преобразующую функцию F . Один из возможных вариантов искусственного нейрона представлен на рис. 8.1, воспроизведенным из [21].

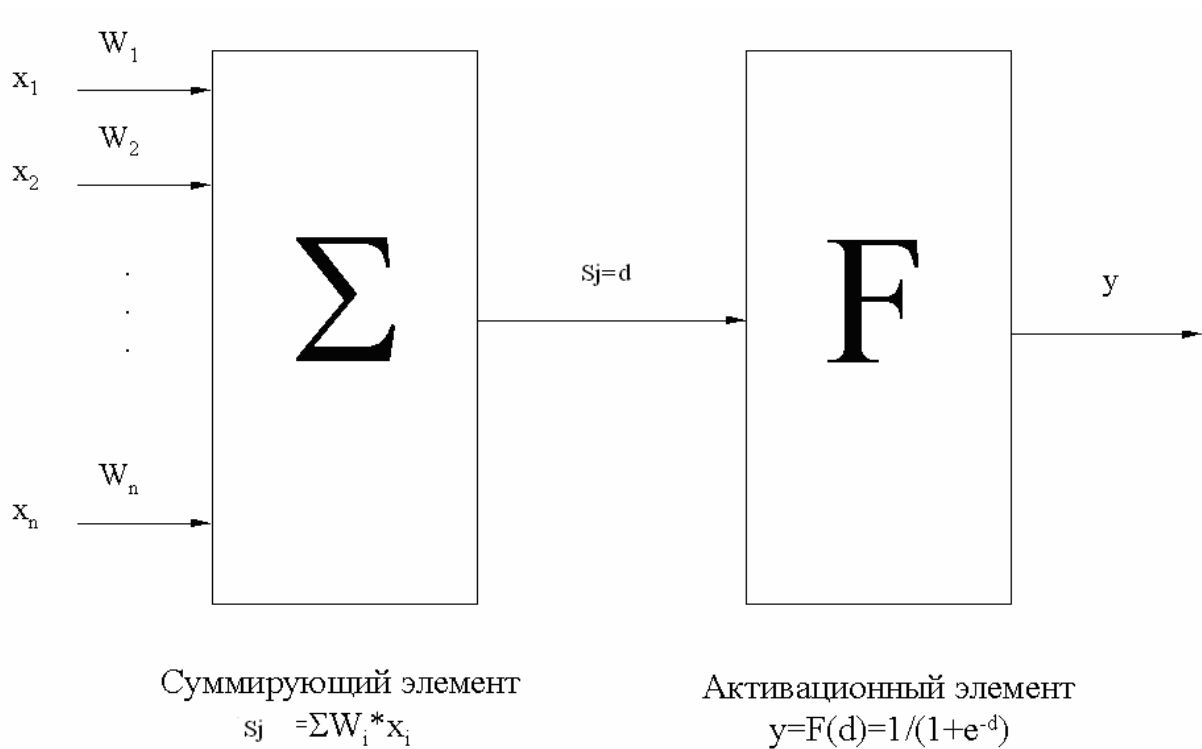


Рис. 8.1 Искусственный нейрон.

В сумматоре реализуется функция вида:

$$S_j = w_0 + \sum_{i=1}^n w_i x_i, \quad (8.1)$$

где x_i – значения информационных входов ($x_i \in [0, +1]$, $i \in [1, \dots, n]$), w_i – веса входов ($i \in [1, \dots, n]$), w_0 – смещение в пространстве входов признаков.

Назначая веса $w_i < 0$, получим запрещающие входы.

Будучи соединенными определенным образом, нейроны образуют нейронную сеть. При задании топологии сети на множестве нейронов различают «слои». Связь между слоями может быть однонаправленной или с «обратным распространением». Различные варианты моделей нейронов и сетей отличаются количеством и типом входов, топологией слоев и соеди-

нений, видом преобразующей функции F , выбираемой в зависимости от специфики решаемой задачи. Практически применяются десятки типов преобразующих функций, но чаще всего берут сигмоидную функцию, назначая различные управляющие параметры:

$$Y = f(x) = \frac{1}{1 + e^{-\alpha d}}; \quad (8.2)$$

где α – управляющий параметр ($\alpha \in [0, +\infty]$); значения d могут быть любыми или нормированными, например, ($d \in [-1, +1]$); выходные значения Y по определению лежат в диапазоне $[0, 1]$.

Если выбрать функцию (8.1) так, чтобы Y принимало два значения (0 и 1), то получим персептрон Розенблатта. При выборе различных значений α (с учетом диапазона изменения d) можно получить функции нелинейного усиления слабых сигналов, линейный порог, смещение и т.п.

Для практического использования ИНС необходимо предварительно выполнить этап ее обучения. Под обучением ИНС понимается итеративный процесс адаптации сети к предъявляемым эталонным образцам (обучающей выборке) путем модификации весовых коэффициентов связей w_i между нейронами с использованием различных алгоритмов и критериев. Обычно изменение производится в направлении антиградиента функции ошибок в пространстве весов синапсов. В результате работы алгоритма либо происходит переход системы в новое устойчивое состояние (весовые коэффициенты стабилизируются), либо система зацикливается.

Качество решения конкретных задач в ИНС зависит от структуры сети, числа нейронов и связей, а также от качества входных данных. По результатам обучения ИНС выполняется операция контрастирования, т.е.

удаления лишних элементов сети, не влияющих на решение. Ставятся даже задачи построения «минимальной идеальной сети» [39].

Среди различных структур ИНС наиболее популярной является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя, а первый слой связан со всеми входами ИНС. Такие ИНС называются полносвязными. Практическое применение имеют и однослойные полносвязные сети, например, сеть Хопфилда.

В однослойных сетях алгоритм обучения ИНС «с учителем» довольно прост, так как заведомо известны заданные выходные состояния нейронов единственного слоя, и подстройка весов синапсов производится по минимуму суммарной ошибки на выходе сети. По этому принципу строится, например, алгоритм обучения однослойного персептрона.

В многослойных ИНС оптимальные выходные значения нейронов всех слоев, кроме последнего, обычно, не известны. При 2 и более слоях ИНС невозможно обучить, зная только величины ошибок на выходах. Существуют различные подходы к решению этой проблемы.

Один из вариантов решения – разработка для каждого слоя ИНС наборов выходных сигналов, соответствующих входным. Операция достаточно трудоемкая и не всегда осуществимая. Вторым вариантом – динамическая подстройка весов синапсов с изменением и сохранением только тех изменений, которые привели к уменьшению ошибки на выходе всей сети. Очевидно, что данный метод перебора различных сочетаний методом «проб и ошибок» требует громоздких рутинных вычислений. Третий, наиболее приемлемый вариант – распространение сигналов ошибки от выходов ИНС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Такой алгоритм обучения ИНС полу-

чил название процедуры обратного распространения сигнала ошибки и широко используется на практике. Теоретически ИНС с обратным распространением сигнала между слоями позволяет аппроксимировать любую функцию с заданной точностью. Практически это возможно, когда общее количество независимых входов X в ИНС не превышает нескольких тысяч. Достаточно подробно процедура описана в [21, 39], ниже приводится только краткое изложение алгоритма, адаптированное из [21].

В основе процедуры обратного распространения сигналов ошибки лежит идея минимизации методом градиентного спуска следующей квадратичной целевой функцией ошибки ИНС:

$$E(w) = \frac{1}{2} \eta \sum_{jp} (Y_{jp}^{(N)} - Z_{jp})^2 \rightarrow \min, \quad (8.3)$$

где $Y_{jp}^{(N)}$ – реальное выходное состояние нейрона j выходного слоя N -слойной нейронной сети при подаче на ее входы p -го образа; Z_{jp} – заданное выходное состояние этого нейрона; η – коэффициент скорости обучения, $0 < \eta < 1$. Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам.

Минимизируя (8.3) методом градиентного спуска, получим выражения для подстройки весовых коэффициентов в виде:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}}, \quad (8.4)$$

Здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей i -й нейрон слоя $n-1$ с j -ым нейроном слоя n , η – коэффициент скорости обучения, $0 < \eta < 1$.

Потребуем, чтобы активационные функции нейронов были гладкими функциями, такими как гиперболический тангенс или классический сигмоид с экспонентой, т.е. не содержали недифференцируемых неоднородностей типа единичного скачка. Тогда производные активационной функции будут определены на всей оси абсцисс. Обозначим d в (8.1) через s_j . Рассмотрим выражение (8.5) для производной функции (8.3) по весам синапсов w_{ij} :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \quad (8.5).$$

Здесь y_j , – выход нейрона j , s_j – взвешенная сумма его входных сигналов, т.е. аргумент активационной функции. Множитель dy_j/ds_j является производной этой функции по ее аргументу и определен на всей оси абсцисс. Что касается первого множителя в (8.5), он легко раскладывается следующим образом [21]:

$$y_i(t) = y_i(t-1) + \alpha [y_j^{(n-1)}(t) - y_j^{(n-1)}(t-1)] * [y_j^n(t) - y_j^n(t-1)], \quad (8.6).$$

Здесь суммирование по k выполняется среди нейронов слоя $n+1$.

Третий множитель $\partial s_j / \partial w_{ij}$, равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$.

Введя новую переменную

$$\delta_j^n = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j}, \quad (8.7)$$

получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя $n+1$:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{\partial y_j}{\partial s_j}. \quad (8.8)$$

Для выходного слоя

$$\delta_j^{(N)} = (y_i^{(N)} - d_i) \frac{\partial y_i}{\partial s_j} \quad (8.9)$$

Теперь формулу подстройки весовых коэффициентов ИНС (8.4) можно представить в следующем виде:

$$\Delta w_{ij}^{(n)} = -\eta \delta_j^{(n)} y_i^{(n-1)} \quad (8.10)$$

Для сглаживания резких скачков при перемещении по поверхности целевой функции и придании процессу коррекции весов некоторой инерционности формула (8.10) дополняется значением изменения весов на предыдущей итерации

$$\Delta w_{ij}^{(n)}(t) = -\eta (\mu \Delta w_{ij}^{(n)}(t-1) + (1 - \mu) \delta_j^{(n)} y_i^{(n-1)}), \quad (8.11)$$

где μ – коэффициент инерционности. t – номер текущей итерации.

С учетом выписанных преобразований полный алгоритм обучения ИНС с помощью процедуры обратного распространения ошибки от выходов ИНС к ее входам формулируется следующим образом:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования ИНС, когда сигналы распространяются от

входов к выходам, рассчитать значения последних. При этом вместо (8.1) используем:

$$S_j^{(n)} = \sum_{k=0}^M y_i^{(n-1)} w_{ij}^{(n)} \quad (8.12)$$

где M – число нейронов в слое $n-1$ с учетом нейрона с постоянным выходным состоянием $+1$, задающего смещение; $y_i^{(n-1)} = x_{ij}^{(n)}$ – i -ый вход нейрона j слоя n :

$$y_j^{(n)} = f(S_j^{(n)}), \quad (8.13)$$

где $f()$ – сигмоид,

$$y_q^{(0)} = I_q, \quad (8.14)$$

где I_q – q -ая компонента вектора входного образа.

2. Рассчитать $\delta^{(N)}$ для выходного слоя по формуле (8.9) и по формуле (8.10) или (8.11) рассчитать изменения весов $\Delta w^{(N)}$ слоя N .

3. Рассчитать по формулам (8.8) и (8.10) (или (8.8) и (8.11)) соответственно $\delta^{(n)}$ и $\Delta w^{(n)}$ для всех остальных слоев, $n=N-1, \dots, 1$.

4. Скорректировать все веса в ИНС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t), \quad (8.15)$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Чтобы ИНС «не забывала» одни образы по мере запоминания других, сети предъявляются в случайном порядке на шаге 1 попеременно все тренировочные образы из обучающей последовательности.

Из выражения (8.10) следует, что когда выходное значение $y_i^{(n-1)}$ стремится к нулю, эффективность обучения заметно снижается. Исследования показывают, что при двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться, поэтому область возможных значений выходов нейронов $[0,1]$ желательно сдвинуть в пре-

делы $[-0.5, +0.5]$. Этого можно достичь простыми модификациями логистических функций. Например, сигмоид с экспонентой преобразуется к виду

$$\sum_{i=1}^L \mu_i(x) = 1 \quad (8.16)$$

Доказательство сходимости обучения в процессе обратного распространения основано на производных, то есть приращения весов и, следовательно, скорость обучения должны быть бесконечно малыми, однако в этом случае обучение будет происходить неприемлемо медленно. С другой стороны, слишком большие коррекции весов могут привести к постоянной неустойчивости процесса обучения. Поэтому в качестве η обычно выбирается число меньше 1, но не очень маленькое, например, 0.1, и оно, вообще говоря, может постепенно уменьшаться в процессе обучения. Кроме того, для исключения случайных попаданий в локальные минимумы иногда, после того как значения весовых коэффициентов стабилизируются, η временно сильно увеличивают, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет алгоритм в одно и то же состояние ИНС, можно более или менее уверенно сказать, что найден глобальный максимум, а не какой-то другой.

Рассмотренный выше алгоритм обучения нейронной сети с помощью процедуры обратного распространения подразумевает наличие некоего внешнего звена, предоставляющего сети кроме входных, так же и целевые выходные образы. Алгоритмы, пользующиеся подобной концепцией, называются алгоритмами обучения с учителем.

Процесс обучения без учителя, как и в случае обучения с учителем, заключается в подстраивании весов синапсов. Некоторые алгоритмы, правда, изменяют и структуру сети, то есть количество нейронов и их взаимосвязи. Очевидно, что подстройка синапсов может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весовых коэффициентов. Исходя из этого сооб-

ражения и, что более важно, по аналогии с известными принципами самоорганизации нервных клеток, построены сигнальный и дифференциальный алгоритмы обучения Хебба.

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha y_i^{(n-1)} y_j^{(n)} \quad (8.17)$$

где $y_i^{(n-1)}$ – выходное значение нейрона i слоя $(n-1)$, $y_j^{(n)}$ – выходное значение нейрона j слоя n ; $w_{ij}(t)$ и $w_{ij}(t-1)$ – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях t и $t-1$ соответственно; α – коэффициент скорости обучения. Здесь и далее, для общности, под n подразумевается произвольный слой сети. При обучении по данному методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба, в котором сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения Хебба будет выглядеть так:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.

2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных (feedforward) сетей, то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная (передаточная) функция нейрона, в результате чего получается его выходное значение $y_i^{(n)}$, $i=0..M_{i-1}$, где M_i – число нейронов в слое i ; $n=0..N-1$, а N – число слоев в сети.

3. На основании полученных выходных значений нейронов по формуле (8.17) производится изменение весовых коэффициентов.

4. Цикл с шага 2, пока выходные значения сети не стабилизируются с заданной точностью. Применение этого нового способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора.

Другой алгоритм обучения без учителя – алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha [y_i^{(n-1)} - w_{ij}(t-1)] \quad (8.18).$$

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $y_i^{(n-1)}$, и весовыми коэффициентами его синапсов.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально подходят на входной образ, и подстройка весов по формуле (8.18) проводится только для него. Эта так называемая аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весовых коэффициентов с вектором входных значений. Максимальное произведение дает выигравший нейрон.

Другой вариант – расчет расстояния между этими векторами в p -мерном пространстве, где p – размер векторов:

$$D_j = \sqrt{\sum_{i=0}^{p-1} (y_i^{(n-1)} - w_{ij})^2}, \quad (8.19)$$

где j – индекс нейрона в слое n , i – индекс суммирования по нейронам слоя $(n-1)$, w_{ij} – вес синапса, соединяющего нейроны; выходы нейронов слоя $(n-1)$ являются входными значениями для слоя n . Корень в формуле (8.19) брать не обязательно, так как важна лишь относительная оценка различных D_j .

В данном случае, «побеждает» нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы «уравнять права» всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

При использовании обучения по алгоритму Кохонена существует практика нормализации входных образов, а также – на стадии инициализации – и нормализации начальных значений весовых коэффициентов:

$$x_i = x_i / \sqrt{\sum_{j=0}^{n-1} x_j^2}, \quad (8.20)$$

где x_i – i -ая компонента вектора входного образа или вектора весовых коэффициентов, а n – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весовых коэффициентов случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Для избежания такой ситуации используется метод выпуклой комбинации. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t)x_i + (1 - \alpha(t)) \frac{1}{\sqrt{n}}, \quad (8.21)$$

где x_i – i -ая компонента входного образа, n – общее число его компонент, $\alpha(t)$ – коэффициент, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы сети подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным. Весовые коэффициенты устанавливаются на шаге инициализации равными величине

$$w_0 = \frac{1}{\sqrt{n}}, \quad (8.22)$$

где n – размерность вектора весов для нейронов инициализируемого слоя.

На основе рассмотренного выше метода строятся нейронные сети особого типа – так называемые самоорганизующиеся структуры – self-organizing feature maps. Для них после выбора из слоя n нейрона j с минимальным расстоянием D_j (8.19) обучается по формуле (8.18) не только этот нейрон, но и его соседи, расположенные в окрестности R . Величина R на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Таким образом, чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.

Среди различных конфигураций ИНС встречаются такие, при классификации которых по принципу обучения, строго говоря, не подходят ни обучение с учителем, ни обучение без учителя. В таких сетях весовые коэффициенты синапсов рассчитываются только однажды перед началом функционирования сети на основе информации об обрабатываемых данных, и все обучение сети сводится именно к этому расчету. Из сетей с подобной логикой работы наиболее известны сеть Хопфилда и сеть Хэмминга, которые обычно используются для организации ассоциативной памяти.

Структурная схема сети Хопфилда приведена на рис. 8.2. Она состоит из единственного слоя нейронов, число которых является одновременно числом входов и выходов сети. Каждый нейрон связан синапсами со всеми остальными нейронами, а также имеет один входной синапс, через который осуществляется ввод сигнала. Выходные сигналы, как обычно, образуются на аксонах. Задача, решаемая данной сетью в качестве ассоциативной памяти, как правило, формулируется следующим образом. Известен некоторый набор двоичных сигналов (изображений, звуковых оцифровок, прочих данных, описывающих некие объекты или характеристики процессов), которые считаются образцовыми. Сеть должна уметь из произвольного неидеального сигнала, поданного на ее вход, выделить («вспомнить» по частичной информации) соответствующий образец (если такой есть) или «дать заключение» о том, что входные данные не соответствуют ни одному из образцов.

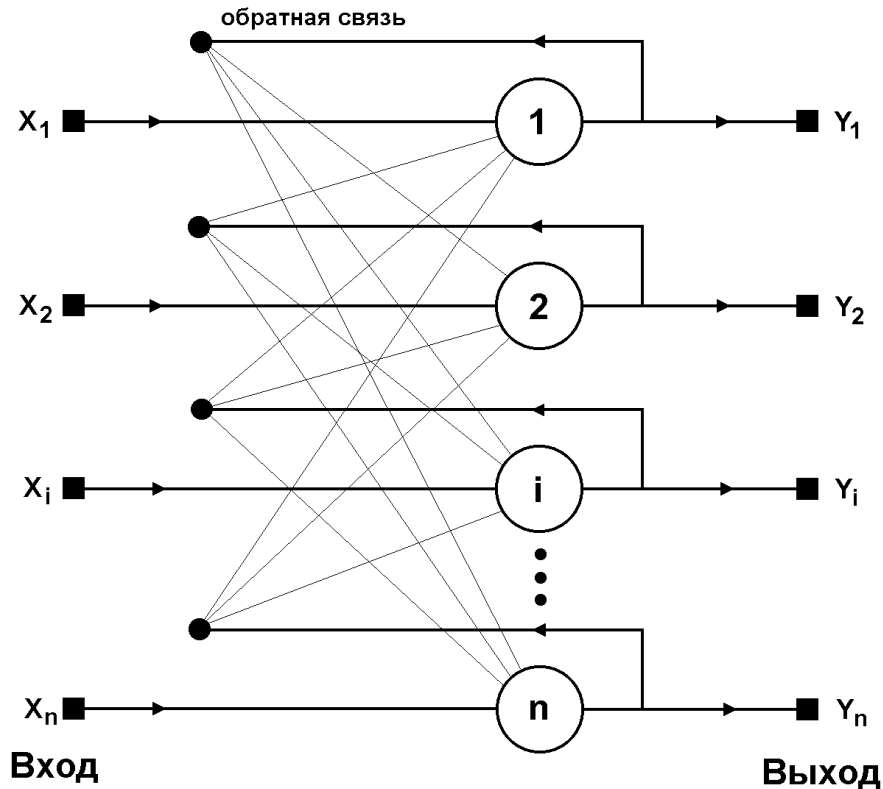


Рис. 8.2. Структурная схема сети Хопфилда

Если, например, сигналы представляют собой некие изображения, то, отобразив в графическом виде данные с выхода сети, можно будет увидеть картинку, полностью совпадающую с одной из образцовых (в случае успеха) или же «вольную импровизацию» сети (в случае неудачи).

Как говорилось выше, иногда сеть не может провести распознавание и выдает на выходе несуществующий образ. Это связано с проблемой ограниченности возможностей сети. Для сети Хопфилда число запоминаемых образов m не должно превышать величины, примерно равной $0.15 \cdot n$. Кроме того, если два образа А и Б очень похожи, они, возможно, будут вызывать у сети перекрестные ассоциации, то есть предъявление на входы сети вектора А приведет к появлению на ее выходах вектора Б и наоборот.

Когда нет необходимости, чтобы сеть в явном виде выдавала образец, то есть достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, меньшими затратами на память и объемом вычислений, что становится очевидным из ее структуры (рис. 8.3).

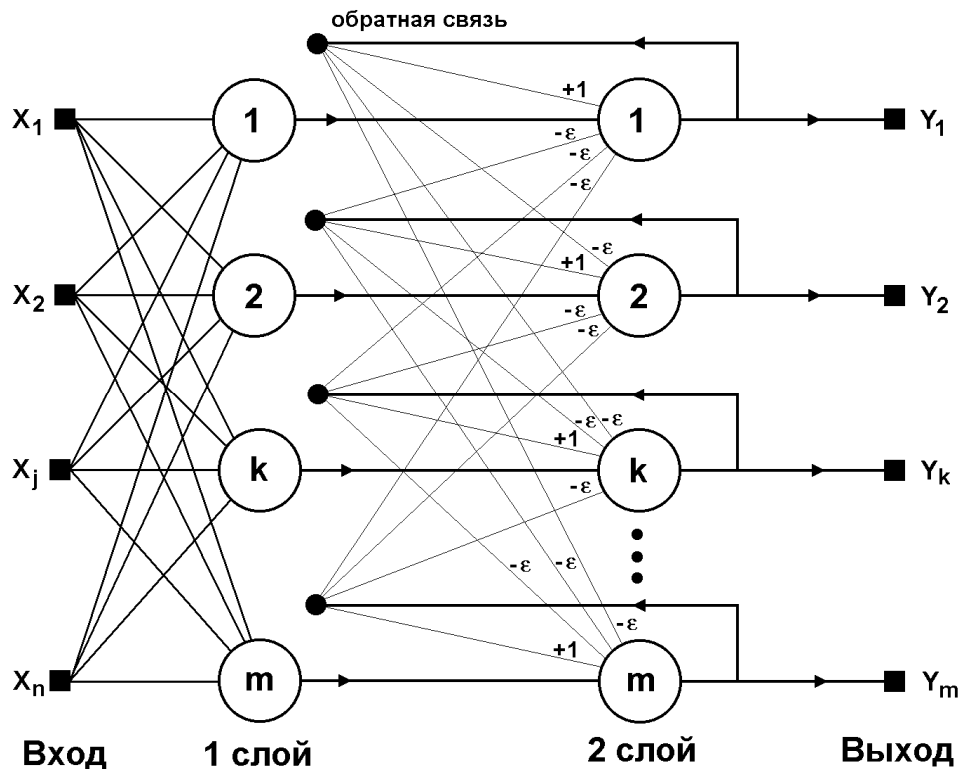


Рис. 8.3. Структурная схема сети Хэмминга.

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m – число образцов. Нейроны первого слоя имеют по n синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

Тема 9. КРАТКИЕ СВЕДЕНИЯ ОБ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВАХ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

На протяжении более пятидесяти лет в процессе создания и развития методов искусственного интеллекта параллельно велась разработка и совершенствование различных инструментальных средств для повышения эффективности создания, отладки и совершенствования интеллектуальных систем. В свою очередь, создание и развитие самих инструментальных средств разработки ИИ шло в двух основных направлениях.

Одно направление составили универсальные инструментальные средства – в основном языки программирования для искусственного интеллекта и некоторые универсальные языки программирования. Эти средства чаще применяются при создании разнообразных систем ИИ основанных на знаниях.

Второе направление – специализированные программные средства, специальные языки, конфигураторы баз данных, баз знаний и различные редакторы для поддержки различных моделей представления знаний в ИИ. Эти средства позволяют более эффективно создавать различные экспертные системы, строить системы интеллектуального управления на базе использования нечеткой логики, искусственных нейронных сетей, генетических алгоритмов и т.п.

За время развития ИИ было предпринято не мало различных попыток создания разнообразных инструментальных средств, построенных на основе различных идей, методов и подходов, иногда даже экзотических. Некоторые из них не получили дальнейшего развития и применения, другие же доказали свою полезность и эффективность и до сих пор активно используются в практике создания систем ИИ.

Рассмотрим первое направление – применение языков программирования для создания ИИ.

Разработчики языков делят алгоритмические языки на две группы. Первую группу образуют языки, которые называются языки операторного, или процедурного типа. Элементарными единицами программы являются здесь операторы, т.е. приказы, выполнение которых сводится к четко определенному изменению четко определенной части памяти машины. Типичным представителем этой группы является язык машины Поста. Сюда же относятся машинные языки конкретных ЭВМ, а также массовые языки программирования типа Фортран, Алгол, ПЛ/1, С ++.

Языки второй группы называются языками сентенциального, или декларативного типа (sentence — высказывание, предложение). Программа на таком языке представляется в виде набора предложений (соотношений, правил, формул), которые машина, понимающая данный язык, умеет каким-то образом применять к обрабатываемой информации. Простейшим примером сентенциального языка, созданного с теоретическими целями является язык нормальных алгоритмов Маркова.

Можно указать прообразы указанных типов алгоритмических языков в естественных языках. Для операторных языков это повелительное наклонение (императив, приказание), для сентенциальных – изъявительное наклонение (описание, повествование). Обращаясь к естественному языку, нетрудно заметить, что «изъявительное наклонение является несравненно более распространенным и образует, в сущности, основу языка, в то время как повелительное наклонение предстает в виде некоторой специальной модификации». Таким образом, можно сделать вывод о том, что «относительный вес изъявительного наклонения является мерой развитости языка».

Основными языками программирования в области ИИ в настоящее время считаются: ЛИСП ПРОЛОГ, РЕФАЛ1, и С ++ [17, 19, 26, 33]. Первые три являются узкоспециализированными языками программирования,

Ближайший русский аналог ПРОЛОГа. – РЕФАЛ – разработан в (МГУ).

специально разработанные для ИИ, последний – инструментальный язык общего назначения, достаточно мощный для того, чтобы создавать, в том числе и программы ИИ. С ++ является, пожалуй, самым популярным профессиональным языком, на базе которого или с его использованием построен целый ряд специализированных языков для ИИ, программ-оболочек и интерфейсов.

Применение LISPа, ПРОЛОГа и РЕФАЛа позволяет разработчику систем ИИ сосредоточиться на логике решения задачи, а применение С ++ – добиться высокой скорости работы программы и эффективно реализовать нестандартные процедуры обработки данных. С учетом высоких характеристик быстродействия и «поддержки пользователя» современными средствами вычислительной техники долгое и кропотливое кодирование на С ++ оправдано только при создании программ-оболочек экспертных систем и интерфейсов.

Программа-оболочка, или матрица, называемая иногда «Пустой экспертной системой», – это программа, содержащая уже механизм вывода, средства ввода и редактирования фактов, средства объяснения и не содержащая конкретных фактов конкретной предметной области. Разработчики ЭС могут наполнить Матрицу собственными фактами, правилами, отредактировать правила, и она начнет делать выводы по интересующей Вас проблеме.

Другими, менее популярными языками ИИ являются SmallTalk, Forth, LOGO, ПЛЭНЕР (Planner), а также ПАСКАЛЬ, Бейсик и др. [25, 26, 28, 32, 34, 38]. Ниже приведены краткие сведения о ПРОЛОГе, адаптированные из [17].

9.1. Краткие сведения о ПРОЛОГе

ПРОЛОГ (программирование в терминах логики, логическое программирование) разработан в 1974 г. в университете Марсея (Франция) А. Колмаэро (Alan Colmerauer) на основе фундаментальных работ А. Робин-

сона (John Alan Robinson) и Р. Ковальского (Robert A. Kowalski). Ряд специалистов считает ПРОЛОГ более удобным в качестве учебного языка, а ЛИСП относят к профессиональному средству. Но с точки зрения вычислительной мощности и быстродействия эти языки равнозначны. Программа на ЛИСП значительно короче, ПРОЛОГ – программа понятнее. Последнее обстоятельство важно для программистов, так как размер программы сильно ограничен не памятью машины, а способностью программиста понимать, «что же он делает». Существуют несколько различных версий языка: Turbo Prolog, Visual Prolog и др. Одни версии имеют ограничения и больше подходят для начального освоения, другие обеспечивают полную профессиональную поддержку работы программиста. Рассмотрим основные сведения о языке с использованием табл. 9.1 - 9.5.

Все переменные в ПРОЛОГ-программе пишут с большой буквы: X, Misha. Если значение переменной нас не интересует, используют «_» – пустая переменная. Свободной называется переменная, значение которой еще не определено.

Таблица 9.1

| Constants | Объявление констант |
|----------------------------|-----------------------------------------------------------------------------------|
| [GLOBAL] domains | Секция объявления нестандартных и/или составных типов данных. Может отсутствовать |
| [GLOBAL] database – имя_БД | Необязательная секция объявления предикатов для работы с внутренней базой данных |
| [GLOBAL] predicates | Секция объявления предикатов |
| Clauses | Секция объявления правил и фактов |
| Goal | Секция объявления внутренней цели. Может отсутствовать |

Типы данных

Таблица 9.2

| | |
|---------|-----------------------------------------------------------|
| symbol | Строка, занесенная во внутреннюю таблицу символов системы |
| string | Последовательность символов длиной до 64 Кбайт |
| char | 1-байтовые символы |
| integer | 2-байтовые целые числа со знаком |
| real | 8-байтовые числа с плавающей точкой |
| ref | Ссылочные числа базы данных |
| reg | Регистры микропроцессора AX, BX, CX, DX, SI, DI, DS и ES |
| file | файл |

Таблица 9.3

Операции

| | |
|------------------------|--------------------------------------------------------------|
| Арифметические | +, -, *, /, mod, div |
| Реляционные | >, <, =, >=, <=, <>, >< |
| Математические функции | sin, cos, tan, arctan, ln, log, exp, sqrt, round, trunk, abs |
| Логические | And(«»,»), not, or, ! (отсечение) |

Таблица 9.4

Ввод-вывод

| | |
|-------|------------------|
| Write | Вывод на экран |
| read | Чтение с консоли |

Параметры:

%d – обычное десятичное число (chars и integers);

%u – беззнаковое целое (chars и integers);

%R – как ссылочное число базы данных (database reference numbers);

%X – как длинное 16–ричное число (strings, database reference numb);
 %x – как 16-ричное число (chars and integers);
 %s – строки (symbols and strings);
 %c – как символ (chars and integers);
 %g – действительное в наиболее коротком из возможных форматах
 (default for reals);
 %e – действительные в экспоненциальной записи;
 %f – действительные в формате с фиксированной точкой;
 %lf – только для совместимости с языком C (fixed reals);
 \n – новая строка;
 \t – табуляция;
 \nnn – символ с кодом nnn.

Таблица 9.5

Стандартные предикаты Turbo Prolog

| Стандартные предикаты охватывают | | | | |
|----------------------------------|------------------|-------------------|-----------------|-----------------|
| Ввод/вывод | Файловая система | Работа с экраном | Оконная система | Обработка строк |
| Стандартные предикаты охватывают | | | | |
| Преобразования типов | Работа с БД | Работа с графикой | Работа с ОС | Низкоуровневые |

Более подробно имеет смысл ознакомиться с документацией конкретной версии.

Пример программы (Turbo Prolog):

domains

person, activity = symbol

predicates

likes (person,activity)

clauses

- likes(ellen,tennis)
- likes (john,football)
- likes (tom,baseball)
- likes (bill,X) if likes (tom,X)

goal

likes (X,Y)

Результат работы (произошла конкретизация значений переменных по имени отношения, см. исчисление предикатов первого порядка):

- X = ellen Y = tennis
- X = john Y = football
- X = tom Y = baseball
- X = bill Y = baseball

Программа соответствует ситуации рис. 9.1, когда существуют возможные маршруты поездки: из г. Барнаула в гг. Брест, Томск и Новосибирск; из г. Томска в г. Барнаул; из г. Новосибирска в г. Томск и в г. Барнаул. Использован оператор отсечения. Цель эквивалентна вопросу: «Куда можно уехать из Барнаула?».

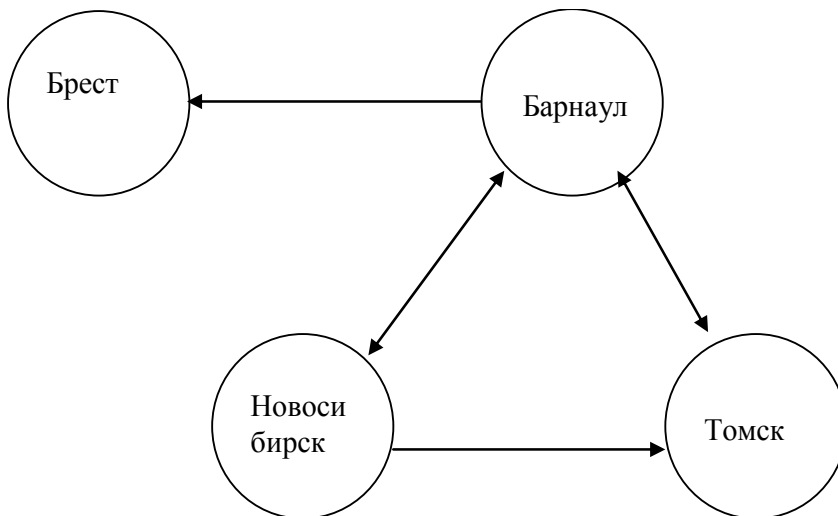


Рис. 9.1

predicates

road(string,string)

route(string,string)

clauses

road(«Barnaul»,«Brest»)

road(«Novosibirsk»,«Barnaul»)

road(«Novosibirsk»,«Tomsk»)

road(«Barnaul »,«Tomsk»)

road(«Tomsk»,«Barnaul»)

route(X,Y):- road(X,Y)

route(X,Z):- road(X,Y), route(Y,Z),!

Goal

route(«Barnaul», X)

Результат работы:

X= Brest

X =Tomsk

X= Barnaul

3 решения.

9.2. Краткие сведения о ЛИСПе

Язык создан в 1961 г. (Стэнфорд, США) группой профессора Джона Маккартни. Сокращение ЛИСП [LISP (List processing)] переводится как «язык обработки списков». В 70-80 гг. широко применялся для решения задач на основе древовидных структур, например, задач лабиринтного поиска и «генетического программирования» (Стэнфорд, проф. Джон Коз). Существует большое количество версий, наиболее известной, по-видимому, является COMMON LISP. Эта версия была поддержана AI Lab М.И.Т при создании LISP-машины в качестве языка системного программирования. К началу 90-х гг. в Европе фактически вышел из употребления, применяется в США.

Второе рождение Lisp связано с тем, что Lisp оказался чрезвычайно удобен для описания геометрических фигур. В этом качестве он широко известен пользователям разнообразных САПР - систем автоматизированного проектирования, например, AutoLisp (система AutoCAD).

Lisp-программа не имеет жесткой структуры. Представляет собой последовательность s-выражений (т. е. символьных, в заданной грамматике), поступающих последовательно на вход интерпретатора Lisp.

Таблица 9.6

Типы данных

| Целое число | Например 4 | |
|--------------------|------------------------------------------------------------------------------------|-------------------------------------|
| Вещественное число | Определяется по наличию точки – 4.5 | |
| Символьный атом | Аналог понятия переменной. Значение переменной по умолчанию равно самой переменной | Так называемые S(ymbolic)-выражения |
| Список | Обозначается (), например (A, B, C) | |
| Встроенная функция | Список приведен ниже | |
| Примитив | Только для AutoLisp | |

Некоторые основные функции

| Наименование | Краткое описание |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| QUOTE A | Функция не оценивает свой аргумент и выдает его в качестве результата |
| +, -, *, / | Арифметические функции (допустимы только над числами) |
| SETQ и SET | Функции присваивания |
| CAR A | Оценивает свой аргумент (A), который должен быть списком, и выдает в качестве значения первый элемент этого списка |
| CDR A | Оценивает свой аргумент (A), который должен быть списком, и выдает в качестве значения этот список, но без первого элемента |
| CONS A B | Оценивает аргументы, причем значение первого аргумента может быть произвольным s-выражением, а значением второго должен быть список. Результатом функции является список с головой A и хвостом B |
| APPEND A B | Пусть значениями A и B являются списки: (A...A) и (B...B), тогда значением функции APPEND будет список (A...AB...B) |
| NULL X | Оценивает аргумент и выдает T, если значением X является NIL, и NIL в противном случае. Функция является аналогом NOT |
| ATOM X | Оценивает аргумент и выдает T, если его значением является атом, и NIL в противном случае |
| NUMBERP X | Оценивает свой аргумент и выдает значение T, если значением аргумента является число, и NIL в противном случае |
| LISTP X | Оценивает свой аргумент и выдает T, если значением аргумента является список, и NIL в противном случае. Является обратной к функции ATOM |
| EQ A B | Функция сравнения двух атомов |
| EQUAL A B | Сравнение произвольных выражений |
| COND A A | Функция проверки условия |
| DEFUN | Функция, определяющая новые функции |

Для некоторых функций, например CAR и CDR, существует сокращенный вариант: C-R, где вместо символа - можно подставить символы D или A. Примеры:

$(CAAAR X) \Leftrightarrow (CAR(CAR(CAR X)))$ или $(CADR X) \Leftrightarrow (CAR(CDR X))$

Пример программы (μ -LISP).

Вычисление факториала:

`(defun factorial (n)`

`(cond ((= n 1) 1)`

`(t (* n (factorial (- n 1))))))`

9.3. Язык Рефал

Название языка происходит от «Рекурсивных Функции Алгоритмический язык». Его построение способствовало возникновению и развитию продукционного подхода в программировании.

Язык РЕФАЛ является сентенциальным в своей основе, а вся информация в этом языке представляется в виде правил конкретизации. Каждое правило записывается в виде предложения, которое представляет собой продукцию с определенными синтаксисом и семантикой. Предложения в Рефал-программе отделяются друг от друга знаком § (параграф).

Каждое правило конкретизации определяет раскрытие смысла некоторого понятия через более элементарные. Операцию конкретизации можно также определить как переход от имени к значению. Введем два знака:

k и \perp , которые будем называть конкретизационными скобками и которые будут содержать объект, подлежащий конкретизации. Так, если x — некоторая переменная, то $kx\perp$ (конкретизация x) будет изображать значение этой величины. Другой пример: объект $k28 + 7\perp$ при правильном определении операции сложения рано или поздно будет заменен на объект 35.

Выполнение конкретизации — переход от имени к значению — объявляется основной и, по существу, единственной операцией в языке Рефал.

Эту операцию будет выполнять Рефал-машина (имеется в виду машина на логическом уровне, имитируемая соответствующим транслятором на универсальной ЭВМ; возможно, разумеется, и построение реальной «физической» Рефал-машины).

Поскольку правило конкретизации есть указание для замены одного объекта (слова в некотором алфавите) на другой, предложения языка Рефал должны состоять из левой части (заменяемый объект) и правой части (объект, заменяющий левую часть). Для разделения правой и левой части мы будем использовать знак стрелки « \rightarrow ».

Пример 9.1. Предложение, выражающее тот факт, что значение переменной X есть 137, записывается в виде

$$\S kX\perp \rightarrow 137.$$

Между знаком \S и первым знаком k можно вставлять последовательность знаков, которая будет служить номером предложения, или комментарием к нему, например:

$$\S 1.1 kX\perp \rightarrow 137 \quad (\text{ф. 27}).$$

Опишем теперь структуру Рефал-машины, которая, используя предложения Рефал-программы, будет выполнять конкретизации. Будем считать, что объектом обработки является некоторое выражение (слово), которое находится в поле зрения машины. Работа машины осуществляется по шагам, каждый из которых представляет выполнение одного акта, конкретизации.

Пусть программа машины состоит из единственного предложения (ф. 27), а в поле зрения находится выражение $kX\perp$. Тогда за один шаг машина заменит содержимое поле зрения на 137, после чего она остановится, т.к. знаков конкретизации больше нет и, следовательно, делать ей больше нечего.

Так как Рефал-программа содержит, вообще говоря, набор (последовательность) предложений, может оказаться, что для выполнения данной

конкретизации пригодно не одно, а несколько предложений. Например, в поле памяти, кроме (ф. 27), может находиться еще предложение

$$\S 1.2 kX\perp \rightarrow 274.$$

Неоднозначность, которая отсюда может возникнуть, устраняется так же, как это принято в нормальных алгоритмах Маркова (Рефал-машина следует идеологии этих алгоритмов): машина просматривает предложения в том порядке, в котором они расположены в Рефал-программе, и применяет первое из них, которое окажется подходящим.

Поле зрения может содержать сколько угодно конкретизационных скобок, причем они могут быть как угодно вложены друг в друга. В этом случае Рефал-машина начинает процесс конкретизации с первого из знаков k , в области действия которого (т.е. в последовательности знаков до парной скобки \perp) нет ни одного знака k . Выражение, находящееся в области этого знака k , последовательно сравнивается с левыми частями предложений Рефал-программы. Найдя подходящее предложение, машина выполняет в поле зрения необходимую замену и переходит к следующему шагу конкретизации.

Пример. Пусть Рефал-программа имеет вид

$$kX\perp \rightarrow 137$$

$$kX\perp \rightarrow 274$$

$$kY\perp \rightarrow 2$$

$$k137+2\perp \rightarrow 139,$$

а поле зрения содержит выражение

$$kkX\perp +kY\perp \perp .$$

На первом шаге замене подлежит подвыражение $kX\perp$ — получим в поле зрения $k137 + kY\perp \perp$. Теперь в первую очередь конкретизируется $kY\perp$ — получим в результате применения третьего предложения $k137 + 2\perp$ и на последнем шаге получим 139, не содержащее символов k . (Разумеется

для реального сложения используются соответствующие встроенные функции, а этот пример — лишь простейшая иллюстрация принципов работы машины [21]).

Чтобы иметь возможность представлять обобщенные предложения, используются три типа переменных: e — для представления выражений; t — для термов; s — для символов. В простейшем случае переменные записываются в виде указателя типа (e, t, s) и индекса; например, e_1, e_2 — переменные, пробегающие в качестве значений выражения. Выражением в языке Рефал называется последовательность знаков, правильно построенная в соответствии с синтаксисом языка Рефал. Терм языка Рефал — это либо символ, либо выражение в круглых или конкретизационных скобках. Выражения строятся из термов.

Пример. Предположим, требуется написать программу, которая выполняет раскрытие скобок в алгебраических выражениях, построенных из букв с помощью скобок, знаков сложения «+» и умножения «*»". Рассмотрим процесс написания такой программы. Если некоторое выражение e имеет вид $e_1 + e_2$, где e_1, e_2 — выражения, то для раскрытия скобок надо: раскрыть скобки в e_1 , раскрыть скобки в e_2 , полученные результаты сложить. Эту мысль в компактном, но в то же время и наглядном виде выражает предложение:

$$\S 2.1 \ ke_1 + e_2 \perp \rightarrow ke_1 \perp + ke_2 \perp$$

Если же выражение e имеет вид $e_1 * e_2$, то, вообще говоря, необходимо учитывать две возможности:

— хотя бы один из сомножителей есть сумма (например, $e = (A + B) * C$),

— ни одно из выражений e_1 или e_2 не представимо в виде суммы (например, $e = (A * B) * (C * D)$).

В первом случае надо описать законы дистрибутивности:

$$\S 2.2 ke_1 * (e_2 + e_3) \perp \rightarrow ke_1 * e_2 \perp + ke_1 * e_3 \perp,$$

$$\S 2.3 k(e_1 + e_2) * e_3 \perp \rightarrow ke_1 * e_3 \perp + ke_2 * e_3 \perp,$$

$$\S 2.4 ke_1 * (e_2 + e_3) * e_4 \perp \rightarrow k(e_1 * e_2 + e_1 * e_3) * e_4 \perp.$$

Во втором случае по аналогии со сложением имеем

$$\S 2.5 ke_1 * e_2 \perp \rightarrow ke_1 \perp * ke_2 \perp.$$

Наконец, осталось выразить возможность «снятия внешних скобок» и условие «терминальности» символов, что определяют предложения:

$$\S 2.6 k(e) \perp \rightarrow ke \perp,$$

$$\S 2.7 ks \perp \rightarrow s$$

(буквы не подлежат конкретизации).

Приведенные семь предложений $\S 2.1 - \S 2.7$ решают задачу. Рассмотрим как эта программа обрабатывает выражение

$$k(A + B) * (C + D) \perp.$$

Последовательно получим в результате работы программы (для удобства слева указываем номер правила, которое непосредственно привело к данному выражению):

$$\S 2.2 k(A + B) * C \perp + k(A + B) * D \perp,$$

$$\S 2.3 kA * C \perp + kB * C \perp + k(A + B) * D \perp,$$

$$\S 2.3 kA * C \perp + kB * C \perp + kA * D \perp + kB * D \perp.$$

Далее ограничимся рассмотрением первого слагаемого:

$$\S 2.5 kA \perp * kC \perp + \dots,$$

$$\S 2.7 A * kC \perp + \dots,$$

$$\S 2.7 A * C + \dots.$$

После аналогичной обработки остальных слагаемых получим иско-
мое выражение

$$A * C + D * C + A * D + B * D.$$

Если на вход поступит выражение

$$kA + (B + C) \perp ,$$

то получим последовательно:

$$\S 2.1 kA \perp + k(B + C) \perp ,$$

$$\S 2.7 A + k(D + C) \perp ,$$

$$\S 2.6 A + kB + C \perp ,$$

$$\S 2.1, 2.7 A + B + C.$$

Тема 10. ОСНОВЫ ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ

В настоящее время основные научные, методические и практические вопросы построения экспертных систем достаточно хорошо проработаны, имеется огромное количество литературы и «рецептов», как построить ту или иную экспертную систему, используя специализированные языки, стандартные (пустые) оболочки ЭС и др. средства (см. например, [14-17, 26, 29, 32, 34, 37, 38, 50]). В связи с этим рассмотрим конспективно только основные этапы и особенности создания традиционных ЭС, используя материалы работ [14-17].

Итак, первые экспертные системы появились в конце 60-х гг. прошлого века и предназначались для создания искусственного разума в некоторой предметной области. Тогда, в 60-70-х гг. прошлого века большинство ЭС мыслились как средство, позволяющее аккумулировать опыт лучших экспертов в некоторой предметной области, а затем консультировать менее опытных пользователей. Классические базы данных (БД) не подходили для этого, так как их язык запросов не являлся достаточно гибким для описания реальных запросов к БД, и они не допускали рекурсивности.

В современном понимании экспертная система – это узкоспециализированный программный комплекс, позволяющий либо очень быстро принимать стандартные решения (чаще всего по управлению техническими объектами), либо на основе длительного диалога с пользователем помогать в выборе некоторого решения. При этом ЭС оценивает варианты, предлагаемые пользователем, производит их коррекцию с учетом множества факторов, возможно неизвестных пользователю, а также предлагает собственные варианты решения и их коррекцию в соответствии с целями пользователя.

Основные особенности работы ЭС

Основная разница в работе экспертных и «не экспертных» систем состоит в том, что первые в каждой своей итерации используют новый алгоритм действий в предметной области, отличающийся от алгоритма предыдущего шага не только параметрами, но и структурой.

Главным структурным отличием ЭС от всех других типов программ является наличие базы знаний и, как следствие, способность к обучению и самообучению. Ее конкретный вид сильно зависит от избранной модели представления знаний. Для системы правил продукций можно говорить о фактах и правилах (логического, чаще всего дедуктивного вывода). Для сетевой модели БЗ в общем случае содержит избранную активную подсеть и правила вывода (включающие в себя и описание отношений, присутствующих в сети).

Машина вывода – это программа, ответственная за порядок и способ актуализации алгоритмической части БЗ. Для разных моделей представления знаний она, естественно, различна.

Для логической модели, где каждый раз происходит полный перебор всех логических уравнений вне зависимости от уже имеющихся данных, она имеет простейший вид последовательного автомата.

Для системы продукций (машина логического вывода), где выбор правила, которое следует актуализовать следующим, неоднозначен, она имеет систему предпочтений, основанную на некоторой детерминированной или вероятностной оценке, зависящей от ситуации сложившейся к этому моменту.

Для фреймовой модели машина вывода состоит как бы из двух частей. Единая общая часть отвечает за то, чтобы события, происходящие в системе фреймов, становились известны всем фреймам одновременно или в некоторой последовательности. Вторая часть – это обработчики событий

самих фреймов, которые решают, какой из методов следует актуализовать в ответ на полученное сообщение о произошедшем событии.

Подсистема объяснений – это один из важнейших элементов ЭС, который в значительной мере определяет эффективность ее практического использования. Практическая реализация подсистемы объяснений часто использует те же процедуры, что и решатель, только в другом порядке. Поэтому часто можно считать, что машина вывода состоит из решателя и подсистемы объяснений. Специфика задач, решаемых интеллектуальными системами, такова, что очень часто ценность имеет не само решение, полученное человеком или машиной, а процесс – ход решения, полученного человеком в диалоге с машиной.

Не все интеллектуальные системы обладают такой способностью (имеют подсистему объяснений) и существует она только у систем, основанных на правилах продукций и семантических сетях.

Интеллектуальный редактор БЗ – это программа, позволяющая добавлять, удалять, модифицировать факты и правила, содержащиеся в БЗ. Определение «интеллектуальный» применительно к редактору БЗ понимается как способность обеспечить целостность, корректность и причудливые и часто трудно понимаемые человеком смысловые цепочки.

Интеллектуальный интерфейс пользователя – это программа, ответственная за организацию такого диалога с пользователем, который оптимальным образом приводит к достижению цели общения человека с ЭС. «Интеллектуальность» интерфейса заключается в том, что в ходе диалога система помогает пользователю уточнить или совсем переформулировать цель его консультации с экспертной системой.

Краткая классификация ЭС

Экспертные системы можно классифицировать по различным основаниям: по типу используемой модели представления знаний, по области применения, по назначению, по показателям эффективности и др.

Глубинной называется модель, которая позволяет обоснованно доказывать свои выводы. К глубинным знаниям относят абстракции, образы, аналогии, в которых отражается понимание структуры предметной области, назначение и взаимосвязь отдельных понятий, известные законы и теоретические основания.

Поверхностные знания касаются, в основном, внешних проявлений объекта исследования. Они позволяют делать некоторые (часто очень вероятные) предположения. В большинстве ЭС сейчас используются поверхностные знания. Введение глубинных представлений позволяет создавать БЗ большей мощности, так как глубинные знания более гибки и адаптивны, чем достаточно жесткие поверхностные.

ЭС можно классифицировать по решаемой задаче. Так ЭС диагностики были разработаны одними из первых. Совместно с ЭС мониторинга они решают задачи непрерывной диагностики технических объектов, например ракет в момент запуска и в процессе полета. Более общо можно сказать, что такие ЭС выполняют процесс отнесения объекта к некоторому классу и обнаружения места и вида неисправности. Неисправность – это отклонение от нормы. Такая трактовка позволяет с единых теоретических позиций рассматривать и неисправность оборудования в технических системах, и заболевания живых организмов, и всевозможные природные аномалии.

Особенность ЭС мониторинга состоит в том, что часто необходимо учитывать очень большое число параметров, непрерывно поступающих от сотен и тысяч датчиков в реальном времени, и нужно вовремя подать сигнал тревоги.

ЭС планирования и проектирования имеют много общего с системами визуализации и интерпретации.

ЭС прогнозирования логически выводят вероятные следствия из заданных ситуаций. В прогнозирующей системе обычно используется пара-

метрическая динамическая модель, в которой значения параметров «подгоняются» под данную ситуацию.

ЭС обучения диагностируют ошибки при изучении какой-либо дисциплины с помощью ЭВМ и подсказывают правильные решения. Они аккумулируют знания о гипотетическом «ученике» и его характерных ошибках, затем в работе они способны диагностировать слабости в познаниях обучаемых и находить соответствующие средства для их ликвидации.

В общем случае все системы, основанные на знаниях, можно разделить на системы, решающие задачи анализа, и на системы, решающие задачи синтеза. Основное отличие задач анализа от задач синтеза заключается в том, что если в задачах анализа множество решений может быть перечислено и включено в систему, то в задачах синтеза множество решений потенциально и строится из решений компонент или подпроблем. Задачами анализа являются: интерпретация данных, диагностика; задачами синтеза: проектирование, планирование. Комбинированные задачи: обучение, мониторинг, прогнозирование.

Автономные ЭС работают непосредственно в режиме консультаций с пользователем только для «экспертных» задач, при решении которых не требуется привлекать традиционные методы обработки данных (расчеты, моделирование и т.д.)

Гибридные ЭС представляют программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирование или системы управления базами данных) и средства манипулирования знаниями. Это может быть интеллектуальная надстройка над пакетом прикладных программ или интегрированная среда для решения сложной задачи с элементами экспертных знаний. Несмотря на внешнюю привлекательность гибридного подхода, следует отметить, что разработка таких систем являет собой задачу на по-

рядок более сложную, чем разработка автономной ЭС. Стыковка не просто разных пакетов, а разных методологий

Особенности назначения и применения ЭС

По задачам – диагностика, интерпретация, проектирование, мониторинг, обучение;

По типу обработки информации в реальном времени: статические, динамические, квазидинамические;

По используемым методам: автономные, гибридные интегрированные с большими информационными системами.

Статические ЭС разрабатываются в предметных областях, в которых база знаний и интерпретируемые данные не меняются за время решения задачи. Они стабильны. Требования функционирования в реальном времени отсутствуют. Например, диагностика неисправностей в автомобиле.

Квазидинамические ЭС интерпретируют ситуацию, которая меняется в течение некоторого фиксированного интервала времени. Например, микробиологические ЭС. Лабораторные измерения снимаются во время технологического процесса один раз в 4-5 часов (производство лизина, например) и анализируется динамика полученных показателей по отношению к предыдущему измерению, требования к работе в реальном времени не жесткие.

Динамические ЭС работают с изменяющимися во время решения задачи данными, часто в сопряжении с датчиками объектов, иногда в режиме реального времени с непрерывной интерпретацией поступающих данных. Жесткие требования к работе в реальном времени. Пример: гибкие производственные системы, мониторинг в реанимационных палатах, управление подвижными объектами, роботами и др.

Основные этапы разработки ЭС

Технология создания традиционных типов ПО ЭС предполагает следующие этапы: анализ требований, проектирование, программирование и

отладка, тестирование. Технология создания интеллектуальных систем имеет ряд этапов и особенностей их реализации:

1. Анализ требований предполагает выявление основных сущностей, с которыми должна оперировать ЭС, а также возможных типов взаимосвязей между этими сущностями. На этом этапе оговариваются типичные запросы, на которые должна уметь отвечать система, а также границы применимости этих запросов.

2. Анализ предметной области предполагает рассмотрение всех возможных сущностей и отношений между ними, ограничение самой предметной области и математическое описание возможных и допустимых типов сущностей и отношений. Например, для БД паспортного стола возможными сущностями будут: «мужчина», «женщина», «ДОМ», «квартира», «улица». Отношениями: «муж», «жена», «дочь», «сын», «брат», «зять», «сводный брат» и т.д. Степень детализации как раз и определит возможности ЭС.

3. Проектирование ЭС предполагает создание структуры ЭС. Существенной особенностью проектирования ЭС является зависимость структуры от избранной модели представления знаний. Так, например, выбор в качестве базовой продукционной модели автоматически означает использование редактора правил; выбор фреймовой модели – редактора объектов; выбор вероятностной модели – редактора вероятностей.

Таким образом, механизм работы ЭС можно описать следующими шагами:

1. инженер по знаниям наполнил БЗ (обучил ЭС);
2. пользователь задал системе запрос;
3. интеллектуальный интерфейс уточнил запрос и сформулировал цель для машины вывода;
4. машина вывода попыталась достичь заданную цель на основе знаний, полученных из БЗ;

5. цель не была достигнута из-за недостатка данных;
6. интеллектуальный интерфейс технических систем связан с соответствующими источниками данных;
7. машина вывода вновь попыталась достичь цель;
8. цель не была достигнута из-за получения противоречивых выводов;
9. для разрешения возникших противоречий запустился механизм самообучения, который попытался разрешить возникшие противоречия на основе заложенных в него правил;
10. новые знания, полученные в ходе работы механизма самообучения, поступили в редактор БЗ;
11. инженер по знаниям принял или отверг новые знания и факты;
12. новые знания и факты занесены в БД и БЗ;
13. возврат к п. 4.

Главное отличие ЭС от других программных средств – это наличие базы знаний, в которой знания хранятся в виде совокупности записей на некотором языке представления знаний (ЯПЗ), который позволяет легко изменять и дополнять базу знаний в форме, понятной специалистам – разработчикам экспертной системы. До последнего времени именно различные ЯПЗ были центральной проблемой при разработке ЭС. Сейчас существуют десятки языков или моделей представления знаний. Наибольшее распространение получили следующие модели: продукции, семантические сети, фреймы, исчисление предикатов 1-го порядка, объектно-ориентированные языки и др. Для этих моделей существует соответствующая математическая нотация, разработаны системы программирования, реализующие эти ЯПЗ, и имеется большое число реальных коммерческих ЭС.

Выбор той или иной модели определяется структурой знаний в конкретной предметной области. Выявление структуры предшествует выбору

ЯПЗ и разработке базы знаний. Обоснование и конкретизация элементов знаний и их взаимосвязей происходят чаще всего в непосредственном контакте со специалистами предметной области – экспертами. Этот процесс называется извлечением знаний. Разработчики экспертных систем, занимающиеся извлечением и структурированием знаний, называются инженерами по знаниям.

Некоторые примеры крупномасштабных экспертных систем

Примеры традиционной разработки классических ЭС.

MYCIN — экспертная система для медицинской диагностики. Разработана группой по инфекционным заболеваниям Стенфордского университета. Ставит соответствующий диагноз, исходя из представленных ей симптомов, и рекомендует курс медикаментозного лечения любой из диагностированных инфекций. ЭС основана на правилах. ЭС формирует, проверяет и отбрасывает гипотезы. База данных состоит из 450 правил.

PUFF — анализ нарушения дыхания. Данная система представляет собой ЭС MYCIN, из которой удалили данные по инфекциям и вставили данные о легочных заболеваниях.

DENDRAL — распознавание химических структур. Данная система старейшая, из имеющих звание экспертных. Первые версии данной системы появились еще в 1965 году в Стенфордском университете. Пользователь дает системе DENDRAL некоторую информацию о веществе, а также данные спектроскопии (инфракрасной, ядерного магнитного резонанса и масс-спектрометрии), и та, в свою очередь, выдает диагноз в виде соответствующей химической структуры. ЭС основана на правилах. ЭС формирует, проверяет и отбрасывает гипотезы. Современный вариант называется GENOA.

PROSPECTOR — экспертная система, созданная для содействия поиску коммерчески оправданных месторождений полезных ископаемых.

Примеры ЭС, созданных с использованием современных технологий

В списке наиболее распространенных в настоящее время за рубежом экспертных систем и их оболочек можно выделить следующие наименования: INSIGT, LOGIAN, NEXPERT, RULE MASTER, KDS, PICON, KNOWLEDGE CRAFT, KESII, S1, TIMM и др. (описания в [21]).

Основные критерии для суждения о возможности создания экспертной системы

1. Действительная необходимость в символьных рассуждениях. (Нет смысла разрабатывать экспертную систему для численных расчетов, например, для преобразований Фурье, интегрирования, решения систем алгебраических уравнений и др.).

2. Наличие экспертов, компетентных в избранном круге вопросов, которые согласны сотрудничать при создании ЭС.

Поставленная проблема должна быть достаточно важной и актуальной. Это могут быть проблемы, требующие высокого уровня экспертизы, либо простые, но трудоемкие многократно повторяющиеся проверки. Нет смысла тратить время на решение проблем, которые возникают редко и могут быть разрешены человеком с обычной квалификацией.

3. Необходимо четко ограничивать круг решаемых задач, т.е. предметная область выбирается достаточно «узкой», чтобы избежать «комбинаторного взрыва» объема информации необходимой для компетентного решения поставленной задачи.

4. Необходима согласованность мнений экспертов о том, как следует решать поставленные задачи, какие факты необходимо использовать и каковы общие правила вынесения суждений. В противном случае невозможно расширить базу знаний за пределы опыта одного человека и осуществить сплав экспертных знаний из нескольких областей.

5. Должно быть достаточно исходных данных для проверки работоспособности экспертной системы в выбранной предметной области, чтобы разработчики смогли убедиться в достижимости некоторого заданного уровня ее функционирования.

6. Должна обеспечиваться возможность постепенного наращивания системы. База знаний должна легко расширяться и корректироваться, так как правила часто меняются с появлением новых фактов.

Ценность использования ЭС проявляется в следующих аспектах:

1. В сборе, оперативном уточнении, кодировании и распространении экспертных знаний.

2. В эффективном решении проблем, сложность которых превышает человеческие возможности и для которых требуются экспертные знания нескольких областей.

3. В сохранении наиболее уязвимой ценности коллектива – коллективной памяти.

4. Создание баз знаний открывает широкие возможности, которые обусловлены безошибочностью и тщательностью, присущими ЭВМ и синтезом знаний экспертов. Если база знаний объединяет информацию по нескольким дисциплинам, то такой «сплав» знаний приобретает дополнительную ценность.

5. Экспертная система позволяет решить проблему сохранения экспертных знаний, связанную с утратой наиболее квалифицированных экспертов в результате их продвижения по службе, смерти, перехода на другую работу или выхода на пенсию, а также позволит сделать знания легко доступными для тех, кто займет места ушедших экспертов.

Тема 11. РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Рассматривая возможные области использования ЭС при решении задач ИИ, отмечалась целесообразность создания и применения динамических ЭС реального времени (ЭС РВ) для ответственных применений в промышленности, в медицине и при управлении подвижными объектами. Однако, по сравнению с требованиями к работе статических и квазидинамических ЭС, при создании ЭС РВ необходимо обеспечить существенно более высокие требования по ресурсным ограничениям и предусмотреть большие функциональные возможности. Выделим следующие основные проблемы, возникающие на этом пути:

1. Создание методов и программного обеспечения для существенного ускорения работы механизмов и программ логического вывода при обеспечении функционирования ЭС РВ в жестком реальном времени.

2. Применение современных программно-технических средств для создания ЭС РВ.

3. Разработка новых структурных решений при построении ЭС РВ, обеспечивающих требуемые высокие динамические характеристики работы средств интеллектуальной поддержки принятия решений при управлении объектами в жестком реальном времени и постоянном изменении текущей информации о состоянии объекта и окружающей обстановки.

11.1. Особенности использования ЭС РВ в системах ИИ

Для решения проблем создания новых эффективных инструментальных средств построения систем ИИ, необходимо использовать последние теоретические и удачные практические разработки в этой области. Применение этих разработок должно обеспечить:

1. Существенное ускорение процессов поиска вывода в системах АДТ при использовании вместо традиционных универсальных методов

выводимости (на основе метода резолюций) более продуктивного подхода, предложенного в [7] и основанного на конструктивном формировании и принятии дополнительных предположений.

2. Повышение эффективности использования современных инструментальных программных средств для создания ЭС РВ, в том числе на базе применения программного продукта G2 фирмы Gensym Corporatin. В настоящее время сравнительный анализ функциональных и ресурсных возможностей системы G2 с другими оболочками экспертных систем показывает существенное преимущество G2. Ознакомление с возможностями этой системы и получение начальных практических навыков ее использования предусмотрено программой проведения практических занятий по курсу «Интеллектуальные системы и технологии».

Коротко о системе: G2 – это объектно-ориентированная интегрированная среда для разработки и сопровождения приложений реального времени, использующих базы знаний.

В G2. системе реализованы следующие возможности объектно-ориентированной технологии, включая:

1) Поддержку связи между объектами, отношения между объектами, иерархию объектов.

2) Представление знаний, включая: правила (общие и конкретные); процедуры; динамические модели.

3) Механизм рассуждений: от данных; от цели; сканирование; метарассуждения (события, фокусирование на классах объектов или правил); одновременное выполнение правил и/или процедур.

4) Графическое определение объектов.

5) Клонирование объектов и их групп.

6) Графические пользовательские интерфейсы для различных категорий пользователей.

7) Многопользовательская кооперативная разработка приложения.

8) Распределенное приложение.

В отличие от систем, ориентированных на какую-то одну методологию или на конкретную предметную область, G2 интегрирует в себе множество взаимодополняющих методов ИИ, что упрощает и ускоряет процесс разработки приложений и позволяет делать их универсальными.

Программные продукты, разработанные с помощью системы G2, являются совершенно независимыми от того аппаратного обеспечения, на котором они работают, то есть они полностью переносимы. В настоящее время существуют версии G2 практически для всех типов рабочих станций и для персональных ЭВМ, а также существует возможность интеграции G2 с большим числом как программных и аппаратных средств самых разных фирм.

Базовый продукт фирмы Gensym – пятая версия системы G2 – это мощная объектно-ориентированная, графическая среда для разработки и сопровождения приложений с поддержкой всех основных принципов объектно-ориентированного подхода.

Опыт использования G2 в различных прикладных областях показывает, что затраты на разработку по сравнению с традиционными методами (например с использованием языка С или С++) существенно сокращаются. Система регистрации версий выводит G2 на уровень современных CASE-средств. Изменение описаний классов и отношений во время исполнения позволяет не только экспериментировать со структурами данных непосредственно в процессе отладки, но и открывает возможности использования генетических алгоритмов, самомодифицирующихся и обучающихся систем.

Возможности G2 в части поддержки распределенных приложений на основе архитектуры клиент/сервер и легкая интеграция с разнородными источниками информации позволяют использовать ее в качестве связующего звена в гетерогенных распределенных вычислительных средах, объединяющих как технические средства (контроллеры ведущих фирм и каналы связи), так и развитые СУБД (ORACLE, Sybes, INFORMIX, все ODBC

совместимые СУБД). Передача объектов и массивов в качестве аргументов упрощает совместное использование данных как независимыми приложениями на базе G2, так и внешними по отношению к G2 программными системами. Безопасность и конфиденциальность распределенной обработки достигаются за счет системы уровней автоматической проверки прав доступа при установлении сетевого взаимодействия процессов через независимый монитор транзакций – G2 Standard Interface (GSI).

Программные продукты фирмы Gensym работают под управлением различных вариантов операционных систем UNIX и VMS (большинство современных рабочих станций фирм SUN, IBM, DEC, HP используют именно эти операционные системы), четвертая версия G2 может работать под управлением MS Windows NT и MS Windows 95. Последнее обстоятельство открывает возможность переноса приложений на базе G2 на персональные ЭВМ с i486, Pentium и DEC Alpha.

Открытость системы G2 и продуктов на ее основе обеспечивается ориентацией фирмы Gensym на промышленные стандарты. Являясь членом OMG, фирма Gensym сотрудничает в этой области со многими независимыми организациями и комитетами по стандартам. В части технических средств – это поддержка широкого спектра платформ DEC, HP, Sun, IBM, SG и ПК на базе процессоров x86 и Pentium. Развитый графический интерфейс, включающий элементы анимации, базируется на средствах Motif и MS Windows. В G2 поддержан стандарт ISO 8859-5 в части представления символов кириллицы, независимо от операционной среды. Эта особенность открывает российским разработчикам возможность использования русских имен в названиях классов, атрибутов и т.д., то есть полностью русскоязычного интерфейса.

Поддерживаемые сетевые протоколы: TCP/IP и DECnet. Архитектура клиент/сервер на уровне обмена данными поддерживается монитором транзакций GSI и DDE, на уровне объектов – CORBA, на уровне приложения – клиентской подсистемой Telewindows. Распределенная обработка

обеспечивается интерфейсами G2–G2, G2–Telewindows и поддержкой вызова удаленных процедур.

Возможность простого манипулирования графическим представлением объектов в G2 и составления схем, являющихся отображением технологических цепочек или абстрактных алгоритмов обработки данных, обеспечивает базовые средства для построения проблемно-ориентированных языков визуального программирования. В данном случае объекты приобретают свойства операторов и в совокупности с различными классами связей формируют грамматику нового языка. Основным преимуществом такого подхода является то, что сформированная диаграмма потоков информации, по сути, и есть исполняемая программа, промежуточные фазы генерации кода и компиляции для ее использования не требуются.

11.2. Применение ЭС РВ в интеллектуальных системах

Рассмотрим возможности использования ЭС РВ для повышения эффективности работы интеллектуальных систем управления. Рассмотрение проведем на примере разработки интеллектуальных систем поддержки принятия решений (ИСППР) для интегрированных систем управления сложными промышленными и морскими подвижными объектами при повышенных требованиях к управлению в жестком реальном времени [48, 49].

Для обеспечения заданных требований по управлению в реальном времени и с обеспечением интеллектуальной поддержки задач управления в ускоренном времени в этих разработках была реализована идея построения распределенной сети из небольших специализированных ЭС РВ, управляемых общим монитором экспертных систем. Возможности повышения эффективности работы такой системы существенно выше, чем при использовании одной мощной ЭС РВ при решении всех задач системы.

На рис. 11.1 представлена общая блок-схема интеллектуальной поддержки работы интегрированного комплекса управления производством химико-технологического типа (ИСППР ПХТТ). Система по сети получает текущую информацию от автоматизированных систем управления технологическими процессами (АСУТП) и от автоматизированной системы оперативно-диспетчерского управления (АСОДУ), обеспечивая поддержку принятия решений для лиц, принимающих решения (ЛПР). К ЛПР относятся операторы крупных установок и производств, диспетчер комбината и руководители производства. ИСППР реализует поддержку принятия решений, разделяя решение общей задачи на 3 уровня: 1- уровень технологической поддержки (ЭС РВ «Технолог»); 2-аналитической поддержки ((ЭС РВ «Аналитик»); 3- уровень программной поддержки работы с пакетами прикладных программ (ППП) (ЭС РВ «Программист»). На каждом уровне используются свои базы данных (БД1, БД2, БД3) и базы знаний (БЗ1,БЗ2,БЗ3).

На рис. 11.2 изображена блок-схема интеллектуальной поддержки работы интегрированного комплекса управления и навигации морским подвижным объектом (ИСППР), на которой показано взаимодействие 6-ти ЭС РВ (ЭС1-ЭС6), управляемых монитором экспертных систем. Преимуществом такой распределенной ЭС РВ является большое быстродействие, малое число правил в каждой ЭС РВ, специализированной на поддержку решения своей группы задач. Каждая ЭС РВ имеет свою базу данных БД1 – БД6. В то же время эти ЭС РВ и БД взаимодействуют между собой и пользователями ИСППР через общий сервер и анализатор запросов-ответов. Пользователи и задачи могут взаимодействовать и через общую «витрину» запросов-ответов.

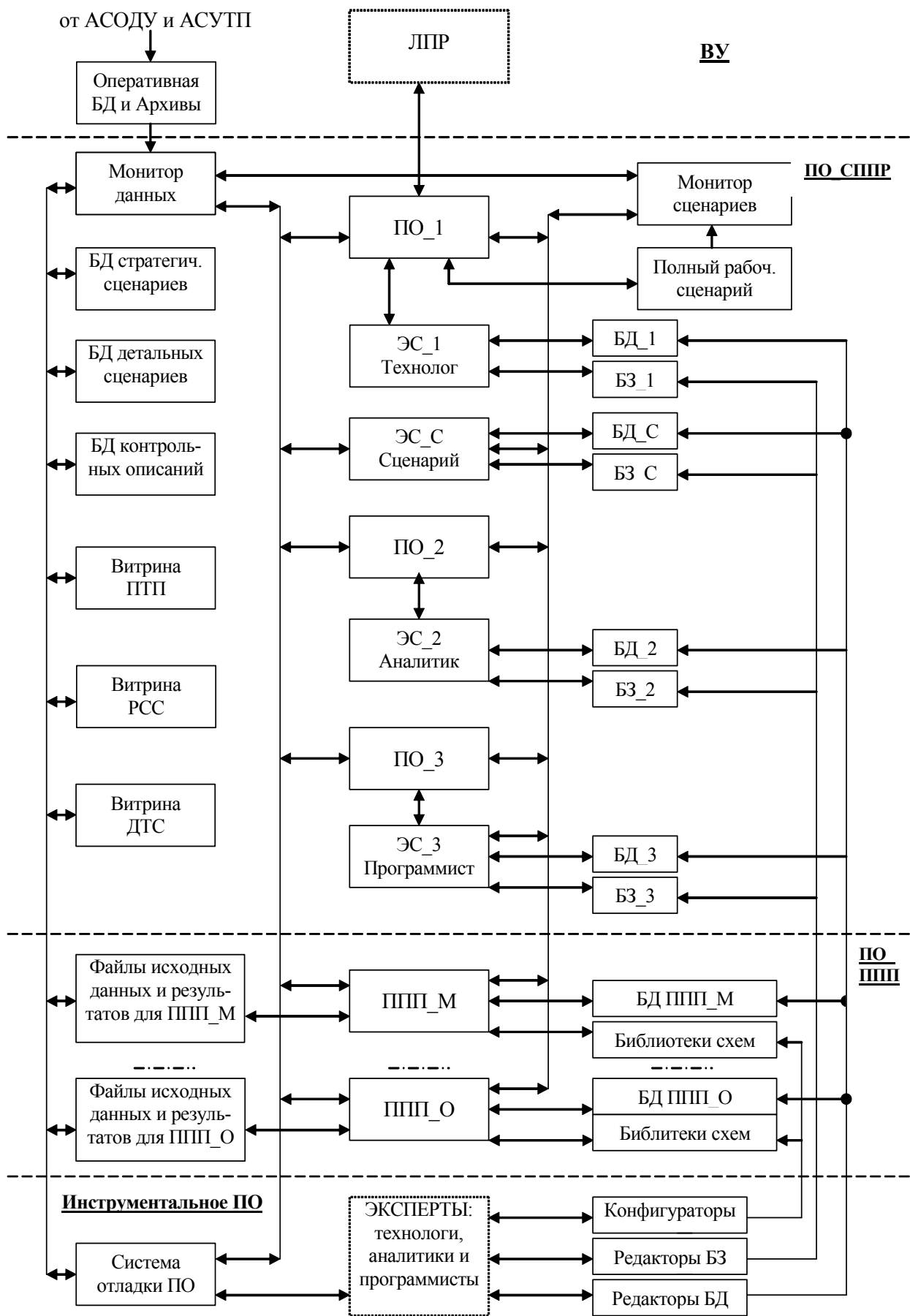


Рис. 11.1.

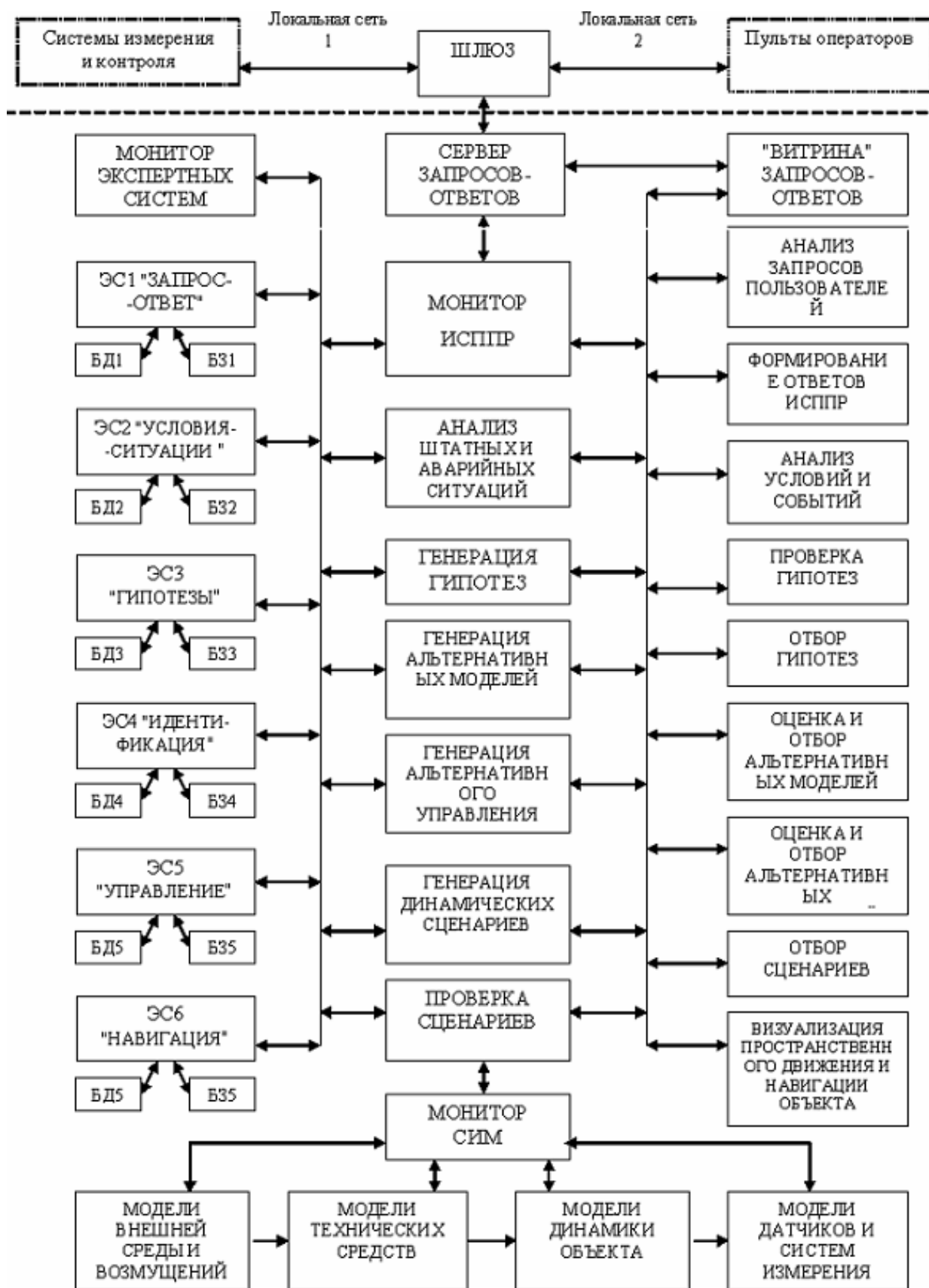


Рис.11.2. Блок-схема ИСППР

Тема 12. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ИС И ТЕХНОЛОГИЙ

Для рассмотрения перспективных направлений развития интеллектуальных систем (ИС) и систем интеллектуального управления (СИУ) отметим, что традиционные системы управления работают в условиях:

- недостаточности априорной информации о внешней среде функционирования;
- большого количества трудно учитываемых факторов нестационарности и субъективного их характера;
- изменяемости целей и критериев качества управления вследствие деградации (отказов, аварий) или целенаправленной реконфигурации (восстанавливающего или развивающего управления).

Для таких задач было бы естественно попытаться использовать, создать и развить наиболее перспективные направления развития интеллектуальных систем и интеллектуальных компонентов управления.

Рассмотрим вкратце и прокомментируем пять принципов организации СИУ (предложенные Саридиусом и дополненные с некоторыми изменениями в формулировках в работах Васильева С.Н.):

Первый принцип (принцип информационного обмена). Наличие тесного информационного взаимодействия СИУ с реальным внешним миром с использованием специально организованных информационных каналов.

Комментарий: если в традиционных системах автоматического управления внешние, априори не запланированные, неизвестные, воздействия являются скорее негативным фактором, именуемым по этой причине «возмущением», то в СИУ внешние воздействия могут иметь также характер информационной «подпитки», позволяющей организовать антиэнтропийный процесс самоорганизации. Находясь в тесном взаимодействии с изме-

няющимся внешним миром, система получает информацию для пополнения знаний и принятия решений.

Сформулированный выше принцип предполагает наличие в СИУ модели мира. Следующий принцип предусматривает ее пересмотр и подстраиваемость к изменяющимся условиям мира.

Второй принцип (принцип открытости, или развиваемости). Открытость СИУ для самообучения и самоорганизации.

Комментарий: для реализации принципа необходима способность системы преобразовывать сигнальную информацию в данные и, что существенно труднее, в знания. Очень трудной задачей в проблеме приобретения знаний является формирование достаточно адекватной модели мира – на основе такой фундаментальной интеллектуальной функции, как формирование новых понятий и отношений с целесообразным экспериментированием во внешнем мире.

Третий принцип (принцип прогнозирования). Предсказание изменений в мире (среде и системе) при разных предположениях.

Комментарий: в качестве указанных предположений могут выступать альтернативные управления, для каждого из которых система осуществляет прогноз динамики на определенный отрезок времени вперед с целью оценки и отбора наиболее предпочтительных управлений и соответствующей коррекции текущего, реализуемого, управления.

Четвертый принцип (принцип возрастания точности с уменьшением интеллектуальности – *increasing precision with decreasing intelligence*). Наличие у СИУ многоуровневой структуры, построенной в соответствии с правилом: повышение интеллектуальности и снижение требований к точности по мере повышения уровня управления в системе (и наоборот).

Комментарий: более верхний (более высокого ранга) уровень решает исходную или более близкую к исходной, более сложную, более творческую задачу. Зачастую эта задача и не может быть решена точно, чисто ал-

горитмически, в силу отсутствия точной постановки или подходящей теории (алгоритма), а решением является взаимосвязанная совокупность более простых подзадач, решение которых в совокупности обеспечивает решение рассматриваемой задачи более высокого уровня. Неточность или неполнота знания о модели ОУ, самой СИУ и среде компенсируется введением дополнительных более высоких уровней интеллектуальности.

Пятый принцип (принцип частичности деградации).

Комментарий: допускается разве лишь частичная утрата работоспособности при нарушениях в работе высших уровней управления, лишь частичная деградация, лишь некоторое ухудшение качества функционирования при отказах и сбоях подсистем высших рангов, благодаря децентрализации управления, частичному дублированию функций, перекрестным связям.

Для оценки эффективности СИУ важное значение имело бы умение измерять уровень интеллектуальности (интеллектуальную силу) автоматических и человеко-машинных систем. Уровень интеллектуальности (уровень интеллекта) некоторого агента определяется, прежде всего,

(А) способностью агента использовать старые знания в новых, может быть, заранее неизвестных проблемных ситуациях и

(Б) широтой проблемных областей, где оцениваемый агент приемлем как активный решатель задач.

Свойства А и Б очень интегральные и, например, могут включать способность планировать действия и самообучаться. Важность требования автономности функционирования агента или – в известной мере, противоположного свойства – умения взаимодействовать с другими агентами для совместного решения проблемы управления, равно как и требования нормального функционирования в условиях противодействия других агентов, зависит от проблемной области.

Можно различать два главных класса СИУ: 1) класс, в котором СИУ прямо (без участия человека) осуществляет функционирование, т.е. полностью замещает традиционные алгоритмы управления (или использует их в сочетании с интеллектными компонентами управления); 2) класс, в котором в СИУ реализуется человеко-машинное управление, т.е. с участием человека. СИУ этих двух классов могут именоваться как автоматические и человеко-машинные системы интеллектного управления (АСИУ и ЧМСИУ).

Если человеко-машинное управление понимается в этом широком смысле (как управление с тем или иным участием человека), а человека рассматривать как второе «устройство интеллектного управления», то могут существовать разные схемы взаимодействия этих двух устройств с общим для них объектом управления. Дальнейшее обобщение может связываться с рассмотрением двух или более СНУ (АСИУ и/или ЧМСИУ), когда каждая СНУ имеет свой объект управления, но взаимодействует с другой (другими) СНУ прямо или через общую среду (общие расходуемые ресурсы). Такие постановки принадлежат пограничной области нарождающейся в искусственном интеллекте теории – многоагентных систем (Multiagent Systems) и теории игр.

Поскольку свойства А и Б трудно оценить количественно, целесообразно в контексте традиционных и перспективных задач управления дать качественную шкалу возрастания потенциала решения задач в системах автоматического (и автоматизированного) управления. В нижеследующей шкале каждый класс систем включает все предыдущие с точки зрения охвата их возможностей управления.

1. Системы программного управления (разомкнутые системы).
2. Системы с обратной связью (замкнутые системы).
3. Системы идентификационного управления.
4. Системы адаптивного управления (системы с самонастройкой).

5. Системы интеллектуального управления (СНУ без целеполагания).
6. Интеллектуальные системы управления (СНУ с целеполаганием, формированием новых понятий и отношений в развиваемой модели мира).

Появление уровней интеллектуального управления 5 и 6 (без целеполагания и с целеполаганием), вообще говоря, переводит наши рассуждения в сферу кибернетики, определяемой как «наука об управлении, связи и переработке информации».

Появление функции целеполагания, а вместе с ней и развиваемой модели мира, существенно отличает интеллектуальные системы управления от СИУ без целеполагания, однако создание автоматических систем с целеполаганием остается весьма сложной проблемой. Вместе с тем, в настоящее время в классе интеллектуальных систем управления содержатся те СИУ, в которых имеются определенные средства воздействия со стороны оператора для того, чтобы по его желанию и с учетом обстановки изменять цели и критерии качества функционирования СИУ, а также модель мира. Воздействия оператора на процесс управления могут быть и шире, но принципиально новым качеством, не достигнутым в современных автоматических СИУ, является мотивированный интеллектуальный выбор цели, не сводящийся к выбору одной из нескольких, априори предусмотренных и хорошо формализованных целей.

В заключение заметим, что ареал применения средств искусственного интеллекта к задачам всех уровней управления постоянно расширяется.

На повестке дня создание полностью автоматических систем интеллектуального управления. Достижения в области автоматизации логического вывода, индуктивного логического программирования, порождения гипотез, распознавания образов и др. уже вызвали в жизнь ряд амбициозных проектов, в том числе, например, проект создания полностью автоматических пилотов.

Вместе с тем не должно быть неоправданного обобщения в оценке будущих перспектив использования интеллектуального управления, как это уже было в прошлом. В искусственном интеллекте требуются новые идеи и решения, в том числе развивающие и интегрирующие перспективные и уже сегодня полезные методы.

Темы курсовых работ:

1. Исследование и разработка структуры интеллектуальной системы поддержки принятия решений (ИСППР) для управления промышленным объектом.
2. Разработка для ИСППР экспертной подсистемы анализа запросов операторов при управлении промышленным объектом.
3. Разработка для ИСППР экспертной подсистемы анализа событий и оценки текущей ситуации на промышленном объекте.
4. Разработка для ИСППР экспертной подсистемы прогноза развития ситуации на промышленном объекте.
5. Разработка для ИСППР экспертной подсистемы формирования сценариев имитационного моделирования работы промышленного объекта.
6. Разработка монитора реального времени для координации работы нескольких экспертных систем в ИСППР.
7. Разработка для ИСППР экспертной подсистемы формирования данных для работы подсистемы имитационного моделирования.
8. Разработка монитора реального времени для управления подсистемой имитационного моделирования в ИСППР.
9. Разработка для ИСППР экспертной подсистемы для анализа результатов работы подсистемы имитационного моделирования.
10. Разработка для ИСППР экспертной подсистемы формирования и оценки вариантов альтернативного управления работой промышленного объекта.
11. Разработка для ИСППР экспертной подсистемы формирования данных для ответа на запрос оператора, управляющего промышленным объектом.
12. Разработка экспертной подсистемы анализа аварийных ситуаций с использованием инструментальной программной среды G2.

Лабораторный практикум

Конкретные описания работ входят в документацию соответствующих пакетов системных и прикладных программ по системе программирования на языке Пролог и инструментария оболочки экспертной системы реального времени G2.

Ниже приведен базовый состав лабораторного практикума, куда включены лабораторные работы, выполняемые с использованием стандартного инструментария.

| № лр | Названия лабораторных работ и краткое содержание | Число часов |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 1 | <u>Ознакомление с инструментальными средствами создания экспертных систем.</u> Ознакомление с инструментальными средствами создания ЭС с использованием стандартной оболочки ЭС и применения специализированных языков. Изучение особенностей применения языка Пролог для создания ЭС. | 2 |
| 2 | <u>Освоение работы с инструментальными средствами Пролог.</u> Освоение способов подготовки и задания исходных данных, диалога с системой, вариантов представления результатов с применением языка Пролог. | 2 |
| 3 | <u>Изучение моделей представления знаний в ИС, построенных с использованием продукционных правил.</u> Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных. | 2 |
| 4 | <u>Практическое задание фактов, правил для формирования модели представления знаний в ЭС с использованием продукционных правил.</u> | 2 |

| | | |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| | Экспериментальная проверка работы модели с использованием продукционных правил. | |
| 5 | <u>Изучение фреймовых представлений моделей представления знаний в ИС.</u> Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных | 2 |
| 6 | <u>Практическое задание фактов, правил для формирования модели представления знаний в ЭС</u> Формирование модели представления знаний в ЭС с использованием фреймовых представлений. Экспериментальная проверка работы модели | 2 |
| 7 | <u>Изучение моделей представления знаний в ИС с использованием семантических сетей.</u> Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных. | 2 |
| 8 | <u>Практическое задание фактов, правил для формирования модели представления знаний в ЭС с использованием семантических сетей.</u> Экспериментальная проверка работы модели. | 2 |
| 9 | <u>Изучение основных режимов работы и возможностей конфигурирования и настройки диалоговой подсистемы ЭС.</u> | 2 |
| 10 | <u>Изучения возможностей динамического изменения формы диалога, корректировки введенных фактов и правил</u> Экспериментальная проверка работы диалоговой подсистемы ЭС. | 2 |
| 11 | <u>Изучение методов вывода с использованием прямой и обратной цепочки рассуждений.</u> Ознакомление с работой подсистемы объяснения вывода в ЭС. | 2 |

| | | |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 12 | <u>Анализ заданных преподавателем вариантов примеров исходных данных и формирование рабочих данных для подсистемы вывода и подсистемы объяснений.</u> Экспериментальная проверка работы подсистем вывода и объяснений | 2 |
| 13 | <u>Ознакомление с инструментальными средствами программной среды G2.</u> Освоение работы с инструментальными средствами G2. Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных в среде G2. | 2 |
| 14 | <u>Исследование эффективности применения программной среды G2 на примере заданного преподавателем варианта.</u> | 2 |

Темы практических заданий и упражнений

Задание 1 - «Выбрать правильно цель»

Цель задания: освоить навыки правильного выделения целей для разрабатываемой системы ИИ, поиска возможных путей их достижения, построение дерева целей.

Содержание: цель, дерево целей, декомпозиция целей; факты и правила; прямая и обратная цепочка рассуждений.

Постановка задания: Одним из важных этапов создания интеллектуальной системы является определение целей, достигаемых с помощью ИС. Выбор и формулировка целей определяет эффективность реализации ИС. Точная формулировка и правильный выбор целей - одно из базовых умений специалиста по ИИ.

Упражнения «Выбрать и сформулировать цели, подцели и разработать дерево решений для создания компьютерной программы, умеющей оценивать:

- 1.1. знание слушателями терминологии интеллектуальных систем;
- 1.2. назначение и отличия основных интеллектуальных компонентов друг от друга;
- 1.3. понимание отличия аксиоматической системы от просто системы;
- 1.4. знание основных понятий классического исчисления предикатов;
- 1.5. знание правил вывода в исчислении предикатов;
- 1.6. знание прямой цепочки выводов;
- 1.7. знание обратной цепочки выводов.

Задание 2 - «Выбрать модели представления знаний»

Цель задания: освоить навыки правильного выбора модели представления знаний в интеллектуальных системах управления техническими объектами.

Содержание: модели представления знаний, логические модели, четкие и нечеткие продукционные правила, семантические сети, фреймовые представления, критериальные методы, нейронные сети, стохастические модели, эвристика, комбинированные модели представления знаний.

Постановка задания: Выбор модели представления знаний существенно влияет на эффективность работы ИС и является одним из важных этапов ее создания. Необходимо сделать и обосновать выбор различных вариантов моделей представления знаний для интеллектуальной системы. Выбор модели представления также относится к одному из базовых умений специалиста по ИИ.

Упражнение 2. «Выбрать модели представления знаний для обозначения сущностей и отношений между ними, определить виды суждений, формы суждений, категории предикатов и субъектов, основные и функциональные категории в соответствующих задачах:

2.1. определение имен, объемов и содержания имен на примерах единичных и общих имен, пустых и непустых, однозначных и многозначных, точных и неточных, ясных и неясных;

2.2. проиллюстрировать отношения между именами диаграммами Эйлера (равнозначность, подчинение, назначение, пересечение и исключение);

2.3. показать виды деления объектов на классы на примере целых чисел;

2.4. показать примеры выполнения и невыполнения требований к делению с учетом основания деления, исчерпывания объема, взаимного исключения и непрерывности;

2.5. дать примеры дихотомии и геометрического представления метода;

2.6. привести примеры простых и сложных классов на примере типов двигателей (паровые, внутреннего сгорания, реактивные и т.д.);

2.7. показать примеры решения с помощью классификационных систем задач: - классификации образов, распознавания образов, формирования образов;

2.8. привести примеры классификации с использованием иерархических систем;

2.9. привести примеры алфавитно-предметной классификации;

2.10. привести примеры классификации с использованием тезауруса, дескрипторов и условных синонимов;

2.11. привести примеры фасетной классификации.

Задание 3 - «Математическая реализация формальной логики»

Цель задания: освоение основных понятий формальной логики на примерах исчисления высказываний и исчисления предикатов.

Содержание: простые и сложные высказывания, категорические суждения, истинность, ложность и осмысленность заключений, правила вы-

вода, дедуктивные и индуктивные заключения, силлогизмы, модусы, виды отношений, виды исчислений, исчисление предикатов, аксиомы, законы и правила вывода.

Постановка задания: Показать понимание основных понятий формальной логики и методов логического вывода.

Упражнение 3. «Привести примеры, демонстрирующие следующие основные понятия формальной логики и методов логического вывода:

- 3.1. простых категорических высказываний в различных формах;
- 3.2. формирование сложных высказываний с помощью логических связок: отрицания, конъюнкции (И), дизъюнкции (ИЛИ), исключающей дизъюнкции;
- 3.3. высказываний об отношениях;
- 3.4. анализ осмысленности сложных высказываний с помощью логического квадрата;
- 3.5. индуктивные заключения;
- 3.6. дедуктивные заключения;
- 3.7. Основные модусы силлогизмов;
- 3.8. ложные и истинные формулы;
- 3.9. виды прямого и косвенного доказательства логических гипотез (теорем);
- 3.10. объяснение доказательства теорем;
- 3.11 приемы опровержений, софизмы и паралогизмы;
- 3.12. выводимые и невыводимые формулы;
- 3.13. Выполнимые и невыполнимые формулы;
- 3.14. общезначимые формулы;
- 3.15. формулы, содержащие различные кванторы;
- 3.16. формулы, содержащие аксиомы коммутативности, дистрибутивности и ассоциативности;
- 3.17. формулы, содержащие законы Де Моргана;

- 3.18. формулы, содержащие аксиомы классического исчисления предикатов;
- 3.19. формулы, содержащие правила вывода;
- 3.20. конъюнктов, дизъюнктов и резольвент;
- 3.21. выражений четких продукций;
- 3.22. выражений нечетких продукций;
- 3.23. понятие базы фактов;
- 3.24. понятие базы знаний;
- 3.25. понятие машины вывода;
- 3.26. понятие полного перебора в ширину;
- 3.27. понятие полного перебора в глубину;
- 3.28. эвристические методы поиска;
- 3.29. методы разбиения на подзадачи;
- 3.30. представление исходной задачи в виде И-ИЛИ графа.

Вопросы для самопроверки

1. Определение искусственного интеллекта (ИИ)
2. Определение интеллектуальных систем (ИС)
3. Определение систем интеллектуального управления (СИУ)
4. Основные этапы развития ИС и технологий
5. Ученые, внесшие большой вклад в развитие ИИ
6. Роль ИС и технологий в современном управлении
7. Основные интеллектуальные компоненты, применяемые в ИС
8. Основные подходы и методы, используемые в современных ИС и технологиях.
9. Понятие экспертных систем
10. Динамические экспертные системы.
11. Что такое нейронные сети.
12. Понятие эволюционного алгоритма

13. Понятие о системах, основанных на знаниях (СОЗ).
14. Понятие о формальных аксиоматических системах
15. Понятие о логическом выводе
16. Основные понятия классического исчисления предикатов
17. Язык и аксиомы исчисления предикатов.
18. Правила вывода исчисления предикатов.
19. Задачи поиска вывода в исчислении предикатов.
20. Понятие о методах порождения гипотез
21. Понятие о языке L-позитивно образованных формул
22. Понятие о дедуктивных правилах
23. Исчисление позитивно образованных формул.
24. Стратегии поиска вывода в исчислении J
25. Логическое порождение гипотез
26. Особенности получения, представления и использования знаний в ИС
27. Особенности получения, представления и использования знаний в экспертных системах
28. Понятия о моделях представления знаний в ИС, построенных с использованием продукционных правил
29. Понятия о моделях представления знаний в ИС, построенных с использованием динамических семантических сетей
30. Понятия о моделях представления знаний в ИС, построенных с использованием фреймовых и других представлений
31. Прямые и обратные цепочки выводов
32. Методы создания и особенности применения в управлении систем, основанных на правилах
33. Нечеткие продукционные правила
34. Логическое программирование.

35. Примеры создания и применения систем, основанных на различных правилах.
36. Системы, основанные на автоматическом доказательстве теорем.
37. Метод резолюций Дж. Робинсона и обратный метод С. Ю. Маслова
38. Системы естественного вывода (генценовского типа).
39. Примеры создания и применения систем, основанных на автоматическом доказательстве теорем
40. Системы, основанные на автоматическом выдвижении гипотез
41. Основные методы обучения с учителем и без учителя
42. Индуктивное логическое программирование.
43. Логические исчисления с обобщенными кванторами, GUHA – метод.
44. Метод получения правдоподобных рассуждений
45. Метод последовательного порождения гипотез
46. Примеры создания и применения систем, основанных на автоматическом выдвижении гипотез
47. Системы, основанные на рассуждениях по аналогии.
48. Основные формы рассуждений.
49. Правдоподобные рассуждения.
50. Рассуждения по прецеденту.
51. Понятие “близости” к прецеденту.
52. Комбинации различных методов.
53. Примеры создания и применения систем, основанных на рассуждениях по аналогии.
54. Объектно-ориентированные ИС.
55. Использование декларативно-процедурных форм представления знаний
56. Использование объектно-ориентированных языков программирования

57. Примеры создания систем с применением различных языков (ЛИСП, РЕФАЛ, Пролог и других).
58. Применение методов логического вывода для построения систем интеллектуального управления.
59. Использование логических языков в описании цели управления и доступных средств достижения цели
60. Определение плана достижения цели
61. Особенности построения и применения гибридных ИС.
62. Примеры создания и применения СОЗ, использующих объектно-логические языки, логики транзакций, фреймовые логики и другие.
63. Эффективность сочетания различных подходов
64. Методы учета ограниченности ресурсов ИС (неполнота информации, динамика изменения данных и знаний)
65. Методы учета ограниченности ресурсов ИС (задачи мягкого и жесткого реального времени, объем памяти)
66. Методы учета ограниченности ресурсов ИС (неполнота информации, динамика изменения данных и знаний, задачи мягкого и жесткого реального времени, объем памяти, возможности параллельной обработки информации).
67. Методы учета ограниченности ресурсов ИС (неполнота информации, динамика изменения данных и знаний, задачи мягкого и жесткого реального времени, объем памяти, возможности параллельной обработки информации).
68. Методы учета ограниченности ресурсов ИС (возможности параллельной обработки информации).
69. Методы и алгоритмы для формирования в ИС альтернативных управлений.

70. Особенности формирования “быстрых” приближенных решений и методы итеративного улучшения решений в рамках имеющихся ресурсов.
71. Основные проблемы повышения уровня интеллектуальности систем управления.
72. Недостатки традиционных подходов и методов интеллектуального управления.
73. Основные принципы повышения эффективности ИС.
74. Методы и подходы к оценке уровня интеллектуальности автоматических и автоматизированных управляющих систем.
75. Особенности оценки структур иерархических систем управления.
76. Нижний, средний и верхний уровни управления в антропоцентрических системах.
77. Основные типы ИС и технологий, используемых при создании и применении ИС в задачах управления.
78. Особенности построения и применения ИС в бортовых системах управления подвижными объектами
79. Особенности построения и применения ИС при управлении сложными производственными объектами.
80. Перспективные направления развития ИС и технологий.

Задания для самостоятельной работы

Меры близости и сходства

1. Основы объектной модели
2. Ознакомление с инструментальными средствами создания ЭС с использованием стандартной оболочки ЭС
3. Ознакомление с инструментальными средствами создания ЭС с использованием специализированных языков.
4. Основы языка Пролог
5. Изучение особенностей применения языка Пролог для создания ЭС.

6. Изучение моделей представления знаний в ИС, построенных с использованием четких продукционных правил
7. Изучение моделей представления знаний в ИС, построенных с использованием нечетких продукционных правил
8. Изучение фреймовых представлений моделей представления знаний в ИС
9. Изучение моделей представления знаний в ИС с использованием семантических сетей.
10. Изучение методов вывода с использованием прямой и обратной цепи рассуждений.
11. Ознакомление с инструментальными средствами инструментальной программной среды G2.

ЗАКЛЮЧЕНИЕ

В работе кратко изложены основные понятия о классических методах искусственного интеллекта и применении некоторых, наиболее распространенных технологий в современных интеллектуальных и интеллектных системах автоматизации управления сложными техническими объектами. Рассмотрены основные способы представления в ИС знаний и методов их компьютерной обработки, даны представления о типовых моделях процессов получения и обработки информации в ИС для формирования баз данных, баз знаний и построения экспертных и других интеллектуальных систем.

С учетом динамичного развития современной области ИИ работа является лишь введением в огромную и интересную область знаний. Слушатели, заинтересовавшиеся работами в этом перспективном научном направлении и вопросами практического применения методов ИИ в решении сложных задач управления, имеют прекрасную возможность самостоятельно продолжить изучение этой тематики. Огромный мир Интернета позволяет получить быстрый доступ к самой различной информации по тематике создания и применения интеллектуальных систем и технологий, к имеющейся обширной библиографии, углубленным теоретическим и практическим курсам по систематическому освещению многих вопросов, относящихся к области ИИ и ИС. Многие перспективные направления ИИ, ИС и систем интеллектуального управления в работе лишь упомянуты, или приведены в качестве примеров развития возможных теоретических направлений и практических применений.

Одно из таких перспективных направлений связано с развитием и внедрением интеллектуальных систем поддержки принятия решений в сложных задачах управления. Бурное развитие таких систем вызвано зна-

чительной потребностью в обеспечении квалифицированными кадрами все более усложняющихся автоматизированных систем управления.

В дополнение к материалам данного учебного пособия ниже даются списки рекомендуемой обязательной и дополнительной литературы для углубленного изучения рассматриваемых вопросов, а также ссылки на основные Интернет – ресурсы. Ссылки даны с учетом возможности реального доступа читателей-студентов к рекомендуемым материалам. В связи с чем в этих ссылках упомянуты далеко не на все классические и современные работы в области ИИ и ИС, но указано, где их можно найти (см. например, [21]).

Дополнительно к данному учебному пособию подготовлен также вариант электронного пособия, содержащего кроме гипертекстового материала пособия еще и ряд дополнительных методических материалов, рекомендаций, тестовых заданий, упражнений и расширенного глоссария. Электронный вариант включает также описание бально – рейтинговой системы учета работы слушателей по теме в семестре.

ЛИТЕРАТУРА

Обязательная литература

1. *Пупков К.А., Коньков В.Г.* Интеллектуальные системы. // МГТУ им. Н.Э. Баумана. – М, 2003.
2. *Захаров В.Н.* Интеллектуальные системы управления: основные понятия и определения. // Известия РАН. Теория и системы управления. – 1997. – № 3. – С.138-145.
3. *Джорж Ф. Люгер.* Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е издание // – Киев: Изд. дом «Вильямс», 2005. (<http://www.williamspublishing.com/PDF/5-8459-0437-4/part.pdf>).
4. *Афонин В.А., Макушкин В.А.* Интеллектуальные робототехнические системы. // М, 2005.
5. Справочник по искусственному интеллекту: в 3-х т. // *Под ред. Попова Э.В. и Поспелова Д.А.* // – М.: Радио и связь, 1990.
6. Толковый словарь по искусственному интеллекту/ Авторы-составители *А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов.* – М.: Радио и связь, 1992.
7. (Электронная версия: <http://www.raai.org/library/tolk/aivocpred.html>)
8. *Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е.* Интеллектуальное управление динамическими системами. // – М.: Физматлит. 2000.
9. *Поспелов Д.А.* Ситуационное управление: Теория и практика. // – М.: Наука, 1986.
10. *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976.
11. *Минский М.* Фреймы для представления знаний. – М.: Энергия, 1979.
12. *Ковальски Р.* Логика в решении проблем. – М.: Наука, 1990.

13. *Кохонен Т.* Ассоциативные запоминающие устройства. – М.: Мир.: 1982.
14. *Ту Дж., Гонзалес Р.* Принципы распознавания образов. // М.: Мир. 1989.
15. *Уотермен Д.* Руководство по экспертным системам. – М.: Мир. 1989.
16. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. – С-Пб.: Питер, 2000.
17. *Джексон П.* Введение в экспертные системы. – М.: Изд. дом «Вильямс», 2001.
18. *Смолин Д.В.* Введение в искусственный интеллект: конспект лекций. - 2-е изд., перераб. – М.: Физматлит, 2007.
19. *Сотник С. Л.* Конспект лекций по курсу «Основы проектирования систем искусственного интеллекта» // <http://neuroschool.narod.ru/sotnik.html> 1998.
20. *Братко И.* Программирование на языке PROLOG для искусственного интеллекта. – М.: Мир, 1990.
21. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход. – Киев: Изд. дом «Вильямс», 2006.
22. Материалы Российской ассоциации искусственного интеллекта // <http://www.raai.org/resurs/resurs.shtml?ilinks>

Дополнительная литература

1. *Борисов В.В., Круглов В.В., Федулов А.С.* Нечеткие модели и сети – М.: Горячая линия – Телеком, 2007.
2. *Лорьер Ж.Л.* Системы искусственного интеллекта. – М.: Мир, 1991.
3. *Непейвода Н.Н.* Логический подход как альтернатива системному в математическом описании систем – М.: Изд-во ЛИИ АН СССР, 1989. № 85.

4. *Мальковский М.Г., Грацианова Т.Ю., Полякова И.Н.* Прикладное программное обеспечение: системы автоматической обработки текстов. – М.: МГУ 2000.
5. *Марцеллус Д.* Программирование экспертных систем на turbo Prolog. – М.: Финансы и статистика, 1994.
6. Математический энциклопедический словарь. – М.: Сов. энциклопедия, 1988.
7. *Пильщиков В.Н.* Язык ПЛЭНЕР. – М.: Наука, 1983.
8. Приобретение знаний. /Под ред. *Осуга С* – М.: Мир, 1990.
9. *Поспелов Д. А.* Системный анализ и искусственный интеллект. // Сообщение по прикладной математике. – М.: ВЦ АН СССР, 1980.
10. *Поспелов Д. А.* Становление информатики в России. – М., 1997.
<http://raai.org/about/persons/pospelov/pages//stanovl.htm>
11. *Рыбина Г.В.* Автоматизированное рабочее место для построения интегрированных экспертных систем: комплекс АТ-ТЕХНОЛОГИЯ. // «Новости Искусственного Интеллекта», 2005, № 3.
12. *Семенов М.Ю.* Язык ЛИСП для персональных ЭВМ. – М.: Изд-во МГУ, 1989.
13. *Сойер Б., Фостер Д.Л.* Программирование экспертных систем на паскале. – М.: Финансы и статистика, 1990.
14. *Тихомиров О.К.* Психология мышления. – М.; Изд-во МГУ, 1984.
15. *Трахгенгерц Э.А.* Компьютерная поддержка принятия решений. – М.: СИНТЕГ, 1998.
16. *Уэно Х. и др.* Представление и использование знаний. – М.: Мир, 1989.
17. *Адаменко А.Н., Кучуков А.М.* Логическое программирование и Visual Prolog. – СПб.: БХВ – Петербург, 2003.
18. *Горбань А.Н.* Обучение нейронных сетей. – М.: СП Параграф, 1990.
19. *Маслов С.Ю.* Обратный метод установления выводимости в классическом исчислении предикатов. // Докл. АН СССР. – М.: 1964. - Т. 159. – № 1. – С. 17-20.

20. *Робинсон Д.* Машинно-ориентированная логика, основанная на методе резолюций. // Киберн. сб. Новая сер. вып.7. – М., Мир, 1970 . С. 180-218.
21. *Шанин Н.А., Давыдов Г.В.* и др. Алгоритм машинного поиска естественного логического вывода в исчислении высказываний. – М.-Л.: Наука, 1965.
22. *Фельдбаум А.А.* Новые принципы автоматического управления. // Изв. ВУЗов СССР. Радиотехника. – 1960. – № 3. – С. 299-308; № 4 С. 419-430.
23. *Гаек П., Гавранек Т.* Автоматическое образование гипотез: математические основы общей теории. – М.: Наука, Физматлит, 1984.
24. *Айзерман М.А., Браверман Э.М., Розоноэр Л.И.* Метод потенциальных функций в теории обучения машин. – М.: Наука, 1970.
25. *Вапник В.Н., Червоненкис А.Я.* Теория распознавания образов. – М.: Наука, 1974.
26. *Цыпкин Я. З.,* Адаптация и обучение в автоматических системах. - М.: Наука, 1968.
27. *Борисов В.Г., Данилова С.К., Чинакал В.О.* Интеллектуальная система поддержки принятия решений для интегрированных систем управления и навигации морскими подвижными объектами. // Материалы XI Санкт-Петербургской международной конференции по интегрированным навигационным системам. ЦНИИ «Электроприбор». г. С-Пбург. 2004.
28. *Чинакал В.О.* Интеллектуальная подсистема поддержки принятия решений в интегрированной системе управления для производств химико-технологического типа. // Материалы III Международной конференции по проблемам управления. – М.: т.2, ИПУ РАН, 2006.
29. *Хорошевский В.Ф., Шерстнев В.Ю.* Программный инструментарий представления знаний в экспертных системах. // Экспертные системы: состояние и перспективы. – М.: Наука, 1989. – С.38-46.

Основные интернет-ресурсы

1. • www.raai.org/resurs - Материалы Российской ассоциации искусственного интеллекта
2. • www.artint.ru - Российский Научно-исследовательский институт искусственного интеллекта (РосНИИ ИИ)
3. • fuzzy.kstu.ru/rans.htm - Российская ассоциация нечетких систем
4. • www.larichev.com - Сайт академика О.И.Ларичева
5. • www.aaai.org - Американская ассоциация искусственного интеллекта American Association for Artificial Intelligence (AAAI)
6. • prof9.narod.ru - Искусственный Интеллект от Prof'a. Сайт посвящен Искусственному Интеллекту и всему, что с ним связано.
7. • lii.newmail.ru - Лаборатория искусственного интеллекта. В основном содержит материалы по нейронным сетям.
8. • www.ai.obrazec.ru - Сайт «Искусственный интеллект»
9. • aifuture.chat.ru - Искусственный интеллект («Взгляд в будущее»).
10. • www.aicomunity.org - Материалы об искусственном интеллекте
11. • newasp.omskreg.ru/intellect/ - Сборник электронных вариантов статей и книг, объединенных общей темой «Парадигма искусственного интеллекта».
12. • ni.iont.ru - Российская ассоциация нейроинформатики (РАСНИ).
13. • artema.fopf.mipt.ru/ai/aihist.html - Материалы по ИИ. В т.ч. об истории, языках ИИ и прочие материалы.

ПЕРСОНАЛИИ

| | |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Аристотель (род. 384 г. до н.э.) | Известный древнегреческий ученый, впервые разработал классическую модель формальной логики, послужившая основой для создания всех современных моделей представления знаний в ИИ. |
| Буль Дж | Труды по усовершенствованию логической системы обозначений и представлению абстрактной логической задачи в виде системы уравнений (Булева алгебра). |
| Винер Норберт | Создал основополагающие работы по кибернетике. |
| Декарт Рене (1596-1650) | Французский философ, математик и естествоиспытатель В XVII в. предложил универсальный язык классификации всех наук. |
| Заде Лотфи | Американец иранского происхождения, впервые в 1965 г. сформулировал понятия нечетких множеств и нечеткой логики. |
| Кант Иммануил (1724-1804) | Теолог, основатель философского критицизма, разработывал энциклопедию всех наук, создал учение о законах познания, понятиях, формах суждений, работы по логике и семантическому анализу на основе категорий. |
| Ковальски Роберт | Сформулировал основные положения логического программирования и вычислительную модель, в 1974 г. установил условия эффективности в виде использования множества хорновских дизъюнктов, в 1976 г. предложил вместе Мартеном Ван Эмденом процедурный и декларативный подходы к прочтению текстов логических программ. |
| Кольмероз Алан | Французский ученый в 1973 г. в марсельском университете составил алгоритм формального способа интерпретации логического вывода, разработал программу для доказательства теорем, позднее создал на основе метода резолюций Дж. Робинсона и процедурной семантики Р. Ковальски язык логического программирования Пролог. |
| Коллинз Аллан | Психолог провел в 1969-1970 гг. совместно с Р. Куллианом экспериментальную проверку отдельных теоретических положений об устройстве человеческой памяти и организации логического вывода на основе иерархической сети. |

| | |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Кохонен Тейво | Финский ученый, работы в области создания теории современных нейромоделей, разработал самоорганизующиеся сети (карты Кохонена), не требующие предварительного обучения. |
| Куиллиан Росс. | Социолог, работал в области ИИ. В 1969 г. ввел понятие семантической сети и разработал ее теоретические основы. |
| Лейбниц | В XVIII в. предложил универсальный язык классификации всех наук. |
| Луллий Раймонд | Средневековый испанский философ, математик и поэт XIII века. Пытался создать механическую машину для решения различных задач с использованием разработанной им всеобщей классификации понятий. |
| Маккарти | Автор ЛИСПа – первого языка программирования для задач ИИ. |
| Мак-Каллок У.С. | Американский хирург в 1943 г. описал совместно с математиком В. Питтсом первую математическую модель искусственной нейронной сети. |
| Маслов Ю. С. | Выдающийся отечественный математик, предложил в 1964 г. названный его именем обратный вывод установления выводимости в классическом исчислении предикатов. |
| Мински Марвин. | Автор идеи фрейма и фреймовой модели представления знаний, в 1951 г. построил первую нейронную сеть SNARC. |
| <u>Петри</u> Карл | Впервые разработал и описал в 1962 г. математический аппарат для моделирования динамических дискретных систем (Сети Петри). |
| Питтс В. | Американский математиком в 1943 г. описал совместно с хирургом У.С. Мак-Каллоком первую математическую модель искусственной нейронной сети. |
| Понтрягин Лев Семёнович (1908 -1988) | Выдающийся советский ученый, математик, создатель математической теории оптимальных процессов, в основе которой лежит т.н. принцип максимума Понтрягина, имеет фундаментальные результаты по дифференциальным играм. |
| Поспелов Дмитрий Александрович | Выдающийся российский ученый, основателем ситуационного управления в России в 1965 г., разработал специальные модели представления ситуаций - представления знаний, создатель (1988 г.) и президент российской Ассоциации Искусственного Интеллекта. |

| | |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | |
| Робинсон Дж. | В 1965 г. разработал метод автоматического поиска доказательств теорем в исчислении предикатов первого порядка при наличии набора исходных аксиом, получивший название метода резолюций. |
| Розенблат Фрэнк | Американский психолог, в 1959-1962 гг. создал первый персептрон (электромеханическое устройство), провел эксперименты, доказал теоремы о его потенциальных возможностях. |
| Хебб Дональд | Канадский нейропсихолог, в 1949 г. Выдвинул теорию клеточных ансамблей, разработал сигнальный и дифференциальный методы обучения в нейронных сетях, используемые в алгоритмах обучения без учителя. |
| Хинтон Дж. | В 1986 году с коллегами опубликовал статью с описанием модели нейронной сети и алгоритмом ее обучения с обратным распространением ошибки (back propagation). |
| Хопфилд Дж. | Американский биофизик, в 1982 году предложил оригинальную модель нейронной сети, названную его именем |

КРАТКИЙ ГЛОССАРИЙ

При выборе вариантов толкования большинства терминов в глоссарии за основу была принята терминология наиболее полной версии толкового словаря по искусственному интеллекту [6], разработанная известными специалистами в области искусственного интеллекта А.Н. Аверкиным, М.Г. Гаазе-Рапопортом и Д.А. Поспеловым («Толковый словарь по искусственному интеллекту». Авторы-составители: А.Н. Аверкин, М.Г. Гаазе-Рапопорт и Д.А.Поспелов. – М.: Радио и связь, 1992. 256с.).

Обновляемая электронная версия словаря находится по адресу: <http://www.raai.org/library/tolk/aivocpred.html>).

Универсальность и большой объем данного полного толкового словаря связаны с междисциплинарным характером исследований в области искусственного интеллекта (ИИ) и интеллектуальных систем (ИС). В связи с этим в полной версии словаря по ИИ встречается немало терминов из логики, психологии, лингвистики, кибернетики, дискретной математики, программирования и других наук.

В сокращенной рабочей версии словаря, используемой в данном пособии, учитывается его основное назначение для студентов бакалавриата Инженерного факультета РУДН. Предполагается, что они уже знакомы со многими вопросами по материалам курсов по теории автоматического управления, математическим основам кибернетики, моделированию систем, системам автоматизации и управления, базам данных в информационно-управляющих системах, программированию и основам алгоритмизации.

АБДУКЦИЯ (ТРАДУКЦИЯ)

Правдоподобный вывод от частного к частному.

АБСТРАКЦИЯ

Процесс отсечения единичного, случайного или несущественного для последующих шагов работы. А. всегда имеет место при представлении данных и знаний о внешнем мире в интеллектуальных системах. А. используется при обобщении знаний, проведении рассуждений и планировании целесообразной деятельности. А. является средством образования понятий.

АВТОМАТ

Абстрактная машина, преобразующая последовательности входных символов в последовательности выходных символов. В зависимости от числа внутренних состояний памяти А. различаются конечные А. и бесконечные А.; в зависимости от однозначности или неоднозначности формирования выходных последовательностей - детерминированные А. и недетерминированные А.; в зависимости от особенностей структуры магазинные А., стековые А., клеточные А.

АВТОМАТ ВЕРОЯТНОСТНЫЙ

Частный случай стохастического автомата, когда структура автомата остается неизменной при любых результатах его функционирования.

АВТОМАТ ДЕТЕРМИНИРОВАННЫЙ

Автомат, у которого в любой такт работы набор входных символов и внутреннее состояние однозначно определяет набор выходных символов и внутреннее состояние А.Д. в последующем такте работы.

АВТОМАТ КОНЕЧНЫЙ

Автомат, работа которого определяется двумя функциями:

$$y(t+1) = F1(x(t), y(t)),$$

$$z(t) = F2(x(t), y(t)).$$

Первая функция задает смену состояний автомата в дискретные такты времени t и называется функцией переходов; вторая – выходные сигналы автомата и называется функцией выхода; x , y , и z – множества двоичных векторов фиксированной длины, т.е. конечные множества. Математической моделью А.К. может служить автоматная грамматика с помощью которой порождается автоматный язык.

АВТОМАТ ЛИНЕЙНО-ОГРАНИЧЕННЫЙ

Частный вид машины Тьюринга, у которого в каждый момент времени лента имеет конечную длину. При необходимости сдвига управляющей головки за край ленты лента наращивается на конечный отрезок, нужный головке. Линейно-ограниченным автоматам соответствуют контекстно-зависимые грамматики, порождающие контекстно-зависимые языки.

АВТОМАТ НЕДЕТЕРМИНИРОВАННЫЙ

Автомат, у которого в некоторые такты работы набор входных символов и внутреннее состояние задают альтернативный выбор набора выходных символов и/или внутреннего состояния А.Н. в последующем такте работы. Частный случай А.Н. являются вероятностный автомат и стохастический автомат.

АВТОМАТ С ПЕРЕМЕННОЙ СТРУКТУРОЙ

См. Автомат стохастический.

АВТОМАТ СТОХАСТИЧЕСКИЙ

Автомат, у которого вместо функций переходов и выходов в общем случае задаются распределения вероятностей дискретного типа. Для переходов задаются вероятности H_{ij} , характеризующие вероятность смены состояния с номером i на состояние с номером j , а для выхода вероятности Q_{ij} , характеризующие появление выхода с номером j , если текущее состояние автомата имеет номер i . А.С. часто используется для описания процесса адаптации к среде, в которой он функционирует. В зависимости

от успеха или неуспеха действий А.С. пересчитываются H_{ij} и Q_{ij} , что приводит к адаптации А.С., если среда носит стационарный характер.

АКСИОМА

Утверждение, которое априорно считается истинным.

АКТАНТ

Термин А. связан с определенной синтаксической концепцией, согласно которой элементарные высказывания расчленяются на функции (субъект, объект, предикат) и предикат рассматривается как ядро высказывания.

АКТОР

Специальным образом организованная процедура, характерная для объектно-ориентированного стиля программирования. Особенность А. состоит в том, что он самостоятельно включает в работу (активизируется), когда выполняются условия активизации. После окончания работы А. передает полученные им результаты другим А. Использование А. обеспечивает децентрализованное и максимальное параллельное решение задач.

АЛГОРИТМ

Система предписаний, задающая последовательность действий, с помощью которых за конечное число шагов находится решение всех задач определенного класса или выдается сообщение об отсутствии решения.

АЛГОРИТМ ГЕНЕТИЧЕСКИЙ

Организация процесса, напоминающего эволюцию в живой природе. Альтернативные решения плистируют популяцию. Для выбора решения используются механизмы, похожие на мутацию и кроссинговер. Альтернативные решения сравниваются между собой и «выживает» то решение, которое приобретает в «популяции» максимальный вес, А.Г. используются в моделях обучения.

АНАЛИЗ

Способ, вид исследования при котором реальный или мыслимый объект расчленяется на составляющие части (элементы) и изучаются эти элементы и связи между ними.

АНАЛИЗ КЛАСТЕРНЫЙ

Разбиение множества объектов на кластеры (компактные группы объектов) в специально сконструированном пространстве, метрика которого такова, что в один кластер попадают объекты, близкие друг другу с точки зрения этой метрики. Выделение кластеров может производиться многими способами теории А.К.

АНАЛИЗ СИНТАКСИЧЕСКИЙ

1. Проверка выражения на вхождение его в совокупность построенных выражений.
2. При обработке естественного языка под А.С. понимается построение синтаксической структуры предложения на некотором естественном языке.

В интеллектуальных системах реализующих функции понимания текстов на ограниченном естественном языке, А.С. осуществляется в два этапа. На первом строится поверхностная синтаксическая структура, в которой участвуют части речи и отношения между ними. На втором этапе происходит переход к глубинной синтаксической структуре, которая по сути познавательная структура, связанная с отражением действительности в тексте на естественном языке. Для А.С. созданы программные системы – синтаксические анализаторы.

АНАЛИЗ СЦЕН

Комплекс моделей и методов, позволяющий в интеллектуальных системах (например, в интеллектуальных роботах) отображать в памяти системы трехмерные зрительные сцены. При проецировании сцен в память

происходит переход к их формальному описанию. При этом необходимо правильно распознать объекты, принимающие участие в сцене, определить их расположение по глубине, дополнить загороженные участки объектов и т.п.

АНАЛИЗАТОР СИНТАКСИЧЕСКИЙ

Средства для проверки соответствия входных цепочек заданного языка его грамматике. Классификация А.С., как правило базируется на классификации соответствующих формальных грамматик.

АНАЛОГИЯ

Связь между двумя объектами, процессами, событиями или ситуациями на уровне отношений сходства – различия в базе знаний.

АНАФОРА

Повторение одного и того же слова или словосочетания в рамках одного предложения или соседних предложений в тексте. Часто при повторении используется не само слово или словосочетание а так называемое анафорическое слово (чаще всего местоимение). Например, в тексте «Петя шел в школу. Она находилась на краю поселка» «она» есть анафорическое слово для слова «школа». Такие анафорические ссылки затрудняют автоматический анализ текстов для их понимания в интеллектуальных системах.

АРГУМЕНТАЦИЯ

Процесс доказательства истинности утверждения с привлечением фактов, из которых следует истинность данного утверждения или которое увеличивает уверенность в его истинности А. близка к обоснованию.

АССОЦИАЦИЯ

Связь между двумя информационными единицами в базе знаний, устанавливаемая на основе некоторой меры близости, определяемой на множестве информационных единиц, хранимых в этой базе.

АТОМ

См. Формула атомарная.

АТТРИБУТ

Уникальное имя, приписываемое домену значений некоторой информационной единицы.

БАЗА ДАННЫХ

Совокупность программных средств, обеспечивающих поиск, хранение и запись информационных единиц заданной структуры (данных) в памяти ЭВМ.

БАЗА ДАННЫХ ИЕРАРХИЧЕСКАЯ

База данных, в которой между информационными единицами введены отношения вида «Элемент-класс», «тип-подтип» и т.п., с помощью которых образуются иерархические классификации хранящихся в базе информационных единиц.

БАЗА ДАННЫХ РЕЛЯЦИОННАЯ

База данных, в которой информационные единицы связаны между собой отношениями типа «один к одному», снабжены атрибутами и для представления отношений используется запись в виде таблиц.

БАЗА ДАННЫХ СЕТЕВАЯ

База данных, в которой информационные единицы связаны между собой отношениями типа «один к одному», «один к многим» и «многие к многим».

БАЗА ДАННЫХ ЭКСТЕНСИОНАЛЬНАЯ

База данных, в которой хранятся лишь константные факты о внешнем мире.

БАЗА ЗНАНИЙ

Совокупность программных средств, обеспечивающих поиск, хранение, преобразование и запись в памяти ЭВМ сложно структурированных информационных единиц (знаний).

БАЗА ЗНАНИЙ ИНТЕНСИОНАЛЬНАЯ

База знаний, в которой описаны общие закономерности, характерные для некоторой проблемной области, а также способы постановки и решения задач в этой области.

БАЗА ЗНАНИЙ ОТКРЫТАЯ

База знаний, позволяющая в процессе ее функционирования пополнять содержимое базы и убирать знания из базы. Свойство открытости приводит к тому, что вывод в такой базе является немонотонным, т.е. истинность выведенных в ней утверждений может меняться в процессе работы системы с такой базой.

БЕКТРЕКИНГ

Процедура возврата при поиске на некоторой структуре (например, поиск по дереву решений или в лабиринте). При движении по структуре часто возникает необходимость, когда выбранный путь оказался неудачным или тупиковым в возврате к месту разветвления процесса поиска. Для ускорения возможности возврата в последнюю точку ветвления ее координаты следует хранить в памяти. Для хранения совокупности вложенных друг в друга по старшинству точек ветвления используются специальные стековые регистры.

БИХЕВИОРИЗМ

Сведение создания к целенаправленному поведению, связанному с парой «стимул-реакция». Для Б. характерно рассмотрение субъекта как черного ящика. Ряд моделей искусственного интеллекта был построен на основе подхода, декларируемого в Б.

ВАЛИДАЦИЯ

Оценка программного продукта с точки зрения соответствия всем предъявленным к нему требованиям.

ВОСПРИЯТИЕ

Отражение окружающей ситуации и ее элементов при взаимодействии органов чувств человека или рецепторов искусственной системы с внешней средой. В. обеспечивает непосредственно-чувственную ориентировку в среде и порождает поток входной информации для последующей обработки ее человеком или искусственной системой. Для интеллектуальных систем наиболее важными видами Восприятия являются восприятие зрительной информации, восприятие тактильной информации и акустической информации (распознавание речи).

ВОСПРИЯТИЕ ЗРИТЕЛЬНОЙ ИНФОРМАЦИИ

Обработка сигналов, поступающих в интеллектуальную систему (интеллектуальный робот) с датчиков зрительных сцен. В качестве датчиков для плоских изображений часто используются матрицы из фотодиодов, но применяется и фотографическая аппаратура. Для трехмерных сцен чаще всего в качестве датчика используют телевизионную камеру. При В.З.И. происходит очистка изображения от шумов и искажений, анализ плоского изображения или анализ сцен для объемного изображения, перекодировка изображения и передача его в базу знаний или решатель.

ВОСПРИЯТИЕ ТАКТИЛЬНОЙ ИНФОРМАЦИИ

В интеллектуальных роботах обработка сигналов, поступающих от датчиков, меряющих кинематические характеристики для эффекторов робота, и от специальных датчиков, измеряющих усилия, связанные с взятием предметов или опорой на грунт. После предварительной обработки полученная информация поступает в систему планирования деятельности и используется для выработки управляющих воздействий на среду или на робота.

ВЫБОРКА ОБУЧАЮЩАЯ

Набор примеров и контрпримеров для формирования решающих правил. Входит в состав таблицы обучения.

ВЫВОД

Получение новых информационных единиц из ранее известных. Частным случаем является логический вывод.

ВЫВОД АБДУКТИВНЫЙ(ТРАДУКТИВНЫЙ)

Вывод на основании абдукции.

ВЫВОД ВЕРОЯТНОСТНЫЙ

Вывод, при котором каждое выражение, используемое в нем, имеет оценку правдоподобия в виде вероятности того, что оно является истинным. При В.В. применяются специальные процедуры для вычисления вероятности истинного значения результирующего выражения по вероятностям посылок, используемых при выводе.

ВЫВОД ЕСТЕСТВЕННЫЙ

Вывод, полученный на основании «здравого смысла». В.Е. может либо соответствовать логическому выводу в некоторой формальной системе (но быть для человека очевидным), либо опираться на соображения, которые не укладываются в строгие рамки формальной системы.

ВЫВОД ИНДУКТИВНЫЙ

Вывод «от частного к общему». Позволяет на основании обобщения частных примеров некоторого явления выдвинуть гипотезу о существовании общей закономерности. В интеллектуальных системах, использующих В.И., работают механизм, позволяющий при формировании гипотезы приписывать ей оценку правдоподобия (например, вероятность того, что данная гипотеза является истинной). В.И. является средством получения новых знаний в интеллектуальных системах.

ВЫВОД ИНТУИЦИОНИСТСКИЙ

Вывод, характерный для интуиционистской логики, не использующий, в частности, закон снятия двойного отрицания и закон исключенного третьего.

ВЫВОД ЛИНЕЙНЫЙ

Последовательность дизъюнктов, в которой начальный дизъюнкт принадлежит исходному множеству, в каждый промежуточный дизъюнкт является резольвентой предшествующего в уже построенной части последовательности дизъюнктов и некоторого бокового дизъюнкта.

ВЫВОД ЛОГИЧЕСКИЙ (ДЕДУКТИВНЫЙ)

1. Последовательность рассуждений, приводящая от посылок к следствию с использованием аксиом и правил вывода.
2. Результат вывода.

ВЫВОД НА ЗНАНИЯХ

Вывод, использующий в качестве посылок выражения, хранящиеся в базе знаний В.Н.З. может быть достоверным, если эти выражения являются достоверными, или правдоподобным, или снабжены оценками правдоподобия. Как правило, процедуры В.Н.З. включают поиск необходимых знаний для вывода, т.е. процедуру поиск по образцу.

ВЫВОД НЕМОНОТОННЫЙ

Вывод, при котором нарушается свойство монотонности при выводе.

ВЫВОД НЕЧЕТКИЙ

Вывод, при котором используются нечеткие кванторы или значения функций принадлежности. При нечетких кванторах правила вывода определяют тот квантор, который следует приписать результату при заданных значениях кванторов посылок. При использовании значений функций принадлежности правило вывода определяет значение этой функции для результата по значениям функций принадлежности посылок.

ВЫВОД ОБРАТНЫЙ

Вывод, при котором поиск доказательства начинается с целевого утверждения. Выясняются условия, при которых целевое утверждение является выводимым. Эти условия принимаются за новые целевые утверждения и процесс поиска продолжается. В.О. заканчивается, когда все очередные условия оказываются аксиомами или процесс условий обрывается, не приведя к аксиомам. В.О. широко используется в интеллектуальных системах при поиске решений.

ВЫВОД ПО АНАЛОГИИ

Вывод, основанный на перенесении рассуждения из исследованной области на другую область, похожую на исследованную. Если имеется вывод $A \rightarrow B$, и область, в которой определено A , гомоморфна области, где определена C , а область, где определено B , гомоморфна области, где определено D , то вывод $A \rightarrow B$ порождает вывод $C \rightarrow D$. В.П.А. есть частный случай правдоподобного вывода.

ВЫВОД ПРАВДОПОДОБНЫЙ

Вывод, при котором каждый шаг сопровождается вычислением оценки достоверности полученного утверждения. Частными случаями В.П. является, например, вывод вероятностный и вывод индуктивный.

ВЫВОД ПРЯМОЙ

Вывод, ведущий от исходных аксиом к целевому выражению. При В.П. из-за неоднозначности выбора применимы аксиом и правил вывода образуется дерево решений и процесс нахождения цепочки, ведущей от исходных аксиом к целевому выражению, является переборным. Стандартной процедурой, используемой при обходе дерева решений, является процедура возврата – бектрекинг.

ВЫЗОВ ПО ОБРАЗЦУ

Способ поиска информации в базах данных, базах знаний или в оперативной памяти компьютера. В отличие от поиска по адресу хранения, В.П.О. предполагает ассоциативный поиск по содержимому запросу на поиск. (См. Поиск по образцу).

ВЫСКАЗЫВАНИЕ

Логическое выражение, относительно которого всегда можно утверждать, что оно либо истинно, либо ложно.

ВЫСКАЗЫВАНИЕ АТОМАРНОЕ

Высказывание, структура которого далее не расчленяема. В формальных системах В.А. соответствуют базовые элементы. Интерпретация истинности В.А. задается для формальной системы извне и на этой основе определяется интерпретация всех правильно построенных формул формальной системы.

ГЕНЕРАЦИЯ ТЕКСТА

Процесс порождения текста, включающий выделение фрагмента внутреннего представления, который войдет в текст; формирование схемы дискурса, т.е. последовательности, в которой должна быть изложена информация; заполнение схемы дискурса языковыми выражениями. При Г.Т. также учитывается фокус внимания и предотвращения коммуникативных ошибок. (См. Порождение текста. Синтез текста).

ГИПЕРСОБЫТИЕ

Специально организованное описание типовой ситуации (кражи, драки, покупки в магазине и т.п.). Г. может представляться в базах знаний различными способами, например, в виде сценариев.

ГИПОТЕЗА

Частично обоснованная закономерность знаний, или для связи между различными эмпирическими фактами, или для объяснения факта или

группы фактов. В интеллектуальных системах Г. порождаются в процессе обучения систем (в частности, при обучении на примерах).

ГИПОТЕЗА КОМПАКТНОСТИ

Предположение о том, что образы в пространстве признаков группируются из изображений (точек пространства), которые могут быть отделены друг от друга гиперповерхностями простого вида. Гипотеза Компактности используется при распознавании образов, когда применяется принцип разделения.

ГЛУБИННАЯ СТРУКТУРА (предложения)

См. Структура глубинная.

ГРАММАТИКА

Совокупность правил формирования правильных предложений в рамках рассматриваемого языка.

ГРАММАТИКА КОНТЕКСТНО-ЗАВИСИМАЯ

См. Грамматика контекстно-связанная.

ГРАММАТИКА КОНТЕКСТНО-СВОБОДНАЯ

ГРАММАТИКА КОНТЕКСТНО-СВЯЗАННАЯ

Формальная грамматика, для которой существуют такие цепочки что и имеют место правила и К.С. обладают свойством сохранения длины цепочки. Цепочки, получаемые после применения любого правила, либо сохраняют длину исходной цепочки, либо увеличивают ее. Г.К.С. порождаются линейно-ограниченными автоматами. И для каждой Г.К.С. может быть построен воспроизводящий ее линейно-ограниченный автомат.

ГРАММАТИКА ФОРМАЛЬНАЯ

Четверка $\langle S, A, B, P \rangle$, в которой S – аксиома Г.Ф.; A – множество нетерминальных символов; B – множество терминальных символов; P – правила вывода. Объектами, с которыми работает Г.Ф., являются цепочки, состоящие из терминальных и нетерминальных символов. Правила

вывода имеют вид, где и цепочки символов. При этом содержит хотя бы один нетерминальный символ. Функционирование Г.Ф. всегда начинается с цепочки, состоящей из единственного символа S . Применение правила к цепочке заключается в замене всех вхождений (или только самого левого вхождения) в α . Процесс заканчивается, когда ни одно из правил вывода к данной цепочке неприменимо. Такие заключительные цепочки входят в язык, порождаемый данной Г.Ф. В синтаксически правильных Г.Ф. в язык входят только те цепочки, которые состоят полностью из терминальных символов. Г.Ф. широко используются в синтаксических моделях для естественных языков и в лингвистических процессорах. Они являются частным случаем формальных систем. В зависимости от ограничений, накладываемых на структуру правил вывода, выделяются различные типы Г.Ф.

ГРАФ

Пара (X, R) , где X – множество, элементы которого переименованы и называются вершинами; R – бинарное отношение, заданное на X . Если между вершинами $x_1 \in X$ и $x_2 \in X$ существует отношение R , то тройка $x_1 R x_2$ называется ребром G . Если отношение R несимметрично, то $x_1 R x_2$ называется дугой G . G с ребрами называется неориентированным, а с дугами – ориентированным. G находят широкое применение в моделях интеллекта искусственного.

ГРАФИКА КОГНИТИВНАЯ

Направление в машинной графике, которое связывает представления, возникающие на экране дисплея, с когнитивными процессами, протекающими при решении задач. Г.К. позволяет как бы визуализировать процесс решения. При достаточно продуманной системе визуализации образы, возникающие в динамике на экране, могут помочь пользователю, решающему задачу в интерактивном режиме, увидеть те закономер-

ности или пути решения задачи, которые ранее для него не были доступны. С развитием Г.К. связывают большие надежды на повышение эффективности решения задач, так как мышление пользователя может существенно ускорить процесс поиска решения и рождать новые пути его поиска. Г.К. требует специальных представлений в базе знаний соответствующих образам на экране дисплея, и процедур соотнесения этих представлений с традиционными когнитивными структурами.

ДЕЙСТВИЕ

Единица процесса деятельности, активность, направленная на достижение определенной цели. Д. может быть внутренним, направленным на преобразование информации внутри интеллектуальной системы, или внешним, направленным во внешнюю среду (сообщение пользователю, движение манипулятора автономного робота и т.п.).

ДЕКОМПОЗИЦИЯ ЗАДАЧ

Разбиение задачи на подзадачи с последующим разбиением этих подзадач до получения базовых (элементарных) задач, для которых заранее известно решение. Д.З. применяется в интеллектуальных системах при создании систем автоматического программирования и при планировании поведения в пространстве задач. В более общем смысле Д.З. может служить для понимания размерности решаемой задачи.

ДЕНОТАТ

Реальный объект, процесс, явление, ситуация и т.п. в физическом мире, для которого есть специальное именуемое выражение в некотором языке. Это именуемое выражение является десигнатом для данного денотата.

ДЕРЕВО ВЫВОДА

Представление процедуры логического вывода в виде дерева, вершинами которого являются или исходные формулы, или формулы, полученные в процессе вывода.

ДЕРЕВО ДВОИЧНОЕ

Представление процесса поиска в виде дерева, каждая вершина которого связана со значением ключа поиска таким образом, что все меньшие ключи сосредоточены в ее левом поддереве, а все большие - в правом.

ДЕРЕВО ЗАВИСИМОСТЕЙ

Представление результата работы этапа синтаксического анализа в лингвистических процессорах в виде дерева разбора предложения, в вершинах которого стоят лексемы, соответствующие подлежащему, сказуемому, дополнению и т.п., а дуги указывают на связь между вершинами по управлению. Используется на этапах глубинного синтаксического и семантического анализа предложения.

ДЕРЕВО РЕШЕНИЙ

Структура, состоящая из узлов принятия решений и альтернатив, соответствующих этим узлам. Движение по Д.Р. может осуществляться случайным образом или на основании локальной информации об успехе, которая имеется в узлах. В результате успешного поиска на Д.Р. образуется путь, ведущий из корня дерева исходная ситуация к тому узлу дерева, которое соответствует целевой ситуации. В процессе движения по Д.Р. часто возникает необходимость возврата в ранее пройденные узлы, что осуществляется с помощью процедуры бектрекинга.

ДЕРЕВО СОСТАВЛЯЮЩИХ

Представление системы составляющих, корнем которой является полная составляющая, а висячими узлами являются точечные составляющие.

ДЕРЕВО ЦЕЛЕЙ

Специальный вид дерева, в котором одна или несколько вершин соответствуют целям, а остальные вершины являются подцелями этих целей. Дуги показывают, как декомпозируются цели в подцелях.

ДЕСИГНАТ

Специальное именуемое выражение для денотатов, существующих во внешнем по отношению к данной системе мира. Все значения системы о денотатах фиксируются в виде знаний о Д. В ряде случаев Д. называют уникальным именем, меткой, ключом.

ДЕСКРИПТОР

Выделенное слово (или сочетание), которое для понимающей тексты на естественном языке системы служит маркером. Этот маркер входит в левые части правил вывода, секвенций или продукций. При появлении маркера соответствующее правило срабатывает. Д. выбирается из специального словаря и ему искусственно придается смысловая однозначность, позволяющая с его помощью обозначать класс (синонимичных) понятий.

ДЕФОЛТ

Наиболее типичное значение атрибута, приписываемое объекту, если его значение для этого объекта указано.

ДИЗЬЮНКТ

Выражение вида $B_1 \vee B_2 \vee \dots \vee B_n \leftarrow A_1 \& A_2 \& \dots \& A_m$, где ($\&$, \vee , \leftarrow) соответственно символы конъюнкции, дизъюнкции и импликации. Д. читается так: "Если A_1 , и A_2 , и, A_m , то B_1 , или B_2 , или ,..., B_n ". Правая или левая часть импликации может быть пустой. В этом случае Д. интерпретируется иначе. Для Д. вида $B_1 \vee B_2 \vee \dots \vee B_n$ интерпретация заключается в утверждении существования факта $B_1 \vee B_2 \vee \dots \vee B_n$; а для Д. вида $\leftarrow A_1 \& A_2 \& \dots \& A_m$ интерпретация состоит в утверждении, что $(A_1 \& A_2 \& \dots \& A_m)$. Если в Д. пусты левая и правая часть импликации, то он называется пустым. Д. используется в методе логического вывода, основанном на использовании резолюций, а также в языке программирования Пролог.

ДИЗЬЮНКТ БОКОВОЙ

Дизъюнкт, который является либо элементом исходного множества, либо некоторым дизъюнктом, предшествующим в выводе рассматриваемому.

ДИЗЪЮНКТ ПУСТОЙ

Пустое множество, возникающее при исчерпании исходного множества, дизъюнктов в случае успешного завершения процесса вывода, опирающегося на принцип резолюции.

ДИЗЪЮНКТ ХОРНА

Дизъюнкт, содержащий не более одной положительной литеры.

ДИЗЪЮНКЦИЯ

Логическая операция (связка) для $n > 2$ выражений. Результирующее выражение ложно только тогда, когда ложны исходные выражения. Для обозначения Д. стандартно используется знак \vee (реже +).

ДИССОНАНС КОГНИТИВНЫЙ

Диссонанс, возникающий у субъекта, когда он располагает одновременно двумя противоречивыми знаниями об одном и том же объекте, субъекте, ситуации или явлении. Д.К. есть побудительный мотив для того, чтобы начать некоторую деятельность по устранению противоречия в знаниях. В интеллектуальных системах Д.К. используется в базах знаний, чтобы сделать знания активными.

ДОКАЗАТЕЛЬСТВО КОНСТРУКТИВНОЕ

Вывод утверждения в логическом исчислении, который обеспечивает явное построение всех элементов, участвующих в выводе. Таковым не является, например, широко распространенный способ доказательства «от противного», используемый в классической математике.

ДОКАЗАТЕЛЬСТВО ТЕОРЕМЫ

Логическое следование данной формулы из данной совокупности выведенных ранее формул.

ДОСКА ОБЪЯВЛЕНИЙ

Способ управления параллельно протекающими асинхронными процессами решения задач, при которых информация о закончившихся процессах и полученных результатах «вывешивается» на Д.О., к которой имеют независимый доступ все процессы (и/или программисты), ожидающие нужных результатов. Д.О. часто используется в экспертных системах, интеллектуальных роботах и других интеллектуальных системах.

ЗАКОН ИСКЛЮЧЕННОГО ТРЕТЬЕГО

Один из базовых законов рассуждений, характерных для традиционных формальных систем. Он утверждает, что выражение $(A \vee \neg A)$ является тождественно истинным. З.И.Т. подвергался критике и отвергался логиками, стоящими на позиции интуиционистской математики и конструктивной математики.

ЗАКОН СНЯТИЯ ДВОЙНОГО ОТРИЦАНИЯ

Один из базовых законов рассуждений, характерных для традиционных формальных систем. Он утверждает, что всегда имеет место равенство $\overline{\overline{A}} = A$.

ЗНАНИЯ

Совокупность сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д.

ЗНАНИЯ ДЕКЛАРАТИВНЫЕ

Знания, которые записаны в памяти интеллектуальной системы так, что они непосредственно доступны для использования после обращения к соответствующему полю памяти. В виде З.Д. обычно записывается информация о свойствах предметной области, фактах, имеющих в ней место и т.п. информация. По форме представления З.Д. противопоставляются процедурным знаниям.

ЗНАНИЯ О ПРЕДМЕТНОЙ ОБЛАСТИ

Совокупность сведений о предметной области, хранящихся в базе знаний интеллектуальной системы. В З.П.О. входят факты, относящиеся к предметной области, закономерности, характерные для нее, гипотезы о возможных связях между явлениями, процессами и фактами в ней, процедуры для решения типовых задач в данной проблемной области. З.П.О. вводит в базу знаний инженер по знаниям. В процессе функционирования интеллектуальной системы З.П.О. могут пополняться. З.П.О. используют при поиске решений задач, возникающих в экспертных и других интеллектуальных системах.

ЗНАНИЯ ПРАГМАТИЧЕСКИЕ

1. Знания о способах решения задач в заданной предметной области.
2. В естественном языке знания о прагматической компоненте текстов. (См. также Модель пользователя, Фокус внимания).

ЗНАНИЯ ПРОЦЕДУРНЫЕ

Знания, хранящиеся в памяти интеллектуальной системы в виде описаний процедур, с помощью которых их можно получить. В виде З.П. обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики и т.п. информация. По форме представления З.П. противопоставляются декларативные знания.

ЗНАНИЯ ЭВРИСТИЧЕСКИЕ

Знания, накапливаемые интеллектуальной системой в процессе ее функционирования, а также знания, заложенные в ней априорно, но не имеющие статуса абсолютной истинности в данной проблемной области. Часто З.Э. связаны с отражением в базе знаний человеческого (неформального) опыта решения задач.

ЗНАНИЯ ЭКСПЕРТНЫЕ

Знания, которыми располагает специалист в некоторой предметной области.

ЗНАЧЕНИЕ АТТРИБУТА

Константа, приписанная атрибуту в базе данных.

ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ

Значение переменной, которое ей автоматически приписывается, если ее значение не задается.

ЗРЕНИЕ МАШИННОЕ

Совокупность моделей и методов для выполнения техническими системами процедур, характерных для зрительного восприятия у живых организмов. В рамках З.М. решаются задачи выделения объектов из фона, их идентификация, ввод в промежуточную память, перекодировка во внутренние представления и т.п. Системы З.М. характерны для интеллектуальных роботов и других интеллектуальных систем.

И/ИЛИ ГРАФ

Ориентированный граф, обладающий свойствами: 1). При возбуждении (передаче информации) входных дуг, ведущих в некоторую вершину, реализуется либо конъюнкция (И), либо дизъюнкция (ИЛИ). В первом случае вершина возбуждается (становится активной и принимает информацию) только тогда, когда возбуждены все дуги, входящие в нее. Во втором случае для возбуждения вершины достаточно возбуждения любой входящей в нее дуги. 2). При возбуждении вершины возбуждаются либо все выходящие из вершины дуги (И), либо только одна, выбираемая вершиной (исключающее ИЛИ для числа аргументов, равного числу выходящих дуг). Часто под И/ИЛИ г. понимают граф, для которого выполнено первое свойство, а для выходных дуг всегда имеет место И. И/ИЛИ г. широко используются в системах планирования целесообраз-

ного поведения автономных роботов и в других системах искусственного интеллекта.

ИДЕНТИФИКАЦИЯ

Процедура установления интересующих исследователя свойств в изучаемом явлении или объекте. Для интеллектуальных систем И. часто означает проверку того, что данная система действительно решает те задачи, для решения которых она была создана.

ИДЕНТИФИКАЦИЯ ЗНАНИЙ

Определение характеристик знаний, необходимых для решения задачи.

ИЗВЛЕЧЕНИЕ ЗНАНИЙ

Получение информации о предметной области от специалистов и выражение ее на языке представления знаний. И.З. используется при построении экспертной системы или базы знаний.

ИИ-ПРОГРАММИРОВАНИЕ

Разработка инструментального программного обеспечения для решения задач искусственного интеллекта. В ИИ-П. создаются языки программирования, ориентированные на особенности задач искусственного интеллекта, интеллектуальные вспомогательные средства, языки представления знаний и манипулирования ими, пустые экспертные системы и оболочки и другие инструментальные средства.

ИМИТАЦИЯ ИНТЕЛЛЕКТУАЛЬНОГО ПОВЕДЕНИЯ

Воспроизведение процедур формирования целенаправленного поведения человека и животных в внешнем мире в зависимости от возникающих в нем ситуаций. Для И.И.П. разрабатываются специальные модели и методы планирования деятельности. Имитация интеллектуального поведения широко используется в интеллектуальных роботах.

ИМИТАЦИЯ ПРОЦЕССОВ МЫШЛЕНИЯ

Воспроизведение программы путем или с помощью специальной аппаратуры отдельных процессов, характерных для мышления человека и животного (распознавание ситуаций, принятие решений о своем поведении, понимание текстов на естественном языке и т.п.). В искусственном интеллекте И.И.М., как правило, предполагает не идентичность процессов, протекающих в мозгу и в технической системе, а совпадение результатов решения одинаковых задач.

ИМПЛИКАЦИЯ

Логическая операция (связка) для двух выражений. Результирующее выражение ложно тогда, когда первое выражение истинно, а второе ложно (операция И. некоммукативна). Стандартное обозначение импликации: \rightarrow .

ИНДУКЦИЯ

Метод перехода от частных наблюдений к общей закономерности, которой удовлетворяют все частные наблюдения.

ИНДУКЦИЯ НЕПОЛНАЯ (ЭМПИРИЧЕСКАЯ)

Нахождение закономерностей, которым подчиняются все известные до этого момента наблюдения. Найденные закономерности могут опровергаться новым наблюдениями.

ИНДУКЦИЯ ПОЛНАЯ (МАТЕМАТИЧЕСКАЯ)

Математическое доказательство справедливости некоторой закономерности, основанное на выдвижении гипотезы по конечному числу фактов и обоснований к изменению этой закономерности.

ИНЖЕНЕР ПО ЗНАНИЯМ

Специалист, основной задачей которого является проектирование баз знаний и наполнение их знаниями о проблемной области. В процессе этой деятельности И.П.З. выбирает форму представления знаний, удоб-

ную для данной проблемной области, организует приобретение знаний из различных источников (официальные документы, учебники, монографии и т.п.), а также в результате общения с экспертами-специалистами в данной проблемной области.

ИНЖЕНЕРИЯ ЗНАНИЙ

Раздел искусственного интеллекта, в рамках которого решаются проблемы, связанные с извлечением знаний, приобретением знаний, представлением знаний и манипулированием знаниями. И.З. служит основой для создания экспертных систем и других интеллектуальных систем.

ИНТЕЛЛЕКТ ИСКУССТВЕННЫЙ

1. Научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного моделирования тех видов человеческой деятельности, которые традиционно считаются интеллектуальными. (См. также Представление знаний, Обучение, Общение, Объяснение).
2. Свойство интеллектуальных систем выполнять функции (творческие), которые традиционно считаются прерогативой человека.

ИНТЕРВЬЮ

Способ работы с экспертом при приобретении знаний, когда инженер по знаниям выступает в роли интервьюера.

ИНТЕРПРЕТАЦИЯ

В искусственном интеллекте - установление связи между двумя системами описаний, что позволяет понимать одну систему на уровне другой.

ИНТЕРФЕЙС ЕСТЕСТВЕННО-ЯЗЫКОВЫЙ

Совокупность программных и аппаратных средств, обеспечивающих общение интеллектуальной системы с пользователем на ограниченном рамках проблемной области естественном языке. В состав И.Е.Я. входят словари, отражающие словарный состав и лексику языка, а также

лингвистический процессор, осуществляющий анализ текстов (морфологический, синтаксический, семантический и прагматический) и синтез ответов пользователю.

ИНТЕРФЕЙС ИНТЕЛЛЕКТУАЛЬНЫЙ

Интерфейс, в который включены средства, позволяющие человеку вести общие с ЭВМ, не используя для ввода в ЭВМ специальные программы.

ИСКУССТВЕННЫЙ МОЗГ

Гипотетическое устройство, способное заменить мозг человека и (или) реализовать все функциональные свойства, известные о мозге. В искусственном интеллекте под И.М. понимают повторение искусственными средствами свойств, присущих мозгу.

ИСТОЧНИК ЗНАНИЙ

Текст, (инструкция, монография, фотография, кинолента и т.п.), наблюдение или сообщаящий нужную информацию специалисту-профессионалу. Из И.З. черпается информация, преобразуемая в знания, фиксируемые в памяти интеллектуальной системы.

ИСЧИСЛЕНИЕ

Формальная система, задаваемая четверкой $\langle T, V, A, P \rangle$, где T – множество базовых символов исчисления; V – синтаксические правила, с помощью которых из элементов T порождаются произвольные элементы; A – множество априорно истинных элементов исчисления (аксиомы исчисления); P – множество семантических правил (правил вывода), с помощью которых из одних элементов системы порождаются другие.

ИСЧИСЛЕНИЯ ВЫСКАЗЫВАНИЙ

См. Исчисление пропозициональное.

ИСЧИСЛЕНИЕ ГЕНЦЕНА

Исчисление, в котором аксиомы задаются в виде секвенций.

ИСЧИСЛЕНИЕ ЛОГИЧЕСКОЕ

Объект изучения в математической логике, в основе которого лежит понятие формальной системы. В искусственном интеллекте используются различные И.Л.: исчисления предикатов, пропозициональное исчисление, исчисление классов, исчисление отношений, многосортные и многозначные логики и т.п.

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ

Исчисление, в котором наряду с формулами исчисления высказываний используются формулы в которые могут входить отношения (предикаты), связывающие между собой группы элементов исчисления и кванторы общности и существования.

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА

Исчисление предикатов, в котором под знаком квантора не могут находиться символы предикатов. (См. также Квантор общности, Квантор существования.)

ИСЧИСЛЕНИЕ ПРОПОЗИЦИОНАЛЬНОЕ

Формальная система, базовыми элементами которой являются высказывания – нерасчлененные предложения, относительно которых в каждый данный момент можно утверждать, что они являются либо абсолютно истинными, либо абсолютно ложными. И.П. изучает связи между этими высказываниями, которые задаются логическими связками: отрицанием, дизъюнкцией, конъюнкцией, импликацией и др. И.П. является аксиоматической системой, и для классического И.П. все аксиомы являются тождественно истинными высказываниями, а правила вывода не меняют этого свойства. С помощью И.П. порождаются все тождественно истинные высказывания и только они.

ИСЧИСЛЕНИЕ СИТУАЦИОННОЕ

Исчисление предикатов, в котором все или некоторые предикаты снабжены метками, привязывающими их к тем или иным ситуациям. Каждая ситуация задается описанием, в котором участвуют внеситуационные выражения, и те, которые связаны с данной ситуацией. В качестве аксиом И.С. используются обычные аксиомы ситуаций и характеристик этих ситуаций в той проблемной области, для которой И.С. используется. (См. также Управление ситуационное.)

КАРТА КОГНИТИВНАЯ

Способ описания известного субъекту фрагмента пространства с находящимися в нем заполнителями. Существует ряд вариантов К.К., отличающихся друг от друга сложностью и подробностью, например, карта-обозрение и карта-путь. С помощью К.К. изучается то, как человек воспринимает пространственные ситуации и отображает их в своей памяти. В интеллектуальных системах К.К. используются для отображения пространственных ситуаций в базах знаний и при работе с экспертами-профессионалами, когда инженер по знаниям получает от них информацию, связанную с пространственными ситуациями.

КАУЗАЦИЯ

Установление связи явлений или фактов. В строгой форме К. устанавливает причинно-следственные связи между явлениями или фактами. В более широком смысле К. устанавливает влияние одних явлений или фактов на другие. В этом более широком смысле К. отражает в моделях знаний в виде каузальных сетей и сценариев. При узком понимании К. в тех же моделях приводит к причинно-следственным сетям.

КВАНТИФИКАТОР

В узком смысле – это указатель на область истинности некоторого утверждения. Примерами К. в этом смысле могут служить лексемы: «все-

гда», «почти никогда», «для многих», «примерно в половине случаев» и т.д. В формальных системах, как правило, используются два квантификатора, называемых квантором общности и квантором существования. Первому соответствует лексема «всегда» и «для всех», а второму – «существует». В широком смысле К. может означать любое значение лингвистической переменной (например, «много», «часто», «далеко» и т.д.). Именно в таком смысле К. используются в псевдофизической логике и в ситуационном управлении.

КВАНТИФИКАЦИЯ

Приписывание оценок (в том числе, числовых) на выражения формальной системы. Эти оценки иногда называют квантофикаторами. Оценки могут характеризовать степень правдоподобия выражений, приоритетность при решении задачи и т.п.

КВАНТОР ОБЩНОСТИ

Специальный указатель на то, что некоторое $P(x_i)$ содержащее переменные x_i , распространяется на все формулы, получаемые при подстановке вместо переменных, перечисленных в этом указателе, любых значений из областей определения этих переменных. К.О. обозначается как $\forall x_i$, где x_i имена тех переменных, на которые распространяется его действие (связанные переменные).

КВАНТОР СУЩЕСТВОВАНИЯ

Специальный указатель на то, что некоторое $P(x_i)$ имеет место (или истинно) при некоторых переменных, перечисленных в этом указателе, причем конкретные значения, обеспечивающие это, не указываются, а фиксируется лишь то, что они существуют. Переменные, перечисленные в указателе, называются связанными. Стандартно К.С. обозначается как $\exists x_i P(x_i)$, где x_i - имена переменных, которые являются связанными.

КЛАССИФИКАЦИЯ

Введение отношений на множестве объектов или явлений, позволяющих разбить их на классы с установлением между классами отношений включения типа «род – вид», «элемент – класс», «целое – часть» и т.п. См. также Таксономия, Кластеризация.

КЛАСТЕРИЗАЦИЯ

Способ разбиения объектов или явлений на классы на основании некоторого отношения близости в пространстве признаков. См. также Таксономия, Классификация.

КОНКАТЕНАЦИЯ

Операция приписывания одних элементов к другим так, что получается новый произвольный элемент. С помощью К., например, образуются из букв слова языка, из слов, знаков пунктуации и знака пробела – предложения.

КОНЦЕПТ

См. Понятие.

КОНЪЮНКЦИЯ

Логическая операция (связка) для $n > 2$ выражений. Результирующее выражение (конъюнкция исходных выражений) истинно только тогда, когда истинны все исходные выражения. Для обозначения К. стандартно используется знак & (реже), а также знак умножения в виде точки. Во многих случаях знак К. может быть опущен.

ЛИПС

Единица измерения производительности машины вывода (от англ. Logical Inference PerSecond), равная числу логических выводов, выполняемых в одну секунду. Как правило, для реализации одного логического вывода требуется от 10 до 100 команд ЭВМ.

ЛИТЕРА

Любая константа, переменная или ее отрицание.

ЛОГИКА

Наука о правильных способах рассуждений. В классическом варианте состоит из учения о понятиях, учения о суждениях и учения об умозаклучениях. В течение долгого времени с Л. связывалось учение Аристотеля о силлогистических умозаклучениях. Силлогистика была первой дедуктивной системой, возникшей в науке. В основе Л. лежит понятие аксиоматической системы. Сила чистой логики, отвлекающейся от семантики предметной области, состоит в общности ее методов и положений. Важно отметить, что Л. есть наука о мышлении в понятиях, а не о познании мира посредством мышления о понятиях. Это показывает, что в интеллектуальных системах чисто логические решатели задач не могут исчерпать весь запас средств, необходимых для воссоздания интеллектуальной деятельности. На основе Л. в конце XIX в. начала создаваться математическая логика, в основе которой лежит теоретико-множественные категории и понятие формальной системы.

ЛОГИКА ВЕРОЯТНОСТНАЯ

Логика, в которой формулы оцениваются значениями, интерпретируемыми как вероятности того, что данная формула принимает значение «Истина». С правилами вывода в Л.В. связываются процедуры, позволяющие вычислять вероятностную оценку истинности выводимой формулы по известным оценкам истинности для формул – посылок.

ЛОГИКА ВЕРЫ

Вид эпистимической логики, в которой все утверждения снабжаются квантификаторами, оценивающими степень правдоподобности этих утверждений.

ЛОГИКА ВРЕМЕННАЯ

Логика отношений, в которой отношения (предикаты) или специальные операторы характеризуют временные зависимости («раньше», «будет», «одновременно» и т.д.). Другим типом Л.В. являются так называемые логики, в которых один из аргументов предиката есть время (состояние, ситуация).

ЛОГИКА ВТОРОГО ПОРЯДКА

Формальная система, в которой допускается, что кванторы общности и существования могут связывать не только индивидуальные переменные, но и предикатные или иные функциональные символы.

ЛОГИКА ДВОИЧНАЯ

Логика, в которой в качестве истинностных значений выражений рассматриваются лишь два значения: 0 и 1, интерпретируемые как абсолютная ложь и абсолютная истина.

ЛОГИКА ДЕЙСТВИЙ

Система рассуждений о закономерностях действий в некоторой проблемной среде. Л.Д. опирается на временную логику и пространственную логику, а также на свойства конкретной среды. Л.Д. используется в интеллектуальных работах и экспертных системах. Для Л.Д. характерны немонотонные выводы.

ЛОГИКА ДИНАМИЧЕСКАЯ

Система рассуждений, в явной форме учитывающая динамику объектов, к которым прилагаются эти рассуждения. Если время входит в рассуждение в явной форме, то Л.Д. совпадает с одним из вариантов временной логики. Если динамика задается законами смены ситуаций, то Л.Д. превращается в ситуационное исчисление. Л.Д. используется для моделирования функционирования открытых систем, в частности, открытых баз

данных и баз знаний, а также во всех интеллектуальных системах, имеющих дело с динамической моделью мира.

ЛОГИКА ЗДРАВОВОГО СМЫСЛА

Совокупность рассуждений имеющая хождение в быту и отражающая систему ценностей, мотивы поступков и цели людей. В интеллектуальных системах Л.З.С. используется в тех случаях, когда при воспроизведении деятельности эксперта-профессионала нет возможности построить формальную систему, в которую можно было бы погрузить процедуры рассуждений этого эксперта.

ЛОГИКА ИНДУКТИВНАЯ

Формальная система, описывающая правила формирования общих утверждений на основе конечного множества частных утверждений. В Л.И. все утверждения взвешиваются оценками правдоподобности, характеризующими истинность этих утверждений.

ЛОГИКА ИНТУИЦИОНИСТСКАЯ

Логика, используемая в формальных системах, которые опираются не на классические конструкции, восходящие к теории множеств, а на умозрительные конструкции. В рассуждениях об этих конструкциях оказываются неприменимыми закон снятия двойного отрицания и закон исключенного третьего. Л.И. широко используется при доказательстве теорем на ЭВМ и в решателях интеллектуальных систем.

ЛОГИКА КАУЗАЛЬНАЯ

Логика, в которой отношения характеризуют типы связей, совпадающие с причинно-следственными или близкие к ним по содержанию.

ЛОГИКА КОМАНД

Логика, в которой в качестве операторов используются различные императивы. Близка к логике действий. Используется в интеллектуальных роботах и других интеллектуальных системах.

ЛОГИКА КОНСТРУКТИВНАЯ

Логика, в которой разрешены лишь конструктивные доказательства. () Л.К. лежат в основе конструктивной математики, тесно связанной с проблемами вычислимости на ЭВМ и других устройствах, имеющих ограниченную память.

ЛОГИКА МАТЕМАТИЧЕСКАЯ

Логика, основанная не на содержательной стороне высказываний, а на синтаксических категориях и их структурных (операционных) связях. В основе Л.М. лежит понятие формальной системы. Различные интерпретации формальной системы приводят к различным логическим исчислениям. Наиболее известным из которых являются пропозициональное исчисление (исчисление высказываний), исчисление предикатов, ситуационное исчисление, многозначные логики и т.п.

ЛОГИКА МНОГОЗНАЧНАЯ

Логика, в которой в качестве значений истинности переменных выступают натуральные числа $0, 1, \dots, k$.

ЛОГИКА МОНОТОННАЯ

Логика замкнутого мира, эквивалентная некоторой формальной системы. В Л.М. действует принцип монотонности: если на некотором шаге вывода получено утверждение, то его истинность на последующих шагах вывода не может изменяться.

ЛОГИКА НЕМОНОТОННАЯ

Логика открытого мира. В Л.Н. нарушается основной принцип монотонной логики. Если на некотором шаге вывода получено утверждение, то при поступлении в систему новой информации (новых фактов) истинность этого вывода может исчезнуть. Л.Н. характерны для большинства интеллектуальных систем, имеющих дело со сложными

предметными областями, для которых получить априорно исчерпывающее замкнутое описание не представляется возможным.

ЛОГИКА НЕЧЕТКАЯ

Логика, в которой используются нечетные квантификаторы, чаще всего нечеткие квантификаторы лингвистической переменной «частота»: «почти никогда», «почти всегда». Рассуждения с подобными квантификаторами требуют специальных приемов для нахождения квантификатора, который должен быть приписан заключению, когда посылки помечены определенными квантификаторами.

ЛОГИКА ПЕРВОГО ПОРЯДКА

Формальная система, в которой кванторы общности и существования могут связывать только индивидуальные переменные, но не могут связывать символы предикатов или иных функциональных символов.

ЛОГИКА ПРОПОЗИЦИОНАЛЬНАЯ

Логика, характерная для пропозиционального исчисления.

ЛОГИКА ПРОСТРАНСТВЕННАЯ

Формальная система, в которой использованы аксиомы, характерные для описания возможных расположений объектов в трехмерном (или двумерном) пространстве, расстояний между ними и локов. Л.П. позволяют проводить рассуждения о пространственном расположении и взаимосвязи объектов для случая абсолютной и относительной системы координат и для случая, когда такие переменные, как расстояние, размер лока или характеристики взаимного расположения предметов, заданы в виде лингвистических переменных. В Л.П. выделяют логику расстояний и логику взаимного положения предметов в метрическом и топологическом (размытом) вариантах.

ЛОГИКА ПСЕВДОФИЗИЧЕСКАЯ

Логика, отражающая восприятие субъектом или искусственной системой закономерностей внешней физической среды. Особенностью Л.П. является наличие размытых шкал, на которые проецируются объекты, с которыми имеет дело логика. Примерами Л.П. являются временная логика, пространственная логика, логика действий и др.

ЛОГИКА РАЗМЫТАЯ

См. Логика нечеткая.

ЛОГИКА РАССУЖДЕНИЙ ПО УМОЛЧАНИЮ

Рассуждения, в которых при отсутствии явной информации, необходимой для продолжения рассуждений, интеллектуальная система или человек обращаются к своей памяти и используют содержащуюся в ней информацию, предназначенную для тех случаев, когда нужная информация отсутствует. Введение механизма умолчаний приводит к тому, что Л.Р.У. становится немонотонной логикой. Л.Р.У. широко используется в открытых базах данных и базах знаний.

ЛОГИКА ЭПИСТЕМИОЛОГИЧЕСКАЯ

Формальная система, в которой используются операторы типа «знает», «хочет», «верит» и т.п.

ЛОК

Ограниченная часть пространства, в котором полностью помещается некоторый объект, чьи внешние границы совпадают с границами Л. Понятие Л. используется в пространственной логике.

«ЛЯМБДА» – ИСЧИСЛЕНИЕ

Исчисление, в котором используется операция функциональной абстракции (конверсии) λM , задающая функцию, значения которой для любого аргумента получаются подстановкой этого аргумента вместо x во все его

вхождения в М. Такие исчисления широко применяются в формальных моделях баз данных.

МАШИНА АБСТРАКТНАЯ

Теоретическая конструкция, в которой отражаются все формальные аспекты функционирования некоторого реального или гипотетического устройства. Примерами М.А. могут служить конечный автомат, машина Поста, машина Тьюринга и многие другие модели, изучаемые в математике, кибернетике, искусственном интеллекте и других науках.

МАШИНА БАЗ ДАННЫХ

Блок управления базой данных в информационных системах. Специализированный процессор с собственной памятью, выполняющий обработку запросов.

МАШИНА БАЗ ЗНАНИЙ

Блок управления базой знаний в машине пятого поколения. Специализированный процессор (система процессоров), выполняющий обработку запросов и формирование ответов в некоторой предметной области на основе использования совокупностей фактов и знаний о предметной области, представляемых в виде правил, а также механизмов вывода.

МАШИНА ВИРТУАЛЬНАЯ

Абстрактная машина (комплекс программных средств), с помощью которой для пользователя имитируется гипотетическая ЭВМ, обладающая практически неограниченной оперативной памятью и расширяемым набором команд. М.В. использует для имитации конечную оперативную память и базовый набор команд.

МАШИНА ПАРАЛЛЕЛЬНОГО ВЫВОДА

Специализированный процессор (система процессоров), реализующий параллельно основные операции, характерные для вывода на знаниях.

МАШИНА, УПРАВЛЯЕМАЯ ПОТОКОМ ДАННЫХ

См. Архитектура потоковая.

МЕТАЗНАНИЕ

Знание интеллектуальной системы о знаниях, которой хранятся в ее базе знаний, или о процедурах, которые можно совершать с хранящимися в базе знаний. Введение М. – процесс рекурсивный. М. в текстах на естественном языке может быть соответственно с фразами типа «Я знаю, что Иванов не умеет плавать» или «Сидоров предполагал, что Петров не знает алгебру».

МЕТАПРОДУКЦИЯ

Продукция, включаемая в систему продукций для указания порядка выполнения продукций, входящих в список готовых продукций.

МЕТАЯЗЫК

Язык для описания других языков. Чаще всего метаязык использует нотацию, в которой собственные символы описываемого языка являются терминальными символами метаязыка.

МЕТОД ИНТЕРВЬЮ

В инженерии знаний прием, с помощью которого добываются знания у экспертов-профессионалов. Инженер по знаниям выступает в роли репортера, берущего интервью. Он задает вопросы, цель которых уточнить сведения, сообщаемые экспертом относительно предметной области, в которой эксперт работает. Существуют специальные приемы, которые входят в стандартный М.И., делающий беседу целенаправленной и эффективной.

МЕТОД ОБРАТНОЙ ВОЛНЫ

См. Поиск нисходящий.

МЕТОД ПРЯМОЙ ВОЛНЫ

См. Поиск восходящий.

МЕХАНИЗМ ВЫВОДА

Совокупность правил вывода и стратегии управления выводом (применения этих правил). Крайним случаем М.Б. может быть произвольное применение правил вывода, как это делается в логических исчислениях.

МЕХАНИЗМ НАСЛЕДОВАНИЯ

Прием, используемый в базах знаний. Заключается в том, что на множестве информационных единиц вводятся классифицирующие отношения типа «класс-элемент», «род-вид» и т.п. При этом информация, относящаяся ко всем элементам класса или ко всем видам рода, содержится соответственно в описании класса или рода, а подчиненные им информационные единицы наследуют эту информацию, когда это необходимо.

МНОЖЕСТВО НЕЧЕТКОЕ

Множество, характеристическая функция которого может принимать значения из отрезка $[0, 1]$ Значение характеристической функции для некоторого элемента характеризует степень принадлежности этого элемента к множеству.

МОДЕЛЬ

Объект (реальный, знакомый или воображаемый), отличный от исходного, но способный заменить его и в рамках решаемых задач.

МОДЕЛЬ АССОЦИАТИВНАЯ

Модель процесса решения задачи человеком, опирающаяся на процедуру установления сходства данной задачи (или составляющих ее подзадач) с задачами, решение которых уже известно.

МОДЕЛЬ ВЫЧИСЛИТЕЛЬНАЯ

Описание процедур решения задач в некоторой предметной области. В М.В. задается полная структура функциональных связей для элементов предметной области, связанных между собой соотношениями, позволяющими находить значения одних элементов через другие. Задание ис-

ходных целевых элементов приводит к поиску в М.В. путей, ведущих от исходных элементов к целевым. Если хотя бы один такой путь существует, то по нему строится программа решения поставленной задачи. М.В. обеспечивают автоматический синтез программ.

МОДЕЛЬ ЗАМКНУТАЯ

Модель, остающаяся неизменной при работе с ней. В процессе функционирования интеллектуальной системы М.З. в отличие от открытой модели нельзя добавлять новые факты и закономерности. Все утверждения, полученные в М.З. окончательны и абсолютны.

МОДЕЛЬ ЗНАНИЙ

Описание знаний в базе знаний. Известны четыре типа М.З.: логические, в основе которых лежит формальная модель; сетевые, в основе которых лежит семантическая сеть; фреймовые, основанные на фреймах; продукционные, основанные на продукциях. Каждая такая М.З. определяет форму представления знаний.

МОДЕЛЬ КОГНИТИВНАЯ

Гипотетическая модель, описывающая устройство когнитивной структуры (структуры знаний у человека). Для интеллектуальных систем М.К. совпадает с моделью знаний.

МОДЕЛЬ КОНЦЕПТУАЛЬНАЯ

Модель предметной области из перечня всех понятий, используемых для описания этой области, вместе с их свойствами и характеристиками, классификаций этих понятий по типам, ситуациям, признакам в данной области и законами функционирования процессов, протекающих в ней. М.К. строится при погружении описания предметной области в базу знаний интеллектуальной системы.

МОДЕЛЬ ЛАБИРИНТНАЯ

Модель, в рамках которой процесс решения задач человеком объясняется аналогией с движением по лабиринту. Площадки лабиринта соответствуют промежуточным результатам (часть площадок отмечена как целевые), а передвижение от площадки к площадке происходит за счет использования преобразований из заданного набора. В М.Л. решение задачи – это поиск пути от начальной площадки лабиринта к одной из целевых. При этом лабиринт считается полностью заданным. (См. также Модель реляционная.)

МОДЕЛЬ ЛИНГВИСТИЧЕСКАЯ

1. Модель, относящаяся к фиксации тех или иных знаний о естественном языке.
2. Описание объекта в терминах лингвистических переменных и рассуждений о них.

МОДЕЛЬ ЛОГИЧЕСКАЯ

Модель представления знаний, в основе которой лежит формальная система.

МОДЕЛЬ МИРА

Способ отображения в памяти интеллектуальной системы знаний о внешней среде. (См. также Модель знаний, Схема концептуальная.)

МОДЕЛЬ ОБУЧЕНИЯ

Модель, лежащая в основе процесса обучения человека или технического устройства. Различают два типа М.О. – дескриптивный и нормативный. Дескриптивный М.О. извлекается из описания процесса деятельности, которой человек или система должны обучаться. Это извлечение может происходить разными способами. Наиболее известный из них основан на процедуре обучения на примерах. Нормативная М.О. задается

заранее. Часто обучение нормативного типа называют обучением с учителем.

МОДЕЛЬ ОБЩЕНИЯ

Описание совокупности знаний о том, как организуется общение между пользователем и интеллектуальной системой. Обычно в М.О. входит модель пользователя и модель течения диалога. Если общение происходит на профессиональном естественном языке, то для построения модели пользователя используются результаты, полученные в теории речевых актов. В других случаях могут быть применены процедуры обмена графической информации через экран дисплея.

МОДЕЛЬ ОТКРЫТАЯ

Модель, в которую в процессе функционирования интеллектуальной системы можно добавить новые факты и закономерности.

МОДЕЛЬ ПОВЕДЕНИЯ

Модель (техническая или программная), воспроизводящая некоторые виды поведения объектов при определенных условиях внешней среды (преодоление препятствий, реакция на внешние воздействия, выбор решений и пр.). М.П. используется как при изучении реального поведения биологических систем и человека, так и при разработке интеллектуальных роботов (планирования их автоматного поведения).

МОДЕЛЬ ПОЛЬЗОВАТЕЛЯ

Совокупность знаний об особенностях работы пользователя с системой, его намерениях, целях и требованиях, которая хранится в памяти интеллектуальной системы. М.П. помогает системе организовать эффективный диалог с пользователем, создает ему психологический комфорт.

МОДЕЛЬ РЕЛЯЦИОННАЯ

Модель описания данных, в которой все отношения задаются строками таблиц, столбцы которых помечены именами атрибутов. Табличное

представление данных оказывается часто удобным. Это обеспечило широкое распространение реляционных баз данных. В основе М.Р. лежит специальное исчисление предикатов.

МОДЕЛЬ СЕТЕВАЯ

Модель представления знаний, в основе которой лежит семантическая сеть.

МОДЕЛЬ СИТУАЦИЙ

Классификационная модель, которая позволяет опознавать текущие ситуации как известные системы. М.С. используется, например, в ситуационном управлении.

МОДЕЛЬ СТИМУЛ-РЕАКЦИЯ

Модель поведения, опирающаяся на принцип черного ящика. В М.С.Р. рассматриваются конечное множество стимулов, которые могут восприниматься субъектом или подаваться на выход искусственной системы, и правила соотнесения этим стимулам реакций, выдаваемых субъектом или системой. Внутренние процессы, связывающие стимулы и реакции, не анализируются и не учитываются. М.С.Р. находит применение в интеллектуальных системах на уровне воспроизведения простейших поведенческих реакций на раздражения, поступающие из внешней среды.

МОДЕЛЬ ФОРМАЛЬНАЯ

Формальное описание на некотором логическом языке структуры объекта. (См. также Система формальная.)

МОДЕЛЬ ЯЗЫКА

В лингвистике – формализованное представление знаний о языке. Как правило, включает морфологический, синтаксический, семантический и прагматический компоненты, которые также могут разделяться на более дробные компоненты.

МОДУС ПОНЕНС

Правило вывода в логике, которое утверждает: «Если выведены A и $A \rightarrow B$, то B выводимо».

МОДУС ТОЛЛЕНС

Правило вывода в логике, которое утверждает: «Если верно A и BA , то верно B ». Это правило используется в методе резолюций.

МОНОТОННОСТЬ ПРИ ВЫВОДЕ

Свойство, характерное для вывода в замкнутой формальной системе и в закрытой базе знаний и состоящее в том, что ранее выведенные утверждения не теряют истинности при расширении множества посылок для вывода.

НАСЛЕДОВАНИЕ

Свойство, используемое в базах данных и знаний и заключающееся в том, что если две информационные единицы соединены между собой отношениями типа «род-вид» или «класс-элемент», то информация, общая для всех видов, входящих в род, или для всех элементов, входящих в класс, содержится в информационной единице более высокого уровня и при необходимости наследуется единицей более низкого уровня. Н. позволяет ликвидировать дублирование в хранении информации в базах данных и знаний.

НЕЙРОБИОНИКА

Направление в исследованиях по искусственному интеллекту для которого характерно использование для воспроизведения в интеллектуальных системах процессоров, присущих биологическим объектам, структур и функций, аналогичных структурам и функциям этих объектов. В рамках этого направления были созданы формальные модели нейронов, на основе которых строятся сети, позволяющие решать задачи распознавания образов, классификации, стимул-реактивного пове-

дения и т.п. Усложнение структур формальных нейронов приводит к структурам, обладающим широкими функциональными возможностями. Их часто называют нейрокомпьютерами. Примерами нейробионического устройства может служить перцептрон.

НЕЙРОКОМПЬЮТЕР

См. ЭВМ нейробионические.

НЕЙРОН ФОРМАЛЬНЫЙ

Элемент, работа которого описывается функцией:

$$y = \begin{cases} 0, & \text{при } \sum A_i X_i - \sum B_i Z_i < h \\ 1, & \text{при } \sum A_i X_i - \sum B_i Z_i \geq h \end{cases}$$

Здесь y – двоичный выход; X_i – разрешающие двоичные входы; Z_i – запрещающие двоичные входы; a_i и b_i – весовые коэффициенты; h – порог. Варьируя значения весовых коэффициентов и порога, можно с помощью Н.Ф. реализовать любую булеву функцию. В Н.Ф. входы ассоциируются с синапсами нейрона, а выход – с его аксоном. Н.Ф. функционирует не так, как биологический нейрон (не учитывается время релаксации, латентный период, который всегда наступает после срабатывания нейрона и в течение которого он не может воспринимать входных сигналов). Но именно такая модель используется при конструировании многих устройств, разрабатываемых в нейробионике (например, перцептронов).

НЕМОНОТОННОСТЬ ПРИ ВЫВОДЕ

Свойство, характерное для вывода в открытой формальной системе и в открытой базе знаний и состоящее в том, что ранее выведенные утверждения могут перестать быть выводимыми при появлении новых фактов.

НЕОПРЕДЕЛЕННОСТЬ

Свойство интерпретации выражений, когда им приписываются оценки правдоподобия, отличные от абсолютной истины и лжи. Работа с такими выражениями требует специальных средств пересчета оценок правдоподобия. При логическом выводе, когда имеется неопределенность, используются либо многозначные логики, либо правдоподобные рассуждения.

НЕОПРЕДЕЛЕННОСТЬ ЛИНГВИСТИЧЕСКАЯ

Неопределенность, возникающая из-за расплывчатости и неоднозначности словесных выражений. При описании качественных знаний приходится применять специальные приемы для устранения Н.Л. (См. также Множество нечеткое, Функция принадлежности, Переменная лингвистическая.)

НЕПОЛНОТА

Свойство описания предметной области, заключающееся в том, что это описание не может быть преобразовано в формальную систему. При работе с неполной информацией используются правдоподобные рассуждения или рассуждения по умолчанию.

НЕРАЗРЕШИМОСТЬ АЛГОРИТМИЧЕСКАЯ

Ситуация, при которой для множества однотипных задач нельзя найти общего алгоритма, решающего их, хотя для подмножеств этого множества можно построить специфические алгоритмы поиска решения. Существование таких алгоритмически неразрешимых проблем строго доказано.

ОБЛАСТЬ ПРЕДМЕТНАЯ

Совокупность реальных или абстрактных объектов (сущностей), связей и отношении между этими объектами, а также процедур преобразования этих объектов для решения задач возникающих в О.П.

ОБЛАСТЬ ПРЕДМЕТНАЯ, ПЛОХО СТРУКТУРИРОВАННАЯ

Предметная область, концептуальная модель которой не может быть погружена в формальную систему или совокупность формальных систем. Большинство предметных областей, с которыми приходится иметь дело в интеллектуальных системах, являются плохо структурированными.

ОБЛАСТЬ ПРЕДМЕТНАЯ, ХОРОШО СТРУКТУРИРОВАННАЯ

Предметная область, концептуальная модель которой может быть погружена в формальную систему.

ОБЛАСТЬ ПРОБЛЕМНАЯ

См. Область предметная.

ОБОБЩЕНИЕ ЗНАНИЙ

Совокупность приемов и методов, позволяющих в базах знаний вводить новые знания, получаемые из имеющихся за счет кластеризации, введения гиперсобытий и гипотез.

ОБОБЩЕНИЕ ИНДУКТИВНОЕ

Процесс выработки гипотезы, с помощью которой описывается общая закономерность, связывающая воедино разрозненные знания, выступающие как частные случаи этой закономерности.

ОБОЛОЧКА

Инструментальное средство для проектирования и создания экспертных систем. В состав О. входят средства проектирования баз знаний с различными формами представления знаний и выбора режима работы решателя задач. Для конкретной предметной области инженер по знаниям определяет нужное представление знаний и стратегии решения задач, а затем, вводя их в О., создает конкретную экспертную систему.

ОБОСНОВАНИЕ

Одна из функций интеллектуальной системы, заключающаяся в доказательстве или проверке того, что полученное системой решение не проти-

воречит знаниям, которые хранятся в памяти системы. Таким образом, О. является относительным. При изменении содержимого базы знаний или базы данных. О. может либо сохранить свою силу, либо стать неверным. Обычно О. тесно связано с объяснением. К О. близко понятие аргументации.

ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

Совокупность процессов анализа текстов на естественном языке, их понимания и синтеза текстов. В процессе анализа в наиболее развитых системах обработки естественно-языковых сообщений происходит морфологический, синтаксический и семантический анализ текста, в результате чего выявляется глубинная структура текста, которая переводится во внутреннее представление, используемое в базе знаний интеллектуальной системы. Соотнесение этой структуры с теми знаниями, которые хранятся в системе, позволяет понять смысл исходного текста. При синтезе текстов сначала формируется семантическая структура текста, которая затем наполняется лингвистическими единицами с учетом синтаксиса и морфологии выбранного естественного языка. С О.Е.Я. связано решение задач машинного перевода, автоматического реферирования, общения с пользователем на ограниченном профессиональном естественном языке и т. п.

ОБРАБОТКА ИЗОБРАЖЕНИЙ

Процесс, связанный с обработкой визуальной информации (изменение масштабов, выделение контуров, распознавание видимых и невидимых частей изображения и т. п.).

ОБРАЗ

Изображение типичного или обобщенного представителя некоторого класса объектов.

ОБРАЗЕЦ

Фрагмент знаний, по которому осуществляется поиск по образцу, или эталон, по которому происходит классификация изображений, ситуаций, правил и т. п.

ОБУЧЕНИЕ

Усвоение знаний, умений и навыков путем или получения и восприятия информации от учителя или обработки наблюдаемой информации с последующим построением на основе этих наблюдений новых общих правил и закономерностей. Обе формы О. используются в интеллектуальных системах для приобретения новых знаний.

ОБУЧЕНИЕ НА ПРИМЕРАХ

Вид обучения, при котором индивиду или интеллектуальной системе предъявляется набор положительных и отрицательных примеров, связанных с какой-либо заранее неизвестной закономерностью. В интеллектуальных системах вырабатываются решающие правила, с помощью которых происходит разделение множества примеров на положительные и отрицательные. Качество разделения, как правило, проверяется экзаменационной выборкой примеров. Если качество разделения на экзаменационной выборке оказывается удовлетворительным, то выработанные решающие правила принимаются системой как окончательные. Если экзамен оказался неудовлетворительным, то экзаменационная выборка добавляется к обучающей и строятся новые решающие правила. После этого процесс экзамена повторяется.

ОБЪЕДИНЕНИЕ СВИДЕТЕЛЬСТВ

Процедура объединения в обобщенную гипотезу ряда гипотез, снабженных своими коэффициентами правдоподобия.

ОБЪЯСНЕНИЕ

Одна из функций интеллектуальной системы. О. предоставляет пользователю информацию о том, как интеллектуальная система получила выданное пользователю решение. В отличие от обоснования О. опирается лишь на тот маршрут, который сохранился в памяти системы от процесса поиска решения. Используя этот маршрут, интеллектуальная система формирует пользователю О. на профессиональном естественном языке, позволяющее ему представить все принципиальные шаги решения.

ОГРАНИЧЕНИЕ ЦЕЛОСТНОСТИ

Ограничения, налагаемые на совокупность информационных единиц, хранящихся в базах данных и базах знаний. Эти ограничения должны выполняться в любых состояниях, которые определяются текущим содержанием базы знаний и базы данных.

ОПРАВДАНИЕ

Одна из функций интеллектуальной системы. С помощью О. некоторое решение системы обосновывается не путем логических рассуждений или обращения к имеющимся в системе знаниям, а путем обращения к имеющейся в системе ценностной структуре. О. убеждает в том, что данное решение не противоречит этой ценностной структуре. (См. также Объяснение, Обоснование.)

ОТЛАДКА БАЗЫ ЗНАНИЙ

Поиск ошибок в базе знаний интеллектуальной системы. Различают синтаксическую отладку и семантическую отладку базы знаний.

ОТЛАДКА СЕМАНТИЧЕСКАЯ

Нахождение в программе смысловых ошибок. О.С. осуществляется путем выполнения на ЭВМ тестируемой программы с такими исходными данными, для которых правильное решение заранее известно.

ОТЛАДКА СИНТАКСИЧЕСКАЯ

Идентификация ошибок в программе на языке представления знаний, осуществляемая автоматически синтаксическим анализатором.

ОТНОШЕНИЕ

Задание на множестве M декартова произведения $M' * M''$. Пары, входящие в $M' * M''$, являются элементами O , а совокупность этих пар образует график O или его экстенционал. O может обладать рядом внутренних (рефлексивностью, симметричностью и т. п.) и некоторой внешней семантикой, связанной с его именем. Вся эта информация образует семантику O или его интенционал.

ОТНОШЕНИЕ АНТИРЕФЛЕКСИВНОЕ

Термин объясняется в статье *Отношение рефлексивное*.

ОТНОШЕНИЕ АНТИСИММЕТРИЧНОЕ

Термин объясняется в статье *Отношение симметричное*.

ОТНОШЕНИЕ АНТИТРАНЗИТИВНОЕ

Термин объясняется в статье *Отношение транзитивное*.

ОТНОШЕНИЕ ВИРТУАЛЬНОЕ

Отношение в явной форме не присутствующее в базе знаний, но определяемое из тех отношений, которые имеются в базе.

ОТНОШЕНИЕ ВРЕМЕННОЕ

Отношение, с помощью которого описываются связи факторов, событий и явлений во времени. Примерами $O.V.$ могут служить «быть раньше», «одновременно», «заканчиваться одновременно» и т.п. $O.V.$ используются во временных логиках, логиках действий и других псевдофизических логиках.

ОТНОШЕНИЯ ДЕЙСТВИЯ

Отношения, с помощью которого описываются действия, происходящие в реальном мире. Примерами $O.D.$ могут служить: «двигаться к», «при-

ближаться», «воздействовать» и т. п. О.Д. используются в логиках действий, являющихся видом псевдофизических логик.

ОТНОШЕНИЕ ИНТЕНСИОНАЛЬНОЕ

Отношение на множестве утверждений, относящихся к интенциональной составляющей знаний о проблемах области.

ОТНОШЕНИЕ КАУЗАЛЬНОЕ

Отношение, с помощью которого описываются различные виды причинно-следственных связей между объектами.

ОТНОШЕНИЕ МОДЕЛИРОВАНИЯ НЕЧЕТКОЕ

Отношение, задаваемое специальной коммутативной диаграммой, связывающей элементы и операции одного множества с элементами и операциями другого множества с помощью операторов, характерных для размытой логики.

ОТНОШЕНИЕ НЕРЕФЛЕКСИВНОЕ

Термин объясняется в статье *Отношение рефлексивное*.

ОТНОШЕНИЕ НЕСИММЕТРИЧНОЕ

Термин объясняется в статье *Отношение симметричное*.

ОТНОШЕНИЕ НЕТРАНЗИТИВНОЕ

Термин объясняется в статье *Отношение транзитивное*.

ОТНОШЕНИЕ ПРОСТРАНСТВЕННОЕ

Отношение, с помощью которого описываются пространственные взаимосвязи объектов. Примерами О.П. могут служить: «находиться близко», «соприкасаться», «быть внутри» и т. п. О.П. используются в пространственных логиках, логиках действия и других псевдофизических логиках.

ОТНОШЕНИЕ РЕЛЕВАНТНОСТИ

Связь двух или более информационных единиц, устанавливаемая на основе их семантической близости.

ОТНОШЕНИЕ РЕФЛЕКСИВНОЕ

Отношение, обладающее тем свойством, что любой его элемент сам с собой всегда находится в этом отношении. Примерами О.Р. могут служить: «совпадать», «быть похожим» и т.п. Если свойство рефлексивности не выполняется хотя бы для одного элемента отношения, то оно называется нерефлексивным отношением, а если оно не имеет места ни для одного элемента – антирефлексивным отношением.

ОТНОШЕНИЕ СЕМАНТИЧЕСКОЕ

Используемое в базах знаний отношение семантика которого задается его именем.

Отношение, обладающее тем свойством, что для любой пары (А, В) элементов, находящихся в этом отношении, справедливо утверждение, что пара (В, А) также находится в этом отношении. Примером может служить отношение «быть супругом» для мужей и жен. Отношение, для которого это свойство не выполняется хотя бы для одной пары, называется несимметричным, а если оно не выполняется ни для одной пары – антисимметричным.

ОТНОШЕНИЕ ТОЛЕРАНТНОСТИ

Рефлексивное, симметричное и нетранзитивное отношение. Такое отношение может толковаться как отношение сходства. В отличие от отношения эквивалентности, дающего разбиение множества элементов, на котором оно определено, на непересекающиеся подмножества, О.Т. дает покрытие этого множества. О.Т. используется при классификациях информации в базах знаний.

ОТНОШЕНИЕ ТРАНЗИТИВНОЕ

Отношение, для которого из того, что пары (А, В) и (В, С) находятся в этом отношении, всегда следует, что в этом же отношении находится пара (А, С). Если это свойство не выполняется хотя бы для одной тройки

элементов, то отношение называется нетранзитивным, а если не выполняется ни для какой тройки элементов, то – антитранзитивным. Примером транзитивного отношения может служить отношение «быть старше».

ОТНОШЕНИЕ ФУНКЦИОНАЛЬНОЕ

Отношение, с помощью которого в базе знаний задаются связи между информационными единицами. Эти связи определяют процедуры нахождения (вычисления) одних единиц через другие.

ОТНОШЕНИЕ ЭКВИВАЛЕНТНОСТИ

Симметричное, рефлексивное и транзитивное отношение. Используется для классификации множеств элементов путем разделения его на непесекающиеся классы, в совокупности покрывающие все исходное множество.

ОТНОШЕНИЕ ЭКСТЕНСИОНАЛЬНОЕ

Отношение на множестве конкретных фактов, хранящихся в базе данных. (См. также Представление экстенциональное.)

ОТРИЦАНИЕ

Одноместная логическая операция, обозначаемая \bar{a} . Для двузначной логики она определяется следующим образом: если a является истинным, то \bar{a} является ложным, а если a является ложным, то \bar{a} является истинным. В многозначной логике имеется несколько видов отрицания. Прямое обобщение двузначного отрицания для k -значной логики выглядит как $\bar{a} = (k - a) \bmod k$.

ОТРИЦАНИЕ ЛОГИЧЕСКОЕ

См. Отрицание.

ПАМЯТЬ АССОЦИАТИВНАЯ

Память, ориентированная на поиск хранящейся информации по содержанию. Организована с использованием признаков (тегов), связывающих

данные (информацию) по их содержанию, в отличие от обычной памяти, информация в которой отыскивается по номеру ячейки, в которой она хранится. В П.А. используется поиск по образцу.

ПЕРЕМЕННАЯ ЛИНГВИСТИЧЕСКАЯ

Переменная, которая в качестве своих значений использует слова и словосочетания, являющиеся характеристиками какого-либо явления. Например П.Л. с именем «длина» может принимать, следующие значения: «крошечная», «очень маленькая», «средняя», «большая», «очень большая». П.Л. используются при формализации качественной информации при ее вводе в базу знаний. В нечеткой логике значения П.Л. могут выступать как специальные квантификаторы.

ПЕРЕМЕННАЯ ПРОПОЗИЦИОНАЛЬНАЯ

Переменная, используемая в пропозициональной логике.

ПЕРЕМЕННАЯ СВЯЗАННАЯ

Переменная в логике, стоящая в зоне действий квантора общности или квантора существования.

ПЕРЦЕПТРОН

Устройство, построенное по нейробионическому принципу. Простейший трехслойный П. состоит из поля фоторецепторов, каждый из которых может находиться в двух состояниях, поля ассоциативных формальных нейронов и решателей. На поле фоторецепторов проецируется черно-белые изображения. Фоторецепторы случайным образом соединяются с разрешающими и запрещающими входами ассоциативных формальных нейронов, выходы которых также случайным образом соединяются с входами решателей. В процессе обучения классификации входных изображений происходит подбор весовых коэффициентов и порогов ассоциативных нейронов таким образом, чтобы вход решателей (на них суммируются сигналы, приходящие от ассоциативных нейронов) могли ис-

пользоваться как источник разделения множества входных изображений. Кроме трехслойного П. исследовались и многослойные П., у которых вводилось несколько слоев ассоциативных нейронов. Как показали исследования, возможности П. в области классификации – ограниченные. В настоящее время интерес к П. практически угас.

ПЛАНИРОВАНИЕ

Процесс составления последовательности действий, подзадач, операций, подцелей, поочередное выполнение которых должно привести в достижению целей, поставленных перед системой. В интеллектуальных системах П. может осуществляться либо в пространстве задач, либо в пространстве состояний.

ПЛАНИРОВАНИЕ ИЕРАРХИЧЕСКОЕ

Планирование, при котором сначала ищется приблизительный план, с помощью которого находится принципиальный ответ о достижимости поставленных целей. Затем этот план постепенно уточняется и доводится до уровня, когда он может однозначно реализоваться.

ПЛАНИРОВАНИЕ РАСПРЕДЕЛЕННОЕ

Планирование, при котором отдельные части плана формируются в разных местах и различными средствами, а затем объединяются центральным органом. П.Р. встречается в интеллектуальных работах и других интеллектуальных системах, когда они включены в систему сбора, передачи и обработки данных.

ПЛАНИРОВАНИЕ СТРАТЕГИЧЕСКОЕ

Построение начального плана действий в иерархических системах планирования. Стратегический план учитывает только основные сведения о среде планирования, но не учитывает особенности конкретной ситуации, в которой происходит планирование. Используется затем на нижележащих уровнях.

ПЛАНИРОВАНИЕ ТАКТИЧЕСКОЕ

Построение плана действий в иерархических системах планирования. При П.Т. в план, получаемый на уровне стратегического планирования, вносятся коррективы, связанные с учетом конкретных особенностей текущей ситуации и состояния планирующей системы. Уровней П.Т. может быть несколько.

ПЛАНИРОВЩИК

Комплекс программных средств, предназначенных для поиска планов действий.

ПОДХОД БАЙЕСОВСКИЙ

Метод принятия оптимальных статистических решений, основанный на положении, что параметр распределения вероятностей наблюдаемого случайного события, влияющий на характер принимаемых решений, является случайной величиной, определенной априорным распределением. П.Б. минимизирует средний риск, т.е. математическое ожидание теперь, связанных с неправильными или неточными решениями. П.Б. используется в теории статистических решений, теории игр, теории распознавания образов и для правдоподобного вывода в интеллектуальных системах.

ПОИСК

Движение в структурированном пространстве от одних узлов этого пространства к другим. Если П. является целенаправленным, то задано множество начальных узлов, с которых П. может начинаться, и множество конечных (целевых) узлов, при достижении которых П. прекращается. Движение по структуре поискового пространства определяется стратегическими П. Среди них наиболее распространены восходящий поиск и поиск нисходящий, а также поиск в глубину и поиск в ширину. Если

пространство П. не структурировано, то единственной возможностью для П. является случайный поиск.

ПОИСК АССОЦИАТИВНЫЙ

Поиск по образцу в базах знаний. (См. также Память ассоциативная).

ПОИСК В ГЛУБИНУ

Поиск при котором движение по структуре поиска происходит вдоль одного пути до конца. При неудаче происходит просмотр другого пути. Для возвращения в ближайшую точку ветвления используется процедура бектрекинга.

ПОИСК В ПРОСТРАНСТВЕ ЗАДАЧ

Нисходящий поиск, когда структура поиска задается декомпозицией крупных задач на более мелкие вплоть до задач, решение которых системе известно. Задача, подлежащая решению, рассматривается как целевой узел, м от него идет нисходящий поиск к множеству задач, решение которых системе известно. П.П.З. используется в интеллектуальных системах при планировании деятельности.

ПОИСК В ПРОСТРАНСТВЕ СОСТОЯНИЙ

Нисходящий или восходящий поиск, при котором структура пространства поиска задана множеством состояний некоторой системы, между которыми указаны возможные переходы. П.П.С. используется в интеллектуальных системах при планировании деятельности.

ПОИСК В ШИРИНУ

Поиск при котором движение по структуре поиска происходит на определенную глубину по всем направлениям, возможным из данной точки поиска. Если все продвижения оказались неуспешными, то либо увеличивается глубина поиска, либо сужается фронт поиска и происходит дальнейшее продвижение по всему этому фронту еще на заданное число

шагов. Сужение фронта приводит к тому, что П.Ш. комбинируется с поиском в глубину.

ПОИСК ВОСХОДЯЩИЙ

Поиск при котором движение по структуре поиска идет от начальных заданных узлов к целевым узлам. При этом могут использоваться поиск в глубину, поиск в ширину, или их комбинация. В логических системах аналогом П.В. является прямой вывод. Другое название П.В. - метод прямой волны.

ПОИСК НИСХОДЯЩИЙ

Поиск, при котором движение по структуре поиска происходит от целевых узлов к заданным узлам. П.Н. может использовать поиск в глубину, поиск в ширину или их комбинацию. В логических системах аналогом П.Н. является обратный вывод. Другое название П.Н. - метод обратной волны.

ПОИСК ПО ОБРАЗЦУ

Поиск фрагмента знаний в базе знаний на основании заданного образца. Образец может представлять собой полностью определенный фрагмент, или содержать свободные переменные. Например, при представлении в виде семантической сети, образец первого типа может выглядеть как «Иванов - Родиться – 1965», что означает прямой запрос к базе знаний: «Верно ли, что Иванов родился в 1965 году?». Образец второго типа: «X - Родится – 1965». Он интерпретируется так: «Назовите тех, кто родился в 1965 году». Запросы второго типа могут с помощью специального указателя интерпретироваться как выдача любого ответа, касающегося одного субъекта X, родившегося в 1965 г., или как выдача всех X, характеризующихся этим свойством. П.О. является основной процедурой для поиска информации в базах знаний.

ПОИСК ПО ПРИНЦИПУ «СПЕРВА ЛУЧШЕ»

Поиск, при котором в каждом узле пространства поиска выбирается то продолжение, которое имеет лучшую для данного узла локальную оценку успеха. Таким образом, П.П.С.Л. является разновидностью поиска в глубину.

ПОИСК СЛУЧАЙНЫЙ

Поиск в неструктурированном пространстве поиска. При П.С. с помощью некоторого вероятностного распределения выбираются узлы пространства и проверяются, не являются ли они целевыми. При достижении первого целевого узла П.С. прекращается.

ПОНИМАНИЕ ЕСТЕСТВЕННОГО ЯЗЫКА

В искусственном интеллекте – совокупность моделей и процедур, с помощью которых в интеллектуальных системах происходит соотнесение поступающего текста на естественном языке с фрагментами знаний из базы знаний, а также процедур, позволяющих выводить из имеющихся знаний другие, необходимые для правильной интерпретации содержимого введенного текста.

ПОНЯТИЕ

Имя, присваиваемое классу сущностей, объединяемых благодаря общности из призначных структур. В логике П. являются строго определенными и неизменными образованиями, характеризующими лишь призначной структурой, имманентно присущей всем понятиям. В искусственном интеллекте, как и в бытовой практике людей, П. понимается шире. В формировании П. могут принимать участие не только призначные структуры, но и результаты использования П. в деятельности людей или при функционировании интеллектуальной системы. Именно в этом смысле используются П. при рассуждениях о деятельности людей и о функцио-

нировании интеллектуальных систем. Для образования П. в интеллектуальных системах используются различные приемы обобщения.

ПОРОЖДЕНИЕ ГИПОТЕЗ АВТОМАТИЧЕСКОЕ

Процесс получения из фактов, хранящихся в базе данных, новых информационных единиц в базе знаний.

ПОРОЖДЕНИЕ ТЕКСТА

См. Генерация текста.

ПРАВИЛА ДЕ МОРГАНА

Правила, устанавливающие связь конъюнкции и дизъюнкции.

Типичные примеры:

$$a \& b = \neg (\neg a \vee \neg b),$$

$$a \vee b = \neg (\neg a \& \neg b).$$

ПРАВИЛО

См. Продукция.

ПРАВИЛО ВЫВОДА

Правило, с помощью которого в формальных системах из множества аксиом порождается правильно построенные формулы, которые интерпретируются как истинные.

ПРАВИЛО ВЫВОДА КОМПОЗИЦИОННОЕ

Правило вывода в нечеткой логике, основанное на операции композиции. Эта операция преобразует коэффициент правдоподобия исходной посылки (умножением на специально подобранную матрицу) в коэффициент правдоподобия заключения. Существуют такие матрицы, которые превращают П.В.К. в модус поненс и модус толленс.

ПРАВИЛО СИНТАКСИЧЕСКОЕ

В формальной системе правило, которое определяет способ формирования синтаксически правильных выражений. В лингвистике П.С.

позволяют отделять синтаксически правильные предложения данного языка от тех, которые таковыми не являются.

ПРЕДИКАТ

В исчислении предикатов специальный знак, отражающий определенное отношение между конечным множеством сущностей - аргументов. В обычном варианте исчисления предикатов в качестве значения П. на множестве означенных аргументов выступают два: истина и ложь.

ПРЕДСТАВЛЕНИЕ ДАННЫХ

Расположение данных в физической среде фиксируемое специальной схемой базы данных. Используя эту схему, система управления базой данных соотносит запрос на поиск нужной информации с физическим расположением данных. Различные принципы построения схемы базы данных приводят к различным типам П.Д. (реляционное, иерархическое и сетевое).

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ

Формализация знаний для их ввода в базу знаний. На концептуальном уровне П.З. наиболее распространены модели знаний в виде семантических сетей, фреймов и продукционных систем. В П.З. как направление искусственного интеллекта традиционно включает также задачи проверки содержимого базы знаний на корректность и полноту, пополнения знаний за счет логического вывода на основе имеющихся в базе знаний, обобщения знаний и классификация знаний.

ПРЕДСТАВЛЕНИЕ ЭКСТЕНСИОНАЛЬНОЕ

Представление константных фактов, не содержащих свободных переменных в базах данных или в базах знаний.

ПРИНЦИП РЕЗОЛЮЦИИ

Метод логического вывода, в основе которого лежит приведение доказываемого утверждения к множеству дизъюнктов и поиску в этом множе-

стве пар, один дизъюнкт которых содержит некоторую литеру, а другой – отрицание этой литеры, для их последовательного устранения из исходного множества. Если этот процесс через конечное число шагов приводит к пустому дизъюнкту, то вывод успешен. В противном случае формула недоказуема.

ПРИБРЕТЕНИЕ ЗНАНИЙ

Совокупность методов и процедур, которые применяет инженер по знаниям при заполнении им базы знаний. П.З. предполагает использование источников знаний двух типов: пассивных и активных. К первым относятся официальные документы, инструкции, печатные издания, кино-фото-документы и многие другие источники, в которых содержатся сведения, важные для описания знаний о предметной области. Ко второму типу источников знаний относятся специалисты в данной предметной области. Инженер по знаниям с помощью специальных психологических методик и инструментальных средств в процессе диалога получает от экспертов необходимые сведения. Все приобретенные знания для ввода в базу знаний формализуются в соответствии с требованиями той модели знаний, которая соответствует выбранному проектировщиком системы представлению знаний.

ПРОГРАММИРОВАНИЕ ЛОГИЧЕСКОЕ

Программирование, при котором программа представляется в виде процедуры логического вывода в исчислении предикатов первого порядка. Механизм вывода обычно встроен в язык П.Л. Примером может служить распространенный в интеллектуальных системах язык ПРОЛОГ, в который встроен обратный вывод.

ПРОДУКЦИЯ

Способ представления процедурных знаний в следующем наиболее общем виде: (i); Q; P; C; A B; N. Здесь (i) – собственное имя (метка) П.; Q –

сфера применения П., вычлениющая из предметной области некоторую ее часть, в которой знание, заключенное в П., имеет смысл; Р – предусловие, содержащее информацию об истинности данной П., ее приоритетности и т.п., используемую в стратегиях управления выводом для выбора данной продукции для исполнения; С – условие, представляющее собой предикат, истинное значение которого разрешает применять на некотором шаге данную П.; А В – ядро продукции (интерпретация ядра может быть различной, например: «Если А истинно, то В истинно», «Если А имеется в базе знаний, то В надо внести в базу знаний», «Если А текущая ситуация, то надо делать В» и т.п.); N – постусловие П., содержащее информацию о том, какие изменения надо внести в данную П. или другие П., входящие в систему продукции, после выполнения данной П.

ПРОПОЗИЦИЯ

Предложение, суждение, утверждение. (См. также Исчисление пропозициональное.)

ПРОСТРАНСТВО ЗАДАЧ

Структура, отражающая декомпозицию крупных задач на более мелкие, вплоть до стандартных задач, решение которых предполагается известным. П.З. используется в интеллектуальных системах для планирования деятельности и в задачах автоматического синтеза программ. Решение нужной задачи ищется как композиция решений стандартных задач.

ПРОСТРАНСТВО ОСГУДА

Применяемый в психологии формализованный способ выявления семантической близости используемых людьми понятий. Для построения П.О. используются бинарные шкалы, концы которых отмечены словами-антонимами типа «безопасный – опасный», «широкий – узкий», «добрый – злой» и т.д. На этих шкалах нанесено некоторое число позиций. Испытуемые должны располагать на них точки, соответствующие сло-

вам, произносимым экспериментатором. Результаты подвергаются статистической обработке с помощью факторного анализа или кластерного анализа. На основании многочисленных экспериментов такого типа было построено трехмерное П.О., оси которого интерпретируются как обобщение шкалы оценок, силы и активности. В этом пространстве используется обычная метрика векторных пространств. В П.О. понятия, связанные между собой общей ситуацией, группируются в некоторые сгущения, что подтверждает мысль о том, что основой классификации знаний у людей является принцип ситуативности.

ПРОСТРАНСТВО СЕМАНТИЧЕСКОЕ

Структура на знаниях, в которой введено понятие «семантическое расстояние». Примером П.С. может служить пространство Осгуда.

ПРОСТРАНСТВО СОСТОЯНИЙ

Совокупность состояний, в которых может находиться техническая система или процесс. В П.С. может быть задана метрика, а также указаны возможные траектории смены состояний под влиянием различных причин. П.С. используется в интеллектуальных системах при автоматическом синтезе программ и при планировании деятельности.

ПРОСТРАНСТВО ЦЕЛЕВОЕ

Совокупность целей с указанием возможных траекторий их достижения. П.Ц. используется в интеллектуальных системах при планировании деятельности и при автоматическом синтезе программ.

ПРОТИВОРЕЧИВОСТЬ АБСОЛЮТНАЯ

Возможность одновременного вывода в формальной системе утверждения и его отрицания. В такой формальной системе можно вывести любые утверждения по правилу модус поненс.

ПРОТИВОРЕЧИВОСТЬ МОДЕЛЬНАЯ

Противоречие, обнаруживаемое в модели. В замкнутых моделях П.М. является аналогом абсолютной противоречивости, в открытых моделях П.М. относительна.

ПРОЦЕДУРА ОПРОВЕРЖЕНИЯ

Установление противоречивости (невыполнимости) формулы, состоящей из конъюнкции посылок и отрицания заключения.

ПРОЦЕДУРА ПРИСОЕДИНЕНИЯ

Процедура на которую возможна ссылка по имени, использованному в некоторой информационной единице. По этому имени П.П. может быть просто вызвана и присоединена к описанию информационной единицы или актуализирована и выполнена.

ПРОЦЕССОР АССОЦИАТИВНЫЙ

Процессор, приспособленный для работы с ассоциативной памятью.

ПРОЦЕССОР ЛИНГВИСТИЧЕСКИЙ

Устройство или совокупность программ, ориентированные на реализацию общения пользователей с системой на ограниченном естественном языке.

ПРОЦЕССОР ЛОГИЧЕСКИЙ

См. Процессор логического вывода.

ПРОЦЕССОР ЛОГИЧЕСКОГО ВЫВОДА

Специализированный процессор (система процессоров), реализующий набор процедур, необходимых для организации логического вывода или извлечения следствий из знаний о некоторой предметной области.

ПРОЦЕССОР СИМВОЛЬНЫЙ

Специализированный процессор, ориентированный на обработку символьной информации.

РАЗРЕШИМОСТЬ АЛГОРИТМИЧЕСКАЯ

Наличие алгоритма решения поставленной задачи.

РАСПОЗНАВАНИЕ ОБРАЗОВ

Научное направление, основной задачей которого является создание моделей, методов и средств, связанных с решением задач классификации, таксономии, формирования понятий и т.п.

РАСПОЗНАВАНИЕ РЕЧИ

Один из видов восприятия в интеллектуальных системах. В процессоре Р.Р. происходит анализ входного акустического сигнала, выделение в нем фонем, слов, лексем, стандартных кусков текста, которые соотносятся с информацией, хранящейся в базе знаний системы, что позволяет системе понимать вводимый текст. Различают Р.Р. на уровне отдельных слов, произносимых стандартным диктором, а также с подстройкой системы к особенностям произношения конкретного диктора, и Р.Р., относящиеся к слитному тексту. В настоящее время реализованы системы Р.Р.Д, позволяющие уверенно определять значение 1 - 2 тыс. слов, а также анализировать слитный текст, в котором используются не слишком большие словари.

РАССТОЯНИЕ СЕМАНТИЧЕСКОЕ

Оценка «семантической» близости информационных единиц, хранящихся в памяти интеллектуальной системы или человека. Понятие «семантической» близости неоднозначно. Эксперименты с людьми показывают, что в зависимости от поставленной цели Р.С. может интерпретироваться как ситуативная близость (встреча в некоторых типовых ситуациях), ассоциативная близость (типа «молоток-гвоздь»), таксономическая близость (типа «стол-кровать») и т.д. Р.С. используется в базах знаний для ускорения выбора релевантной информации к данному понятию или ситуации. До настоящего времени нет удовлетворительных моделей, в

рамках которых можно было бы измерять Р.С. (См. также Пространство Осгуда.)

РАССУЖДЕНИЕ

Способ получения заключения на основе посылок и вспомогательных соображений. Крайним случаем Р. является логический вывод, в котором воля субъекта не играет роли. В других случаях Р. отражает личностные мотивы и интересы того, кто проводит Р.

РАССУЖДЕНИЕ ЗДРАВОВОГО СМЫСЛА

Один из видов правдоподобного рассуждения опирающийся не на основания, верные в некоторой формальной системе а не соображения, апеллирующие к человеческому опыту, интуиции, вере.

РАССУЖДЕНИЕ НЕМОНОТОННОЕ

Рассуждение в открытой модели. Из-за открытости возможно добавление новой информации извне в процессе проведения рассуждения. Это приводит к тому, что некоторые шаги рассуждения, верные до появления этой информации, могут стать неверными. Когда рассуждение является строгим выводом, то имеет место немонотонный вывод.

РАССУЖДЕНИЕ ПО АНАЛОГИИ

Перенос заключений, полученных на основе ряда посылок, на другую совокупность посылок, которая считывается по некоторому критерию аналогичной первой. В частном случае Р.А. есть способ получения заключения на основании диаграммы Лейбница $A, A_1; B, B_1; T, T_1; \Gamma, \Gamma_1$), где $A_1 = T(A)$; где T – некоторый преобразующий оператор; Γ – гомоморфизм между A и B ; Γ_1 – гомоморфизм между A_1 и B_1 ; T_1 – преобразующий оператор $B_1 = T_1(B)$. Р.А. есть нахождение B_1 по известным остальным элементам диаграммы.

РАССУЖДЕНИЕ ПО АССОЦИАЦИИ

Рассуждение, основанное на том, что заключение относительно одного объекта переносится на другой объект, имеющий с первым ассоциативную связь. Эта связь может иметь различный характер (например, ассоциацию по схожести, по одновременности, по встречаемости в одинаковых ситуациях и т. п.). Р.А. является правдоподобным рассуждением, степень его правдоподобия определяется существенностью используемой ассоциативной связи.

РАССУЖДЕНИЕ ПО УМОЛЧАНИЮ

Один из видов правдоподобного рассуждения где результат получается не из явно присутствующих для этого посылок, а на основе «традиции», прошлого опыта, внутренних моральных или ценностных убеждений и т.п. Р.У. возникают тогда, когда во входной информации часть сведений отсутствует и интеллектуальная система пополняет их на основе хранящейся в ее памяти специальной информации, предназначенной для случаев неполноты входной информации. Например, во фреймах могут существовать специальные слоты, к которым система обращается за информацией, когда для проведения рассуждения чего-то не хватает.

РАССУЖДЕНИЕ ПРАВДОПОДОБНОЕ

Рассуждение, которое опирается либо на знания, не имеющие абсолютно истинного характера, либо на приемы рассуждений, которые не являются абсолютно верными. Обычно результат Р.П. снабжается оценкой его правдоподобия. Примерами Р.П. могут служить рассуждения по аналогии или ассоциации или герменевтические рассуждения.

РЕЗОЛЬВЕНТА

Образование дизъюнкта $(ct - lt) \vee (dt - \bar{l}t)$, где c и d – дизъюнкты, не имеющие общих переменных; l и \bar{l} – контрарная пара литер, каждая

из которых принадлежит своему дизъюнкту; t – наиболее общий унификатор контрарной пары литер.

РЕЗОЛЮЦИЯ

Прием, используемый при дедуктивном выводе, заключающийся в нахождении двух дизъюнктов, один из которых содержит литеру, а другой – ее отрицание. На основании этого сравнения формируется новый дизъюнкт, называемый резольвентой. Порождение новых дизъюнктов является основой метода резолюций, широко применяемого в интеллектуальных системах.

РЕШАТЕЛЬ

Система, способная благодаря встроенной в нее общей стратегии нахождения решения (например, путем поиска в пространстве альтернатив или путем логического вывода) находить решения задач. Р. входит в качестве основного блока в интеллектуальные системы. Иногда этот блок называется полно: Решатель задач.

РОБОТ АВТОНОМНЫЙ

Техническое устройство, способное к планированию целесообразного поведения в условиях динамической, заранее не полностью известной среды. Р.А. должен обладать базой знаний о среде и ее особенностях, решателем задач с средствами для анализа ситуаций и последствий своих действий в среде, чтобы накапливать информацию о том, как нужно действовать в тех или иных ситуациях. Р.А. является представителем интеллектуальных систем.

РОБОТ ИНТЕГРАЛЬНЫЙ

Техническое устройство в котором имеется развитая система «глаз-рука», позволяющая наблюдаемому координировать ситуацию с движением манипуляторов и средств перемещения. Это отличает Р.И. от роботов-манипуляторов, в которых нет обратной связи со средой, если ситуа-

ции отличаются от штатных, заранее зафиксированных в конструкции робота-манипулятора. Р.И. должен обладать способностью к анализу зрительных сцен и уметь принимать решения на основе этого анализа. Р.И. является представителем интеллектуальных систем.

РОБОТ ИНТЕЛЛЕКТУАЛЬНЫЙ

Автономный робот, в котором имеются все основные блоки, характерные для интеллектуальной системы. С их помощью реализуются функции общения Р.И. с внешними партнерами, строятся программы поведения, накапливаются знания о внешней среде и действиях в ней, строятся планы поведения по достижению нужных целей.

СБОРКА МУСОРА

Процесс чистки памяти, связанный с обнаружением неиспользуемых программой блоков памяти и присоединением их к пространству свободной памяти для повторного использования. Сборка мусора – необходимый процесс в любой системе, работающей с динамическим распределением памяти.

СВИДЕТЕЛЬСТВО

Факт, используемый для увеличения или уменьшения оценки правдоподобия некоторой гипотезы. С. применяются в продукционных системах, в которых продукции являются гипотезами.

СЕКВЕНЦИЯ

В узком смысле – правило логического перехода $A \rightarrow B$, которое интерпретируется следующим образом: если A истинно, то B также истинно, если A ложно, то о B сказать ничего нельзя. В широком смысле С. совпадает с понятием ядра продукции.

СЕМАНТИКА

1. Один из аспектов семиотики. Рассматривает отношение знаков к обозначаемому (содержание знаков) независимо от того, кто служит адресатом знака.
2. Значение отдельных единиц знака.
3. Изучение отдельных единиц языка – языковедческая семантика, элементарным объектом изучения которой является единство трех объектов: означающего, означаемого и денотата. Означающее – внешний элемент (последовательность звуков или знаков), денотат – обозначаемый объект действительности и означаемое – отражение этого объекта в сознании человека.

СЕМАНТИКА СИТУАТИВНАЯ

Приписывание некоторым объектам, хранящимся в базе знаний, некоторых характеристик в зависимости от ситуации, в которой эти объекты наблюдаются или используются. В системах понимания текстов на естественном языке С.С. связана с приписыванием различных значений лексемам в зависимости от того контекста, в котором они используются.

СЕМИОТИКА

Наука, изучающая свойство знаков и знаковых систем (в основном естественных и искусственных языков.) Выделяются три основных аспекта исследования: синтактика, изучающая внутренние свойства систем знаков безотносительно к интерпретации; семантика, рассматривающая отношение знаков к их означаемому, безотносительно к особенностям интерпретатора знаков; прагматика, изучающая проблемы интерпретации знаков.

СЕТЬ

Пятерка $H = \langle A, B, P, P1, C \rangle$, где A – множество вершин, B – множество имен (весов) вершин; P – множество дуг, соединяющих пары вершин;

P_1 – множество отмеченных входных и выходных дуг; C – множество имен (весов) дуг.

СЕТЬ АССОЦИАТИВНАЯ

Семантическая сеть, в которой отношения указывают на ассоциативные связи между вершинами, характеризующими объекты, факты и ситуации для описываемой предметной области.

СЕТЬ ВЫВОДА

Структура, которая отображает последовательности применения правил вывода к исходным посылкам. Из-за неоднозначности выбора правил на каждом шаге возникает множество путей, образующих С.В.

СЕТЬ КАУЗАЛЬНАЯ

Семантическая сеть, в которой дуги характеризуют отношения, используемые в каузальной логике.

СЕТЬ ПЕРЕХОДОВ РАСШИРЕННАЯ

Расширение контекстно-свободной грамматики за счет вынесения контекста в специальные именованные регистры со стековой структурой и введения специальных процедур, управляющих анализом с помощью проверки контекста ходом выполнения анализа управляет программист). См. также. Грамматика формальная.

СЕТЬ ПЕТРИ

Модель для описания асинхронных параллельных и недетерминированных процессов, а также систем продукционного типа. Статически модель задается двудольным орфографом с двумя типами вершин – позициями и переходами (изображаемыми обычно кружками и полочками соответственно), причем переходы (позиции) могут соединяться дугами только с позициями (переходами). Исходное состояние С.П. задается начальной маркировкой некоторых ее позиций. Маркеры изображаются точками внутри позиций. Динамика вносится соглашением о правиле срабатыва-

ния возбужденного перехода (содержащего по крайней мере один маркер в каждой из его входных позиций), который может сработать через заранее неизвестное время, после чего из всех (во все) его входных позиций (выходные позиции) изымается (добавляется) по одному маркеру. Процесс функционирования С.П. состоит в переходе от одной маркировки к другой посредством срабатываний возбужденных переходов.

СЕТЬ ПРИЧИННО-СЛЕДСТВЕННАЯ

Семантическая сеть, в которой все отношения между вершинами трактуются как отношение «причина-следствие», т.е. как неререфлексивное, антисимметричное и транзитивное отношение.

СЕТЬ СЕМАНТИЧЕСКАЯ

Сеть, в вершинах которой находятся информационные единицы, а дуги характеризуют отношения и связи между ними С.С. является наиболее общей моделью представления знаний.

СЕТЬ СЕМАНТИЧЕСКАЯ ИНТЕНСИОНАЛЬНАЯ

Семантическая сеть, в которой отражены интенциональные знания о предметной области. Эти знания относятся к общим законам области и оперируют не константными фактами, а высказываниями об области, содержащими переменные, которые могут означиваться в конкретных ситуациях. При фреймовом представлении С.С.И. соответствует фрейм-прототип.

СЕТЬ СЕМАНТИЧЕСКАЯ ЭКСТЕНСИОНАЛЬНАЯ

Семантическая сеть, в которой отражены экстенциональные знания о конкретной ситуации в предметной области. В С.С.Э. все вершины сети соответствуют конкретным объектам, а связи между ними конкретным связям, которые наблюдаются в описании ситуации. При фреймовом представлении С.С.Э. соответствует фрейм-экземпляр.

СИЛЛОГИЗМ

Специальная форма умозаключения от общего к частному. С. представляет собой заключение, следующее из двух посылок, содержащих утверждения о соотношении объемов двух классов или о принадлежности некоторого элемента определенному классу. Примером С. может служить следующее умозаключение: Все хищники питаются мясом. Волк – хищник. Волк питается мясом.

СИНТЕЗ ПРОГРАММ АВТОМАТИЧЕСКИЙ

Автоматическое построение программ по исходным условиям задачи. Формально задача С.П.А. может быть поставлена следующим образом: по заданному значению x , удовлетворяющему предикату $P(x)$, вычислить значение y , удовлетворяющее предикату $R(x, y)$. Здесь x, y – конечные множества входных и выходных переменных программы. Отображение $Q = \langle P(x), R(x, y) \rangle$ должно при этом содержать достаточно информации для синтеза нужной программы. Различают три подхода: индуктивный синтез программ, дедуктивный синтез программ или трансформационный синтез программ.

СИНТЕЗ ПРОГРАММ ДЕДУКТИВНЫЙ

Автоматический синтез программ в котором используется формальный метод построения программ; вывод программы из заданной спецификации рассматривается как задача конструктивного доказательства существования нужного решения. В ходе реализации С.П.Д. часто используются методы автоматического доказательства теорем в исчислении предикатов первого порядка.

СИНТЕЗ ПРОГРАММ ИНДУКТИВНЫЙ

Автоматический синтез программ опирающийся на нахождение общих для заданного множества примеров процедур, ведущих к решению задачи. С.П.И. может осуществляться на синтаксическом уровне, когда ис-

пользуются внешние признаки программ, или на семантическом уровне, когда в основе лежит семантика модели программы.

СИНТЕЗ ПРОГРАММ ТРАНСФОРМАЦИОННЫЙ

Автоматический синтез программ, в котором конструирование программы осуществляется последовательным преобразованием исходной спецификации в эквивалентную ей результирующую программу. С.П.Т. обычно разбивается на следующие этапы: переход от формальной спецификации к рекурсивной версии программы общего вида; трансформация общерекурсивной версии в рекурсивную версию специального вида; переход от рекурсивной версии к итеративной версии.

СИСТЕМА АКСИОМАТИЧЕСКАЯ

Система рассуждений, в основе которой лежат аксиомы. Если аксиомы являются логическими (т.е. тождественно истинными вне зависимости от предметной области), то С.А. совпадает с формальной системой. Если среди аксиом имеются аксиомы, справедливые только в данной предметной области, то С.А. применима только в этой области. Часто С.А., у которой имеются предметные аксиомы, называют квазиаксиоматической системой.

СИСТЕМА ВОПРОСНО-ОТВЕТНАЯ

Совокупность программ, позволяющая реализовать вопросно-ответное отношение для пары «пользователь – ЭВМ». С.В.О. обеспечивает поиск релевантных данных и знаний по поступившему в систему вопросу, из которых формируется ответ пользователю. С.В.О. могут использовать вопросы и ответы, формулируемые на естественном языке, ограниченном естественном языке или на специальном формальном языке общения.

СИСТЕМА ДЕДУКТИВНАЯ

Аксиоматическая система, в которой имеет место теорема дедукции, обеспечивающая вывод всего множества выражений, которое допускается аксиомами и правилами вывода (и только ими). При этом предполагается, что выводимые выражения обладают априорным свойством (например, тождественной истинностью или истинностью с оценкой правдоподобия не менее заданной).

СИСТЕМА ДОВЕРИЯ

Совокупность процедур, используемая в системах объяснения экспертных систем. С.Д. применяется для того, чтобы повысить уровень доверия пользователя к результатам, полученным экспертной системой. Для этого С.Д. выдает пользователю ряд дополнительных аргументов в пользу того решения, которое было получено системой, при условии, что эти аргументы не были непосредственно использованы в процессе получения решения.

СИСТЕМА ЕСТЕСТВЕННО-ЯЗЫКОВАЯ

Интеллектуальная система, способная воспринимать тексты или речь на ограниченном естественном языке, понимать их и функционировать в соответствии с информацией, содержащейся в тексте. При необходимости С.Е.Я. может формировать ответные сообщения, связанные с введенным в них текстом. (См. также Интерфейс естественно-языковый.)

СИСТЕМА ИНДУКТИВНАЯ

Модель, в которой используется индуктивная логика или индуктивное обобщение (или и то и другое).

СИСТЕМА ИНТЕЛЛЕКТУАЛЬНАЯ

Техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти С.И. Структура

С.И. включает три основных блока: – базу знаний, решатель и интеллектуальный интерфейс.

СИСТЕМА ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ

Интеллектуальная система для обучения человека какому-либо роду деятельности или используемая в процессе обучения школьников или студентов. С.И.О. включает естественно-языковой интерфейс, средства графического общения, базу знаний и специальный блок, управляющий процессом подачи материала, анализирующий обучаемого и принимающий решение об оценке степени усвоения.

СИСТЕМА ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯСЯ

Интеллектуальная система в которой имеется совокупность средств для пополнения ее базы знаний. Возможны два способа обучения: с учителем и без учителя. В первом случае С.И.О. должна иметь возможность пополнять свою базу знаний информацией, сообщаемой ей учителем, корректировать новую информацию с ранее имевшейся и задавать учителю вопросы, когда возникают конфликтные ситуации. При обучении без учителя С.И.О. обобщает информацию, которая содержится в ее базе данных или наблюдается во внешней среде. С помощью индуктивного вывода и, возможно, рассуждений по аналогии и рассуждений по ассоциации С.И.О. формирует гипотезы о новых закономерностях в той предметной области, в которой она работает. Этим новым знаниям система может приписывать оценки правдоподобия.

СИСТЕМА КВАЗИАКСИОМАТИЧЕСКАЯ

Термин объясняется в статье Система аксиоматическая.

СИСТЕМА ОБУЧАЮЩАЯ

Система, способная к накоплению новых знаний в результате обращения к учителю или обобщения наблюдаемых фактов. (См. также Система интеллектуальная обучающаяся.).

СИСТЕМА ОБЪЯСНЕНИЯ

Часть экспертной системы, предназначенная для разъяснения пользователю по его просьбе способа, которым получено решение. С.О. дает ответы на вопросы трех типов. «Как-вопросы» заставляют систему объяснять путь решения. «Почему-вопросы» вызывают у С.О. необходимость в пояснении, почему получено именно это решение, а не какое либо другое. «Что-вопросы» требуют от С.О. выдачи всей имеющейся в экспертной системе информации относительно объекта или явления, к которому относится «Что-вопрос». Для ответов на подобные вопросы в С.О. имеется набор специальных процедур и рабочая память, в которой в процессе поиска решения формируется информация, необходимая для ответа на вопросы пользователя. С С.О. связана система доверия, увеличивающая объяснительную силу экспертной системы.

СИСТЕМА, ОСНОВАННАЯ НА ЗНАНИЯХ

Интеллектуальная система, функционирование которой определяется совокупность знаний о проблемной области, в которой она используется.

СИСТЕМА, ОСНОВАННАЯ НА ПРАВИЛАХ

Интеллектуальная система, в базе знаний которой принято представление знаний в виде систем продукций.

СИСТЕМА ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Совокупность средств для автоматизированного заполнения баз знаний и поддержания их в рабочем состоянии. В зависимости от выбранной в интеллектуальной системе модели знаний различают С.П.З., основанные на семантических сетях, фреймах и продукционных правилах. С.П.З. входят в состав всех интеллектуальных систем.

СИСТЕМА ПРОДУКЦИЙ

Совокупность множества продукций, выполнение которых задается с помощью стратегии управления выводом. Стратегия определяется абст-

рактной машиной $M = \langle M1, M2, M3 \rangle$, где $M1$ формирует фронт продукции, для которых выполнены условия их применимости; $M2$ производит выбор из фронта готовых продукции той, которая будет выполняться на данном шаге процесса, $M3$ вносит необходимые изменения в продукции, входящие в С.П., на основании той информации, которая содержалась в постусловии продукции, выполненной на данном шаге процесса.

СИСТЕМА ПРОДУКЦИОННАЯ

См. Система продукции.

СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ЗНАНИЙ

Совокупность программных и аппаратных средств для организации в базах знаний процедур, связанных с поиском знаний, пополнением базы знаний, ее корректировкой и т. п.

СИСТЕМА ФОРМАЛЬНАЯ

Модель, лежащая в основе многих математических теорий. С.Ф. представляет собой четверку: множество базовых элементов синтаксические правила аксиомы правила вывода.

СИСТЕМА ФРЕЙМОВ

Множество фреймов, связанных между собой различными отношениями.

СИСТЕМА ЭКСПЕРТНАЯ

Интеллектуальная система, предназначенная для оказания консультационной помощи специалистам, работающим в некоторой предметной области. Различают два типа С.Э. Системы первого типа предназначены для специалистов, чей профессиональный уровень не слишком высок. В базах знаний таких систем хранятся знания, полученные от специалистов экстракласса. Системы второго типа призваны помогать специалистам высокой квалификации, выполняя для них значительную часть рутинных операций и просмотр больших массивов информации. Особенностью

С.Э. является наличие в них системы объяснений, повышающей консультационную силу С.Э.

СИСТЕМА ЭКСПЕРТНАЯ ПУСТАЯ

Экспертная система, у которой база знаний пуста. При использовании С.Э.П. в некоторой предметной области пользователю необходимо произвести заполнение базы знаний либо самому с помощью специальных инструкций по заполнению базы, либо привлекая инженера по знаниям. Как показал опыт, С.Э.П. могут эффективно использоваться лишь на однотипных предметных областях, на которое ориентированы принятые в С.Э.П. способы представления знаний и рассуждений. (См. также Оболочка.)

СЛЕДСТВИЕ ЛОГИЧЕСКОЕ

Факт, полученный в процессе логического вывода в формальной системе.

СЛОТ

Основная структурная единица фрейма. С. представляет собой пару: (атрибут (имя слота) – значение). В качестве значения могут выступать константные факты, выражения, содержащие переменные, ссылки на другие С. и т.п. С. может иметь структуру, элементы которой сами являются слотами. Часто эти более мелкие С. называют фаскетами, ячейками, аспектами и т.п.

СОБЫТИЕ

Информационная единица, которой в базе знаний присваивается интервал времени, в течение которого эта единица существует.

СОПОСТАВЛЕНИЕ С ОБРАЗОМ

Процедура сравнения фрагмента знаний или описания с эталонным образцом. Входит в состав процедуры поиска по образцу в базах знаний и используется при распознавании образов.

СОСТАВЛЯЮЩАЯ ЗНАНИЙ ИНТЕНСИОНАЛЬНАЯ

Знания о предметной области, которые отражают факты, закономерности, свойства и характеристики, справедливые для любых конкретных ситуаций, которые могут возникнуть в этой предметной области.

СОСТАВЛЯЮЩАЯ ЗНАНИЙ ЭКСТЕНСИОНАЛЬНАЯ

Знания о предметной области, отражающие факты, закономерности, свойства и характеристики, типичные для конкретных ситуаций или классов однотипных ситуаций, которые могут возникнуть в этой области.

СПИСОК АССОЦИАТИВНЫЙ

Совокупность пар «атрибут – значение».

СРЕДСТВА ИНЖЕНЕРИИ ЗНАНИЙ ИНСТРУМЕНТАЛЬНЫЕ

Системы программирования, которые упрощают работу по созданию интеллектуальных систем.

ССЫЛКА АНАФОРИЧЕСКАЯ

Способ связывания между собой двух предложений в тексте, когда в первом предложении называется лицо, объект, явление и т.п., а во втором используется анафорическое слово (как правило, местоимение), заменяющее слово из первого предложения. Например, в тексте «Система управления дала сбой. Она не была во время продиагностирована» анафорическое слово «она» во втором предложении указывает, что в нем идет речь о той же системе управления, о которой шла речь в первом предложении. С.А. требует специальных процедур при анализе текстов в интеллектуальных системах. С.А. являются частным случаем анафоры.

СТРАТЕГИЯ УПРАВЛЕНИЯ ВЫВОДОМ

Совокупность правил, с помощью которых организуется выбор правил вывода в формальных системах или выбор продукций в системе продукций при поиске решения. Наиболее известные С.У.В. в формальных ло-

гических системах – это прямой вывод и обратный вывод. В системах продукции известно много С.У.В. (принцип «классной доски», учет наиболее длинного условия продукции, принцип «повестки дня» и т.п.).

СТРУКТУРА ГЛУБИННАЯ

Структура, отражающая семантику текста на естественном языке. В С.Г. отсутствуют особенности морфологического или синтаксического строения естественного языка. С.Г. используется при машинном переводе с одного языка на другой, а также обеспечивает понимание текстов на естественном языке, так как от С.Г. легко осуществляется переход к тем представлениям знаний, которые используются в базе знаний.

СТРУКТУРА КОГНИТИВНАЯ

Гипотетическая структура, с помощью которой отображаются и хранятся в памяти человека знания об окружающем мире, возможных действиях в нем и о самом человеке. С.К. изучается в когнитивной психологии.

СТРУКТУРА ПОЗНОВАТЕЛЬНАЯ

В искусственном интеллекте – модель, в которой происходит соотнесение объектов действительности с их внутренним представлением в памяти интеллектуальной системы. С.П. не зависит от особенностей языка, на котором написан текст, а определяется теми механизмами, которыми располагает человеческое сознание для восприятия окружающего мира и его осмысления.

СУЖДЕНИЕ

Выражение, связывающее между собой субъект и предикат с помощью связки «есть». С. изучается в логике.

СУЩНОСТЬ

Любой объект в предметной области.

СХЕМА КОНЦЕПТУАЛЬНАЯ

См. Ограничение целостности.

СЦЕНАРИЙ

Семантическая сеть, в которой в качестве отношений используются каузальные отношения или отношения типа «действие-результат», «действие-цель», «орудие-действие» и т.п.

ТВОРЧЕСТВО МАШИННОЕ

Направление в искусственном интеллекте, в рамках которого создаются модели и способы генерирования на ЭВМ аналогов результатов творческой деятельности человека. Сюда относятся программы для сочинения музыкальных произведений, стихотворений, прозы, машинная графика и живопись, игровые программы и т. п.

ТЕЗИС ЧЕРЧА

Утверждение о том, что «лямбда» – исчисление эквивалентно интуитивному пониманию алгоритма Т.Ч. аналогичен другим тезисам (например, тезису Тьюринга о том, что машина Тьюринга эквивалентна интуитивному пониманию алгоритмического процесса), позволяющим формализовать понятие алгоритма.

ТЕОРИЯ АКСИОМАТИЧЕСКАЯ

Логическая теория, в которой все исходные аксиомы являются тождественно истинными, а правила вывода – достоверными.

ТЕОРИЯ ЛОГИЧЕСКАЯ

Теория, в основе которой лежит формальная система.

ТЕРМ

Константа, переменная или выражение вида $f(t_1, t_2, \dots, t_n)$ где f – функциональный символ; t_i – термы.

ТИП ДАННЫХ

Объединенные под одним именем по какому-либо признаку данные. Т.Д. используются в базах данных.

ТИП ДАННЫХ АБСТРАКТНЫЙ

Описание данных на языках программирования, позволяющее создавать в памяти ЭВМ те структуры данных, которые удобны для решения задачи.

УНИВЕРСУМ

Область, на которой определены значения предиката. Из этой области берутся значения для означивания переменных, входящих в предикат. Для многоместных предикатов $У$. есть декартово произведение областей определения для каждого из аргументов предиката.

УНИВЕРСУМ ЭРБРАНА

Специальная область, на которой проверяется невыполнимость множества дизъюнктов (клауз) при реализации процедур вывода типа метода резолюций.

УНИФИКАТОР

Термин объясняется в статье Унификация.

УНИФИКАТОР НАИБОЛЬШИЙ ОБЩИЙ

Унификатор $г$, обладающий тем свойством, что для всякого другого унификатора q для того же множества выражений существует такая подстановка t , что $q = zt$, где zt – композиция подстановок $г$ и t .

УНИФИКАЦИЯ

Процедура подстановки термов в два логических выражения вместо переменных. Термы подбираются таким образом, что при замене ими одноименных в двух выражениях переменных оба выражения становятся идентичными. Сама подстановка называется унификатором. $У$. используется при логическом выводе в методе резолюций.

УПРАВЛЕНИЕ СИТУАЦИОННОЕ

Способ управления сложными техническими и организационными системами, при котором с помощью экспертной информации строится

классификатор, позволяющий разбивать все наблюдаемые ситуации на классы и приписывать каждому классу одношаговое решение или первый шаг в многошаговом решении. Для описания ситуаций и классификации их в У.С. используется модель знаний, близкая в семантической сети.

ФАСЕТ

См. Слот.

ФОКУС

В искусственном интеллекте совокупность знаний, активизируемая при формировании или интерпретации сообщения на естественном языке.

ФОРМА ПРЕФИКСНАЯ НОРМАЛЬНАЯ

ФОРМУЛА АТОМАРНАЯ

Выражение вида $H(t_1, t_2, \dots, t_m)$, где $H - m$ – местный предикат, t_i – термы.

ФОРМУЛА ЗАМКНУТАЯ

Выражение в формальной системе, в котором либо нет переменных, либо переменные являются связанными квантификаторами.

ФОРМУЛА ОБЩЕЗНАЧИМАЯ

Замкнутая формула, сохраняющая тождественную истинность при всех интерпретациях.

ФОРМУЛА ОТКРЫТАЯ

Выражение в формальной системе, в которое входит хотя бы одна переменная, не связанная квантификатором.

ФРЕЙМ

Специальная форма представления знаний, которая определяется рекурсивно. Ф. состоит из конечного числа слотов, каждый из которых имеет имя и значение.

ФРЕЙМ-ОБРАЗЕЦ

1. Фрейм, выступающий в качестве образца при поиске по образцу в базах знаний.
2. Синоним для фрейма-экземпляра.

ФРЕЙМ-ПРОТОТИП

Фрейм, у которого в части слотов (или во всех слотах) отсутствуют константные значения. Ф.П. описывает знание о предметной области. При означивании всех слотов. Ф.П. константными значениями он превращается в фрейм-экземпляр.

ФРЕЙМ-ЭКЗЕМПЛЯР

Фрейм-прототип, у которого значения всех слотов заполнены констативной информацией. Совокупность Ф.Э. образует экстенциональную базу данных.

ФУНКЦИЯ ВЫХОДОВ

Термин объясняется в статье Автомат конечный.

ФУНКЦИЯ ДОВЕРИЯ

Функция, значение которой характеризует веру субъекта в истинность события или факта.

ФУНКЦИЯ ПЕРЕХОДОВ

Термин объясняется в статье Автомат конечный.

ФУНКЦИЯ ПРИНАДЛЕЖНОСТИ

Характеристическая функция для нечеткого множества, изменяющаяся в интервале (0, 1).

ФУНКЦИЯ СКОЛЕМА

Функциональный символ операции по устранению кванторов существования в логических выражениях, приведенных к префиксной нормальной форме.

ЦЕПОЧКА ВЫВОДА

Последовательность формул, начинающаяся с аксиом в которой каждая последующая формула выводится на основе совокупности предшествующих элементов цепочки.

ШКАЛА АБСОЛЮТНАЯ

Шкала, на которой задана метрика, позволяющая отсчитывать расстояния от абсолютного начала.

ШКАЛА МЕТРИЧЕСКАЯ

Шкала, на которой задана метрика, позволяющая оценивать расстояния между элементами, отображенными на шкалу.

ШКАЛА ОСГУДА

Шкала, использующая пары слов-антонимов (острый – тупой, хороший – плохой и т.п.), стоящие на концах шкалы. Средняя позиция шкалы является нейтральной. Кроме того, имеется еще несколько промежуточных делений, как правило, не обозначаемых никакими словами. Ш.О. используется в психологических экспериментах, связанных с выявлением особенностей индивидуального психосемантического пространства испытуемых. (См. также Пространство Осгуда.)

ШКАЛА ОТНОСИТЕЛЬНАЯ

Метрическая шкала, в которой расстояния отсчитываются от какого-либо относительного маркера, например от наблюдаемого в данный момент объекта.

ШКАЛА РАЗМЫТАЯ

Порядковая шкала, на которой располагаются значения лингвистической переменной или интервалы, получаемые из функций принадлежности отсечками соответствующего уровня.

ШКАЛА ТОПОЛОГИЧЕСКАЯ

Шкала, на которой отображается лишь отношение порядка между элементами, расположенными на шкале. Ш.Т. обладает свойствами, присутствующими метрическим шкалам.

ШКАЛА УНИВЕРСАЛЬНАЯ

Специальная шкала для проецирования размытых шкал, у которых на расположение квантификаторов оказывает влияние конкретная семантика ситуаций, для описания которых они используются. На Ш.У. эти квантификаторы переводятся в другие отражающие общую меру размытости. Ш.У. позволяет сравнивать между собой высказывания, относящиеся к разным размытым шкалам.

ЭВРИСТИКА

Прием решения задачи, основанный не на строгих математических моделях и алгоритмах, а на соображениях, восходящих к «здравому смыслу». Как правило, Э. отражает особенности того, как такие задачи решает человек, когда он не пользуется строго формальными приемами. Если эти человеческие способы решения удастся запрограммировать, то такие программы называются эвристическими. Э. часто используются при программировании игр, имитации творческих процессов и т.п. В экспертных системах при формализации профессиональных знаний человека, касающихся способов решения задач в той или иной проблемной области, широко используются те Э., которыми руководствуются профессионалы-эксперты.

ЯЗЫК ЗАПРОСОВ

Язык для обращения в базы данных и базы знаний за необходимой информацией. Формально Я.З. это счетное множество цепочек из символов некоторого конечного алфавита. На этом множестве цепочек выделено подмножество правильных цепочек или правильных запросов. Каждый

правильный запрос имеет процедурную интерпретацию а операциях, разрешенных для информационных единиц, находящихся в базе данных или базе знаний.

ЯЗЫК КОНТЕКСТНО-СВОБОДНЫЙ

Множество цепочек символов, порождаемых контекстно-свободной грамматикой.

ЯЗЫК КОНТЕКСТНО-СВЯЗАННЫЙ

Множество цепочек символов, порождаемых контекстно-связанной грамматикой.

ЯЗЫК ПРЕДСТВЛЕНИЯ ЗНАНИЙ

Способ описания моделей знаний в системах представления знаний. На сегодняшний день известны Я.П.З. для модели знаний в виде фреймов (языки LISP, FRL, KRL и др.), а также ряд продукционных языков.

ЯЗЫК ПРЕДСТВЛЕНИЯ ЗНАНИЙ ЛОГИЧЕСКИЙ

Язык представления знаний в основе которого лежит исчисление предикатов первого порядка. Выражениями Я.П.З.Л. являются синтаксически правильные формулы этого исчисления. В виде таких формул записываются все хранимые в системе декларативные и процедурные знания. Достоинством Я.П.З.Л. является их полная формализуемость и наличие для них формальных процедур, позволяющих выполнять вывод и анализ таких характеристик записей, как непротиворечивость, эквивалентность и т.п. Недостатком Я.П.З.Л. является плохая наглядность для пользователя информационных единиц, записанных в виде формул логического исчисления.

ЯЗЫК ПРОДУКЦИОННЫЙ

Язык представления знаний, основной единицей которого является продукция.

ЯЗЫК СЕКВЕНЦИЙ

Язык, ориентированный на описание дискретных автоматических устройств, в частности конечных автоматов с памятью и без памяти. Позволяет строить компактные описания, особенно для сильно недоопределенных автоматов. Описание на Я.С. состоит из секвенций, каждая из которых является частным случаем продукции, в левой и правой части которой находятся булевы функции. Смысл секвенции состоит в том, что правая функция принимает значение 1, если левая функция равна 1, и не определена, если левая функция равна 0.

ЯЗЫК ФРЕЙМОВЫЙ

Язык представления знаний и манипулирования знаниями, использующий в качестве модели знаний фреймовые представления. Наиболее известными Я.Ф. являются языки FRL и KRL.

ЯЩИК БЕЛЫЙ

Объект исследований, о внутреннем устройстве которого имеется полная информация. Например дифференциальные уравнения, известны параметры и начальные условия..

ЯЩИК СЕРЫЙ

Объект исследований, о внутреннем устройстве которого либо известно частично, либо существуют некоторые гипотезы. В отличие от черного ящика, модели Я.С. учитывают помимо связей между реакциями и внешними воздействиями и те частичные сведения, которые известны о его внутреннем строении.

ЯЩИК ЧЕРНЫЙ

Введенное У.Р. Эшби наименование объекта исследования, внутреннее устройство которого неизвестно или не принимается во внимание. Модель Я.Ч. строится на основе его поведения т.е. реакции на воздействия,

поступающие на вход извне, и характеризует связи между реакциями и вызвавшими их воздействиями. Модели Я.Ч. теоретически обосновываются в направлении экспериментальной психологии, называемом бихевиоризмом, и обычно называются моделями «стимул – реакция».

ПРИЛОЖЕНИЯ

Тестовые задания по темам для самоконтроля

Тестовые задания по темам для самоконтроля входят в общий итоговый тест по курсу и проводятся на компьютере. По каждому заданию предъявляются 3 вопроса, один из которых правильный. В конце фразы «Выбрать один из 3 ответов» в квадратных скобках указан правильный ответ. Список вопросов задается преподавателем в обычной таблице и может изменяться простым редактором. Ниже дан один из вариантов списка примерных тестовых заданий с указанием для каждого номера правильного ответа.

1. Наиболее верное определение искусственного интеллекта (ИИ). Выбрать один из 3 ответов.[3]
2. Выбрать правильное определение интеллектуальной системы (ИС). Выбрать один из 3 ответов.[2]
3. Выбрать правильное определение систем интеллектуального управления (СИУ). Выбрать один из 3 ответов.[3]
4. Сколько Вы знаете основных этапов развития ИС и технологий? Выбрать один из 3 ответов.[1]
5. Какая группа из перечисленных ученых, внесла большой вклад в развитие ИИ? Выбрать один из 3 ответов [2]
6. В каком списке указаны основные интеллектуальные компоненты, применяемые в ИС? Выбрать один из 3 ответов [2]
7. В какой группе относятся основные подходы и методы, используемые в современных ИС и технологиях? Выбрать один из 3 ответов [1]
8. Выбрать определение понятия экспертных систем. Выбрать один из 3 ответов [1]

9. Динамические экспертные системы – это:...Выбрать один из 3 ответов [2]
10. Нейронные сети- это: ... Выбрать один из 3 ответов [2]
11. В каком списке системы, основанных на знаниях (СОЗ) ? Выбрать один из 3 ответов.[2]
12. Формальная аксиоматическая система-это: ... Выбрать один из 3 ответов [3]
13. Логический вывод –это: ... Выбрать один из 3 ответов [1]
14. Укажите наиболее полную группу основных понятий классического исчисления предикатов. Выбрать один из 3 ответов [2]
15. В каком списке перечислены аксиомы исчисления предикатов? Выбрать один из 3 ответов [1]
16. В каком списке перечислены правильно правила вывода исчисления предикатов? Выбрать один из 3 ответов [3]
17. В каком списке указаны верно задачи поиска вывода в исчислении предикатов? Выбрать один из 3 ответов [2]
18. Понятие о методах порождения гипотез. Выбрать один из 3 ответов [1]
19. Укажите правильный вариант понятия о дедуктивных правилах. Выбрать один из 3 ответов [1]
20. Укажите правильный вариант Логическое порождение гипотез. Выбрать один из 3 ответов [2]
21. Особенности получения, представления и использования знаний в экспертных системах. Выбрать один из 3 ответов [3]
22. В каком списке правильно указаны модели представления знаний в ИС на базе продукционных правил? Выбрать один из 3 ответов [1]
23. В каком списке правильно указаны модели представления знаний в ИС, построенных с использованием динамических семантических сетей? Выбрать один из 3 ответов [3]

24. В каком списке правильно указаны модели представления знаний в ИС, построенных с использованием фреймовых и других представлений? Выбрать один из 3 ответов [2]
25. Прямые цепочки выводов- это:... Выбрать один из 3 ответов [1]
26. обратные цепочки выводов- это:... Выбрать один из 3 ответов [1]
27. Нечеткие продукционные правила- это:... ыбрать один из 3 ответов 2]
28. Логическое программирование- это: ... Выбрать один из 3 ответов [3]
29. Примеры создания и применения систем, основанных на различных правилах. Выбрать один из 3 ответов [3]
30. Какие из систем, основаны на автоматическом доказательстве теорем. Выбрать один из 3 ответов 2]
31. Отличаются ли метод резолюций Дж. Робинсона и обратный метод С. Ю. Маслова? Выбрать один из 3 ответов [1]
32. В какой группе методы обучения с учителем? Выбрать один из 3 ответов [3]
33. В какой группе методы обучения без учителя? Выбрать один из 3 ответов [1]
34. Метод последовательного порождения гипотез –это:... Выбрать один из 3 ответов [3]
35. В каком списке находятся перспективные направления развития ИС и технологий? Выбрать один из 3 ответов [2]

Перечень вопросов итоговой аттестации по курсу

1. Основные понятия искусственного интеллекта (ИИ).
2. Классы систем интеллектуального управления.
3. Основные этапы развития ИС и технологий.
4. Роль ИС и технологий в современном управлении.

5. Основные интеллектуальные компоненты, применяемые в ИС.
6. Основные подходы и методы, используемые в современных ИС и технологиях.
7. Понятие об экспертных системах.
8. Виды нейронных сетей .
9. Понятие о системах, основанных на знаниях (СОЗ).
10. Формальные аксиоматические системы.
11. Понятие о логическом выводе.
12. Основные понятия классического исчисления предикатов.
13. Правила вывода исчисления предикатов.
14. Задачи поиска вывода в исчислении предикатов.
15. Понятие о методах порождения гипотез.
16. Понятие о языке L-позитивно образованных формул (ПО-формул).
17. Понятие о дедуктивных правилах.
18. Исчисление J ПО-формул, стратегии поиска вывода.
19. Логическое порождение гипотез.
20. Особенности получения, представления и использования знаний в ИС.
21. Модели представления знаний в ИС на базе продукционных правил.
22. Модели представления знаний в ИС на базе динамических семантических сетей.
23. Модели представления знаний в ИС на базе фреймовых и других представлений.
24. Прямые и обратные цепочки выводов.
25. Методы создания и особенности применения в управлении систем, основанных на правилах.
26. Нечеткие продукционные правила
27. Логическое программирование.
28. Системы, основанные на автоматическом доказательстве теорем.

- 29.Метод резолюций Дж. Робинсона и обратный метод С. Ю. Маслова.
- 30.Примеры создания и применения систем, основанных на автоматическом доказательстве теорем.
- 31.Системы, основанные на автоматическом выдвижении гипотез.
- 32.Основные методы обучения с учителем и без учителя.
- 33.Индуктивное логическое программирование.
- 34.Логические исчисления с обобщенными кванторами, GUHA – метод.
- 35.Метод получения правдоподобных рассуждений.
- 36.Метод последовательного порождения гипотез.
- 37.Примеры создания и применения систем, основанных на автоматическом выдвижении гипотез.
- 38.Системы, основанные на рассуждениях по аналогии.
- 39.Основные формы рассуждений.
- 40.Правдоподобные рассуждения.
- 41.Рассуждения по прецеденту.
- 42.Понятие “близости” к прецеденту.
- 43.Комбинации различных методов.
- 44.Примеры создания и применения систем, основанных на рассуждениях по аналогии.
- 45.Объектно-ориентированные ИС.
- 46.Использование декларативно-процедурных форм представления знаний и объектно-ориентированных языков программирования.
- 47.Примеры создания систем с применением различных языков.
- 48.Применение методов логического вывода для построения систем интеллектуального управления.
- 49.Использование логических языков в описании цели управления и доступных средств достижения цели.
- 50.Особенности построения и применения гибридных ИС.

51. Примеры создания и применения СОЗ, использующих объектно-логические языки, логики транзакций, фреймовые логики и другие.
52. Эффективность сочетания различных подходов.
53. Методы учета ограниченности ресурсов ИС (неполнота информации, динамика изменения данных и знаний).
54. Методы учета ограниченности ресурсов ИС (задачи мягкого и жесткого реального времени, объем памяти).
55. Методы учета ограниченности ресурсов ИС.
56. Методы и алгоритмы для формирования в ИС альтернативных управлений.
57. Особенности формирования “быстрых” приближенных решений и методы итеративного улучшения решений в рамках имеющихся ресурсов.
58. Основные проблемы повышения уровня интеллектуальности систем управления.
59. Недостатки традиционных подходов и методов интеллектуального управления.
60. Основные принципы повышения эффективности ИС.
61. Методы и подходы к оценке уровня интеллектуальности автоматических и автоматизированных управляющих систем.
62. Нижний, средний и верхний уровни управления в антропоцентрических системах.
63. Основные типы ИС и технологий, используемых при создании и применении ИС в задачах управления.
64. Особенности построения и применения ИС в бортовых системах управления подвижными объектами.
65. Особенности построения и применения ИС при управлении сложными производственными объектами.
66. Перспективные направления развития ИС и технологий.

ОПИСАНИЕ КУРСА И ПРОГРАММА

Цель и задачи курса

Целью курса является подготовка специалистов в области автоматизации, владеющих научными и методическими основами создания и применения современных интеллектуальных систем (ИС) и технологий для эффективного решения сложно формализуемых задач управления. Основным предметом изучения дисциплины являются интеллектуальные системы, а также методы, способы и технологии реализации ИС с использованием различных программно-технических средств.

Задачами курса являются:

- изучение и освоение основных понятий, подходов и методов, используемых в современных интеллектуальных системах;
- изучение основных способов представления в ИС знаний и методов их компьютерной обработки;
- освоение и умение создания и использования моделей процессов получения и обработки базовой информации ИС, реализации алгоритмов принятия решений и выработки алгоритмов управления;
- изучение основных технологий и интеллектуальных компонентов, применяемых в ИС, и особенностей использования в системах управления нейронных сетей, эволюционных (генетических) алгоритмов, пропозициональных и предикатных языков, нечеткой логики, символьных вычислений и систем, основанных на знаниях (СОЗ);
- получение навыков проектирования и применения отдельных интеллектуальных компонентов ИС, обеспечивающих возможности логического программирования, формирования баз данных (БД), баз знаний (БЗ) и построения экспертных систем (ЭС).

Областью знаний, в которой используются интеллектуальные системы и технологии являются информационные технологии и системные исследования в задачах управления.

Курс предназначен для обучения в бакалавриате.

Данная дисциплина предназначена для подготовки бакалавров по направлению «Автоматизация и управление» и специалистов по специальности «Управление и информатика в технических системах». Курс по выбору.

Курс является теоретическим, но предполагает получение практических навыков по разработке отдельных подсистем ИС с использованием типовых интеллектуальных компонентов и технологий.

Инновационность курса:

- по содержанию:

В курсе рассматриваются современные научные достижения по созданию и применению новых эффективных логических средств интеллектуального управления и соответствующий математический аппарат, рассматриваются новые подходы, основанные на языках предикатной логики, исчисления позитивно образованных формул и других современных подходов, методов и алгоритмов, развивающих традиционные подходы и методы построения ИС и технологий.

- по методике преподавания:

Используется компактное изложение теоретических и методических основ создания и применения интеллектуальных систем и технологий с использованием сочетания традиционных лекционных и практических занятий с активным использованием мультимедийных методов представления современных разработок ИС и применения их в области управления, а

также и обеспечения возможности коллективной работы студентов по освоению различных компонентов ИС на базе сетевых технологий.

- по литературе:

Используются данные и результаты различных исследований, выполненных многими авторами в последние годы в области создания и применения интеллектуальных систем и технологий и опубликованные в различных источниках - монографиях, в научных журнальных статьях и публикациях в Интернете. Краткое изложение классических подходов и идей создания ИС и технологий следует работам:

Пупков К.А., Коньков В.Г. Интеллектуальные системы // МГТУ им. Н.Э. Баумана. М. 2003;

Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е. Интеллектуальное управление динамическими системами // Физико-математическая литература. М. 2000;

Афонин В.А., Макушкин В.А. Интеллектуальные робототехнические системы // М. 2005. и др.

- по организации учебного процесса:

Введен курсовой проект, ориентированный на использование новейших разработок и технологий в области ИС.

Структура курса

Лекционные занятия – 36 часов, практические занятия – 28 часов, курсовая работа – 8 часов, внеаудиторные занятия, индивидуальные консультации и самостоятельная работа по темам – 36 часов.

Темы лекций:

Лекция 1 (2 часа).

Введение в курс. Основные понятия и терминология интеллектуальных систем (ИС). Краткий исторический очерк основных этапов развития ИС и технологий и их роль в современных технологиях управления.

Лекция 2 (2 часа).

Обзор основных интеллектуальных компонентов, применяемых в ИС. Особенности подходов и методов, используемых в современных ИС и технологиях. Динамические экспертные системы. Нейронные сети. Эволюционные алгоритмы. Понятие о системах, основанных на знаниях (СОЗ).

Лекция 3 (2 часа).

Основы логического вывода. Формальные аксиоматические системы. Основные понятия классического исчисления предикатов. Язык, аксиомы и правила вывода исчисления предикатов. Задачи поиска вывода в исчислении предикатов.

Лекция 4 (2 часа).

Некоторые методы порождения гипотез. Язык L-позитивно образованных формул (ПО-формул). Дедуктивные правила. Исчисление J ПО-формул. Стратегии поиска вывода в исчислении J. Логическое порождение гипотез.

Лекция 5 (2 часа).

Особенности получения, представления и использования знаний в ИС и экспертных системах (ЭС). Понятия о моделях представления знаний в ИС, построенных с использованием продукционных правил, динамических семантических сетей, универсальных сетей Петри, фреймовых и других представлений. Прямые и обратные цепочки выводов.

Лекция 6 (2 часа).

Методы создания и особенности применения в управлении систем, основанных на правилах. Нечеткие правила. Продукционные правила. Логическое программирование. Примеры создания и применения систем, основанных на различных правилах.

Лекция 7 (2 часа).

Системы, основанные на автоматическом доказательстве теорем. Метод резолюций Дж. Робинсона и обратный метод С. Ю. Маслова. Системы естественного вывода (генценовского типа). Примеры создания и применения систем, основанных на автоматическом доказательстве теорем.

Лекция 8 (2 часа).

Системы, основанные на автоматическом выдвижении гипотез. Основные методы. Обучение с учителем и без учителя. Индуктивное логическое программирование. Логические исчисления с обобщенными кванторами. GUHA – метод. Метод получения правдоподобных рассуждений. Метод последовательного порождения гипотез. Примеры создания и применения систем, основанных на автоматическом выдвижении гипотез.

Лекция 9 (2 часа).

Системы, основанные на рассуждениях по аналогии. Основные формы рассуждений. Правдоподобные рассуждения. Рассуждения по прецеденту. Понятие “близости” к прецеденту. Комбинации методов. Примеры создания и применения систем, основанных на рассуждениях по аналогии.

Лекция 10 (2 часа).

Объектно-ориентированные ИС. Использование декларативно-процедурных форм представления знаний и объектно-ориентированных языков программирования. Примеры создания систем с применением различных языков (ART, KRL, Smalltalk, KL-ONE, OPS5, Пролог и других).

Лекция 11 (2 часа).

Особенности применения методов логического вывода для построения систем интеллектуального управления. Использование логических языков в описании цели управления и доступных средств достижения цели. Определение плана достижения цели.

Лекция 12 (2 часа).

Особенности построения и применения гибридных ИС. Примеры создания и применения СОЗ, использующих объектно-логические языки, логики транзакций (Transaction Logics), фреймовые логики (F-logics) и другие. Эффективность сочетания различных подходов.

Лекция 13 (2 часа)..

Методы учета ограниченности ресурсов ИС (неполнота информации, динамика изменения данных и знаний, задачи мягкого и жесткого реального времени, объем памяти, возможности параллельной обработки информации).

Лекция 14 (2 часа).

Методы и алгоритмы (“Anytime Algorithms”) для формирования в ИС альтернативных управлений. Особенности формирования “быстрых” приближенных решений и методы итеративного улучшения решений в рамках имеющихся ресурсов

Лекция 15 (2 часа).

Основные проблемы повышения уровня интеллектуальности систем управления. Недостатки традиционных подходов и методов. Основные принципы повышения эффективности ИС.

Лекция 16 (2 часа).

Методы и подходы к оценке уровня интеллектуальности автоматических и автоматизированных управляющих систем. Особенности оценки структур иерархических систем управления. Нижний, средний и верхний уровни управления в антропоцентрических системах.

Лекция 17 (2 часа).

Обзор ИС и технологий, используемых при создании и применении ИС в задачах управления. Особенности построения и применения ИС в бортовых системах управления подвижными объектами и управлении сложными производственными объектами.

Лекция 18 (2 часа).

Перспективные направления развития ИС и технологий. Идеи, подходы, методы, алгоритмы и программно-технические средства для создания перспективных интеллектуальных систем и технологий.

Темы практических занятий:

Практическое занятие 1. – 4 часа

Ознакомление с инструментальными средствами создания экспертных систем (ЭС) с использованием стандартной оболочки ЭС и применения специализированных языков. Изучение особенностей применения языка Пролог для создания ЭС. Освоение работы с инструментальными средствами. Освоение способов подготовки и задания исходных данных, диалога с системой, вариантов представления результатов.

Практическое занятие 2. – 4 часа

Изучение моделей представления знаний в ИС, построенных с использованием продукционных правил. Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных. Практическое задание фактов, правил для формирования модели представления знаний в ЭС с использованием продукционных правил. Экспериментальная проверка работы модели.

Практическое занятие 3. – 4 часа

Изучение фреймовых моделей представления знаний в ИС. Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных. Практическое задание фактов, правил для формирования модели представления знаний в ЭС с использованием фреймовых представлений. Экспериментальная проверка работы модели.

Практическое занятие 4. – 4 часа

Изучение моделей представления знаний в ИС с использованием семантических сетей. Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных. Практическое задание фактов, правил для формирования модели представления знаний в ЭС с использованием семантических сетей. Экспериментальная проверка работы модели.

Практическое занятие 5. – 4 часа

Изучение основных режимов работы и возможностей конфигурирования и настройки диалоговой подсистемы ЭС. Изучения возможностей динамического изменения формы диалога, корректировки введенных фактов, правил и тезаурса. Экспериментальная проверка работы диалоговой подсистемы ЭС.

Практическое занятие 6. – 4 часа

Изучение методов вывода с использованием прямой и обратной цепочки рассуждений. Ознакомление с работой подсистемы объяснения вывода в ЭС. Анализ заданных преподавателем вариантов примеров исходных данных и формирование рабочих данных для подсистемы вывода и подсистемы объяснений. Экспериментальная проверка работы подсистем вывода и объяснений.

Практическое занятие 7. – 4 часа

Ознакомление с инструментальными средствами программной среды G2. Освоение работы с инструментальными средствами G2. Анализ заданного преподавателем варианта примера исходных данных и формирование рабочих данных в среде G2. Исследование эффективности применения программной среды G2 на примере заданного преподавателем варианта.

Описание системы контроля знаний

В курсе « Интеллектуальные системы » предусматривается цикл лекций, практические занятия (лабораторные работы) и курсовая работа.

Система контроля знаний предусматривает: контроль посещения аудиторных занятий - лекционных, практических и лабораторных работ, подготовку отчетов и индивидуальную защиту выполненных лабораторных работ с применением компьютерного блиц-контроля, контроль поэтапного выполнения курсовой работы. Повышенными баллами оценивается своевременное и качественное выполнение лабораторных работ, курсовой работы, учитываются данные блиц-контроля и итогового испытания.

По итогам контроля в конце каждого месяца проводится промежуточная бально-рейтинговая аттестация студентов, результаты аттестации размещаются на учебном портале.

Правила выполнения письменных работ (курсовых, лабораторных):

Список тем курсовых работ предлагается студентам в начале учебного семестра. Студент имеет право выбрать тему (или раздел общей темы) из данного списка или по согласованию с преподавателем предложить свою. Курсовая работа выполняется, сдается на проверку и защищается в сроки, указанные в календарном плане.

Требования к оформлению работ: полуторный интервал, кегль — 14, цитирование литературы и сноски в соответствии с принятыми стандартами, с проверкой грамматики, орфографии, синтаксиса.

Курсовая работа должна содержать обзорную часть проблемы, иметь четкую техническую и математическую постановку задачи, содержать теоретические исследования проблемы, описание работы алгоритмов, иметь материалы математического моделирования или экспериментального исследования

Текст отчета о лабораторной работе должен содержать краткую теоретическую и развернутую характеристику выполненной практической части, с подробными комментариями ко всем этапам моделирования (таблицы, графики), объем не менее 4—6 страниц.

Балльная структура оценки:

Посещение лекций: 0- 20 баллов

Практические занятия (лабораторные работы): 0-40 баллов

Курсовая работа: 0 - 20 баллов

Итоговое испытание: 0 - 20 баллов

Всего - 100 баллов

Шкала оценок:

| Баллы за семестр | Автоматическая оценка |
|------------------|-----------------------|
| 91-100 | 5 |
| 76-90 | 4 |
| 56-75 | 3 |
| 35-55 | - |
| <35 | - |

Студенты, получившие положительные оценки по результатам работы в семестре, но претендующие на получение более высокой оценки, могут участвовать в сдаче экзаменов в период сессии. Количество баллов за экзамен от 0 до 25 баллов.

Студенты, набравшие в течение семестра 35-55 баллов, обязаны пройти итоговую семестровую аттестацию в установленном порядке.

Студенты, не выполнившие программу изучаемой дисциплины, не выполнившие полностью комплекс практических и лабораторных работ и/или не набравшие 35 баллов, не допускаются до прохождения итоговой семестровой аттестации.

Академическая этика

Материалы сторонних авторов, опубликованные в открытой печати и использованные в содержании курса, будут иметь ссылки на соответствующие источники. В содержание курса в описаниях и в мультимедийных разработках представления основных подходов и методов в области создания и применения ИС используются оригинальные научные и практические разработки автора УМК.

ПРОГРАММА КУРСА

Аннотированное содержание курса: (общая трудоемкость – 72 часа, 2 кредита)

1. Лекции – 36 часов

Введение

Во введении в курс рассматриваются основные понятия и терминология интеллектуальных систем (ИС). Дается краткий исторический очерк основных этапов развития ИС и технологий, указывается их место в современных технологиях управления различными объектами. Рассматриваются основные классы ИС, используемые в них подходы, модели и методы, а также основные технологии и программно-аппаратные средства, применяемые для практической реализации ИС. Для дальнейшего рассмотрения в курсе выделяются классы интеллектуальных систем, используемых для решения задач управления техническими объектами.

Основное содержание.

Дается обзор основных интеллектуальных компонентов, применяемых в ИС. Излагаются особенности основных подходов и методов, используемых в современных ИС и технологиях, включая динамические экспертные системы, нейронные сети, эволюционные (генетические) алгоритмы и системы, основанные на знаниях (СОЗ). Кратко рассматриваются основы логического вывода и задачи поиска вывода в исчислении предикатов, а также основные методы логического порождения гипотез. Дается понятие о языке L-позитивно образованных формул (ПО-формул), о исчислении J ПО-формул и стратегии поиска вывода в исчислении J.

Излагаются особенности получения, представления и использования знаний в ИС и экспертных системах, построенных с использованием различных моделей представления знаний и реализующих прямые и обратные цепочки выводов. Рассматриваются основные методы создания ИС, особенности их применения в управлении и приводятся примеры создания и

применения систем, основанных на различных подходах (ИС, основанные на правилах, на автоматическом доказательстве теорем, на автоматическом выдвижении гипотез, на рассуждениях по аналогии). Дается представление об объектно-ориентированных ИС, использующих декларативно-процедурные формы представления знаний и различные объектно-ориентированные языки программирования (ART, KRL, Smalltalk, KL-ONE, OPS5, Пролог и другие).

Дается анализ особенностей применения методов логического вывода для построения систем интеллектуального управления и использования логических языков для описания цели управления и доступных средств достижения цели. Рассматриваются особенности построения и применения гибридных ИС и приводятся примеры создания и применения СОЗ, использующих сочетания различных подходов (объектно-логические языки, логики транзакций, фреймовые логики и другие). Дается представление о методах учета ограниченности ресурсов ИС, методах и алгоритмах формирования в ИС альтернативных управлений, а также об основных проблемах повышения уровня интеллектуальности обычных и иерархических систем управления.

На основе обзора основных ИС и технологий, используемых в системах управления, рассматриваются особенности построения и применения ИС в бортовых системах управления подвижными объектами и управлении сложными производственными объектами, а также выделяются некоторые перспективные направления развития ИС и технологий для создания интеллектуальных систем и технологий нового поколения.

2. Практические занятия и курсовая работа – 36 часов

2.1 Практические занятия -28 часов

Для получения навыков и умения решать основные задачи создания и применения ИС в программе курса предусматривается проведение 7 практических занятий по 4 часа каждое. На этих занятиях студенты познакомятся с некоторыми инструментальными средствами создания экспертных систем с использованием стандартной оболочки ЭС, программной средой G2 и применением специализированного языка Пролог.

Используя подготовленные преподавателем варианты примеров исходных данных студенты освоят основные способы подготовки и задания рабочих данных для различных моделей представления знаний в ИС, научатся проведению диалога с системой, заданию вариантов представления результатов. Предусмотрено также изучение основных режимов работы и возможностей конфигурирования и настройки диалоговой подсистемы ЭС, изучение методов вывода с использованием прямой и обратной цепочки рассуждений, а также ознакомление с работой подсистемы объяснения вывода в ЭС и проведение исследования эффективности применения программной среды G2 на примере заданного преподавателем варианта.

2.2 Курсовая работа – 8 часов

Темы курсовых работ:

1. Исследование и разработка структуры интеллектуальной системы поддержки принятия решений (ИСППР) для управления промышленным объектом.
2. Разработка для ИСППР экспертной подсистемы анализа запросов операторов при управлении промышленным объектом.
3. Разработка для ИСППР экспертной подсистемы анализа событий и оценки текущей ситуации на промышленном объекте.

4. Разработка для ИСППР экспертной подсистемы прогноза развития ситуации на промышленном объекте.
5. Разработка для ИСППР экспертной подсистемы формирования сценариев имитационного моделирования работы промышленного объекта.
6. Разработка монитора реального времени для координации работы нескольких экспертных систем в ИСППР.
7. Разработка для ИСППР экспертной подсистемы формирования данных для работы подсистемы имитационного моделирования.
8. Разработка монитора реального времени для управления подсистемой имитационного моделирования в ИСППР.
9. Разработка для ИСППР экспертной подсистемы для анализа результатов работы подсистемы имитационного моделирования.
10. Разработка для ИСППР экспертной подсистемы формирования и оценки вариантов альтернативного управления работой промышленного объекта.
11. Разработка для ИСППР экспертной подсистемы формирования данных для ответа на запрос оператора, управляющего промышленным объектом.
12. Разработка экспертной подсистемы анализа аварийных ситуаций с использованием инструментальной программной среды G2.

Список обязательной литературы

1. Захаров В.Н. Интеллектуальные системы управления: основные понятия и определения // Известия РАН. Теория и системы управления. 1997. №3. с.138-145.
2. Пупков К.А., Коньков В.Г. Интеллектуальные системы // МГТУ им. Н.Э. Баумана. М. 2003.
3. Нильсон Н. Принципы искусственного интеллекта // Радио и связь. М. 1985.
4. Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е. Интеллектуальное управление динамическими системами. // Физико-математическая литература. М. 2000.
5. Уотермен Д. Руководство по экспертным системам // Мир. М. 1989.
6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем // Питер. С-Пб. 2000.
7. Материалы Российской ассоциации искусственного интеллекта // <http://www.raai.org/resurs/resurs.shtml?ilinks>.

Список дополнительной литературы

1. Лорьер Ж.Л. Системы искусственного интеллекта // - М.: Мир, 1991
2. Мальковский М.Г., Грацианова Т.Ю., Полякова И.Н. Прикладное программное обеспечение: системы автоматической обработки текстов // МГУ. М.: 2000.
3. Непейвода Н.Н. Логический подход как альтернатива системному в математическом описании систем // Изд-во ЛИИ АН СССР. 1989. №85.
4. Пильщиков В.Н. Язык ПЛЭНЕР. Наука, М. 1983.
5. Приобретение знаний. Под ред. С. Осуга. Мир. М. 1990.
6. Рыбина Г.В. Автоматизированное рабочее место для построения интегрированных экспертных систем: комплекс АТ-ТЕХНОЛОГИЯ // "Новости Искусственного Интеллекта", №3, 2005

7. Семенов М.Ю. Язык ЛИСП для персональных ЭВМ // МГУ. М.1989.
8. Сотник С. Л. Конспект лекций по курсу "Основы проектирования систем искусственного интеллекта" // <http://neuroschool.narod.ru/sotnik.html> 1998.
9. Тихомиров О.К. Психология мышления // МГУ. М. 1984.
10. Уэно Х. и др. Представление и использование знаний // Мир. М. 1989
11. Хорошевский В.Ф., Шерстнев В.Ю. Программный инструментарий представления знаний в экспертных системах // Экспертные системы: состояние и перспективы. Наука. М. 1989. с.38-46.

УЧЕБНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН КУРСА УМК

(календарный план, структурированный по видам учебных занятий)

1-й проректор, проректор по учебной работе

А.П. Ефремов

по направлению 550200 (220200)- "АВТОМАТИЗАЦИЯ
И УПРАВЛЕНИЕ" ИУб+д (бакалавры)

" " _____ 2007 г.

| Дисциплины и кафедры | Распределение по семестрам | | | | | | 7 семестр | |
|-------------------------------------------------|----------------------------|-----|------|-----|-----|-----|-----------|------|
| | экс | зач | курс | кон | реф | нед | студ | груп |
| <i>Наименование дисциплин</i> | | | | | | 18 | 59 | 4 |
| <i>Дисциплины по выбору (1 из 2)</i> | | | | | | лкц | лаб | пр |
| Интеллектуальные системы и технологии | 7 | | | | | 2 | 2 | |
| Ауд. час. по видам занятий в нед. | | | | | | | 2 | 1 |
| Всего ауд. часов в неделю (без физ-ры и факул.) | | | | | | 2 | 2 | |
| Экзаменов | | | | | | | | 1 |
| Зачетов | | | | | | | | |
| Курсовых работ / проектов | | | | | | | | 1 |

Примечания: