

## **Л 1. Цели и задачи курса. История развития и основные направления искусственного интеллекта**

**Цели и задачи курса. Структура дисциплины. История развития ИИ. Этап 1 (50-60-е годы). Этап 2 (60-70-е годы). Этап 3 (70-80-е годы). Этап 4 (80-90-е годы). Этап 5 (1990-2000-е годы). Основные направления развития ИИ.**

См. презентацию.

### **НАЧАЛЬНЫЙ ЭТАП — ЭВРИСТИЧЕСКИЕ ПРОГРАММЫ**

Начало исследований в области искусственного интеллекта относится к концу 50-х годов и связывается с работами Ньюэлла, Саймона и Шоу [Newell. 1972], исследовавших процессы решения различных задач. Результатами их работ явились такие программы, как Логик-Теоретик, предназначенная для доказательства теорем в исчислении высказываний, и Общий Решатель Задач. Эти работы положили начало первому этапу исследований в области искусственного интеллекта, связанному с разработкой программ, решающих задачи на основе применения разнообразных эвристических методов.

Эвристический метод решения задачи при этом рассматривался как свойственный человеческому мышлению "вообще", для которого характерно возникновение "догадок" о пути решения задачи с последующей проверкой их. Ему противопоставлялся используемый в ЭВМ алгоритмический метод, который интерпретировался как механическое осуществление заданной последовательности шагов, детерминировано приводящей к правильному ответу. Трактовка эвристических методов решения задач как сугубо человеческой деятельности и обусловила появление и дальнейшее распространение термина *искусственный интеллект*. Исследователям того времени казалось, что создание компьютера, обладающего человеческим разумом, вопрос 10-15 лет.

В это же время был сформирован бионический подход к ИИ. Он был представлен только простейшими нейронными сетями, поэтому не получил своего развития. Другие направления (генетические алгоритмы, нечеткая логика и др.) датируют свое появление тем же промежутком времени, но тогда они были мало популярны и также не нашли области применения.

Примерно в то время, когда работы Ньюэлла и Саймона стали привлекать к себе внимание, в Массачусетском технологическом институте, Стэнфордском университете и Стэнфордском исследовательском институте также сформировались исследовательские группы в области ИИ. В противоположность ранним работам Ньюэлла и Саймона эти исследования больше относились к формальным математическим представлениям. Способы решения задач в этих исследованиях развивались на основе расширения математической и символьной логики. Моделированию же человеческого мышления придавалось второстепенное значение. К исследователям этого направления можно отнести таких известных в области ИИ ученых, как Минский, Мак-Карти, Слейгл, Рафаэль, Бобров, Бенерджи и др.

На дальнейшие исследования в этой области ИИ большое влияние оказало появление метода резолюции, предложенного Робинсоном, основанного на доказательстве теорем в логике предикатов и являющегося теоретически исчерпывающим методом доказательства.

Методологическое значение работ Робинсона и других аналогичных работ заключалось в том, что основное внимание в исследованиях по ИИ переместилось с разработки методов воспроизведения в ЭВМ человеческого мышления на разработку машинно-ориентированных методов решения задач.

При этом определение термина "искусственный интеллект" претерпело существенное изменение. Целью исследований, проводимых в направлении ИИ, стало не моделирование способов мышления человека, а разработка программ, способных решать "человеческие задачи". Так, один из видных исследователей ИИ того времени Р. Бенерджи в 1969 г. писал: "Область исследований, обычно называемую искусственным интеллектом, вероятно, можно представить, как совокупность методов и средств анализа, конструирования машин, способных выполнять задания, с которыми до недавнего времени мог справиться только человек. При этом по скорости и эффективности машины должны быть сравнимы с человеком" [Бенерджи. 1972].

Исследовательским полигоном для развития методов ИИ на первом этапе являлись все возможные игры, головоломки, математические задачи. Выбор таких задач для исследований обуславливался простотой и ясностью проблемной среды, ее относительно малой размерностью, возможностью достаточно легко подбора решений. В то же время такие среды подходили для моделирования достаточно сложных процессов решения и исследования всевозможных стратегий решения с относительно небольшими затратами как человеческих, так и машинных ресурсов.

Основной расцвет такого рода исследований приходится на конец 60-х годов, после чего стали делаться первые попытки применения разработанных методов для задач, решаемых не в искусственных, а в реальных проблемных средах. Однако такие попытки натолкнулись на большие трудности, обусловленные главным образом необходимостью моделирования внешнего мира. Эти трудности были связаны с проблемами описания знаний о внешнем мире, организации их хранения, эффективного поиска, введения в память ЭВМ новых знаний и устранения устаревших, проверки полноты и непротиворечивости и т. п.

## **ВТОРОЙ ЭТАП — ИНТЕГРАЛЬНЫЕ РОБОТЫ**

Необходимость исследования систем искусственного интеллекта при их функционировании в реальном мире привела к постановке задачи создания интегральных роботов. При разработке проектов таких роботов использование термина "искусственный интеллект" стало звучать более обоснованно, так как в них решались не отдельные задачи ИИ, а исследовался и реализовывался необходимый спектр "интеллектуальных" функций, таких, как организация целенаправленного поведения, восприятие информации о внешней среде, формирование действий, обучение, общение с человеком и другими роботами.

Для формирования целенаправленного поведения, т.е. программы решения некоторой внешней по отношению к роботу задачи, он должен обладать необходимым комплексом знаний об окружающем мире и среде функционирования. Эти знания должны быть заложены в робота в виде модели проблемной среды, т.е. той части внешнего мира, которая существенна для решения задач, ставящихся перед роботом.

Проведение работ, связанных с созданием интегральных роботов, можно считать вторым этапом исследований по искусственноому интеллекту.

В нескольких научно-исследовательских институтах были разработаны экспериментальные роботы, функционирующие в лабораторных условиях. Проведение экспериментов в реальных условиях показало необходимость кардинального пересмотра вопросов, связанных с проблемой представления знаний о среде функционирования; и недостаточную исследованность таких проблем, как зрительное восприятие, построение сложных планов поведения в динамических средах, общение с роботами на естественном языке и т.д.

## **ТРЕТИЙ ЭТАП — ЭКСПЕРТНЫЕ СИСТЕМЫ**

Проблемы интеллектуальных роботов были более или менее ясно сформулированы и поставлены перед исследователями в середине 70-х годов и связаны с началом третьего этапа исследований систем ИИ. Его характерной чертой явилось смещение центра внимания исследователей от создания автономно функционирующих систем, к созданию человеко-машинных систем, интегрирующих в единое целое интеллект человека и способности вычислительных машин.

Такое смещение обусловливалось двумя причинами:

во-первых, к этому времени выяснилось, что даже простые на первый взгляд задачи, возникающие перед интегральным роботом при его функционировании в реальном мире (например, движение по пересеченной местности, распознавание объектов на сложном фоне с естественным освещением, организация сложного поведения и т. п.), не могут быть решены методами, разработанными для экспериментальных задач в специально сформированных проблемных средах;

во-вторых, стало ясно, что сочетание дополняющих друг друга возможностей человека и ЭВМ позволяет "обойти острые углы" путем перекладывания на человека тех функций, которые пока еще недоступны для ЭВМ. Вычислительная машина, со своей стороны, способна

обрабатывать большие объемы информации с использованием регулярных методов, многократно просматривать различные пути решения, предлагаемые человеком, предоставлять ему всевозможную справочную информацию.

На первый план выдвигалась не разработка отдельных методов машинного решения задач, а разработка методов и средств, обеспечивающих тесное взаимодействие человека и вычислительной системы в течение всего процесса решения задачи с возможностью оперативного внесения человеком изменений в ходе этого процесса.

Развитие исследований ИИ в данном направлении обусловливалось также резким ростом производства средств вычислительной техники и таким же резким их удешевлением, делающим их потенциально доступными для более широких кругов пользователей. Однако эта доступность для большинства реальных пользователей так и осталась "потенциальной", поскольку требовала для реализации овладения большими объемами специальных знаний по использованию ЭВМ.

Создание человеко-машинных систем ИИ нашло свое наиболее яркое выражение в экспертных системах (ЭС). Для их реализации разрабатывались многочисленные модели и языки представления знаний, специальные языки программирования и символьные ЭВМ.

После появления первых ЭС и интенсивного их развития в литературе появились утверждения о том, что ЭС не нашли своего применения и не оправдали надежд. Некоторые также утверждали, что ЭС систем не существует вообще, что это обычные расчетно-логические программы.

Причины первого заблуждения состоят в том, что ЭС рассматривается как альтернатива традиционному программированию, т.е. ЭС должна в изоляции от других программных средств решать задачи, стоящие перед заказчиком. Надо отметить, что на заре появления ЭС специфика используемых в них языков, технологии разработки приложений и используемого оборудования (например, Lisp-машины) давала основания предполагать, что интеграция ЭС с традиционными, программными системами является сложной и, возможно, невыполнимой задачей при ограничениях, накладываемых реальными приложениями. Однако в настоящее время коммерческие инструментальные средства для создания ЭС разрабатываются в полном соответствии с современными технологическими тенденциями традиционного программирования, что снимает проблемы, возникающие при создании интегрированных приложений.

Причина второго заблуждения вытекает из того, что любой программный продукт является эргодический (человеко-машинной) системой, и создают его, по большей мере, эксперты и специалисты. Все программы (кроме некоторых компьютерных игр) можно отнести к системам поддержки деятельности человека. Каждый программный продукт реализует вычисление формул и алгоритмов и, следовательно, обладает, как минимум, базой знаний и машиной вывода. Известно, что гипертекстовые системы CASE-системы и многие другие классы программ впервые сформировались именно в ЭС. Поэтому можно справедливо утверждать, что большинство современных программ включают в себя элементы (компоненты) ЭС. Однако, обратное утверждение, что любая программа является ЭС, неверно. Можно предположить, что в ближайшем будущем многие компьютерные программы будут содержать интеллектуальные компоненты, близкие по своей структуре и технологиям к экспертным системам.

Появление ЭС сыграло важную роль в развитии ИИ, т.к. позволило перевести разработки в области ИИ из исследовательской плоскости в область реализации практических программных комплексов. Первые успехи систем ИИ для решения коммерческих задач можно отнести к 1985 году, однако массовое распространение они получили только в середине 90-х годов. Так, например, коммерческий рынок продуктов искусственного интеллекта в мире в 1993 году оценивался примерно в 0.9 млрд. долларов и разделялся на следующие основные направления [Попов. 1995]:

- экспертные системы (системы, основанные на знаниях);
- нейронные сети и нечеткая логика;
- естественно-языковые системы.

В США в 1993 году рынок между этими направлениями распределился так: экспертные системы - 62%, нейронные сети и нечеткая логика - 26%, естественно-языковые системы - 12%. Рынок этот можно разделить и иначе: на системы искусственного интеллекта (приложения) и

инструментальные средства, предназначенные для автоматизации всех этапов существования приложения. Например, на рынке США доля приложений составила примерно две трети, а доля инструментария - одну треть.

Экспертные системы, показав практическую ценность, оказали большую услугу всей области искусственного интеллекта. Увеличение финансирования позволило породить новые и оживить старые направления ИИ.

Большие успехи в области ЭС (инженерии знаний) привели к идеи постепенной интеллектуализации машинных функций, направленной в конечном итоге на создание автономных систем ИИ - интеллектуальных агентов. Одними из основных проблем, препятствующих созданию автономных ИС, являются ограниченность базы знаний, модели представления знаний и несовершенство интерфейса взаимодействия с окружающей средой и пользователями.

## ЧЕТВЕРТЫЙ ЭТАП — НЕЙРОННЫЕ СЕТИ

Отправной точкой четвертого этапа развития в области ИИ можно назвать резкое увеличение интереса к бионическому направлению и, в первую очередь, к нейронным сетям в середине-конце 80-х годов. Это было обусловлено несколькими причинами. Во-первых, сложности представления знания и обучения в ЭС заставили обратиться к более пристальному и подробному изучению того, как это делает человек.

Во-вторых, нейробиологи и нейроанатомы к этому времени достигли значительного прогресса. Усердно изучая структуру и функции нервной системы человека, они значительно расширили свои знания об организации восприятия, мышления, памяти и моторике.

В-третьих, к этому времени был решен ряд теоретических проблем в области обучения многослойных нейронных сетей, сформулированных в конце 60-х годов, и преодолен ряд технологических вопросов. Немаловажную роль сыграл и коммерческий интерес к ИС, который был направлен на поиск инновационных подходов и подогрет возможностью патентования аппаратной реализации нейросетевых парадигм.

Одной из проблем ЭС в частности, и ИС в целом является сложность в приобретении новых знаний, обучение. Вопросами обучения и понимания традиционно занималась психология. Развитие вычислительной техники, появление теории информации, интеллектуализация компьютеров оказали сильное влияние на гуманитарные науки и не только как инструментарий. Появились, например, такие науки (направления) как когнитивная психология и компьютерная лингвистика.

Искусственные нейронные сети (НС) имитируют естественный прототип, который играет главенствующую роль в организации высшей нервной деятельности человека и его интеллектуальных способностях.

Лучшее понимание функционирования нейрона и картины его связей позволило исследователям создать математические модели для проверки своих теорий. Появилась возможность проводить эксперименты на цифровых компьютерах без привлечения человека или животных, что решает многие практические и морально-этические проблемы. В первых же работах выяснилось, что эти модели не только повторяют свойства мозга, но и способны выполнять функции, имеющие свою собственную ценность. Поэтому возникли и остаются в настоящее время две взаимно обогащающие друг друга цели нейронного моделирования: первая - понять функционирование нервной системы человека на уровне физиологии и психологии и вторая - создать вычислительные системы (искусственные нейронные сети), выполняющие сходные с человеческим мозгом функции.

Параллельно с прогрессом в нейроанатомии и нейрофизиологии психологами были созданы модели человеческого обучения. Одной из таких моделей, оказавшейся наиболее плодотворной, была модель Д. Хэбба [Hebb, 1949], который в 1949 г. предложил закон обучения, явившийся стартовой точкой для алгоритмов обучения искусственных нейронных сетей. Дополненный сегодня множеством других методов он продемонстрировал ученым того времени, как сеть нейронов может обучаться.

В пятидесятые и шестидесятые годы группа исследователей, объединив эти биологические и физиологические подходы, создала первые искусственные нейронные сети. Выполненные первоначально как электронные сети, они были позднее перенесены в более гибкую среду

компьютерного моделирования, сохранившуюся и в настоящее время. Первые успехи вызвали взрыв активности и оптимизма. Минский. Розенблatt, Уидроу [Widrow, 1959 — цит. по XXXXXX. 1995] и другие разработали сети, состоящие из одного слоя искусственных нейронов. Часто называемые персепtronами, они были использованы для такого широкого класса задач, как предсказание погоды, анализ электрокардиограмм и искусственное зрение. В течение некоторого времени казалось, что ключ к интеллекту найден, и воспроизведение человеческого мозга является лишь вопросом конструирования достаточно большой сети.

Но эта иллюзия скоро рассеялась. Сети не могли решать задачи, внешне весьма сходные с теми, которые они успешно решали. С этих необъяснимых неудач начался период интенсивного анализа. Минский, используя точные математические методы, строго доказал ряд теорем, относящихся к функционированию нейронных сетей.

Его исследования привели к написанию книги, в которой он вместе с Пайпертом доказал, что используемые в то время однослойные сети теоретически неспособны решить многие простые задачи, в том числе реализовать функцию «Исключающее ИЛИ». Минский также не был оптимистичен относительно потенциально возможного здесь прогресса.

Персепtron показал себя заслуживающим изучения, несмотря на жесткие ограничения. У него много привлекательных свойств: линейность, теорема об обучении, простота модели параллельных вычислений.

Блеск и строгость аргументации Минского, а также его престиж породили огромное доверие к книге - ее выводы были неуязвимы. Разочарованные исследователи остались поле исследований ради более обещающих областей, а правительства перераспределили свои субсидии, и искусственные нейронные сети были забыты почти на два десятилетия.

Тем не менее, несколько наиболее настойчивых ученых, таких как Кохонен. Гроссберг, Андерсон продолжили исследования. Наряду с плохим финансированием и недостаточной оценкой ряд исследователей испытывал затруднения с публикациями. Поэтому исследования, опубликованные в семидесятых и начале восьмидесятых годов, разбросаны в массе различных журналов, некоторые из которых малоизвестны. Постепенно появился теоретический фундамент, на основе которого сегодня конструируются наиболее мощные многослойные сети. Оценка Минского оказалась излишне пессимистичной, многие из поставленных в его книге задач решаются сейчас сетями с помощью стандартных процедур.

За несколько лет теория стала применяться в прикладных областях, появились новые корпорации, занимающиеся коммерческим использованием этой технологии. Нарастание научной активности носило взрывной характер. В 1987 г. было проведено четыре крупных совещания по искусственным нейронным сетям и опубликовано свыше 500 научных сообщений - феноменальная скорость роста.

С одной стороны, блестящая научная работа Минского задержала развитие искусственных нейронных сетей, но с другой стороны, нет сомнений в том, что область пострадала вследствие необоснованного оптимизма и отсутствия достаточной теоретической базы. И возможно, что шок, вызванный книгой «Персептроны», обеспечил необходимый для созревания этой научной области период.

В настоящее время имеется много впечатляющих демонстраций возможностей искусственных нейронных сетей: сеть научили превращать текст в фонетическое представление, которое затем с помощью уже иных методов превращалось в речь; другая сеть может распознавать рукописные буквы: сконструирована система сжатия изображений, основанная на нейронной сети. Все они используют сеть обратного распространения - наиболее успешный, по-видимому, из современных алгоритмов. Обратное распространение, независимо предложенное в трех различных работах, является систематическим методом для обучения многослойных сетей, и тем самым преодолевает ограничения, указанные Минским.

Разработано много других сетевых алгоритмов обучения, имеющих свои специфические преимущества. Следует подчеркнуть, что ни одна из сегодняшних сетей не является панацеей, все они страдают от ограничений в своих возможностях обучаться и вспоминать.

Область нейронных сетей, продемонстрировала свою работоспособность и имеет уникальные потенциальные возможности, но в то же время и много ограничений, множество открытых вопросов. Такая ситуация настраивает на умеренный оптимизм. Существует, однако, опас-

ность, что искусственные нейронные сети начнут продавать раньше, чем придет их время, обещая функциональные возможности, которых пока невозможно достичнуть. Если это произойдет, то область в целом может пострадать от потери кредита доверия и вернется к застойному периоду семидесятых годов. Для улучшения существующих сетей требуется много основательной работы. Должны быть развиты новые технологии, улучшены существующие методы и расширены теоретические основы, прежде чем данная область сможет полностью реализовать свои потенциальные возможности.

Искусственные нейронные сети предложены для задач, простирающихся от управления боем до присмотра за ребенком. Потенциальными приложениями являются те, где человеческий интеллект малоэффективен, а обычные вычисления трудоемки или неадекватны. Этот класс приложений не меньше класса, обслуживаемого обычными вычислениями, и можно предполагать, что искусственные нейронные сети займут свое место наряду с обычными вычислениями в качестве дополнения такого же объема и важности.

Одним из перспективных направлений развития нейронных сетей является их интеграция с экспертными системами и другими ИС.

Прежде чем искусственные нейронные сети можно будет использовать там, где поставлены на карту человеческая жизнь или ценное имущество, должны быть решены вопросы, относящиеся к их надежности. Подобно людям, структуру мозга которых они копируют, искусственные нейронные сети сохраняют в определенной мере непредсказуемость. Единственный способ точно знать выход состоит в испытании всех возможных входных сигналов. В большой сети такая полная проверка практически неосуществима, и должны использоваться статистические методы для оценки функционирования.

Одной из ключевых проблем НС заключается в их неспособности "объяснить", как они решают задачу\*. Внутреннее представление, получающееся в результате обучения, часто настолько сложно, что его невозможно проанализировать, за исключением самых простых случаев. Это напоминает неспособность объяснить, как происходит процесс узнавания человека, несмотря на различие в расстоянии, угле, освещении и на прошедшие годы. Эта особенность НС противопоставляется возможностям экспертных систем, которые могут проследить процесс своих рассуждений в обратном порядке для проверить разумности полученного результата.

Нейронные сети традиционно считаются второй волной интеллектуальных программ, которая постепенно оттесняет ЭС и занимает значительное место на рынке систем искусственного интеллекта. Наибольшее распространение нейронные сети получили в прогнозирующих и аналитических программах. Меньшее распространение они получили в системах распознавания. Тем не менее, задачу обучения новым знаниям нейронные сети не могут решать эффективно. Причины этого сложность представления и организации знаний (значительные размеры сети), трудность извлечения знаний и неоптимальные процедуры (алгоритмы) обучения.

## **ПЯТЫЙ ЭТАП — НЕЧЕТКАЯ ЛОГИКА**

Начало пятого этапа развития области ИИ связано с активным применением математического аппарата нечеткой логики (НЛ) для решения практических и теоретических задач. По времени увеличение внимания к этому направлению совпадает с четвертым этапом, однако пик интереса смешен по времени на 5-10 лет вперед (начало-середина 80-х).

При этом основы нечеткой логики были заложены в конце 60-х годов в трудах известного математика Заде [Заде. 1976]. В ту пору весьма популярными были эксперименты с "мажоритарными" пространствами, в которых намеренно устранилось понятие меры и вместо него вводился ряд качественных факторов (типа квантора "большинства") - прообраз первых нечетких утверждений. Аналогичное явление можно наблюдать в модальных логиках. Социальный заказ на исследования подобного рода был вызван растущим недовольством экспертными системами и связан с отсутствием математического аппарата, переводящего невнятные и неоднозначные житейские утверждения на язык четких математических формул. Первым серьезным шагом в этом направлении явилась теория нечетких множеств, разработанная Заде. Он же дал и название для новой области науки -"fuzzy logic".

Аппарат теории нечетких множеств, продемонстрировав ряд многообещающих возможностей применения - от систем управления летательными аппаратами до прогнозирования итогов выборов, оказался вместе с тем чрезмерно сложен для воплощения при тогдашнем уровне технологии - и на многие годы нечеткая логика заняла свое место в ряду других специальных научных дисциплин - где-то посередине между экспертными системами и нейронными сетями.

Свое второе рождение теория нечеткой логики пережила в начале восьмидесятых годов, когда сразу несколько групп исследователей (в основном в США и Японии) всерьез занялись созданием электронных систем различного применения, использующих нечеткие управляющие алгоритмы. Теоретические основы для этих попыток были заложены в ранних трудах Коцко [Kosko. 1994. 1997] и других ученых. Наибольшую роль сыграли, пожалуй, два научных результата: доказательство FAT-теоремы и комбинация нечеткой логики с нейронными сетями Кохонена, указавшая путь к преодолению наиболее критического "узкого места" новой теории - автоматизированного формирования системы нечетких правил по содержимому входных данных. Теорема FAT (Fuzzy Approximation Theorem), доказывает, что любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике.

К 90-му году появилось около 40 патентов, относящихся к нечеткой логике. Сорок восемь японских компаний образовали совместную лабораторию LIFE (Laboratory for International Fuzzy Engineering), японское правительство финансировало 5-летнюю программу по нечеткой логике, включающую 19 различных проектов - от систем оценки глобального загрязнения атмосферы и предсказания землетрясений до АСУ заводских цехов и складов [Intemetl, 2002].

Результатом выполнения этой программы явилось появление целого ряда новых массовых микрочипов, основанных на нечеткой логике. Сегодня их можно найти в стиральных машинах и видеокамерах, цехах заводов и моторных отсеках автомобилей, в системах управления складскими роботами и боевыми вертолетами.

Новый подход к представлению знаний не только повысил эффективность многих технических решений (фаззи-контроллеров), но и упростил задание правил в системах, основанных на знаниях. Нечеткие ЭС явились третьей волной коммерческих интеллектуальных систем.

Например, в США развитие нечеткой логики идет по пути создания систем, служащих большому бизнесу и военной промышленности. Нечеткая логика применяется при анализе новых рынков, биржевой игре, опенке политических рейтингов, выборе оптимальной ценовой стратегии и т.п. Появились и коммерческие системы массового применения (CubiCalc, FEDE, FuziCalc и др.).

В настоящее время НЛ активно используется не только в СИИ, но и во многих традиционных информационных системах. Появились и стали активно развиваться другие математические аппараты (теория доверия, возможности, интервальных множеств и др.), направленные на преодоление ограничений классической логики.

## ЭВОЛЮЦИОННЫЙ ПОДХОД

Начало шестого этапа развития ИИ связано с ростом интереса к использованию генетических алгоритмов (ГА) для эффективного обучения широкого класса нейронных сетей. В связи с этим этот этап можно было исходно назвать *нейрогенетическим*, однако необходимость усовершенствования ГА оказала влияние на увеличение интереса к другим направлениям и полученным достижениям в области эволюционного подхода. Наряду с тенденцией к автономности интеллектуальных систем (создания интеллектуальных агентов, синергетических систем, ботов, роботов и т.д.), восходящей к третьему этапу развития ИИ, это привело к значительному расширению нейрогенетической парадигмы.

Возможность того, что вычислительная система, наделенная простыми механизмами изменчивости и отбора, могла бы функционировать по аналогии с законами эволюции в природных системах, была очень привлекательна. И это стало причиной появления ряда вычислительных систем, построенных на принципах естественного отбора.

История эволюционных вычислений началась с разработки ряда независимых моделей, основными из которых были генетические алгоритмы и классификационные системы Холланда (Holland), опубликованные в начале 60-х годов. Они получили всеобщее признание после

выхода в свет книги, ставшей классикой в этой области - "Адаптация в естественных и искусственных системах" [Adaptation in Natural and Artificial Systems, 1975]. В 70-х годах в рамках теории случайного поиска Растригиным Л.А. был предложен ряд алгоритмов, использующих идеи бионического поведения особей. Развитие этих идей нашло отражение в цикле работ Букатовой И.Л. по эволюционному моделированию. Развивая идеи Цетлина М.Л. о целесообразном и оптимальном поведении стохастических автоматов. Неймарк Ю.И. предложил осуществлять поиск глобального экстремума на основе коллектива независимых автоматов, моделирующих процессы развития и элиминации особей. Большой вклад в развитие эволюционного программирования внесли Фогель (Fogel) и Уолш (Walsh). Несмотря на разницу в подходах, каждая из этих "школ" взяла за основу ряд принципов, существующих в природе, и упростила их до такой степени, чтобы их можно было реализовать на компьютере.

Главная трудность с построения вычислительных систем, основанных на принципах естественного отбора, и их применением в прикладных задачах, состоит в том, что природные системы достаточно хаотичны и не носят четкую направленность. Обычно компьютер используется как инструмент для решения определенных задач, в которых акцентируется внимание на максимально быстром выполнении действий при минимальных затратах. Природные системы не имеют никаких таких целей или ограничений, во всяком случае, они не очевидны. Выживание в природе не направлено к некоторой фиксированной цели, вместо этого эволюция совершает шаг вперед в любом доступном направлении.

Условно усилия, направленные на моделирование эволюции по аналогии с природными системами, к настоящему времени можно разбить на две большие категории:

- системы, которые смоделированы на биологических принципах. Они успешно использовались для задач типа функциональной оптимизации и могут легко быть описаны на небиологическом языке
- системы, которые являются биологически более реалистичными, но которые не оказались особенно полезными в прикладном смысле. Они больше похожи на биологические системы и менее направлены. Они обладают сложным и интересным поведением, и, видимо, вскоре получат практическое применение.

Конечно, на практике трудно разделять эти вещи так строго. Эти категории — просто два полюса, между которыми лежат различные вычислительные системы. Ближе к первому полюсу — эволюционные алгоритмы, такие как Эволюционное Программирование (Evolutionary Programming), Генетические Алгоритмы (Genetic Algorithms) и Эволюционные Стратегии (Evolution Strategies). Ближе ко второму полюсу — системы, которые могут быть классифицированы как Искусственная Жизнь (Artificial Life).

Генетические Алгоритмы - адаптивные методы поиска, которые в последнее время часто используются для решения задач функциональной оптимизации. Они основаны на генетических процессах биологических организмов: биологические популяции развиваются в течении нескольких поколений, подчиняясь законам естественного отбора и по принципу "выживает наиболее приспособленный", открытому Чарльзом Дарвином. Подражая этому процессу генетические алгоритмы способны "развивать" решения реальных задач, если те соответствующим образом закодированы. Например, ГА могут использоваться, чтобы проектировать структуры моста, для поиска максимального отношения прочности/веса, или определять наименее расточительное размещение для нарезки форм из ткани. Они могут также использоваться для интерактивного управления процессом, например, на химическом заводе, или балансировании загрузки на многопроцессорном компьютере.

Основные принципы ГА были сформулированы Холландом [Holland. 1975] и хорошо описаны во многих работах. В отличие от эволюции, происходящей в природе, ГА только моделируют те процессы в популяциях, которые являются существенными для развития. Точный ответ на вопрос: какие биологические процессы существенны для развития, и какие нет? - все еще открыт для исследователей.

В природе особи в популяции конкурируют друг с другом за различные ресурсы, такие, например, как пища или вода. Кроме того, члены популяции одного вида часто конкурируют за привлечение брачного партнера. Те особи, которые наиболее приспособлены к окружающим условиям, будут иметь относительно больше шансов воспроизвести потомков. Слабо приспо-

собленные особи либо совсем не произведут потомства, либо их потомство будет очень немногочисленным. Это означает, что гены от высоко адаптированных или приспособленных особей будут распространяться в увеличивающемся количестве потомков на каждом последующем поколении. Комбинация хороших характеристик от различных родителей иногда может приводить к появлению "суперприспособленного" потомка, чья приспособленность больше, чем приспособленность любого из его родителя. Таким образом, вид развивается, лучше и лучше приспособливаясь к среде обитания.

ГА используют прямую аналогию с таким механизмом. Они работают с совокупностью "особей" - популяцией, каждая из которых представляет возможное решение данной проблемы. Каждая особь оценивается мерой ее "приспособленности" согласно тому, насколько "хорошо" соответствующее ей решение задачи. Например, мерой приспособленности могло бы быть отношение силы/веса для данного проекта моста. (В природе это эквивалентно оценке того, насколько эффективен организм при конкуренции за ресурсы). Наиболее приспособленные особи получают возможность "воспроизводить" потомство с помощью "перекрестного скрещивания" с другими особями популяции. Это приводит к появлению новых особей, которые сочетают в себе некоторые характеристики, наследуемые ими от родителей. Наименее приспособленные особи с меньшей вероятностью смогут воспроизвести потомков, так что те свойства, которыми они обладали, будут постепенно исчезать из популяции в процессе эволюции.

Так и воспроизводится вся новая популяция допустимых решений, выбирая лучших представителей предыдущего поколения, скрещивая их и получая множество новых особей. Это новое поколение содержит более высокое соотношение характеристик, которыми обладают хорошие члены предыдущего поколения. Таким образом, из поколения в поколение, хорошие характеристики распространяются по всей популяции. Скрещивание наиболее приспособленных особей приводит к тому, что исследуются наиболее перспективные участки пространства поиска. В конечном итоге, популяция будет сходиться к оптимальному решению задачи.

В последние годы, реализовано много генетических алгоритмов. По этой причине в настоящее время под термином "генетические алгоритмы" скрывается не одна модель, а достаточно широкий класс алгоритмов, подчас мало похожих друг от друга. Исследователи экспериментировали с различными типами представлений, операторов кроссовера и мутации, специальных операторов, и различных подходов к воспроизведству и отбору.

Хотя модель эволюционного развития, применяемая в ГА, сильно упрощена по сравнению со своим природным аналогом, тем не менее. ГА является достаточно мощным средством и может с успехом применяться для широкого класса прикладных задач, включая те, которые трудно, а иногда и вовсе невозможно, решить другими методами. Однако ГА, как и другие методы эволюционных вычислений, не гарантирует обнаружения глобального решения за полиномиальное время. ГА не гарантируют и того, что глобальное решение будет найдено, но они хороши для поиска "достаточно хорошего" решения задачи "достаточно быстро". Там, где задача может быть решена специальными методами, почти всегда такие методы будут эффективнее ГА и в быстродействии, и в точности найденных решений. Одним из главных преимуществ ГА является то, что они могут применяться как в сложных задачах, где не существует никаких специальных методов, так и там, где хорошо работают существующие методики, эффективность которых можно повысить за счет сочетания с ГА.

## ШЕСТОЙ ЗТАП — СВЕРТОЧНЫЕ НЕЙРОСЕТИ

### Тенденции дальнейшего развития области искусственного интеллекта

Характеризуя всю область в целом, можно отметить, что рассмотренные выше этапы и направления демонстрируют быстрое развитие и расширение области ИИ, поэтому четкое выделение новых (будущих) этапов станет еще более трудной задачей.

В рамках **первой тенденции**, которая остается неизменяемой в течение всей истории развития ИИ, можно отметить процесс выделения и смешения направлений ИИ в отдельные или другие научные дисциплины. На сегодняшний день информационные технологии получили широкое распространение, а уровень их интеллектуализации сильно возрос, что привело не просто к возможности, а к постоянно растущей потребности в интеллектуальных системах. Слово интеллект (intelligence) стало крайне популярным и активно эксплуатируется в самых

разных аспектах: Business Intelligence (BI), интеллектуальная камера, интеллектуальный дом, интеллектуальный анализ данных и т.д., а апогеем этой идеи можно назвать слоган компании Motorola «Intelligence everywhere».

Особенностью современной формы этой тенденции является разделение области не только по направлениям (методологиям) исследований и разработки, но и по прикладным аспектам (ИС в экономике, в медицине, в военной отрасли, в сфере развлечений и т.д.). Кроме того, можно заметить наметившиеся разделение ИИ на два уровня, направленных с одной стороны, на широкую аудиторию, а с другой стороны — на узкоспециализированных специалистов-профессионалов.

**Второй тенденцией** является обновление парадигмы второго этапа развития ИИ, соответствующей задачам создания интегральных роботов. Создание автономных роботов, интегрирующих множество различных интеллектуальных функций и способных функционировать в реальных условиях, является одной из самых быстро развивающихся областей ИИ.

В качестве **третьей тенденции** можно выделить смежную с робототехникой область создания ботов и интеллектуальных компьютерных программ, называемых *интеллектуальными агентами (ИЛ)*. Они выступают в роли виртуальных помощников или советчиков в рабочих программах, в компьютерных играх используются как AI-персонажи. в чатах и форумах — как модераторы и т.д. На сегодня актуальными являются идеи о создании сообществ ИА, интеллектуальных организаций и предприятий.

**Четвертой тенденцией** в области ИИ является создание естественно-языковых (ЕЯ) систем. ЕЯ-процессор позволит преодолеть многие сложности взаимодействия человека и компьютера, даст возможность извлечения знаний из текстов без использования специальных устройств, увеличит производительность и эффективность взаимодействия компьютерных программ с помощью более гибких языков.

Особенно актуальна эта задача потому, что пользователи ИТ все больше времени проводят в виртуальном информационном мире (Интернет, компьютерные игры, мобильная телефония и т.д.). Необходимость эффективного взаимодействия внутри информационной сети выдвигает задачу создания ЕЯ - модулей на передний план всей области ИТ.

**Пятой тенденцией** развития искусственного интеллекта, особенно его бионической составляющей, является внедрение в область ИИ новых достижений нейронауки и генетики. Вопросы об организации системы обработки информации, мышления и памяти остаются до сих пор мало изученными, поэтому основные усилия исследователей человеческого мозга направлены сейчас на изучение интеллектуальных функций, и, особенно, памяти человека.

**Шестой тенденцией** в области ИИ является внедрение разработок в творческую деятельность людей. Так, например, уже сейчас существуют виртуальные музыкальные группы, которые имеют не только компьютерный вид, но и компьютерный голос, неотличимый от человеческого.

## **Основные направления исследований ИИ**

Интеллектуальные информационные системы проникают во все сферы нашей жизни, поэтому трудно провести строгую классификацию направлений, по которым ведутся активные и многочисленные исследования в области ИИ. Рассмотрим кратко некоторые из них.

**1. Разработка интеллектуальных информационных систем или систем, основанных на знаниях.** Это одно из главных направлений ИИ. Основной целью построения таких систем являются выявление, исследование и применение знаний высококвалифицированных экспертов для решения сложных задач, возникающих на практике. При построении систем, основанных на знаниях (СОЗ), используются знания, накопленные экспертами в виде конкретных правил решения тех или иных задач. Это направление преследует цель имитации человеческого искусства анализа неструктурированных и слабоструктурированных проблем. В данной области исследований осуществляется разработка моделей представления, извлечения и структурирования знаний, а также изучаются проблемы создания баз знаний (БЗ), образующих ядро СОЗ. Частным случаем СОЗ являются экспертные системы (ЭС).

**2. Разработка естественно-языковых интерфейсов и машинный перевод.** Проблемы компьютерной лингвистики и машинного перевода разрабатываются в ИИ с 1950-х гг. Системы машинного перевода с одного естественного языка на другой обеспечивают быстроту и систе-

матичность доступа к информации, оперативность и единообразие перевода больших потоков, как правило, научно-технических текстов. Системы машинного перевода строятся как интеллектуальные системы, поскольку в их основе лежат БЗ в определенной предметной области и сложные модели, обеспечивающие дополнительную трансляцию «исходный язык оригинала - язык смысла - язык перевода». Они базируются на структурно-логическом подходе, включающем последовательный анализ и синтез естественно-языковых сообщений. Кроме того, в них осуществляется ассоциативный поиск аналогичных фрагментов текста и их переводов в специальных базах данных (БД). Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с компьютером на естественном языке (так называемые системы ЕЯ-общения).

**3. Генерация и распознавание речи.** Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, "разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии. В таких системах под текстом понимают фонемный текст (как слышится).

**4. Обработка визуальной информации.** В этом научном направлении решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений связана с трансформированием графических образов, результатом которого являются новые изображения. В задаче анализа исходные изображения преобразуются в данные другого типа, например, в текстовые описания. При синтезе изображений на вход системы поступает алгоритм построения изображения, а выходными данными являются графические объекты (системы машинной графики).

**5. Обучение и самообучение.** Эта актуальная область ИИ включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К данному направлению относятся не так давно появившиеся системы добычи данных (Data-mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery).

**6. Распознавание образов.** Это одно из самых ранних направлений ИИ, в котором распознавание объектов осуществляется на основании применения специального математического аппарата, обеспечивающего отнесение объектов к классам, а классы описываются совокупностями определенных значений признаков.

**7. Игры и машинное творчество.** Машинное творчество охватывает сочинение компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов. Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки программного обеспечения. Кроме того, компьютерные игры предоставляют мощный арсенал разнообразных средств, используемых для обучения.

**8. Программное обеспечение систем ИИ.** Инstrumentальные средства для разработки интеллектуальных систем включают специальные языки программирования, ориентированные на обработку символьной информации (LISP, SMALLTALK, РЕФАЛ), языки логического программирования (PROLOG), языки представления знаний (OPS 5, KRL, FRL), интегрированные программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (KE, ARTS, GURU, G2), а также оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют создавать прикладные ЭС, не прибегая к программированию.

**9. Новые архитектуры компьютеров.** Это направление связано с созданием компьютеров не фон-неймановской архитектуры, ориентированных на обработку символьной информации. Известны удачные промышленные решения параллельных и векторных компьютеров, однако в настоящее время они имеют весьма высокую стоимость, а также недостаточную совместимость с существующими вычислительными средствами.

**10. Интеллектуальные роботы.** Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном используются программируемые манипуляторы с жесткой схемой управления, названные роботами первого поколения. Несмотря на очевидные успехи отдельных разработок, эра интеллектуальных автономных роботов пока не наступила. Основными сдерживающими факторами в разработке автономных роботов являются нерешенные проблемы в области интерпретации знаний, машинного зрения, адекватного хранения и обработки трехмерной визуальной информации.

## **Л 2. Интеллектуальные системы (ИС), как основа новых информационных технологий**

1. Понятие ИИ.
2. Определения и пути создания ИИ.
3. Бионическое, эвристическое и эволюционное направления.
4. Классификация систем ИИ.

Искусственный интеллект (ИИ) как наука существует около полувека. Первой интеллектуальной системой считается программа «Логик-Теоретик», предназначенная для доказательства теорем и исчисления высказываний. Ее работа впервые была продемонстрирована 9 августа 1956 г. В создании программы участвовали такие известные ученые, как А. Ньюэлл, А. Тьюринг, К. Шенонн, Дж. Лоу, Г. Саймон и др. За прошедшее с тех пор время в области ИИ разработано великое множество компьютерных систем, которые принято называть интеллектуальными. Области их применения охватывают практически все сферы человеческой деятельности, связанные с обработкой информации.

На сегодняшний день не существует единого определения, которое однозначно описывает эту научную область. Среди многих точек зрения на нее доминируют следующие три.

Согласно **первой** исследования в области ИИ относятся к фундаментальным, в процессе которых разрабатываются новые модели и методы решения задач, традиционно считавшихся интеллектуальными и не поддававшихся ранее формализации и автоматизации.

Согласно **второй** точке зрения это направление связано с новыми идеями решения задач на ЭВМ, с разработкой новых технологий программирования и с переходом к компьютерам не фон-неймановской архитектуры.

**Третья** точка зрения, наиболее прагматическая, основана на том, что в результате исследований, проводимых в области ИИ, появляется множество прикладных систем, способных решать задачи, для которых ранее создаваемые системы были непригодны. По последней трактовке ИИ является экспериментальной научной дисциплиной, в которой роль эксперимента заключается в проверке и уточнении интеллектуальных систем, представляющих собой аппаратно-программные информационные комплексы.

### **1. Понятие ИИ**

Под **интеллектом** мы понимаем способность любого организма (или устройства) достигать некоторой измеримой степени успеха при поиске одной из многих возможных целей в обширном многообразии сред. Необходимо отличать знания от интеллекта, имея в виду, что знания - полезная информация, накопленная индивидуумом, а интеллект - это его способность предсказывать состояние внешней среды в сочетании с умением преобразовывать каждое предсказание в подходящую реакцию, ведущую к заданной цели.

## 2. Определения и пути создания ИИ

1. Искусственный интеллект — наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ. ИИ связан со сходной задачей использования компьютеров для понимания человеческого интеллекта, но не обязательно ограничивается биологически правдоподобными методами.
2. Искусственный интеллект - это возможность решения задач, которые до сих пор не удавалось решить человеку, машинным способом с помощью программных средств.

Основной проблемой искусственного интеллекта является разработка методов представления и обработки знаний.

К программам искусственного интеллекта относятся:

1. игровые программы (стохастические, компьютерные игры);
2. естественно-языковые программы - машинный перевод, генерация текстов, обработка речи;
3. распознающие программы - распознавание почерков, изображений, карт;
4. программы создания и анализа графики, живописи, музыкальных произведений.

Выделяются следующие направления искусственного интеллекта:

1. экспертные системы;
2. нейронные сети;
3. естественно-языковые системы;
4. эволюционные методы и генетические алгоритмы;
5. нечеткие множества;
6. системы извлечения знаний.

**Экспертные системы** - система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации, ориентирована на решение конкретных задач.

**Нейронные сети** реализуют нейросетевые алгоритмы. Делятся на:

- сети общего назначения, которые поддерживают около 30 нейросетевых алгоритмов и настраиваются на решение конкретных задач;
- объектно-ориентированные - используемые для распознания символов, управления производством, предсказание ситуаций на валютных рынках;
- гибридные - используемые вместе с определенным программным обеспечением.

**Естественно-языковые (ЕЯ) системы** делятся на:

- программные продукты естественного языкового интерфейса в БД (представление естественно-языковых запросов в SQL-запросы);
- естественно-языковой поиск в текстах, содержательное сканирование текстов (используется в поисковых системах Internet, например, Google);

- масштабируемые средства распознания речи (портативные синхронные переводчики);
- средства голосового ввода команд и управления (безлюдные производства);
- компоненты речевой обработки, как сервисные средства программного обеспечения (ОС Windows).

**Нечёткие множества** - реализуют логические отношения между данными. Эти программные продукты используются для управления экономическими объектами, построения экспертных систем и систем поддержки принятия решений.

**Генетические алгоритмы** - это методы анализа данных, которые невозможно проанализировать стандартными методами. Как правило, используются для обработки больших объёмов информации, построения прогнозных моделей. Используются в научных целях при имитационном моделировании.

**Системы извлечения знаний** - используются для обработки данных из информационных хранилищ.

### **3. Бионическое, эвристическое и эволюционное направления.**

Принято различать три основные пути моделирования интеллекта и мышления:

- классический, или (как его теперь называют) бионический;
- эвристического программирования;
- эволюционного моделирования.

#### **Бионическое моделирование**

Непосредственное моделирование человеческого мозга (т.е. моделирование каждой нервной клетки и связей между ними) с целью создания автоматов, обладающих интеллектом, чрезвычайно сложно. Мозг представляет собой самую сложную и лишь частично изученную структуру. Сложнейшее переплетение связей коры головного мозга практически не поддаются расшифровке. Известно лишь примерное расположение зон мозга, отвечающих за ту или иную функцию. В настоящее время неизвестен и принцип работы мозговых элементов нейронов, многочисленные связи которых имеют внешне хаотический характер. Попытки смоделировать работу головного мозга соединением между собой множества процессоров подобно нейронной сети, показали, что некоторое увеличение скорости и потока обрабатываемой информации идет лишь до уровня одного - двух десятков процессоров, а затем начинается резкий спад производительности. Процессоры как бы "теряются", перестают контролировать ситуацию или проводят большую часть времени в ожидании соседа. Некоторых успехов удалось добиться лишь в приборах, работающих в "двумерном варианте", т.е. обрабатывающих не последовательную, а параллельную информацию, например, в системах распознаваниях образов. В них одна плоскость данных одновременно взаимодействует с другой, причем количество единиц

информации может достигать нескольких миллионов. Таким образом происходит единовременный охват изучаемого объекта, а не последовательное изучение его частей.

### **Эвристическое программирование**

Второй подход к решению задачи искусственного интеллекта связан с эвристическим программированием и решает задачи, которые в общем можно назвать творческими.

Практичность этого метода заключается в радикальном уменьшении вариантов, необходимых при использовании метода проб и ошибок. Правда, всегда существует вероятность упустить наилучшее решение, так что говорят, что этот метод предлагает решения с некоторой вероятностью правильности.

Обычно используют два метода: метод анализа целей и средств и метод планирования. Первый заключается в выборе и осуществлении таких операций, которые последовательно уменьшают разницу между исходным и конечным состоянием задачи. Во втором методе вырабатывается упрощенная формулировка исходной задачи, которая также решается методом анализа целей и средств. Один из полученных вариантов дает решение исходной задачи.

### **Эволюционное моделирование**

Третий подход является попыткой смоделировать не то, что есть, а то, что могло бы быть, если бы эволюционный процесс направлялся в нужном направлении и оценивался предложенными критериями.

Идея эволюционного моделирования сводится к экспериментальной попытке заменить процесс моделирования человеческого интеллекта моделированием процесса его эволюции. При моделировании эволюции предполагается, что разумное поведение предусматривает сочетание способности предсказывать состояние внешней среды с умением подобрать реакцию на каждое предсказание, которое наиболее эффективно ведет к цели.

Этот метод открывает путь к автоматизации интеллекта и освобождению от рутинной работы. Это высвобождает время для проблемы выбора целей и выявления параметров среды, которые заслуживают исследования. Такой принцип может быть применен для использования в диагностике, управлении неизвестными объектами, в игровых ситуациях.

Итак, существуют три пути моделирования интеллекта: бионический, эвристический и эволюционный. В зависимости от использованных средств можно выделить **три фазы в исследовании**.

**Первая фаза** - создания устройств, выполняющих большое число логических операций с высоким быстродействием.

**Вторая фаза** включает разработку проблемно-ориентированных языков для использованного на оборудовании, созданном в первой фазе.

**Третья фаза** наиболее выражена в эволюционном моделировании. В ходе развития этой фазы отпадает необходимость в точной формулировке постановки задачи, т.е. задачу можно сформулировать в терминах цели и допустимых затрат, а метод решения будет найден самостоятельно по этим двум параметрам.

Работы по искусственному интеллекту во многом тесно связаны с философской проблемой кибернетического моделирования. Эти работы часто связывают с построением точной копии человеческого мозга. Однако такой подход можно назвать "некибернетическим". Каковы же черты кибернетического метода мышления, какие вопросы вносит кибернетика в человеческое познание? В своей "Истории западной философии" Б. Рассел ставит вопрос о факторах, позволивших европейцам создать тип культуры, в котором ведущее место заняла наука. Причину этого Рассел усматривает, как он выражается, в двух великих интеллектуальных изобретениях: **изобретение дедуктивного метода** (частное заключение выводится из общего) древними греками (Эвклид) и **изобретение экспериментального метода** в эпоху возрождения (Галилей). Именно эти два интеллектуальных изобретения - дедуктивный метод (а тем самым математика) и эксперимент - позволили создать классическую науку. К этим двум основным интеллектуальным орудиям современное развитие познания добавляет третье - **математическую модель и математическое моделирование**. Соединение дедуктивных построений математики с данными, добтыми экспериментальным методом, создает естествознание, в центре которого стоит понятие научного закона. Совокупность законов - это основное содержание естествознания; их установление его основная задача.

Закон претендует на точное (в рамках данного уровня познания) описание хода явлений. Закон либо верен, либо неверен, бессмысленно говорить о хороших и плохих законах. Модель в этом отношении противоположна закону. Модель может быть плохой или хорошей, она не претендует на точное воспроизведение сложной системы, а ограничивается описанием отдельных аспектов, причем для одного и того же аспекта могут быть предложены модели, одновременно имеющие право на существование.

В изучении сложных систем (в т.ч. диффузных - нельзя выделить отдельные части без повреждения системы) формулировка относительно простых законов оказывается невозможной и заменяется построением эскизных моделей. Образно говоря, здесь мы имеем дело с математическим описанием, напоминающим современную абстрактную живопись. Можно сказать, что попытки реалистичного описания сложных систем иллюзорны такое описание не воспринималось бы из-за чрезмерной сложности.

Это не означает, что категория закона утрачивает смысл в науке, но то, что дополнительно к ранее известным интеллектуальным орудиям - строгой дедукции и эксперименту рождается третье орудие - математическое моделирование, в котором по-новому выступает математика и появляется новый вид эксперимента - машинный эксперимент, в котором проигрываются различные модели с последующим сопоставлением с реальным экспериментом.

Путь, который предлагает кибернетика, состоит в построении эскизных моделей, охватывающих все более и более широкий диапазон функций мышления. Задачи раскрыть "в лоб" "сущность мышления" не ставится, а ставится задача построения эскизных моделей, позволяющих описать отдельные его стороны, воспроизведены отдельные его функции и, двигаясь в этом направлении, строить системы, все более приближающиеся к человеческому мозгу.

Отсутствия жесткой связи способа функционирования (поведения) со строго определенным субстратом означает, что если две системы обнаруживают одинаковое поведение в достаточно широкой области, то они должны рассматриваться как системы сходные, аналогичные по этому способу поведения. Имеет смысл рассмотреть этот вопрос в связи с проблемой кибернетического моделирования.

Иногда встречается утверждение, что кибернетическое моделирование вообще неприменимо к изучению мышления, т.к. моделирование основана на понятиях соответствия и изоморфизма (тождественности), а мышление есть чисто человеческая способность, якобы не могущая быть описана на основе понятий соответствия. Иногда говорят, что понимание познания, мышления как соответствия образа предмету означает ни много ни мало как дуалистическую точку зрения, внешне сопоставляющую предмет и образ.

Понимание сознания как отражения неизбежно означает понимание его как соответствия, возникающего в ходе приспособления организма к среде. Причем это соответствие не есть просто внешнее соответствие вещи и образа как самостоятельного по отношению к вещи идеального предмета. Это действительно была бы дуалистическая точка зрения, но она не может монополизировать понятие соответствия, именно как соответствие определенных состояний мозга определенным состояниям внешнего мира. Это соответствие и несет информацию о внешнем мире.

В приведенном утверждении не проводится различие между информационным моделированием информационных процессов и информационным моделированием неинформационных процессов. Информационная модель прибора не будет работать, а будет только моделировать работу, однако в отношении мышления этот тезис представляется спорным. По отношению к информационным процессам их моделирование является функционально полным, т.е. если модель дает те же самые результаты, что и реальный объект, то их различие теряет смысл.

Многие споры вокруг проблемы "кибернетика и мышление" имеют эмоциональную подоплеку. Признание возможности искусственного разума представляется чем-то унижающим человеческое достоинство. Однако нельзя смешивать вопросы возможности искусственного разума с вопросом о развитии и совершенствовании человеческого разума. Разумеется, искусственный разум может быть использован в негодных целях, однако это проблема не научная, а скорее морально-этическая. Однако развитие кибернетики выдвигает ряд проблем, которые все же требуют пристального внимания. Эти проблемы связаны с опасностями, возникающими в ходе работ по искусенному интеллекту. Первая проблема связана с возможной потерей стиму-

лов к творческому труду в результате массовой компьютеризации или использования машин в сфере искусств. Однако в последнее время стало ясно, что человек добровольно не отдаст самый квалифицированный творческий труд, т.к. он для самого человека является привлекательным. Вторая проблема носит более серьезный характер и на нее неоднократно указывали такие специалисты, как Н. Винер, Н. М. Амосов, И. А. Полетаев и др. Состоит она в следующем. Уже сейчас существуют машины и программы, способные в процессе работы самообучаться, т.е. повышать эффективность приспособления к внешним факторам. В будущем, возможно, появятся машины, обладающие таким уровнем приспособляемости и надежности, что необходимость человеку вмешиваться в процесс отпадет. В этом случае возможна потеря самим человеком своих качеств, ответственных за поиск решений. Налицо возможная деградация способностей человека к реакции на изменение внешних условий и, возможно, неспособность принятия управления на себя в случае аварийной ситуации. Встает вопрос о целесообразности введения некоторого предельного уровня в автоматизации процессов, связанных с тяжелыми аварийными ситуациями. В этом случае у человека, "надзирающим" за управляющей машиной, всегда хватит умения и реакции таким образом воздействовать на ситуацию, чтобы погасить разгорающуюся аварийную ситуацию. Таковые ситуации возможны на транспорте, в ядерной энергетике. Особо стоит отметить такую опасность в ракетных войсках стратегического назначения, где последствия ошибки могут иметь фатальный характер. Несколько лет назад в США начали внедрять полностью компьютеризированную систему запуска ракет по командам суперкомпьютера, обрабатывающего огромные массивы данных, собранных со всего света. Однако оказалось, что даже при условии многократного дублирования и перепроверки, вероятность ошибки оказалась бы столь велика, что отсутствие контролирующего оператора привело бы к непоправимой ошибке. От системы отказались. Люди будут постоянно решать проблему искусственного интеллекта, постоянно сталкиваясь со всеми новыми проблемами. И, видимо, процесс этот бесконечен.

#### **4. Классификация систем ИИ**

Интеллектуальная информационная система (ИИС) основана на концепции использования базы знаний для генерации алгоритмов решения прикладных задач различных классов в зависимости от конкретных информационных потребностей пользователей.

Для ИИС характерны следующие признаки:

- развитые коммуникативные способности;
- умение решать сложные плохо формализуемые задачи;
- способность к самообучению;
- адаптивность.

Каждому из перечисленных признаков условно соответствует свой класс ИИС. Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Средства ИИ могут использоваться для реализации различных функций, выполняемых ИИС. На рис. 1.1 приведена классификация ИИС, признаками которой являются следующие интеллектуальные функции:



Рис. 1.1. Классификация интеллектуальных информационных систем

- коммуникативные способности - способ взаимодействия конечного пользователя с системой;
- решение сложных плохо формализуемых задач, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;
- способность к самообучению - умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
- адаптивность - способность системы к развитию в соответствии с объективными изменениями области знаний.

### **Л 3. Формализация знаний в ИС**

1. Основные понятия и определения. Проблемная область.
2. Данные и знания.
3. Свойства, характеристики знаний.
4. Процедурные и декларативные знания.
5. Классификация знаний
6. Формальные языки.
7. Модели представления знаний
8. Классификация моделей знаний
9. Логические, продукционные, сетевые, фреймовые модели представления знаний

#### **Вступление**

Понятие искусственного интеллекта можно трактовать в **широком и узком** смыслах слова.

В **широком** смысле слова искусственный интеллект есть научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного планирования тех видов человеческой деятельности, которые традиционно считаются интеллектуальными (разумными). При этом термин интеллектуальные в применении к видам человеческой деятельности означает, что их до недавнего времени способен был осуществлять только человек.

В **узком** смысле слова под искусственным интеллектом понимается программа система, имитирующая на компьютере мышление человека.

**Предметом изучения ИИ** являются методы анализа и синтеза ЭВМ с интеллектуальными способностями.

**Объект изучения в дисциплине ИС** — рассмотрение процессов решения задач, характеризующихся высокой степенью сложности и эвристикой.

В теории ИИ можно выделить **два основных направления** их исследования: **общетеоретическое**, связанное с разработкой моделей и методов реализации отдельных аспектов получения и преобразования знаний и **прикладное**, нацеленное на разработку комплексных технологий конечных пользователей.

Технически интеллектуальные системы - технические и программные системы, ориентированные на решение большого и очень важного класса задач, называемых неформализованными.

Система становится интеллектуальной, если в ней: **данные заменяются на знания**; алгоритмы функционирования заменяются на методы искусственного интеллекта.

ИС особенно эффективны в применении к слабо структурированным задачам, в которых отсутствует строгая формализация, и для решения которых применяются эвристические процедуры, позволяющие в большинстве случаев получить решение.

# 1. Основные понятия и определения. Проблемная область

**Формализация знаний** — приведение знаний, поступающих из разных источников, к одинаковой форме, для повышения их доступности.

## Понятие «Знание» в ИИ

**Знания** - это совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной *предметной области*.

Иными словами, **знания** - это выявленные закономерности *предметной области* (принципы, связи, законы), позволяющие решать задачи в этой области. С точки зрения *ИИ* знания можно определить как формализованную информацию, на которую ссылаются в процессе логического вывода.

В этом случае, под *ПрО* понимается область человеческих знаний, в терминах которой формулируются задачи и в рамках которой они решаются. Т.е. *ПрО* представляется описанием части реального мира, которое в силу своей приближенности рассматривается как ее *информационная модель*.

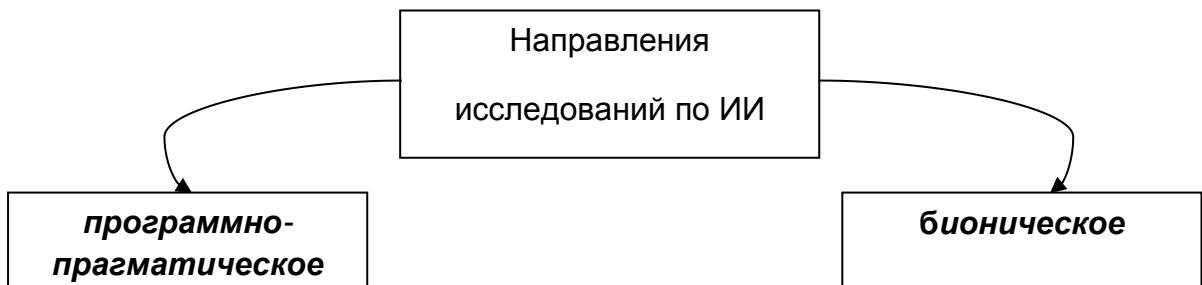
**Проблемная область** - это содержательное описание в терминах *ПрО* проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших ее возникновение.

В исследованиях по *ИИ* можно выделить два основных направления:

**программно-прагматическое** («не имеет значения, как устроено «мыслящее» устройство, главное, чтобы на заданные входные воздействия оно реагировало, как человеческий мозг»).

К ярким представителям программно-прагматического направления можно отнести **экспертные системы** - сложные программные комплексы, аккумулирующие знания специалистов - экспертов для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области.

**бионическое** («единственный объект, способный мыслить - это человеческий мозг, поэтому любое «мыслящее» устройство должно каким-то образом воспроизводить его структуру»). В рамках данного подхода сформировалась новая наука *нейроинформатика*.



- эвристического программирования;
- эволюционного моделирования

## **2. Данные и знания**

**Общепризнанного определения знания**, как и определения **искусственного интеллекта**, не существует. Известные трактовки этого понятия отражают его различные аспекты, поэтому приведем несколько определений.

Наиболее общее определение трактует **знание как всю совокупность данных (информации), необходимую для решения задачи**. В этом определении подчеркивается, что данные в привычном понимании также являются знаниями. Однако знания в информационном плане не ограничиваются рамками данных.

(Данные — представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.)

В полном объеме **информация**, содержащаяся в **знаниях**, должна включать сведения о: **системе понятий предметной области, в которой решаются задачи; системе понятий формальных моделей, на основе которых решаются задачи; соответствии систем понятий, упомянутых выше; методах решения задачи; текущем состоянии предметной области.**

Из перечисленных компонентов только последний (**текущее состояние предметной области**) в явном виде соответствует понятию "**данные**".

В целом обо всей приведенной выше информации иногда говорят, что она составляет **проблемную область** решаемой задачи.

## **3. Свойства, характеристики знаний**

Несмотря на сложности формулировки определения **знания** считается общепризнанным, что знания имеют ряд свойств, позволяющих отличать их от данных: **внутреннюю интерпретируемость; внутреннюю (рекурсивную) структурированность; внешнюю взаимосвязь единиц; шкалирование; погружение в пространство с семантической метрикой; активность.**

Если данные обладают этими свойствами, можно говорить о перерастании данных в знания.

Раскроем подробнее приведенные выше **свойства** знаний.

**Внутренняя интерпретируемость** означает наличие в памяти ЭВМ сведений не только о значении, но и о наименовании информационной единицы. Следует отметить, что это свойство присуще некоторым моделям представления данных, например, реляционной.

**Внутренняя (рекурсивная) структурированность** отражает вложенность одних информационных единиц в другие или в самих себя. Она предусматривает установку отношений принадлежности элементов к классу, родовидовые отношения типа «часть-целое» и т.п. В целом внутренняя структурированность характеризует структуру знания.

**Внешняя взаимосвязь единиц** определяет, с какой информационной единицей имеет связь данная информационная единица и какова эта связь. С помощью этого свойства устанавливается связь различных отношений, отражающих семантику и прагматику связей понятий, а также отношений, отражающих смысл системы в целом.

Отдельные информационные единицы не могут описывать динамические ситуации, когда некоторые факты, содержащиеся в структуре одной единицы, вступают в ситуативную

связь с фактами или явлениями, описанными в структуре другой единицы. Для описания таких связей используются специальные информационные элементы, в которых указываются имена взаимосвязанных информационных единиц и имена существующих отношений.

**Шкалирование** означает использование шкал, предназначенных для фиксации соотношения различных величин. Прежде всего шкалирование необходимо для фиксации соотношений качественной информации.

**Погружение в пространство с семантической метрикой** используется для задания меры близости информационных единиц.

**Пример.** Пусть разработано несколько вариантов построения системы связи. Требуется определить, насколько структура существующей системы связи близка к одному из имеющихся вариантов. Для этого можно использовать метод матриц сходства - различия, в соответствии с которым матрицы заполняются оценками попарного сходства и различия элементов структур реальной системы связи и рассматриваемого варианта. В качестве оценок обычно выступают числа в диапазоне от -1 до +1 при условии, что -1 характеризует полное различие, а +1 - полное сходство.

На основании матриц сходства - различия определяется степень сходства текущей ситуации с заранее заданной (планируемой).

**Активность** знаний выражается в возможности вызова той или иной процедуры в зависимости от структуры, сложившейся между информационными единицами.

Активность знаний обусловлена тем, что в отличие от обычных программ, в которых процедуры играют роль активаторов данных, в интеллектуальных системах определенная структура данных активизирует выполнение той или иной процедуры. Практически это осуществляется включением в состав информационной единицы элемента, содержащего имя процедуры, или представлением знаний в виде правил, причем правила записываются в следующем виде: "если произошли события А<sub>1</sub> и А<sub>2</sub> и ... и А<sub>k</sub>, то необходимо выполнить процедуру В". Использование правил значительно упрощает объяснение того, как и почему получено то или иное заключение (вывод).

Перечисленные особенности информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в **базы знаний** (БЗ). Совокупность средств, обеспечивающих работу со знаниями, образует **систему управления базой знаний** (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы все пять особенностей знаний.

**Базы данных** фиксируют **экстенсиональную** семантику заданной проблемной области, состояние конкретных объектов, конкретные значения параметров для определенных моментов времени и временных интервалов. **База знаний** определяет **интенсиональную** семантику моделей и содержит описание абстрактных сущностей: объектов, отношений, процессов.

Если рассматривать знания с точки зрения решения задач в некоторой предметной области, то их удобно разделить на две большие категории - **факты и эвристику**.



Первая категория указывает обычно на хорошо известные в данной предметной области обстоятельства, поэтому знания этой категории иногда называют текстовыми, подчеркивая их достаточную освещенность в специальной литературе или учебниках. Вторая категория знаний основывается на собственном опыте специалиста (эксперта) в данной предметной области, накопленном в результате многолетней практики.

#### 4. Процедурные и декларативные знания

Знания можно разделить на процедурные и декларативные. Декларативные знания - это совокупность сведений о качественных и количественных характеристиках конкретных объектов, явлений и их элементов, представленных в виде фактов и эвристик. Традиционно такие знания накапливались в виде разнообразных таблиц и справочников, а с появлением ЭВМ приобрели форму информационных массивов (файлов) и баз данных. Процедурные знания хранятся в памяти ИИС в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описывается информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики и тому подобная информация. Другими словами, процедурные знания - это методы, алгоритмы, программы решения различных задач, последовательности действий (в выбранной проблемной области) - они составляют ядро баз знаний.

Таким образом, при использовании знаний происходит переход к формуле  
**знания + вывод = система.**

**Работа со знаниями**, иначе называемая **обработкой знаний**, лежит в основе всего современного периода развития ИИ. В свою очередь обработка знаний включает в себя:

**извлечение знаний** из источников (под источниками понимаются материальные средства хранения знаний, а также события и явления, но при этом считается, что человек источником не является);

**приобретение знаний** от профессионалов (экспертов); **представление знаний**, т.е. их формализация, позволяющая в дальнейшем использовать знания для проведения логического вывода на ЭВМ;

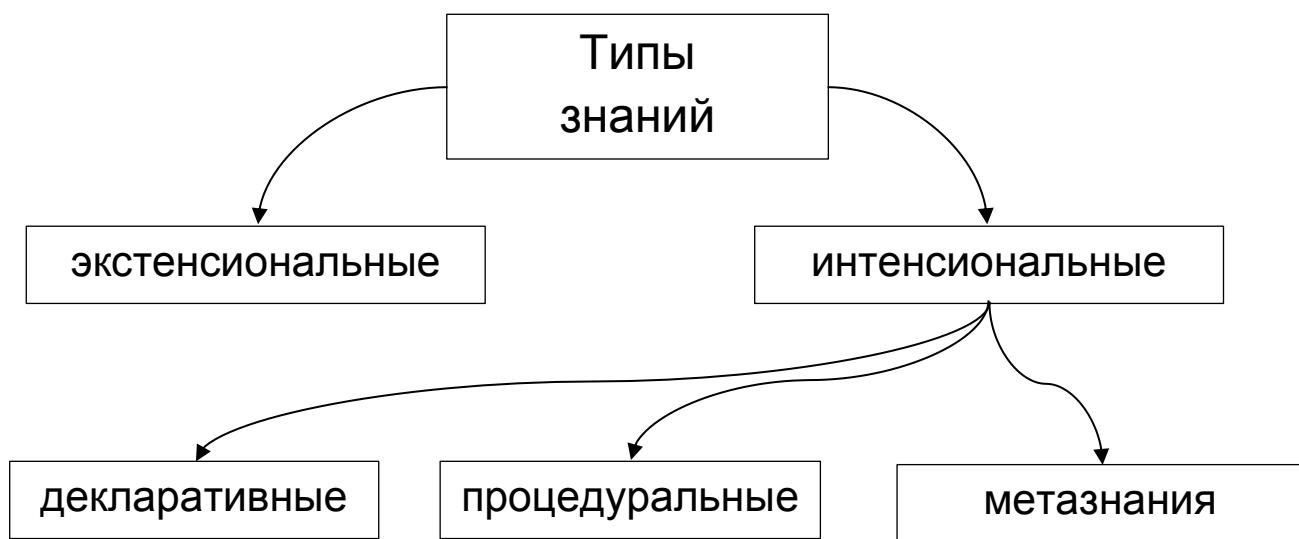
**манипулирование знаниями**, включающее пополнение, классификацию, обобщение знаний и вывод на знаниях;

**объяснение на знаниях**, позволяющее дать ответ, как и почему проведен тот или иной вывод.

В памяти ЭВМ знания представляются в виде некоторой знаковой системы. С понятием "знак" связываются понятия "экстенсионал" и "интенсионал".

**Экстенсионал знака** - это его конкретное значение или класс допустимых значений.

**Интенсионал знака** - это его смысл, характеристика содержания. Интенсионал знака определяет содержание связанного с ним понятия.



Соответственно различают два типа знаний: экстенсиональные и интенсиональные.

**Экстенсиональные знания** - это набор количественных и качественных характеристик различных конкретных объектов. Они представляются перечислениями объектов предметной области, экземпляров объектов, свойств объектов. Иными словами, экстенсиональные знания - это данные, хранящиеся в базах данных.

Иногда экстенсиональные знания называются предметными, или фактографическими знаниями.

**Интенсиональные знания** - это совокупность основных терминов, применяемых в проблемной области, и правил над ними, позволяющих получать новые знания. Интенсиональные знания описывают абстрактные объекты, события, отношения.

Интенсиональные знания подразделяются на декларативные, процедуральные и метазнания.

**Декларативные знания** отражают понятия проблемной области и связи между ними. Они не содержат в явном виде описания каких-либо процедур. Иначе декларативные знания называются понятийными, или концептуальными.

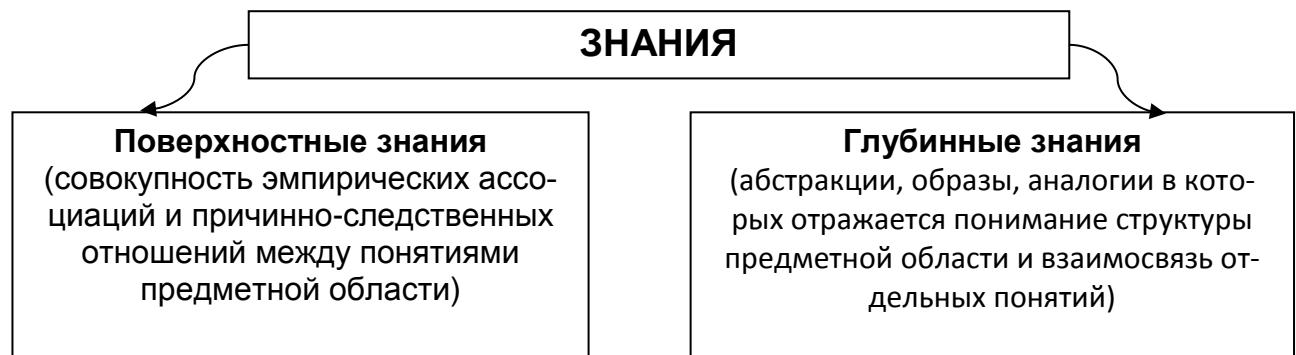
**Процедуральные знания** описывают процедуры, т.е. указывают операции над понятиями, позволяющие получать новые понятия. В отличие от декларативных знаний они содержат в явном виде описания процедур. Примером процедуральных знаний является программа, хранящаяся в памяти ЭВМ. Иногда процедуральные знания называются алгоритмическими.

**Метазнания** - это знания об организации всех остальных типов знаний. Иначе они называются специальными. Метазнания содержат признаки декларативных и процедуральных знаний.

**Поверхностные** — знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области.

**Глубинные** — абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной области. Эти знания объясняют явления и могут использоваться для прогнозирования поведения объектов.

## 5. Классификация знаний (по глубине и жесткости)



Классификация знаний по «жесткости»

## 6. Формальные языки

Формализованный (формальный) язык - язык, характеризующийся точными правилами построения выражений и их понимания. Он строится в соответствии с четкими правилами, обеспечивая непротиворечивое, точное и компактное отображение свойств и отношений изучаемой предметной области (моделируемых объектов).

В отличие от естественных языков формальным языкам присущи четко сформулированные правила семантической интерпретации и синтаксического преобразования используемых знаков, а также то, что смысл и значение знаков не изменяется в зависимости от каких-либо pragматических обстоятельств (например, от контекста).

Большинство формальных языков (созданных конструкций) строится по следующей схеме: сначала выбирается алфавит, или совокупность исходных символов, из которых будут строиться все выражения языка; затем описывается синтаксис языка, то есть правила построения осмысленных выражений. Буквами в алфавите формального языка могут быть и буквы алфавитов естественных языков, и скобки, и специальные знаки и т.п. Из букв, по определенным правилам можно составлять слова и выражения. Осмысленные выражения получаются в формальном языке, только если соблюдены определенные в языке правила образования. Для каждого формального языка совокупность этих правил должна быть строго определена и модификация любого из них приводит чаще всего к появлению новой разновидности (диалекта) этого языка.

Формальные языки широко применяются в науке и технике. В процессе научного исследования и практической деятельности формальные языки обычно используются в тесной взаимосвязи с естественным языком, поскольку последний обладает гораздо большими выразительными возможностями. В то же время формальный язык является средством более точно-го представления знаний, чем естественный язык, а следовательно, средством более точного и объективного обмена информацией между людьми.

Формальные языки часто конструируются на базе языка математики. Веком бурного развития различных формальных языков можно считать XX век.

С точки зрения информатики, среди формальных языков наиболее значительную роль играют формальный язык логики (язык алгебры логики) и языки программирования.

Возникновение языков программирования приходится на начало 50-х годов XX в.

Языков программирования и их диалектов (разновидностей) насчитывается несколько тысяч. Классифицировать их можно по-разному. Некоторые авторы разбивают все многообразие языков программирования на процедурные и декларативные. В процедурных языках преобразование данных задается с помощью описания последовательности действий над ними. В декларативных языках преобразование данных задается посредством описания отношений между самими данными. Согласно другой классификации, языки программирования можно разделить на процедурные, функциональные, логические, объектно-ориентированные. Однако любая классификация несколько условна, поскольку, как правило, большинство языков программирования включает в себя возможности языков разных типов. Особое место среди языков программирования занимают языки, обеспечивающие работу систем управления базами данных (СУБД). Часто в них выделяют две подсистемы: **язык описания данных и язык манипулирования**.

## 7. Модели представления знаний

Центральной проблемой обработки знаний, является проблема представления знаний.

Проблема представления знаний - это проблема представления взаимосвязей в конкретной предметной области в форме, понятной системе искусственного интеллекта. Представление знаний - это их формализация и структурирование (в целях облегчения решения задачи), с помощью которых отражаются характерные признаки знаний: внутренняя интерпретируемость, структурированность, связность, семантическая метрика и активность. Представление знаний - это процесс (способ) описания знаний человека о проблемной области посредством выражений на формальном языке, называемом языком представления знаний.

При проектировании модели представления знаний следует учитывать такие факторы, как - однородность представления и простота понимания. Однородность представления приводит к упрощению механизма управления логическим выводом и управлением знаниями. Простота понимания предполагает доступность понимания представления знаний и экспертам, и пользователем системы. В противном случае затрудняется приобретение знаний и их оценка.

Способ представления знаний определяет, каким образом знания описываются в памяти ЭВМ, а также каковы возможности БЗ. Для того чтобы ЭВМ имела возможность манипулирования знаниями о проблемной области, они должны быть представлены в виде модели.

Модель представления знаний (МПЗ) - это способ и результат формального описания знаний в БЗ. Она должна быть понятной пользователю и обеспечивать однородность представления знаний, за счет чего упрощаются управление знаниями и логический вывод, а также удовлетворять ряду других требований.

К настоящему времени разработано достаточно много различных МПЗ, и работа по созданию новых моделей продолжается. Однако наибольшее распространение получили четыре модели: **модель семантической сети, фреймовая, продукционная и логические**.

В основе использования МПЗ лежит **аксиоматический метод**. Аксиоматический метод в любой науке состоит в том, что выделяется некоторое небольшое множество истинных утверждений, опираясь на которые можно вывести все истинные утверждения данной науки. Классическим примером аксиоматического метода является аксиоматическое построение геометрии Евклида и Лобачевского, каждая из которых базируется на ряде постулатов. Следует отметить, что одну и ту же теорию можно строить, исходя из различного набора аксиом.

Таким образом, необходимо:

1. Построить алфавит теории, т.е. задать счетное множество символов (буквы и знаки включены в этот термин) и определить множество объектов языка - **выражений**. Под выражением имеется в виду конечная последовательность символов языка. Понятно, что символ может несколько раз появляться в выражении, а может и не появляться (роль формальных выражений в символическом языке аналогична роли слов в обычном языке).

2. Выделить подмножество таких выражений, которые будем называть **формулами** (обычно имеется хорошо разработанная процедура, позволяющая по данному выражению определить, является ли оно формулой). С формулами не будем связывать никакого значения (смысла); никакого значения не будем связывать и с входящими в них буквами и символами. Все операции будут формализованы.

3. Из **бесконечного множества истинных формул (тавтологий)** выделим небольшую группу (1 -10) так называемых аксиом теории (как правило, всегда имеется возможность эффективно выяснить, является ли данная формула аксиомой). За аксиомы берутся некоторые тавтологии, из которых по формальным правилам выводятся все остальные тавтологии.

4. Указать конечное множество отношений между формулами, которые называют правилами вывода. Правила вывода сопоставляют некоторым последовательностям формул новые формулы. Записывают правила вывода в форме фигуры, где формулы, стоящие над чертой называются посылками, а формулы, стоящие под чертой, называется следствием посылок по данному правилу вывода. С помощью правил вывода из аксиом получаются новые истинные формулы, называемые теоремами.

Доказательством (выводом) называется конечная последовательность формул  $A_1, A_2, \dots, A_p$  такая, что каждая  $A_k$  есть либо аксиома теории, либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода. Теоремой называется такая формула  $A$  теории, что существует вывод, в котором последней формулой является формула  $A$ . Теоремы, как правило, выражаются равенствами, импликациями и эквивалентностями. Доказательство теорем превращается в последовательность таких формул, и построение формальных доказательств можно поручить ЭВМ.

К МПЗ предъявляются требования полноты и непротиворечивости.

## 8. Классификация моделей знаний



## 9. Логические, продукционные, сетевые, фреймовые МПЗ

### 1. Логическая МПЗ

Логические МПЗ - это модели, основанные на правилах формальной логики.

### 2. Продукционная МПЗ

МПЗ, основанные на правилах, являются наиболее распространенными и более 80% ЭС используют именно их. Продукционная модель основана на правилах, позволяющая представить знания в виде предложений типа "Если (условие), то (действие)".

### 3. Фреймовая МПЗ

Фреймовая МПЗ базируется на понятии функционального программирования - способа составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программ на части является введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции - оператор суперпозиции других функций.

### 4. Семантические сети

Термин "семантическая" означает "смысловая", а сама семантика - наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. **наука, определяющая смысл знаков**.

## Л 4. Инженерия знаний в ИС

1. Методы инженерии знаний. Теоретические аспекты получения знаний. Методы извлечения явных знаний. Методы выявления скрытых знаний.
2. Проблемы структурирования знаний.
3. Проблема сборки знаний в единую модель предметной области.

### Вступление

Выявление и сбор нужных знаний - центральный вопрос, который встает перед разработчиками не только ЭС, но и других, относящихся к системам с интеллектом. Проблема в том, что эксперт и, вообще говоря, любой человек несет в себе огромное количество «скрытых» знаний, то есть знаний, которые им не осознаются и часто проявляются в сложных ситуациях, когда, возможно, решается вопрос жизни и смерти. Это знания нашего бессознательного «Я», которые мы получаем, рождаясь на свет. По некоторым оценкам они составляют до 70% нашего общего знания. Мы пользуемся им, не задумываясь, не осознавая, как мы управляем руками, ногами и всем прочим, часто выполняя одновременно несколько действий. С детства мы тренируем руки, ноги и другое, доводя алгоритмы управления до автоматизма. Высочайших успехов спортсмены добиваются тоже благодаря немыслимым по сложности тренировкам. Теперь же мы задались целью создать машинный интеллект, который был бы сопоставим с нашим и даже выше него, хотя бы только рациональный и для узких сфер деятельности. Иначе мы не можем доверять машине принятие решений и многое другое. Нам необходим весь объем знаний и скрытых, и явных, но разбросанных по разным источникам, который бы обеспечивал решение задач, ранее решаемых только высококлассными специалистами.

### 1. Методы инженерии знаний

Сложность проблемы привела к разработке целого арсенала методов, обобщенная классификация которых приведена на рис. 1.

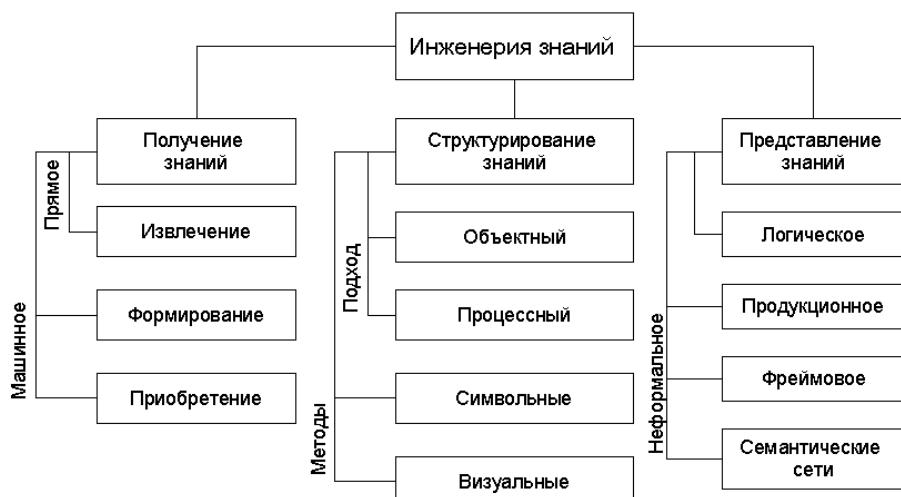


Рис. 1. Структура поля методов инженерии знаний

## **1.1. Теоретические аспекты получения знаний**

Инженеру по знаниям необходимо четко понимать природу и особенности процесса извлечения знаний. Из множества аспектов проблематики получения знаний выделяют три основных: **психологический, лингвистический и методологический (гносеологический)**.

Считается, что **психологический** аспект является ведущим из трех аспектов извлечения знаний, поскольку он определяет успешность и эффективность взаимодействия инженера по знаниям с источником знаний первого типа - **экспертом**. Психологический аспект выделяется еще и потому, что извлечение знаний происходит чаще всего в процессе непосредственного общения разработчиков ЭС, а законы психологии общения, к сожалению, часто недооцениваются или игнорируются начинающими инженерами по знаниям.

Упрощенно выделяют следующие структурные компоненты: **участники общения (партнеры), средства общения (процедура), предмет общения (знания)**.

В соответствии с этой структурой выделяют **три «слоя» психологических проблем**, возникающих при извлечении знаний:

- **контактный** (проблемы взаимодействия партнеров);
- **процедурный** (проблемы грамотного проведения процедуры извлечения);
- **когнитивный** (проблема получения знания как такового).

**На взаимодействие партнеров влияют следующие их параметры:**

**демографические** (пол, возраст, национальность, образование);

**личностные** (особенности личностного портрета, темперамент, мотивация).

На проведение процедуры извлечения влияют профессиональные параметры:

**ситуация** (место, время, продолжительность);

**оборудование** (вспомогательные средства, освещенность, мебель);

**профессиональные приемы** (темп и громкость речи, мимика, поза, интонации).

Наименее исследованы проблемы когнитивного слоя, связанные с изучением пространства памяти эксперта и реконструкцией его понятийной структуры и модели рассуждений.

**Лингвистический** аспект касается исследования языковых проблем, так как язык - это основное средство общения в процессе извлечения знаний.

**Гносеологический** аспект извлечения знаний объединяет методологические проблемы получения нового научного знания, поскольку при создании БЗ эксперт часто впервые формулирует некоторые закономерности, до того момента составлявшие его личный опыт.

**Внутренняя согласованность эмпирического знания** характеризуется понятиями модальности, противоречивости и неполноты. Модальность знания означает возможность его существования в различных категориях. Возможные противоречия не всегда должны разрешаться в поле знаний, а напротив, именно они служат чаще всего отправной точкой в рассуждениях экспертов. Неполнота знания связана с невозможностью полного описания ПрО.

**Системность** ориентирует инженера по знаниям на рассмотрение любой ПрО с позиций закономерностей системного целого и взаимодействия составляющих его частей.

Процесс познания глубоко субъективен, то есть он существенно зависит от особенностей самого познающего субъекта, поэтому более корректно говорить о глубине понимания, чем об **объективности** знания. Таким образом, инженер по знаниям должен сосредоточиться на понимании проблемы.

**Историзм** связан с развитием и изменением представлений о ПрО с течением времени.

Методологическая структура познания может быть представлена как некоторая последовательность этапов:

- **описание и обобщение фактов;**
- **выявление связей между фактами**, формулирование правил и закономерностей;
- **построение модели ПрО;**
- **объяснение и прогнозирование явлений.**

## 1.2. Методы извлечения явных знаний

Рассмотрим обобщенную классификацию этих методов (рис. 2).

**Первый класс** образуют методы, которые ориентированы на непосредственный контакт инженера по знаниям с экспертом (источником знаний),

**второй класс** - текстологические методы, основанные на приобретении знаний из документов и специальной литературы.

**Коммуникативные методы** разделяются на пассивные и активные. В **пассивных** методах ведущую роль играет эксперт, в **активных** - инженер по знаниям. При решении конкретных задач, как правило, используются как пассивные, так и активные методы. Активные методы, в свою очередь, делятся на **индивидуальные и групповые**. В групповых методах знания получают от множества экспертов, в индивидуальных - от единственного. Индивидуальные методы получили более широкое применение на практике.



Рис. 2. Классификация методов извлечения данных

**Пассивные** коммуникативные методы включают наблюдение, анализ протоколов «мыслей вслух», процедуры извлечения знаний из лекций.

**Метод наблюдения** является одним из наиболее применяемых на начальных этапах разработки экспертных систем. Его суть заключается в фиксировании всех действий эксперта, его реплик и объяснений. При этом аналитик не вмешивается в работу эксперта, а только наблюдает за процессом решения реальных задач либо за решением проблем, имитирующих реальные задачи. Наблюдения за процессом решения реальных задач позволяют инженеру по знаниям глубже понять предметную область. Однако эксперт в этом случае испытывает большое психологическое напряжение, понимая, что осуществляет не только свою профессиональную деятельность, но и демонстрирует ее инженеру по знаниям. Наблюдение за имитацией процесса снимает это напряжение, но приводит к снижению полноты и качества извлекаемых данных. Наблюдения за имитацией незаменимы в тех случаях, когда наблюдения за реальным процессом невозможны из-за специфики изучаемой предметной области.

**Метод анализа протоколов «мышлений вслух»** отличается от метода наблюдения тем, что эксперт не только комментирует свои действия, но и объясняет цепочку своих рассуждений, приводящих к решению. Основной проблемой, возникающей при использовании этого метода, является **принципиальная сложность для любого человека словесного описания собственных мыслей и действий**. Повысить полноту и качество извлекаемых знаний можно за счет многократного уточняющего протоколирования рассуждений эксперта.

**Метод извлечения знаний из лекций** предполагает, что эксперт передает свой опыт инженеру по знаниям в форме лекций. При этом инженер по знаниям может заранее сформулировать темы лекций. Если этого не удается сделать, то инженер конспектирует лекции и задает вопросы. Качество информации, предоставленной экспертом в ходе лекции, определяется четкостью сформулированной темы, а также способностями лектора в структурировании и изложении своих знаний и рассуждений.

**Активные индивидуальные методы** включают **методы анкетирования, интервьюирования, свободного диалога и игры с экспертом**.

**Преимуществом методов анкетирования является то, что анкета или вопросник составляются инженером по знаниям заранее** и используются для опроса экспертов. Составление анкеты следует проводить с учетом рекомендаций, выработанных в социологии и психологии. Основные требования к анкетам такие:

Анкета не должна быть монотонной и однообразной, чтобы не вызывать скуку или усталость. Для этого необходимо разнообразить тематику и формы задания вопросов, включить вопросы-шутки и применить стиль игры.

Анкета должна быть приспособлена к языку эксперта.

Следует учитывать, что вопросы влияют друг на друга, поэтому важно расположить их в правильной последовательности.

В анкете должно содержаться оптимальное число избыточных вопросов, часть которых предназначена для контроля правильности ответов, а другая часть - для снятия напряжения.

**Метод интервьюирования** отличается от метода анкетирования тем, что позволяет аналитику опускать ряд вопросов в зависимости от ситуации, вставлять новые вопросы в анкету, изменять темы и разнообразить ситуацию общения. Важную роль в методе интервьюирования играют типы вопросов, которые классифицируются по форме, по функциям, по воздействию.

По форме вопросы классифицируются следующим образом:

открытый вопрос называет тему или предмет, оставляя эксперту полную свободу в отношении формы и содержания ответа;

закрытый вопрос предлагает эксперту выбрать ответ из предложенного набора;

личный вопрос касается личного опыта эксперта;

безличный вопрос направлен на выявление наиболее распространенных закономерностей предметной области;

прямой вопрос непосредственно указывает на интересующий предмет или тему (используется при «зажатости» эксперта);

косвенный вопрос исподволь затрагивает рассматриваемую проблему;

верbalный вопрос - традиционный устный вопрос;

вопрос с использованием наглядного материала позволяет разнообразить интервью и снять усталость эксперта (используются фотографии, рисунки, карточки).

**По функциям вопросы делятся на основные, зондирующие и контрольные:**

основной вопрос направлен на выявление знаний;

зондирующий вопрос направляет рассуждения эксперта в нужную сторону;

контрольный вопрос проверяет достоверность и объективность информации, полученной в интервью ранее.

**По воздействию выделяют вопросы нейтральные и наводящие:**

**нейтральный вопрос** подчеркивает беспристрастность инженера по знаниям к предмету исследования;

**наводящий вопрос** ориентирует эксперта принять во внимание позицию инженера по знаниям.

Дополнительно в интервью рекомендуется включать следующие вопросы: контактные (снимающие психологический барьер между аналитиком и экспертом); буферные (разграничающие отдельные темы интервью); оживляющие память экспертов (реконструирующие отдельные случаи из практики); «провоцирующие» (способствующие получению неподготовленных ответов).

При использовании метода интервьюирования следует иметь в виду, что его эффективность во многом определяется языком вопросов (понятностью, лаконичностью, терминологи-

ей); порядком вопросов (логическая последовательность); уместностью вопросов (этичностью и вежливостью).

Прежде чем готовить вопросы, аналитик должен овладеть ключевым набором знаний исследуемой предметной области, поскольку любой вопрос имеет смысл только в контексте.

**Метод свободного диалога** позволяет извлекать знания в форме беседы с экспертом, поэтому здесь не предусматривается использование жесткого вопросника или плана. В то же время подготовка к свободному диалогу должна проводиться по специальной методике, в которую входит общая, специальная, конкретная и психологическая подготовка. Общая подготовка направлена на повышение научной эрудиции, овладение общей культурой, знакомство с системной методологией. Специальная подготовка сводится к овладению теорией и навыками интервьюирования. Конкретная подготовка предполагает изучение предметной области, подготовку ситуации общения, знакомство с экспертом, тестирование эксперта. Психологическая подготовка включает знакомство с теорией общения и с когнитивной психологией.

**Активные групповые методы включают «мозговой штурм», дискуссии за круглым столом и ролевые игры. Групповые методы позволяют творчески интегрировать знания множества экспертов.**

**Метод «мозгового штурма»** - один из наиболее известных и широко применяемых методов генерирования новых идей путем творческого сотрудничества группы специалистов. Являясь, в некотором смысле, единым мозгом, группа пытается штурмом преодолеть трудности, мешающие разрешить рассматриваемую проблему. В процессе такого штурма участники выдвигают и развивают собственные идеи, стимулируя появление новых и комбинируя их. Для обеспечения максимального эффекта «мозговой штурм» должен подчиняться определенным правилам и основываться на строгом разделении во времени процесса выдвижения идей и процесса их обсуждения и оценки. **На первой стадии штурма запрещается осуждать выдвинутые идеи и предложения** (считается, что критические замечания уводят к частностям, прерывают творческий процесс, мешают выдвижению идей). **Роль аналитика состоит в том, чтобы активизировать творческое мышление участников заседания и обеспечить выдвижение возможно большего числа идей.**

После выдвижения идей выполняются тщательное их обсуждение, оценка и отбор лучших. На стадии обсуждения участники «мозгового штурма» должны сконцентрироваться на положительных сторонах идей, найти в них рациональные зерна и предложить направления их развития. Выдвигаемые в процессе обсуждения дополнительные идеи могут базироваться на идеях других участников или, наоборот, служить для них фундаментом, катализатором. Значительный эффект дает комбинирование идей при одновременном выявлении преимуществ и недостатков синтезируемых при этом вариантов.

**Метод «мозгового штурма» эффективен при решении не слишком сложных задач общего организационного характера**, когда проблема хорошо знакома всем участникам заседания и по рассматриваемому вопросу имеется достаточная информация.

**Индивидуальный «мозговой штурм»** проводится по тем же правилам, что и коллективный, но выполняется одним экспертом, который одновременно генерирует идеи, дает им объективную оценку и критикует их.

**Массовый «мозговой штурм»** проводится в массовой аудитории (до нескольких десятков человек). Отбор идей проводится на промежуточных этапах. Эксперты группируются по 6-8 человек, при этом важно, чтобы непосредственное отношение к задаче имел лишь руководитель группы, а остальные были лишь знакомы с нею (иначе амбиции могут сыграть негативную роль). Штурм проводится в два этапа. На первом этапе оперативные группы осуществляют прямой коллективный «мозговой штурм».

При этом желательно, чтобы каждая группа работала над задачей, наиболее близкой по тематике к профилю вошедших в нее специалистов. На втором этапе руководители каждой группы в течение нескольких минут оценивают выдвинутые идеи, отбирают из них наиболее интересные и сообщают их на «плenарном заседании».

**Двойной «мозговой штурм»** органически соединяет в себе процессы генерирования идей и их доброжелательной позитивной критики.

**Обратный «мозговой штурм»** отличается от прямого тем, что в нем больше внимания уделяется критике высказанных идей.

**Метод дискуссии за круглым столом** предполагает равноправное обсуждение экспертами поставленной проблемы. Отличительной особенностью метода дискуссии является коллективное рассмотрение предметной области с разных точек зрения и исследование спорных гипотез.

**Экспертные игры** предназначены для извлечения знаний и базируются на деловых, диагностических и компьютерных играх (рис. 3).



Рис. 3. Классификация экспертных игр

По числу участников игры подразделяют на **индивидуальные** (игры с экспертом) и **групповые** (ролевые игры в группе). По применению специального оборудования - игры с тренажерами и игры без реквизита. Особый класс представляют собой компьютерные игры.

**В играх с экспертом инженер по знаниям** берет на себя чью-нибудь роль в моделируемой ситуации. Ролевые игры в группе предусматривают участие в игре нескольких специали-

стов. Участники игры наделяются определенными ролями, а собственно игра проводится по составленному когнитологом сценарию. В целях повышения эффективности ролевых игр в них необходимо вводить элементы состязательности.

**Игры с применением тренажеров** позволяют фиксировать трудноуловимые знания, которые возникают в реальных ситуациях и могут быть потеряны при выходе из них.

**Компьютерные экспертные игры** в настоящее время используются в основном в целях обучения. Они полезны для «разминки» экспертов перед сеансом извлечения знаний.

**Текстологические методы** включают методы извлечения знаний, основанные на изучении текстов учебников, специальной литературы и документов.

**Текстология** - это наука, целью которой является практическое прочтение текстов, изучение и интерпретация литературных источников, а также рассмотрение семиотических, психолингвистических и других аспектов извлечения знаний из текстов.

**Особую сложность представляет извлечение знаний из специальной литературы и методик, поскольку в них очень высока степень концентрации специальных знаний.**

### **1.3. Методы выявления скрытых знаний**

Для выявления скрытых знаний обычно рекомендуются методы **психосемантики**, которые позволяют исследовать структуры сознания через моделирование индивидуальной системы знаний человека и выявлять элементы знаний, которые могут им не осознаваться (латентные, скрытые, имплицитные). Эта наука объединяет методы когнитивной психологии, психолингвистики, психологии восприятия и исследования индивидуального сознания.

**Методы многомерного шкалирования** основаны на статистических методах обработки экспертных оценок сходства между анализируемыми объектами, которые выбираются из определенной шкалы. Результаты обработки представляются в виде точек некоторого координатного пространства. Возможность визуализации результатов является безусловным преимуществом метода, однако она быстро утрачивается с увеличением размерности пространства.

**Шкалированием** называют поиск подпространства, для которого величина  $D^m - D^{m*}$  имеет минимальное значение. Здесь  $D^m$ ,  $D^{m*}$  - матрицы расстояний между объектами (признаками) в исходном пространстве  $D^m$  и в искомом подпространстве  $D^{m*}$ , где  $m$  и  $m^*$  - размерности соответствующих пространств. Если  $m^* = 2$ , шкалирование превращается в проецирование на плоскость. Значения расстояний в матрицах  $D$  могут выбираться экспертом из предложенной шкалы либо вычисляться по совокупности признаков, описывающих объект. Во втором случае расстояния можно вычислить разными способами. Одной из самых популярных метрик является евклидово расстояние:

$$d_{i,j} = \left( \sum_{k=1}^K (x_i^k - x_j^k)^2 \right)^{\frac{1}{2}},$$

где  $x_i^k$ ,  $x_j^k$  – значения  $k$ -го признака у  $i$ -го и  $j$ -го объектов соответственно;  $K$  - общее число признаков.

Расстояния-метрики должны удовлетворять следующим условиям:

$$d(x, y) \geq 0; d(x, x) = 0; d(x, y) = d(y, x); d(x, y) + d(y, z) \geq d(x, z).$$

**Метрическим шкалированием** называют образование новых классов с использованием метрических расстояний. Этот тип обработки данных ориентирован на максимальное сближение числовых значений матриц  $D^m$  и  $D^{m*}$ . Существует также **неметрическое шкалирование**, которое не предъявляет жестких требований к сближению пространств и во многих случаях более оправдано в связи с условностью понятия «расстояние».

Важно отметить, что в шкалировании отыскиваются не новые признаки, а новые пространства, поэтому его результаты следует интерпретировать как восстановленную (на плоскости или в объеме) структуру расположения точек.

**Главными недостатками метода многомерного шкалирования являются:**

- субъективные оценки сходства между объектами и признаками обрабатываются как расстояния в пространстве, а результаты анализируются на основе геометрической интерпретации. Это ограничивает размерность выявляемых пространств и требует серьезного упрощения реальных знаний эксперта, следствием которого могут стать неадекватные БЗ;
- выделенные подпространства не имеют иерархической организации, что затрудняет их интерпретацию;
- используется только один вид отношений между понятиями (отношение сходства).

**Метафорический подход** ориентирован на выявление скрытых составляющих практического опыта эксперта и основан на сравнении объектов предметной области с абстрактными объектами из мира метафор, в результате чего можно выявить новые свойства анализируемых объектов и определить отношение эксперта к ним. Используя метафорические сравнения, эксперт выходит за рамки объективности и действует в соответствии со своими субъективными представлениями.

**Метод репертуарных решеток** широко применяется в психологических исследованиях для выявления личностных свойств, которые проявляются через систему личностных конструктов. Этот метод может применяться и для извлечения знаний. **Репертуарная решетка представляет собой матрицу, которая заполняется экспертом.** Столбцам матрицы соответствуют определенные группы объектов (элементов), в качестве которых могут выступать люди, предметы, понятия, отношения, звуки и др. Строки матрицы соответствуют конструктам,

которые представляют собой биполярные признаки, параметры, шкалы, отношения или способы поведения. **Дж. Келли, автор этого метода**, называл конструктом признак или свойство, определяющие сходство двух или нескольких объектов и их отличие от других объектов. Иными словами, конструкты - это признаки, которые могут использоваться для обобщения и разделения объектов на классы. Конструкты можно применить не к любым объектам, а только в некотором «диапазоне их пригодности».

Примерами личностных конструктов могут быть «умный - глупый», «мужской - женский», «хороший - плохой».

**Самым распространенным и простым методом анализа репертуарной решетки является кластерный анализ.** Слабым местом в теории Дж. Келли является предположение о том, что человек может точно описать конструкты, которые он использует, чтобы объяснить, чем сравниваемые объекты похожи друг на друга и чем отличаются. Процедура выявления и вербализации конструктов очень утомительна для экспертов, поэтому во многих методиках используются готовые наборы конструктов, релевантные рассматриваемым объектам.

## 2. Проблемы структурирования знаний

Ясно, что объем и содержание нужных знаний должны определяться принятыми разработчиками понятием модели предметной области (МПрО) и методикой ее построения. А понятия эти у разных авторов - разные, зависят от целей, которые ставят перед собой авторы, принятых подходов, субъективных предпочтений, ориентации на определенные классы задач, методы формализации и т.д. Сходятся они в одном: нужно выделить объекты предметной области (ПрО), их свойства и отношения между ними. Сегодня чаще говорят о необходимости построения онтологии ПрО как неформальной системы концептуализации знаний, как метода их структурирования. Выделяются три основных подхода к структурированию знаний: структурный, объектный и объектно-структурный.

**Структурный подход** основан на идее алгоритмической декомпозиции, когда каждый модуль системы выполняет один из этапов общего процесса. В рамках этого подхода разработано большое число выразительных средств: диаграммы потоков данных, структурированные словари (тезаурусы), языки спецификации системы, таблицы решений, стрелочные диаграммы, деревья переходов, деревья целей и т.д.

**Объектно-ориентированный подход** связан с декомпозицией, при которой каждый объект рассматривается как экземпляр определенного класса. К базовым понятиям подхода относятся такие как: абстрагирование, класс, иерархия, наследование, типизация, инкапсуляция, модульность, полиморфизм.

- Абстрагирование - это упрощенное описание системы, в которой выделяются ее наиболее существенные свойства и детали, а незначительные аспекты опускаются. Реальность представляется моделью сущности (объекта) и моделью (методом) ее поведения. При

этом объекты соответствуют понятиям ПрО, а методы - операциям, которые должны выполняться над объектами.

- Класс - множество объектов, у которых структура и свойства одинаковы.
- Иерархия - это упорядоченная система абстракций - классов.
- Наследование - соотношение между классами, когда один класс использует структурную или функциональную часть другого класса (или нескольких других).
- Типизация - ограничение, накладываемое на класс, которое препятствует взаимозаменяемости объектов, принадлежащих, разным классам.
- Инкапсуляция - ограничение доступа к внутренней структуре и механизмам функционирования объекта.
- Модульность - свойство системы, допускающее возможность ее декомпозиции на ряд взаимосвязанных частей (модулей).
- Полиморфизм - возможность наделения объекта различными свойствами и стратегиями поведениями. Другими словами, предполагается, что одно имя может соответствовать различным классам объектов, входящим в один суперкласс. То есть объект, обозначенный этим именем, может по-разному реагировать на некоторые действия.
- Объектно-структурный подход предполагает проведение последовательного анализа информации о рассматриваемой ПрО и ее представления в виде стратифицированной модели, в которой все знания раскладываются по ролевым стратам: кто, что, зачем, как, где, когда, почему, сколько и т.п. (табл. 1).

Таблица 1.

Уровень страты	Категория страты	Вид анализа
1	КТО	Организационный анализ: коллектив разработчиков
2	ЧТО	Концептуальный анализ: основные принципы, понятийная структура
3	ЗАЧЕМ	Стратегический анализ: назначение и функции системы
4	КАК	Функциональный анализ: гипотезы и модели принимаемых решений
5	ГДЕ	Пространственный анализ: окружение, оборудование, коммуникации
6	КОГДА	Временной анализ: временные параметры и ограничения
7	ПОЧЕМУ	Причинно-следственный (каузальный) анализ
8	СКОЛЬКО	Экономический анализ: ресурсы, прибыль, окупаемость

### 3. Проблема сборки знаний в единую модель предметной области

Из приведенного обзора поля методов инженерии знаний становится ясно, что рядовой инженер по знаниям (или когнитолог), и не только он, должен все перечисленное реально знать и понимать настолько, чтобы свободно ориентироваться, когда, что и как применять при разработке БЗ ЭС реальной сложности.

Очевидно, что в работе с экспертом (экспертами) должны принимать участие психологи, методологи, игротехники-математики и другие люди, умеющие четко взаимодействовать меж-

ду собой, что само по себе не просто. Но если они даже справляются со своими задачами, возникают вопросы согласования полученных знаний, их представления, визуализации, структурирования.

В итоге формируется весьма запутанная общая картина. Необходим единый метод, с единой концепцией и структурой, который сквозным образом приведет к нужному результату, который в начале никто себе и не представляет и для которого все перечисленные методы будут играть лишь ту роль, для которой они более всего подходят. То есть будут использоваться тогда, когда это действительно нужно и понятно зачем. Очевидно, это должны быть специальный метод, методика, соответствующие программные средства, которые будут увязывать все в единое целое: процессы работы с экспертом при выполнении всех типов анализов, фиксацию получаемых знаний, их обработку, выдачу на выходе результатов в виде готовой к использованию модели предметной области (Б3).

Вопросы.

1. Какие проблемы изучает инженерия знаний?
2. Как интерпретируется понятие «извлечение знаний», какие основные методы используются для его получения?
3. В чем сущность психологических проблем взаимодействия с экспертами?
4. Опишите структуру поля методов получения знаний.
5. Какие методы применяются для получения явных знаний?
6. Что понимается под «скрытыми» знаниями, каковы основные подходы к их получению?
7. Какие основные подходы существуют для решения проблемы структурирования знаний?
8. Какие основные типы моделей представления знаний существуют?

## Л 5. Формально-логические модели

1. Логическая модель представления знаний.
2. Исчисление высказываний.
3. Основы исчисления предикатов.

**Представление знаний** — вопрос, возникающий в когнитологии (науке о мышлении), в информатике и в исследованиях искусственного интеллекта. В когнитологии он связан с тем, как люди хранят и обрабатывают информацию. В информатике — с подбором представления конкретных и обобщённых знаний, сведений и фактов для накопления и обработки информации в ЭВМ. Главная задача в ИИ — научиться хранить знания таким образом, чтобы программы могли осмысленно обрабатывать их и достигнуть тем подобия человеческого интеллекта.

Под термином «представление знаний» чаще всего подразумеваются способы представления знаний, ориентированные на автоматическую обработку современными компьютерами, и, в частности, представления, состоящие из явных объектов ('класс всех мерчендайзеров, 'Вован - индивид') и из суждений или утверждений о них ('Вован - мерчендайзер', 'все мерчендайзеры - лузеры'). Представление знаний в подобной явной форме позволяет компьютерам делать дедуктивные выводы из ранее сохранённого знания ('Вован - лузер').

### 1. Логическая модель представления знаний — модель в представлении знаний.

Основная идея подхода при построении логических моделей представления знаний — вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике.

Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории, задаваемое кортежем:

$S = \langle B, F, A, R \rangle$ , где:

$B$  — счетное множество базовых символов (алфавит);

$F$  — множество (перечислимое) формул (называемых также правильно построенными формулами), построенных из элементов  $B$  с использованием некоторого набора синтаксических правил;

$A$  — выделенное подмножество априори истинных формул (аксиом);

$R$  — конечное множество отношений между формулами, называемое правилами вывода.

Достоинства логических моделей представления знаний:

1. В качестве «фундамента» здесь используется классический аппарат математической логики, методы которой достаточно хорошо изучены и формально обоснованы.

2. Существуют достаточно эффективные процедуры вывода, в том числе реализованные в языке логического программирования Пролог, использующие механизмы автоматического доказательства теорем для поиска и логически осмысленного вывода информации.

3. В базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода.

## **2. Исчисление высказываний**

Рассмотрим логические модели, основанные на классической теории исчисления предикатов 1-го порядка, когда предметная область описывается в виде набора аксиом.

Предикат, в логике — понятие, определяющее предмет суждения (субъект) и раскрывающее его содержание.

Рассмотрение логики предикатов начнем с **исчисления высказываний**.

**Вызованием** называется некоторое предложение, смысл которого можно выразить значениями ИСТИНА (TRUE) или ЛОЖЬ (FALSE).

**Например**, предложения «собака серая» и «собака черная» будут высказываниями. Из простых высказываний можно составить более сложные:

«собака серая или собака черная», «собака серая и собака черная», «если собака не серая, то собака черная».

При этом элементарными высказываниями считаем те, которые нельзя разделить на части. Элементарные высказывания рассматриваются как переменные логического типа.

**Исчисление высказываний.** Символами языка логики высказываний, составляющими ее алфавит, являются:

- логические переменные  $P, Q, R, S, \dots$ ,
- логические константы TRUE (ИСТИНА) и FALSE (ЛОЖЬ),
- логические операции  $\wedge$  («и», конъюнкция),  $\vee$  («или», дизъюнкция),  $\neg$  («не», отрижение),  $\equiv$  («тогда и только тогда, когда», эквиваленция),  $\rightarrow$  («следует», импликация) и круглые скобки.

С помощью элементов алфавита можно построить разнообразные логические формулы. Будем называть выражение, составленное из обозначений высказываний и связок (и, разумеется, скобок), логической формулой, если оно удовлетворяет следующим условиям:

- каждая логическая переменная и константа истинности являются формулами. Например: TRUE,  $P, Q$  и  $R$  — формулы;
- если  $P$  и  $Q$  — формулы, то  $\neg P, P \wedge Q, P \vee Q, P \equiv Q, P \rightarrow Q$  - тоже формулы (запись  $\neg P$  будем заменять на  $\bar{P}$ ).
- других формул не бывает.

Формула, давая некоторое описание мира, может быть как истинной, так и ложной. **Интерпретация** — это утверждение относительно правдивости высказывания в некотором возможном мире.

Интерпретация определяет семантику формулы путем сопоставления символов формул со свойствами объектов среды. Значение формулы ИСТИНА говорит о наличии некоторого свойства и ЛОЖЬ - об отсутствии. Каждое возможное отображение значения истинности высказывания соответствует возможной интерпретации мира. Например, если Р обозначает высказывание «идет дождь», а Q - «я на работе», то набор высказываний {P, Q} имеет четыре различных отображения в таблице истинности {ИСТИНА, ЛОЖЬ}. Эти отображения соответствуют четырем различным интерпретациям.

### **3. Основы исчисления предикатов (ИП)**

Исчисление высказываний является весьма грубой моделью представления знаний. Основной ее недостаток в том, что высказывание здесь рассматривается как единое целое, без анализа его внутренней структуры. Это ограничивает возможности ИВ при моделировании сложных силлогических построений.

Силлогизм (образовано от греческого слова: συλλογισμός — подытоживание, умозаключение). В традиционной формальной логике силлогизмом называют дедуктивное умозаключение, в котором из двух ранее установленных суждений, называемых посылками, получается третье суждение, называемое выводом.

#### **a. Понятие предиката**

Если высказывание отражает какой-либо факт и далее оперирует с ним как с единой формулой, не разделяя его, скажем, на субъекты и объекты, то предикатная форма, напротив, отображает данный факт уже как взаимодействие, отношение или свойство некоторых существ. Это отношение принято выделять прописными буквами перед скобками, в которых указываются те или иные существа, находящиеся в данном отношении.

Рассмотрим несколько предложений:

- a) Лена и Таня сестры;
- b) грибы в лесу;
- c) капля долбит камень;
- d) снег белый;
- e) мальчик послал книгу брату.

В правилах исчисления предикатов эти предложения можно записать следующим образом:

- a') СЕСТРЫ (Лена, Таня);
- б') В (грибы, лес);
- в') ДОЛБИТ (капля, камень);

- г') БЕЛЫЙ (снег);  
д') ПОСЛАЛ (мальчик, книга, брат).

В первом предложении выделено отношение родства, во втором - предлогом В - пространственные отношения. В предложении в') выделено действие между субъектом и объектом, в предложении г') - свойство (в данном случае - цвет), в предложении д') - также действие.

То, что стоит перед скобками, называется **предикатным символом** (предикатной константой), то, что стоит в скобках, - термами. Каждый терм занимает свое место. **Предикатные символы могут быть предлогами, существительными, глаголами, прилагательными и т.п.** Терм, как правило, существительное или то, что его заменяет. Все это вместе образует предикатную формулу (или, короче, - предикат).

Термов может быть несколько. По их количеству предикаты разделяются на одноместные (г'), двуместные (а', б', в'), трёхместные (д') и т.д. Предикатная формула еще называется атомом.

## **б. Алфавит ИП**

ИП - аксиоматическая система, построенная согласно формальной теории  $F = (A, V, W, R)$ .

Алфавит ИП - А содержит:

индивидуальные константы  $a, b, c, \dots, k$ ;

предметные переменные  $x, y, z, \dots, u$ ;

функциональные константы  $f, g, h, \dots, w$ ;

высказывания  $p, q, r, s, \dots, l$ ;

предикатные константы  $P, Q, R, \dots, V$ .

Исчисление предикатов в определенном смысле продолжение и расширение исчисления высказываний, поэтому в алфавит включены все те же пропозициональные связки  $\Lambda, V, \rightarrow, \equiv$ . Но перечень логических знаков в ИП расширяется еще двумя, называемых кванторами:  $\forall$  и  $\exists$ . Квантор  $\forall$ , читается, как «все», «для всех», «всякий», «каков бы ни был» и т.н. Поэтому он называется квантором всеобщности (общности). Квантор всеобщности является сверткой бесконечной конъюнкции:  $\forall x P(x) = P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_k)$ . Квантор существования -  $\exists x P(x)$  - читается, как «некоторый», «хотя бы один», «существует», и является сверткой бесконечной дизъюнкции:  $\exists x P(x) = P(x_1) \vee P(x_2) \vee \dots \vee P(x_k)$ . Выражение  $\forall x P(x)$  читается: «для любого  $x$  выполняется условие  $P(x)$ ». Выражение  $\exists x P(x)$  - «существует хотя бы один  $x$ , при котором выполняется  $P(x)$  (то есть  $P(x) = I$ )».

## **в. Синтаксис ИП**

Множество синтаксических правил  $V$  ИВ применимо и в ИП. Правильно построенные формулы (ППФ) в рамках исчисления высказываний остаются ППФ и в исчислении предикатов. Но формулы ИВ, для того, чтобы участвовать во всех преобразованиях формул ИП должны быть представлены в предикатной форме. То есть в каждом высказывании должен быть выделен предикат и его аргументы.

Добавляются еще правила:

1. атом есть ППФ;
2. если  $P(x)$  - ППФ и  $x$  – переменная, то  $\forall x P(x)$  и  $\exists x P(x)$  - ППФ;
3. если  $P(x, y)$  и  $x, y$  - переменные и  $f$  - функция, то  $\forall x \forall y P(x, f(y))$  и  $\exists x \exists y P(x, f(y))$  - ППФ. Последние формулы верны и для смешанных кванторов.

Каждому квантору соответствует только одна переменная, в наших примерах  $x$  или  $y$ . Эта переменная называется связанной или квантифицированной, она пишется сразу за квантором. Область действия квантора - формула, к которой применяется эта квантификация. Каждое вхождение квантифицированной переменной в область действия квантификации является **связанным**, любая другая переменная в данной области, не являющаяся связанной, называется свободной.

Каждую предикатную формулу можно интерпретировать, то есть оценить ее как И или Л. При этом можно оценить «перекрытие» кванторов на одну и ту же переменную:

$\forall x \exists x P(x)$  интерпретируется как  $\exists x P(x)$ , а  $\exists x \forall x P(x)$  интерпретируется как  $\forall x P(x)$ .

Будем понимать под  $A$  предикат  $A(x, y)$  и отметим важные соотношения:

$$\forall x \forall y A = \forall y \forall x A \quad (6.1)$$

$\exists x \exists y A = \exists y \exists x A$ , то есть одноименные кванторы можно менять местами. Иное дело разноименные кванторы. Здесь выполняется только такое условие:

$$\exists x \forall y A \rightarrow \forall y \exists x A$$

Последняя импликация поясняется следующим примером. Пусть имеем для целых чисел истинное утверждение:  $\forall y \exists x (x + y = 0)$  (для любого  $y$  найдется такой  $x$ , что выполняется равенство  $x + y = 0$ ). Переставим кванторы:  $\exists x \forall y (x + y = 0)$ . Получим выражение: существует такой  $x$ , при котором выполняется условие  $(x + y = 0)$  для всех  $y$ , что некорректно.

#### 4. Базовые аксиомы ИП

Система базовых аксиом  $W$  в ИП может быть принята такой же, как и в ИВ. Однако к ней необходимо добавить аксиомы, учитывающие появление кванторов:

$$(A4) \forall x P(x) \rightarrow P(y),$$

(A5)  $P(y) \rightarrow \exists x P(x)$ .

A4 говорит, что если  $P(x)$  истинен для всех  $x$ , то он истинен и для некоторого  $y$  из этого же универсума (если все яблоки в данном ящике красные, то одно-то красное уж найдется всегда).

A5 говорит, что если найдется  $y$ , при котором  $P(y)$  истинен, то верно, что найдется хотя бы один  $x$ , для которого предикат  $P(x)$  тоже истинен (даже если  $x$  совпадает с  $y$ ). (Если среди яблок в данном ящике нашлось одно сладкое, то уже существует, по крайней мере, одно сладкое.)

## 5. Правила вывода в ИП

Правила вывода  $R$  остаются прежними: правило подстановки и правило заключения, но они дополняются еще одним правилом, учитывающим свойства кванторов. Это правило называется правилом специализации. Суть его в следующем: если ППФ  $\forall x P(x)$  истинна и  $b$  - некоторая константа, то формула  $P(b)$  также истинна, то есть справедливо  $\forall x P(x) = P(b)$ .

Пусть, например, имеются формулы  $\forall x(P(b) \rightarrow Q(x))$  и  $P(b)$ . Если они истинны, то, применяя специализацию, имеем ряд теорем:

$\forall x(P(b) \rightarrow Q(x))$ , (специализация)

то есть  $P(b) \rightarrow Q(b)$  (modus ponens с  $P(b)$ )

## 6. Примеры предикатов

Разберем несколько примеров построения предикатов.

1. «**А вы, друзья, как ни садитесь, все ж в музыканты не годитесь».**

Обозначим:  $x$  - способ рассаживания музыкантов,  $y$  - качество исполнения,  $P(x, y)$  - предикат, связывающий способ рассаживания и качество исполнения. Окончательная формула:  $\forall x \bar{P}(x, y)$ .

2. «**Кто не работает, тот не ест»:**  $\forall x(P(x) \rightarrow E(x))$ .

Здесь  $x$  - человек,  $P$  - предикатная константа РАБОТАЕТ,  $E$  - предикатная константа ЕСТЬ.

3. «**Болтун - находка для шпиона**»:  $\forall x \exists y P(x, y)$

где «роли исполняют»:  $x$  - болтун,  $y$  - шпион,  $P$  - НАХОДКА.

4. Силлогизм о Сократе можно переписать так: для всех  $x$ , если  $x$  - человек, то  $x$  - смертен; Сократ человек; (следовательно) Сократ смертен. Обозначим через  $M$  «быть смертным», через  $Я$  «быть человеком». Мы приходим к следующей формуле:

$\forall x((H(x) \rightarrow M(x)) \wedge H(\text{Сократ})) \rightarrow M(\text{Сократ})$ .

(Сократ здесь - индивидная константа).

5. «В каждом городе найдется краевед, который покажет достопримечательности»:

$$\forall x(G(x) \rightarrow \exists y(S(x, y) \wedge P(z, y))$$

(Если  $x$  - город ( $G$ ), то найдется такой краевед ( $y$ ), ЖИВУЩИЙ\_В ( $S$ ), который ПОКАЖЕТ ( $P$ ) достопримечательности ( $z$ ).)

Формулы ИП часто можно писать по-разному на один и тот же словесный текст, пользуясь разной степенью «детализации». Например, «Болтун - находка для шпиона» можно записать более подробно:

$$8. \forall x((M(x) \wedge B(x)) \rightarrow \exists y P(x, y)),$$

где  $M$  - ЧЕЛОВЕК,  $B$  - БОЛТУН,  $P$  - НАХОДКА,  $y$  - шпион.

### Упражнения.

1. Попробуйте представить в виде предикатных формул следующие фразы:

- Кто весел, тот смеется;
- Кто-то привык за победу бороться;
- И никто ему по-дружески не спел;
- Всяк сверчок знай свой шесток;
- И никто не узнает, где могилка моя;
- Все цветы мне надоели, кроме роз;
- Есть многое на свете, друг Гораций, что и не снилось нашим мудрецам;
- Немногие вернулись с поля, не будь на то Господня воля, не отдали б Москвы.

2. Пусть  $L$  означает **ЛЮБИТ**,  $\mathfrak{ц}$  - цветы,  $\mathfrak{к}$  - конфеты,  $\mathfrak{x}$  - девушка.

Переведите на русский язык выражения:

- $\forall x L(x, \mathfrak{ц})$ ,
- $\exists x \bar{L}(x, \mathfrak{к})$ ,
- $\exists x L(x, \mathfrak{к})$ ,
- $\forall x L(x, \mathfrak{ц}) \wedge L(x, \mathfrak{к})$ ,
- $\exists x (\bar{L}(x, \mathfrak{к}) \rightarrow \bar{L}(x, \mathfrak{ц}))$ .

### Преобразование формул в ИП

Стандартизация переменных	Подстановки и унификация
Исключение квантора существования	Логический вывод в ИП
Предваренная форма	Стратегии резолюции
Исключение квантора общности	Дерево опровержения
Приведение матрицы к КНФ	

## Старая версия лекции (2016 год)

**Логика предикатов первого порядка** является более выразительным средством, чем логика высказываний, и позволяет представлять знания о среде более компактно.

Приведем понятие **предикат**, данное Д. А. Поспеловым: «Под предикатом будем понимать некоторую связь, которая задана на наборе констант или переменных». Пример предиката: «*A* больше *B*». При задании семантики (т. е. определении переменных *A* и *B*) можно будет судить об истинности предиката.

Предикат принимает только два значения ИСТИНА или ЛОЖЬ.

Исчисление предикатов. Основными синтаксическими единицами логики предикатов являются константы, переменные, функции, предикаты, кванторы и логические операторы. Формальный синтаксис логики предикатов первого порядка можно представить с помощью **языка Бэкуса—Наура**, который обычно применяется для записи грамматик языков программирования:

```

⟨константа⟩ → ⟨идентификатор1⟩
⟨переменная⟩ → ⟨идентификатор2⟩
⟨функция⟩ → ⟨идентификатор3⟩
⟨предикат⟩ → ⟨идентификатор4⟩
⟨терм⟩ → ⟨константа⟩ | ⟨переменная⟩ | ⟨функция⟩(⟨список термов⟩)
⟨список термов⟩ → ⟨терм⟩ | ⟨терм⟩{, ⟨терм⟩}
⟨атом⟩ → ⟨предикат⟩ | ⟨предикат⟩ | (⟨список термов⟩)
⟨литера⟩ → ⟨атом⟩ | ¬⟨атом⟩
⟨оператор⟩ → ∧ | ∨ | → | ≡
⟨список переменных⟩ → ⟨переменная⟩ | ⟨переменная⟩ | {, ⟨переменная⟩}
⟨квантор⟩ → (Э⟨список переменных⟩) | (В⟨список переменных⟩)
⟨формула⟩ →
⟨литера⟩ | ((⟨формула⟩)) | ⟨квантор⟩(⟨формула⟩)|((⟨формула⟩))⟨оператор⟩(⟨формула⟩))

```

В данной записи слева в угловых скобках приводятся **типы синтаксических объектов**, в правой части записи приводятся возможные **способы организации синтаксически корректных объектов определяемого типа**. При этом знак «|» интерпретируется как знак «или»; заключение конструкции в скобки {} означает, что эта конструкция может повторяться нуль или более раз.

**Номера идентификаторов** следует понимать в том смысле, что идентификаторы, используемые для обозначения объектов разных типов, должны быть различны.

Например, ⟨идентификатор1⟩ обозначает константу, которая формируется из строчных букв.

**Имена переменных** (идентификатор2) должны начинаться с заглавных букв.

**Имена предикатов** (идентификатор4) должны состоять из прописных букв.

**Имена функций** (идентификатор3) состоят из строчных букв, при этом первой буквой является f, g, h, p, q.

**Функции, как и предикаты**, задают связь между переменными и константами, но эта связь не характеризуется логическим значением. С помощью функции можно представить сложный объект, представляющий набор информации о мире. **Предикат и функция отличаются также и на синтаксическом уровне, так как функция может являться аргументом предикатов, а предикат — нет.**

Математически строго формулы логики предикатов определяются рекурсивно:

- 1) предикат есть формула;
- 2) если A и B - формулы, то  $\neg A$ ,  $A \wedge B$ ,  $A \vee B$ ,  $A \rightarrow B$ ,  $A \equiv B$  - тоже формулы;
- 3) других формул не бывает.

Таблица истинности связок предикатов (И — ИСТИНА, Л — ЛОЖЬ), позволяющая определить, истинно или ложно значение формулы - связки при различных значениях входящих в нее предикатов A и B.

A	B	$A \wedge B$	$A \vee B$	$\neg A$	$A \rightarrow B$	$A \equiv B$
И	И	И	И	Л	И	И
И	Л	Л	И	Л	Л	Л
Л	И	Л	И	И	И	Л
Л	Л	Л	Л	И	И	И

Здесь выражение  $A \rightarrow B$  аналогично «ЕСЛИ A, ТО B» в естественном языке.

Многие формулы логики предикатов требуют использования **кванторов**, определяющих область значений переменных — аргументов предикатов.

Условимся, что выражение «все течет, все изменяется» является правильным. Если определить, что предикат ТЕЧИЗМ(x) описывает «x течет, изменяется», то этот предикат является истиной **при подстановке любой сущности реального мира в x**.

Выделенная фраза обозначается через  $(\forall x)$  (перевернутое A от английского All — все), она обычно читается как «для всех» и записывается перед предикатом в виде  $(\forall x) F(x)$ .

В логике предикатов имеется еще одна конструкция для суждения о значении предикатов по их переменным. Этот квантор эквивалентен суждению **«существует, по крайней мере, одно x такое, что F(x) — истина»**.

Выделенная фраза записывается в виде  $(\exists x)$  (перевернутое E от английского Exists — существует), а все суждение представляется как  $(\exists x)F(x)$ .

Символы, которые означают «для всех» –  $\forall$  и «существует» -  $\exists$  называются соответственно **квантором общности** и **квантором существования**. В логике предикатов первого порядка не разрешается применение кванторов к предикатам.

Переменная, которая **проквантифицирована**, называется связанной переменной, а переменная, которая не связана кванторами, называется свободной переменной.

Формула, в которой все переменные связаны, называется предложением. Каждому предложению ставится в соответствие определенное значение — «истина» или «ложь».

Сложные формулы в логике предикатов получаются путем комбинирования атомарных формул с помощью логических операций. Такие формулы называются правильно построенным логическим формулами (ППФ). Интерпретация ППФ возможна только с учетом конкретной области интерпретации, которая представляет собой множество всех возможных значений термов, входящих в ППФ. Для представления знания в данной предметной области необходимо установить область интерпретации, т. е. выбрать константы, которые определяют объекты в данной области, а также функции и предикаты, которые определяют зависимости и отношения между объектами. После этого уже можно построить логические формулы, описывающие закономерности данной предметной области. Все это возможно, когда знания являются полными, четкими и надежными, в противном случае записать знания с помощью логической модели не удается.

### **Пример.**

Введем обозначения:

$A(x)$  = «студент  $x$  учится отлично»;

$B(x)$  = «студент  $x$  получает повышенную стипендию».

Формула  $A(\text{Иванов}) \rightarrow B(\text{Иванов})$  означает: «студент Иванов учится отлично, следовательно, студент Иванов получает повышенную стипендию».

Формула с квантором общности  $(\forall x)(A(x) \rightarrow B(x))$  означает: «каждый студент, который учится отлично, получает повышенную стипендию».

## Л 6. Сетевые и фреймовые модели представления знаний

1. Семантические сети. Понятия, события и свойства. Виды семантических сетей. Семантические сети Куиллиана. Основной способ интерпретации семантической сети.
2. Фреймы. Слоты. Структура фрейма. Указатели наследования и типа данных. Значение слота. Присоединенные процедуры.

### 1. Семантические сети

Термин "семантическая" означает "смысловая", а сама семантика - это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков.

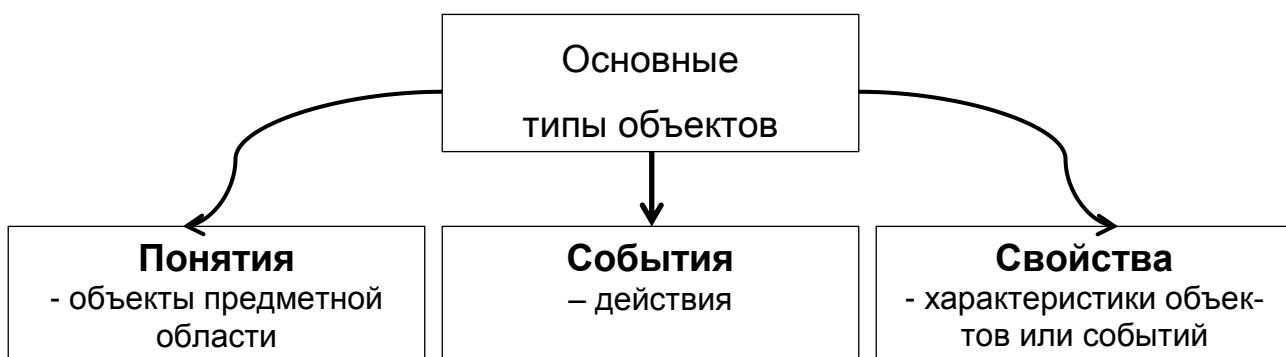
**Основная идея подхода к представлению знаний, основанных на аппарате семантических сетей, состоит в том, чтобы рассматривать проблемную среду как совокупность сущностей (объектов) и (отношений) связей между ними.** Система знаний представляется семантической сетью. Сущности представляются при этом поименованными вершинами, а отношения - направленными поименованными ребрами.

Семантическая сеть наиболее близка к тому, как представляются знания в текстах на естественном языке. В ее основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек (**a r b**), где **a** и **b** - два объекта или понятия, а **r** двоичное отношение между ними.

Графически семантическая сеть представляется в виде помеченного ориентированного графа, в котором **вершинам** соответствуют **объекты** (понятия), а **дугам** - их **отношения**. Дуги помечаются именами соответствующих отношений.

### Понятия, события и свойства

Используются три основных типа объектов: **понятия, события и свойства**.



### Виды семантических сетей

Семантическая сеть является моделью широкого предназначения. Выделяют следующие виды семантических сетей:

**ситуационные** сети, которые описывают временные, пространственные и причинно-следственные (**каузальные** - зависимость между переменными или понятиями, при которой изменение в одной (или более) переменной или понятии влечет за собой изменение в другой (других)) отношения;

**целевые** сети, используемые в системах планирования и синтеза, которые описывают отношения "цель-средства" и "цель-подцель";

**классификационные** сети, использующие отношения "род-вид", "класс-подкласс";

**функциональные** сети, использующие отношения "аргумент-функция" и т.д.

Особенность семантической сети как модели представления знаний, которая может одновременно считаться и ее достоинством, и ее недостатком, заключается в невозможности в явном виде разделить БЗ и механизм логического вывода. Поэтому интерпретация семантической сети осуществляется только с помощью использующих ее процедур.

### **Семантические сети Куиллиана**

При построении семантической сети отсутствуют ограничения на число связей и сложность сети. Семантическая сеть должна быть систематизирована для того, чтобы формализация оказалась возможной. Семантические сети TLC (Куиллиана) систематизируют функции отношений между понятиями с помощью следующих признаков:

- множество-подмножество (типы отношений «целое-часть»; род-вид» и т. д.);
- отношений «близости»;
- отношений «сходства-различия»;
- логических связей (И, ИЛИ, НЕ);
- количественных связей (больше, меньше, равно, ...);
- пространственных связей (далеко от, за, над, ...);
- временных связей (раньше, позже, в течение, ...);
- атрибутных связей (иметь свойство, иметь значение);
- лингвистических связей и др.

**Основной идеей семантической сети TLC** было описание значений класса, к которому принадлежит объект, его прототипа и установление связи со словами, отображающими свойства объекта.

Пример семантической сети приведен на рис. 6.1.

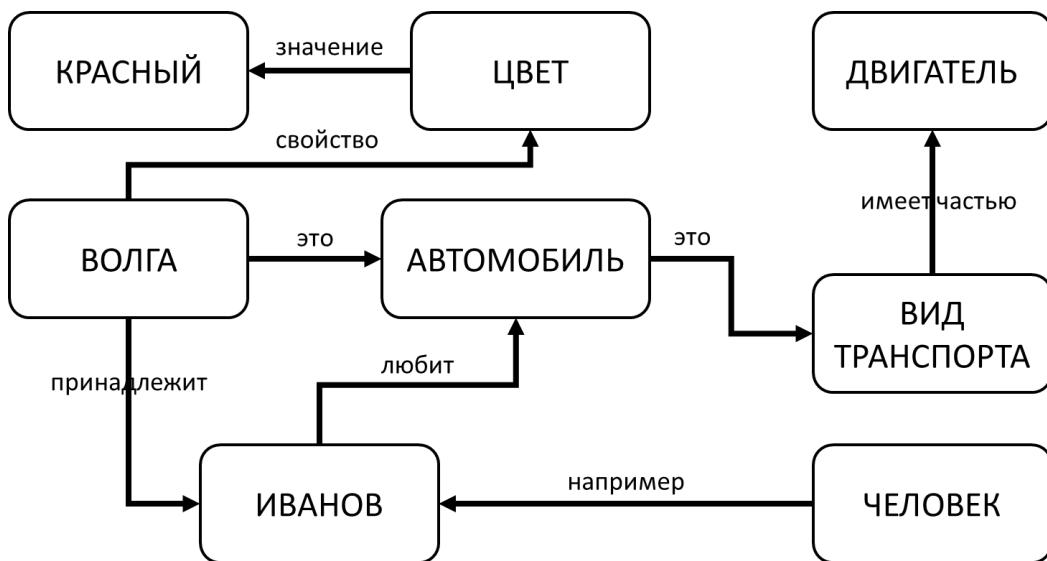


Рис. 6.1. Семантическая сеть

Для реализации семантических сетей существуют специальные языки: **.NET**, язык **SIMER+MIR** и др.

**PROSPECTOR, TORUS** — известные экспертные системы, использующие семантические сети в качестве модели представления знаний.

#### **Основной способ интерпретации семантической сети.**

Основным способом интерпретации семантической сети является **способ сопоставления частей сетевой структуры**. Он основан на построении подсети (подграфа), соответствующей задаваемому вопросу, и сопоставлении ее с общей сетью, имеющейся в БЗ. Запросная подсеть накладывается на имеющийся в базе знаний фрагмент.

Для поиска отношений между концептуальными объектами используется другой способ перекрестного поиска. Согласно этому способу ответ на вопрос выводится путем обнаружения в имеющейся сети узла, в котором пересекаются дуги, исходящие из различных узлов запросной подсети.

Систематизация отношений семантической сети является сложной задачей, зависящей от специфики знаний предметной области. Общезначимые отношения, присутствующие во многих предметных областях, являются основой концепции семантических сетей. Семантические сети получили широкое распространение в системах распознавания речи и экспертных системах. Необходимость структуризации семантических сетей привела к появлению концепции фреймов.

## **2. Фреймы**

Фреймовая модель представления знаний основана на теории фреймов М. Минского, которая представляет собой психологическую модель памяти человека и его сознания.

**Фрейм** - это абстрактный образ для представления некоего стереотипа восприятия (объекта, понятия или ситуации) путем сопоставления факта с конкретными элементами и

значениями в рамках, определенных для объекта, в структуре БЗ. Под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Фреймом также называется и формализованная модель для отображения образа.

**Фреймовая МПЗ** базируется так же на понятии функционального программирования - способа составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программ на части является введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции - оператор суперпозиции других функций. Оно следует из такого раздела математики, как лямбда-исчисление, созданное американским ученым А. Чёрчем.

Фрейм имеет имя, служащее для идентификации описываемого им понятия. Различают фреймы-образцы, или прототипы, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма отображает все многообразие знаний о мире через:

- фреймы-структуры, которые применяются для обозначения объектов и понятий (заем, залог, вексель);
- фреймы-роли (менеджер, кассир, клиент);
- фреймы-сценарии (банкротство, собрание акционеров, и т. д.);
- фреймы-ситуации (тревога, авария, рабочий режим устройства и др.).

## Слоты

**Фрейм** - структура, состоящая из имени фрейма, слотов (имена и значения) и присоединенных процедур, связанных с фреймом или со слотами. Основные структурные элементы фрейма определяются с помощью слотов. Текущие значения слотов помещаются в **шпации**. Число слотов в каждом фрейме устанавливается проектировщиком системы, при этом часть слотов определяется самой системой для выполнения специфических функций (**системные слоты**), примерами которых являются: **слот-указатель родителя данного фрейма (АКО)**, слот-указатель дочерних фреймов, слот для определения даты создания фрейма и т. д.

Слот имеет уникальное в пределах фрейма имя; в качестве имени слота может выступать любой произвольный текст. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения. Системные слоты могут иметь имена: AKO, RELATIONS и т. д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

Слот может содержать не только конкретное значение, но также имя процедуры, позволяющей вычислить это значение по заданному алгоритму. Такие процедуры называются присоединенными или связанными процедурами. Вызов связанной процедуры осуществляется при обращении к слоту, в котором она помещена. Например, слот «возраст» может одержать имя процедуры, которая вычисляет возраст по дате рождения, записанной в другом слоте, и текущей дате.

В слоте могут содержаться данные сложных типов: массивы, списки, множества, фреймы и т. д. Значение слота может представлять собой арифметическое значение, фрагмент текста и т. д.

Формально структура фрейма может быть представлена как список свойств:

```
(<имя_фрейма>
<имя_слота1> (<значение_слота1>) <тип_значения_слота1>
<имя_слота2> (<значение_слота2>) <тип_значения_слота2>
.
.
.
<имя_слота N> (<значение_слота N>) <тип_значения_слота N>)
```

Зададим, например, фрейм для объекта «Преподаватель»:

(Преподаватель		
ФИО	(Петров И. П.)	строка_символов
Уч_степень	(канд)	строка_символов
Уч_звание	(доцент)	строка_символов
Должность	(доцент)	строка_символов
Возраст	age(дата_рожд)	процедура
Адрес	Дом_адрес	фрейм)

### Структура фрейма

Фрейм имеет структуру, представленную в табл. 6.1.

Таблица 6.1

Структура фрейма

Имя фрейма		
Имя слота	Значение слота	Значение типа данных слота

Совокупность данных предметной области может быть представлена множеством взаимосвязанных фреймов, образующих единую фреймовую систему, в которой объединяются декларативные и процедурные знания, такая система имеет, как правило, иерархическую структуру, в которой фреймы соединены друг с другом с помощью родовидовых связей. На верхнем уровне находится фрейм, содержащий наиболее общую информацию, истинную для всех фреймов.

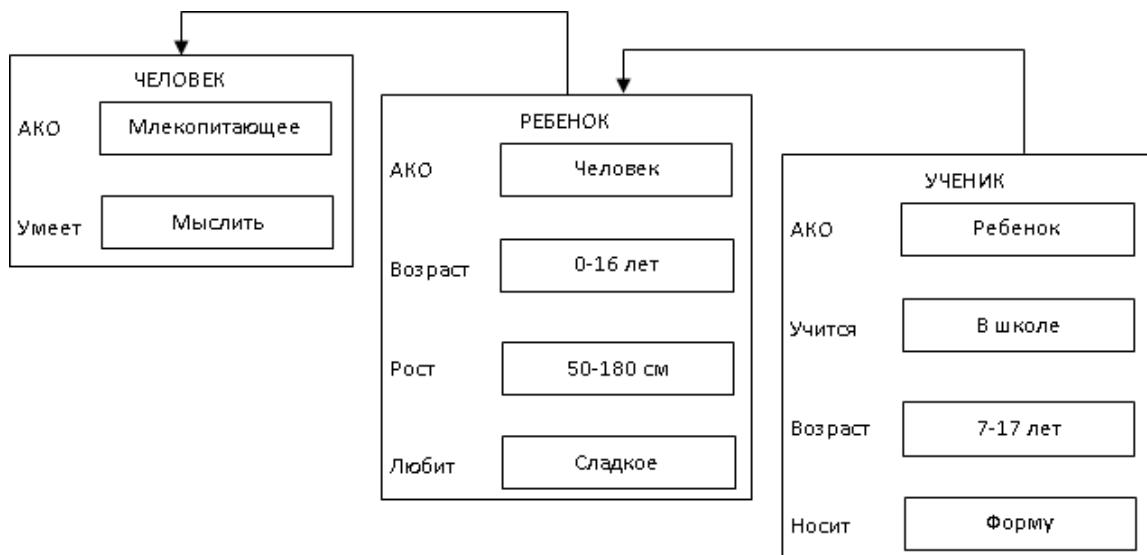
## Указатели наследования и типа данных. Значение слота.

**Важнейшим свойством теории фреймов** является заимствование из теории семантических сетей - так называемое наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО -связям. Слот связи указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов.

**Указатели наследования** показывают, какую информацию об атрибутах слотов из фрейма верхнего уровня наследуют слоты с одинаковыми именами в данном фрейме. Указатели наследования характерны для фреймовых систем иерархического типа. Наследование происходит по АКО-связям (A-Kind-Of). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда наследуются значения одинаковых слотов.

В конкретных системах указатели наследования могут быть организованы различными способами: U (Unique) - значение слота не наследуется; S (Same) - значение слота наследуется; R (Range) - значение слота должно находиться в пределах интервала значений, указанных в одноименном слоте родительского фрейма; O (Override) - при отсутствии значения в текущем слоте оно наследуется из фрейма верхнего уровня].

Например, в сети фреймов на рис. 6.2 понятие «ученик» наследует свойства фреймов «ребенок» и «человек», которые находятся на более высоком уровне иерархии. На вопрос «любят ли ученики сладкое?» следует ответ «да», так как этим свойством обладают все дети, что указано во фрейме «ребенок». Наследование свойств может быть частичным, так как возраст для учеников не наследуется из фрейма «ребенок», поскольку указан явно в своем собственном фрейме.



**Указатель типа данных** показывает тип значения слота. Наиболее употребляемые типы: frame - указатель фрейма; real, integer, Boolean, text, table, expression, lisp - вещественный, целый, логический текстовый тип, таблица, выражение, связанная процедура и т. д.

Значение слота должно соответствовать указанному типу данных и условию наследования. Существует несколько способов получения слотом значений во фрейме-экземпляре:

- по умолчанию от фрейма-образца (значение по умолчанию);
- через наследование свойств от фрейма, указанного в слоте АКО;
- по формуле, указанной в слоте;
- через присоединенную процедуру;
- явно из диалога с пользователем;
- из базы данных.

## **Присоединенные процедуры**

Во фреймах различают два типа присоединенных процедур: **процедуры-демоны** и **процедуры-слуги**.

Демоны (процедуры) автоматически запускаются при обращении к соответствующему слоту. Типы демонов связаны с условием запуска процедуры. Демон с условием IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено. Демон IF-ADDED запускается при попытке изменения слота. Демон IF-REMOVED запускается при попытке удаления значения слота. Возможны также другие типы демонов. Процедуры-слуги запускаются явно.

В качестве значения слота может быть присоединенная процедура, называемая **служебной** в языке LISP или **методом** в языках объектно-ориентированного программирования. Присоединенная процедура запускается по сообщению, передаваемому из другого фрейма. Демоны и присоединенные процедуры являются процедурными знаниями, объединенными с декларативными в единую систему.

Фреймовые системы в последние годы реализуют на объектно-ориентированных языках, но в этих языках нет средств для реализации присоединенных процедур, поэтому требуется интеграция объектно-ориентированного языка с другими средствами обработки знаний (например, с языком PROLOG). Существуют также специализированные языки, такие как семейство KL(1) представления знаний на основе фреймовой модели.

Основным преимуществом фреймов как модели представления знаний является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность. Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language).

## **Достоинства и недостатки фреймового представления**

К **достоинствам** фреймового представления знаний следует отнести то, что оно:

- 1) обеспечивает эффективную реализацию процедур вывода;

- 2) имеет возможность логических скачков, т.е. немонотонного вывода;
- 3) обеспечивает возможность образования СС фреймов, что дает большую экономию памяти при представлении информации за счет наследования свойств фреймов более высоких уровней во фреймах более низких уровней;
- 4) обеспечивает хорошее соответствие реальной действительности;
- 5) позволяет комбинировать различные модели представления знаний, объединяя их достоинства и компенсируя их недостатки.

К недостаткам отнесем:

- 1) каждый фрейм представляет собой достаточно сложный фрагмент знаний. Поэтому удаление или включение нового фрейма - весьма болезненная процедура, так как должна предусматривать и удаление всех составляющих элементов, которые могут быть составными частями других фреймов;
- 2) достаточно сложно осуществлять на фреймах представление временных процессов;
- 3) отсутствует формальная теория вывода на фреймах. Поэтому на инженере по знаниям целиком лежит ответственность за корректность организации иерархии фреймов и их заполнения.

### **Гибридные модели представления знаний**

При проектировании баз знаний сложных предметных областей, в которых необходимо представление знаний всех известных типов - иерархического, декларативного, процедурного, разрабатываются гибридные модели, сочетающие продукционную, семантическую и фреймовую модели.

Наиболее популярными являются следующие **гибридные модели**: семантико-продукционная, семантико-фреймовая, фреймо-продукционная, семантико-фреймо-продукционная. В таких моделях разработчики стремятся компенсировать недостатки моделей одного типа достоинствами других. Например, недостатком продукционной модели является неэффективность вывода и нечитабельность (т.е. трудность понимания того, что делает система в процессе вывода) при больших БЗ. Этот недостаток хорошо компенсируется фреймовой моделью, когда декларативные знания представляются в виде фреймов иерархического типа, а процедурные знания - в виде слотов обращения к присоединенным процедурам. В итоге формируется иерархия, вершинами которой являются фреймы, моделирующие левую часть правил-продукций, а правая часть правил моделируется слотами - присоединенными процедурами. Если знания имеют слабую иерархичность или вложенность, то удобнее семантико-фреймовая модель, в которой вершины задаются фреймами, а связи между ними описываются семантической сетью.

Во фреймо-продукционной модели база знаний состоит из продукции, задающих причинно-следственные отношения между простыми и сложными объектами (любыми сущностями) моделируемой предметной области. При этом в качестве сложных объектов выступают фреймы. На основе объектов-условий определяется значение выделенного объекта-цели, играющего роль одной из гипотез решения. Вывод истинной (т.е. наилучшей с точки зрения имеющейся БЗ) гипотезы-решения может осуществляться как по прямой, так и по обратной стратегиям. При прямом выводе обход дерева условий выполняется от корня до соответствующей заданным условиям гипотезы. При обратном выводе выполняется проверка истинности условий, при которых предполагаемая гипотеза оказывается истинной. Возможно применение и смешанной стратегии вывода.

## **Л 6-1. Представление знаний в виде семантической сети (дополнительная лекция 1)**

Общие понятия и определения. Роль отношений в СС. Свойства отношений. Предикатные семантические сети. Атрибутивные семантические сети. Вывод на семантических сетях. Достоинства и недостатки семантических сетей.

### **Общие понятия и определения**

Само название «семантическая сеть» (СС) акцентирует внимание на смысле (семантика - смысловая сторона языка, «семантический» - связанный со смыслом), т.е. на значении тех слов, предложений, ситуаций, состояний, которые входят в модель предметной области и которые необходимо описать адекватно нашему (человеческому) пониманию таким образом, чтобы с этим смыслом можно было работать, т.е. моделировать процессы, рассуждения, выводы, сравнения, сопоставления и т.п. Очевидно, это непросто, хотя и возможно многими способами. Поэтому, не вдаваясь в языковые, психологические и другие подробности, под семантикой будем понимать отношения между некоторыми сущностями, в качестве которых могут выступать отдельные символы, слова, предложения, пакеты, пентаграммы, образы, т.е. все, что необходимо для описания объектов (процессов, состояний и т.п.) предметной области.

В психологии, например, изучались семантические объекты как структурные модели долговременной памяти человека, которые затем использовались для создания программ, понимающих естественный язык (ЕЯ). Эта проблема стала особенно актуальной в связи с необходимостью разработки систем автоматического перевода с одного языка на другой. Формализовать текст - значит построить по тексту на ЕЯ соответствующий текст на некотором формальном языке, описывающий языковую ситуацию. Таким образом, можно говорить о моделях представления семантики знаковых (языковых) систем и процедурах их обработки (понимания, распознавания, классификации и т.п.). С появлением СИИ такой подход стал использоваться для формализации знаний о предметной области. Семантическая сеть (СС) является удобным способом графического описания объектов предметной области. При этом под объектом может пониматься процесс, состояние, какая-либо конкретная сущность или сущность обобщённая (класс) и т.п.

Более строго определим СС как направленный граф с помеченными вершинами и дугами. При этом вершины отождествляются с сущностями (объектами) предметной области, а дуги - с отношениями между ними.

Большое разнообразие СС формируется исходя из следующих соображений:

- является ли вершина СС простой или сложной?
- сколько и какие отношения используются в СС?
- какие процессы необходимо моделировать на СС?

Будем говорить, что вершина СС является простой (ординарной), если она не имеет внутренней структуры. Другими словами, простые вершины отождествляются с терминальными

(т.е. более не раскрываемыми) сущностями моделируемой среды. В качестве таких вершин могут, очевидно, фигурировать числа, лексемы, пентаграммы и т.п.

Сложные вершины допускают раскрытие вплоть до семантической сети, в которой участвуют только простые вершины. Таким образом, СС со сложными вершинами может разворачиваться, как матрёшка.

В связи с многообразием отношений и сложностью исследования их свойств на практике стараются использовать минимальное число отношений для упрощения СС. И в зависимости от того, какие типы отношений используются, можно выделить несколько видов СС.

1. *Однородные СС*. В них используется один тип отношений (неважно какой, но один).

В этом смысле любые ориентированные графы являются примерами однородных СС.

2. *Иерархические СС*. Здесь используются отношения: часть - целое (Part-of-while), класс - подкласс или род - вид (АКО - A Kind Of), экземпляр класса - класс (Is A). Каждое из этих отношений образует свою иерархию. Яркими примерами иерархических СС являются организационные структуры, выстраиваемые по отношению «целое - часть». (Например, схема центральной избирательной комиссии: областная, городская, окружная, районная, участковая.)

3. *Сценарные СС*. Используются отношения строгого или нестрогого порядка. Сюда же могут быть отнесены СС с отношениями времени, причина - следствие.

4. *Функциональные СС*. В них используются функциональные отношения типа «вход - выход» (рис. 1).

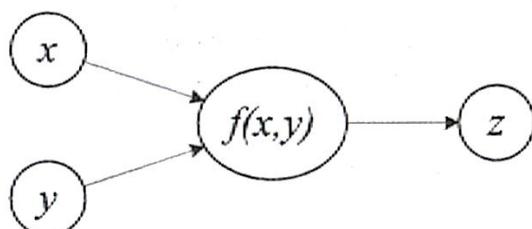


Рис. 1. СС типа «вход - выход»

На рис. 2 приведен пример СС с простой структурой.

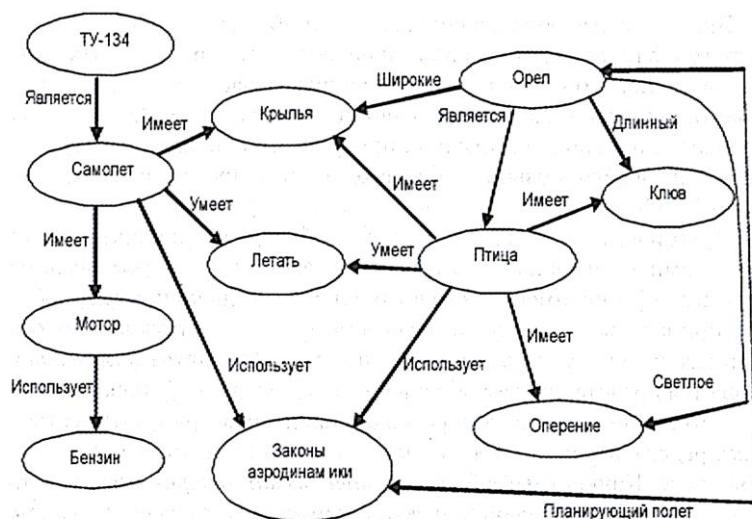


Рис. 2. СС самолета и птицы

## Роль отношений в СС

Выделим наиболее важные (фундаментальные) типы отношений между объектами, образующие «скелет (структуру)» и позволяющие более свободно ориентироваться в сложной и большой СС. Так, между двумя классами объектов, представленными в СС своими именами, может существовать родовидовое отношение *AKO* (*A kind of*). Наличие его между классами объектов *A* и *B*, например, означает, что любой объект, отображаемый понятием *B*, отображается и понятием *A*, и существует такой объект, который отображается понятием *A*, но не отображается понятием *B*. Иными словами, понятие *A* более общее, чем *B*. В таком случае говорят, что *A* является родом *B* или что понятие *A* - родовое для *B*. Так, понятие *животное* (и, соответственно, обобщенный объект *животное* в некоторой проблемной среде) родовое для понятия «человек», а понятие *транспорт* - родовое для понятия *самолет*.

Видовая связь обратна родовой. Если объект *A* является родом *B*, то говорят также, что *B* вид *A*, и наоборот. Из приведенных примеров видно, что родовое понятие не охватывает всех свойств видового, так как видовое понятие богаче содержанием. Все свойства родового понятия, как правило, присущи и видовому (наследование свойств). Это позволяет более компактно представить систему знаний СС.

Другой важный вид связи между обобщенными и конкретными объектами - отношение *является представителем* (элементом) (связь *IS A*). Она имеет место в том случае, когда конкретный объект принадлежит классу, отображаемому соответствующим обобщенным объектом. Так, конкретный объект *Третьяковская галерея* является представителем обобщенного объекта *музей*. Важно отметить, что один и тот же конкретный объект может рассматриваться как представитель нескольких классов в одной и той же проблемной среде. Конкретный объект *Третьяковская галерея* может быть представителем классов объектов *картичная галерея* и *здание в Москве*. Первый из них имеет родовидовую связь с объектом *музей*, второй - нет, так как музей может быть размещен в нескольких зданиях или на открытом воздухе и не всякое здание в Москве - музей. Свойства, присущие обобщенному объекту - классу, характеризуют и любой конкретный объект представитель. Таким образом, множество свойств конкретного объекта содержит в себе подмножество свойств, которыми он наделяется как представитель тех или иных обобщенных объектов (или совпадает с этим подмножеством).

Между сложным (составным) объектом и каким-либо другим объектом проблемной среды может существовать связь «являться частью целого» (*Part of while - POW*). Ясно, что частью конкретного составного объекта не может являться объект класс.

Типизация объектов и фундаментальные отношения не решают всех проблем представления знаний, но создают хорошую основу для построения прикладной базы знаний. Типизация обеспечивает и реальные осязаемые преимущества, позволяя сжать базу знаний, сделать ее более компактной. Так, для объекта класса, являющегося видом некоторого другого, нет необходимости представлять в базе все свойства, по крайней мере, часть их можно получить благодаря механизму наследования. Для такого объекта целесообразно

представлять в базе лишь его собственные (ненаследуемые) свойства, а также свойства, которые хотя и являются наследуемыми, но несут дополнительную информацию.

Отношения многообразны по типу и свойствам. Основные типы отношений, используемых в СИИ:

- лингвистические (падежные или ролевые: кто, что, где, когда, зачем, почему, условие, место, время и т.п.);
- атрибутивные: форма, размер, цвет, вкус и т.п.;
- действия, когда каждый глагол рассматривается как отношение между объектом (субъектом) действия и объектом, выражющим некоторую способность (летает, ходит, стоит и т.п.), реализующуюся во времени как процесс;
- логические:  $\neg, \vee, \wedge, >$ ;
- квантифицированные:  $\exists, \forall$ ;
- теоретико-множественные (множество, подмножество, объединение, дополнение, пересечение).

Наиболее многообразна группа лингвистических отношений, в которой могут быть выделены подгруппы временных (*раньше, позже, одновременно* и т.п.), пространственных (*над, под, внутри, рядом* и т.п.), причинно-следственных (*каузальных*) и других типов отношений.

## Свойства отношений

Классификация отношений важна, поскольку разные типы отношений обладают различными свойствами, которые необходимо учитывать при построении моделей вывода в СС. Речь идет о следующих свойствах отношений: рефлексивность (нерефлексивность, антирефлексивность), симметричность (несимметричность, антисимметричность), транзитивность (нетранзитивность, антитранзитивность). Соответственно, каждое отношение характеризуется определенным набором этих свойств.

Определим некоторые важные в дальнейшем свойства отношений.

**Рефлексивность.** Если  $x$  и  $y$  находятся в отношении  $R(x,y)$ , то это отношение рефлексивно, если оно сохраняется при замене  $y$  на  $x$ , т.е. само с собой. Т.е. отношение  $R$  называется **рефлексивным**, если  $xRx$ . Так, отношения «быть похожим на», «быть не старше» (я не старше самого себя) рефлексивны. Отношения «быть братом», «быть старше» не рефлексивны. В графе  $G(R)$  рефлексивного отношения при каждой вершине имеется петля.

Отношение  $R$  называется **антирефлексивным**, если  $R$  может выполняться лишь для несовпадающих объектов, т. е. из  $xRy$  следует, что  $x \neq y$ .

Антирефлексивными являются приведенные выше не рефлексивные отношения. В графе  $G(R)$  антирефлексивного отношения петли отсутствуют.

**Симметричность.** Отношение  $R$  называется симметричным, если при выполнении  $xRy$  выполняется и  $yRx$ . Отношения «быть похожим на», «быть родственником» (но не «быть мужем») симметричны. Другими словами, отношение  $xRy$  сохраняется, если объекты  $x$ ,  $y$  поменять местами:  $xRy \sim yRx$  (знак  $\sim$  означает логическую эквивалентность).

Отношение  $R$  называется **асимметричным**, если  $R \cap R^{-1} = \emptyset$ . Иначе говоря, из двух выражений  $xRy$  и  $yRx$  но меньшей мере одно несправедливо. Граф  $G(R)$  не может содержать одновременно дуги вида  $(x_i, x_j)$  и  $(x_j, x_i)$ . Так, отношение «быть братом» не является ни симметричным, ни асимметричным. Действительно, если Петр - брат Федора, то Федор - брат Петра; но если Игорь - брат Ольги, то неверно, что Ольга - брат Игоря.

Верно утверждение, что если отношение  $R$  асимметрично, то оно антрефлексивно.

Отношение  $R$  называется антисимметричным, если выражения  $xRy$  и  $yRx$  справедливы одновременно только тогда, когда  $x = y$ . Граф  $G(R)$  не может содержать одновременно дуги вида  $(x_i, x_j)$  и  $(x_j, x_i)$  при  $i \neq j$ .

**Транзитивность.** Отношение  $R$  называется транзитивным, если  $xRz$  и  $zRy$ , то  $xRy$ . Отсюда по индукции получаем: если  $xRz_1, z_1Rz_2, \dots, z_{k-1}Ry$ , то  $xRy$ . Например, если выполняются отношения  $x < y$  и  $y < z$ , то выполняется и отношение  $x < z$ . Или: *Ваня старше Пети, Петя старше Кати, следовательно, Ваня старше Кати*; если *Пётр пришёл раньше Ивана, Иван пришёл раньше Фёдора*, то *Пётр пришёл раньше Фёдора*.

Свойство транзитивности отношений играет особую роль, поскольку позволяет отображать в базе знаний не все отношения между объектами, а лишь небольшую часть их. Так, перечисленные выше отношения *Is A*, *AKO* (*A kind of*), *POW* (*Part-of-while* (часть - целое)) являются транзитивными и образуют «скелет» СС, позволяющий более свободно ориентироваться даже в сложной и большой модели. Транзитивность родовидового отношения обеспечивает также возможность наследования свойств не только у представленного в базе знаний родовидового объекта, но и родового для родового. Возможности «сжатия» базы знаний обеспечиваются также транзитивностью родовидового отношения *POW* (является частью).

Свойство транзитивности позволяет отображать в базе знаний не все фундаментальные отношения между объектами, а лишь небольшую часть их. Остальные могут быть получены из представленных в СС с помощью специального механизма поддержки транзитивности, типичного для семантических сетей.

**Ацикличность.** Важным также являются такие свойства, как **ацикличность** и **эквивалентность**. Отношение  $R$  называется **ацикличичным**, если из  $xRz_1, z_1Rz_2, \dots, z_{k-1}Ry$  следует, что  $x \neq y$ .

Например, отношение «быть отцом» ацикличично, но не транзитивно; если  $x$  лучше, чем  $y$ , а  $y$  лучше, чем  $z$ , то естественно считать, что в этом смысле  $x$  лучше  $z$ .

**Утверждается**, что:

- всякое ацикличичное отношение асимметрично;
- всякое антирефлексивное и транзитивное отношение ацикличично.

**Эквивалентность.** Отношение  $R$  называется **отношением эквивалентности** (эквивалентностью), если оно рефлексивно, симметрично и транзитивно (обозначается символами:  $\equiv, \sim$ ).

Например, отношениями эквивалентности являются: отношение «быть на одном курсе» на множестве студентов одного факультета; отношение «иметь одинаковый остаток при делении на 3» на множестве натуральных чисел; отношение подобия на множестве треугольников.

**Обращение** отношений. Ещё одно очень интересное свойство, которым мы часто пользуемся. Если шляпка дороже сумочки, то сумочка дешевле шляпки. Иначе говоря, истинно выражение  $xR_iy \sim yR_jx$ . Здесь вместе с переменой мест объектов меняется и сам тип отношений (*дороже – дешевле, выше – ниже, быстрее – медленнее*). Возникает так называемая смысловая симметричность, т. е. не надо в БЗ вносить всевозможные отношения. Достаточно ввести операцию обращения, а часть отношений, являющихся результатом этой операции, можно исключить. Записывается операция обращения так:  $R_j = R_i^{-1}$ .

## Предикатные семантические сети

Рассмотрим тип СС, формируемых на основе анализа текста естественного языка (ЕЯ).

Элементарное предложение в предикатных СС состоит из следующих компонентов:

- пропозициональная вершина, обозначающая конкретное предложение (высказывание, суждение);
- множество концептуальных вершин, число которых должно совпадать с количеством мест того предиката, который описывается в предложении. Дуги, ведущие к концептуальным вершинам, помечаются метками, означающими порядок записи в предикате (рис. 3).

P1 - обозначение предиката, описываемого предложением; в среднем кружочке - имя предикатного символа (стрелка *Pred*); в остальных кружках - термы, первый - A, второй - B.

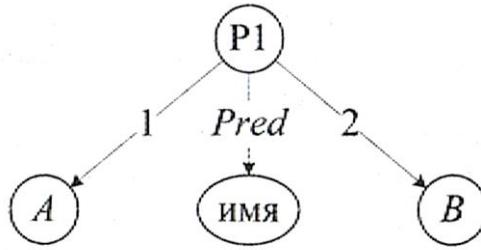
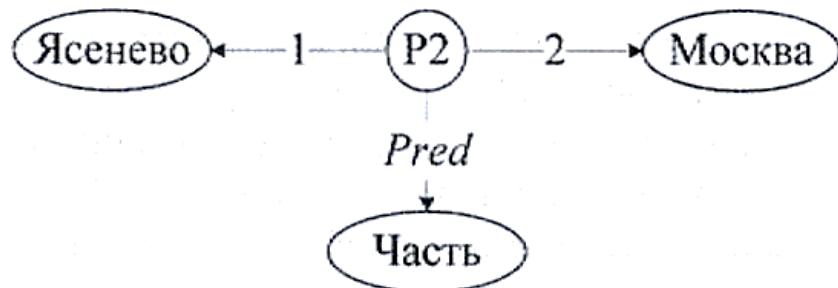


Рис. 3. «Ячейка» предикатной сети

*Пример 1.* Москва есть город (на языке исчисления предикатов - *Город (Москва)*). Предикат Р1 одноместный, имя предикатного символа - *Город*, терм - *Москва*.



*Пример 2.* Ясенево - часть Москвы (Часть (Ясенево, Москва)). Здесь предикат Р2 {Часть} двухместный (Ясенево, Москва).



*Пример 3.* Рассмотрим текст, состоящий из следующих предложений. Ваня взял книгу у Маши. Лена взяла книгу у Пети. Петя любит Машу, а Маша любит конфеты.

Предикатная форма: (*Взял (Ваня, Книга, Маша)*; *Взял (Лена, Книга, Петя)*; *Любит (Петя, Маша)*; *Любит (Маша, Конфеты)*).

Семантическая сеть, соответствующая этим предложениям, представлена на рис. 4.

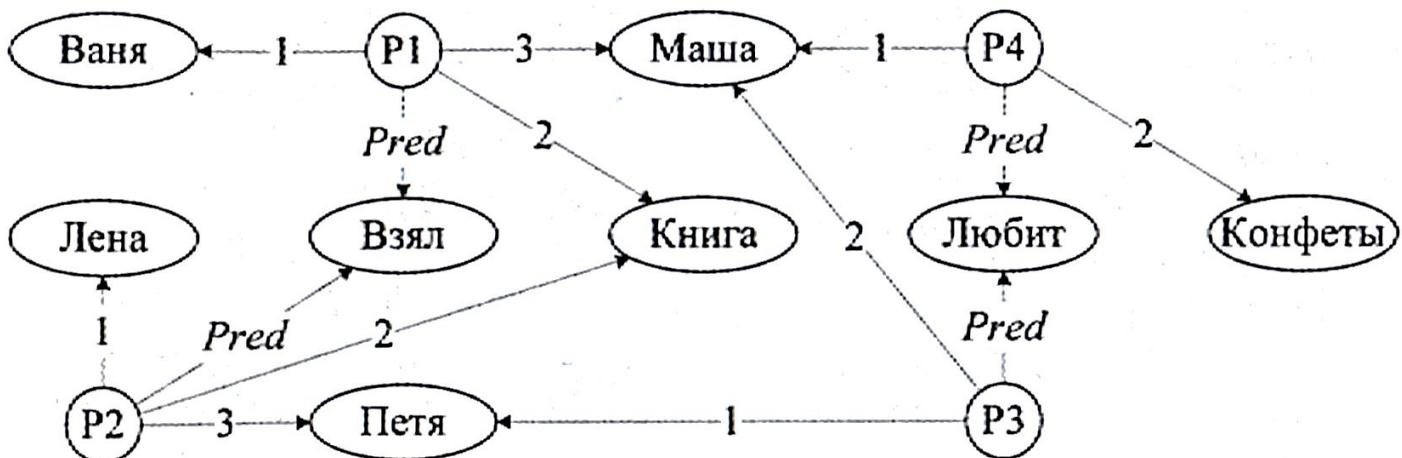


Рис. 4. Предикатная семантическая сеть про книги, любовь и конфеты

Из примеров видно, что:

- 1) частные факты и общие высказывания представляются в виде вершин графа;
- 2) предложение состоит из подграфов (подсетей) специального вида со связями между вершинами-понятиями и предикатными вершинами.

Подграф называется фрагментом сети или предложением. Для объединения составных предложений вводятся специальные вершины «И», «ИЛИ», «НЕ» ( $\wedge, \vee, \neg$ ). Их предикатная форма:  $P1(\wedge(A, B))$ ;  $P1(\vee(D, E, F))$ ;  $P1(\neg G)$  рис. 4.

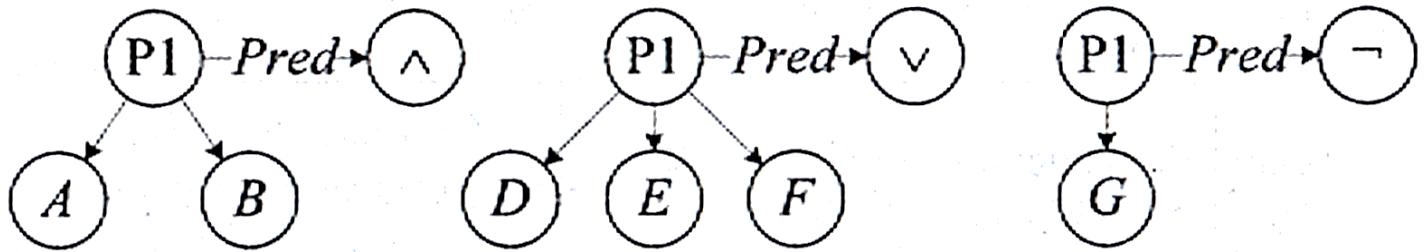


Рис. 4. Примеры «ячеек» предикатной сети, моделирующих пропозициональные связи

**Пример 5.** Говорят, что: ИЛИ Петя любит Машу ( $P1$ ), ИЛИ Паша любит Машу ( $P2$ ), ИЛИ Ваня любит Машу ( $P3$ ), И Ваня выше Пети ( $P4$ ). Рисунок 5 фиксирует эту ситуацию. Здесь введены дополнительные вершины:  $P5(\vee(P1, P2, P3))$  и  $P6(P5 \vee P4)$  для предложений типа «ИЛИ» и «Я».

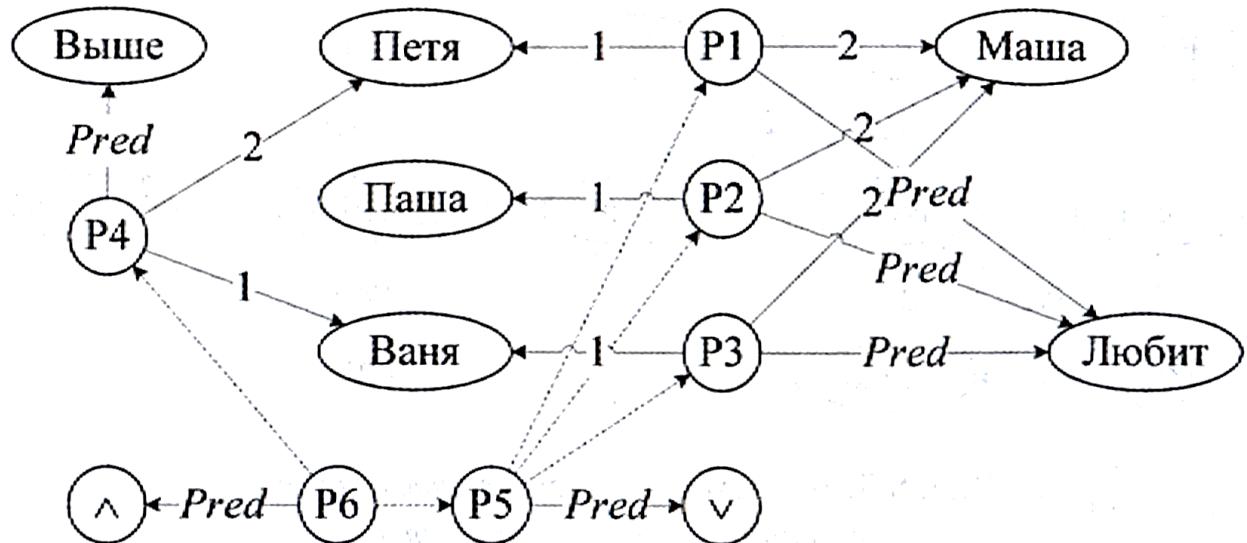


Рис. 5. Предикатная сеть с вершинами типа «И» и «ИЛИ»

По такой сети легко восстановить первоначальный смысл каждого предложения, составить текст на каком-либо другом языке. Действительно, СС подобного типа чаще всего используют в системах машинного перевода, а также для обработки научно-технической информации, например, для аннотирования, реферирования. При этом процесс обработки текста можно представить в виде схемы.



Рис. 6. Схема системы машинного перевода

Лингвистический процессор приводит текст на естественном языке (ЕЯ) к виду СС, которая затем обрабатывается системами представления и обработки знаний для получения ответов на запросы на ЕЯ. Перевод с ЕЯ на язык СС и обратно также осуществляется лингвистическим процессором (рис. 6).

### Атрибутивные семантические сети

Этот популярный тип семантических сетей определяется через систему унарных отношений. Как известно, унарные отношения определяются одноместным предикатом или свойством, описываемого понятия. Например, мы говорим: яблоко красное и вкусное. Это отношение соответствует записям: ЯБЛОКО (ЦВЕТ - красный), ЯБЛОКО (ВКУС - сладкий). Такого рода отношения называют атрибутами, а соответствующие СС - атрибутивными. Здесь предполагается известным: имя объекта (класса, подкласса, единичного экземпляра), перечень его атрибутов и область задания каждого атрибута. Например, атрибут яблока ЦВЕТ можно задать перечнем (красный, желтый, белый, зеленый,...). Совокупность атрибутов вместе с их множествами задания определяют объект как нечто целое в виде отношения:

$$R = \{(A_1 = a_{11}, a_{12}, \dots, a_{1n}), (A_1 = a_{21}, a_{22}, \dots, a_{2n}), \dots\}.$$

При этом очевидно, что атрибуты составляют постоянную (статическую) часть определения объекта, а область задания - переменную (динамическую) часть. Атрибутивное задание в конкретных реализациях чаще всего осуществляется путем позиционирования атрибутов, т.е. фиксации места каждого атрибута в общем описании:

поз. 1 поз. 2 поз. 3 поз. 4

$$R (\text{ЯБЛОКО}) = (\text{ЦВЕТ СОРТ ВКУС ВЕС})$$

$$R (\text{имя объекта}) = (A_1 (\text{имя атрибута}_1), A_2 (\text{имя атрибута}_2), \dots).$$

Такое представление задает как бы свертку *n* унарных отношений в одном. Это чрезвычайно экономит память для сложных описаний и существенно повышает быстродействие. Имя атрибута при этом понимается по умолчанию, на местах значений атрибутов указываются их конкретные значения. Другими словами, мы определяем описание каждого объекта или вершины СС как отношение с перечнем атрибутов и областями их определения. Если

обозначить область определения каждого атрибута как DOM (DOMain - область определения), то можно написать:

$$R = \{(A_1 \text{ DOM}(A_1), A_2 \text{ DOM}(A_2), \dots, A_n \text{ DOM}(A_n)\}.$$

где  $R$  - имя объекта;  $A_1, A_2, \dots, A_n$  - имена атрибутов.

На основе такого представления можно построить множество конкретных описаний. Будем называть каждое такое описание фактом (или примером). В нашем примере мы можем определить, например, факты:

$F_1$ : ЯБЛОКО<sub>1</sub> ЦВЕТ = ЗЕЛЕНЫЙ, ВКУС = КИСЛЫЙ ;

$F_2$ : ЯБЛОКО<sub>2</sub>, ЦВЕТ = КРАСНЫЙ, ВКУС = СЛАДКИЙ.

Максимальное число фактов при трех значениях атрибута ЦВЕТ и двух - атрибута ВКУС - очевидно равно 6.

$R$  задает статическую, или *интенсиональную (внутреннюю)* составляющую описания объекта; конкретные факты:  $F_1, F_2, \dots, F_k$  - динамическую, т.е. *экстенсиональную (внешнюю)* составляющую описания.

Теперь мы можем дать формальное определение атрибутивной семантической сети. Назовем *схемой* или *интенсионалом (INT – INTensional)* некоторого отношения  $R_i$  набор пар вида

$$\text{INT}(R_i) = \langle A_j \text{ DOM}(A_j) \rangle,$$

где  $R_i$  – имя некоторого отношения;  $A_j$  – атрибут ( $j = 1, 2, \dots, n$ ); домен  $\text{DOM}(A_j) = \{a_{ij}\}$  – область значений  $A_j$ .

*Экстенсионалом (EXT – EXTensional)* отношения  $R_i$  называется выражение

$$\text{EXT}(R_i) = \{F_1, F_2, \dots, F_k, \dots, F_n\},$$

где  $F_k = (R_i \langle A_{ijk}, a_{ijk} \rangle)$ , т.е.  $F_k$  – есть факт, описывающий  $j$ -й атрибут отношения  $R_i$  атрибутивной парой,  $a_{ijk}$  – значения  $j$ -го атрибута  $i$ -го отношения. Тогда атрибутивная СС описывается как множество пар вида  $\langle \text{INT}(R_i), \text{EXT}(R_i) \rangle$ , ( $i = 1, 2, 3, \dots, n$ ) по всему множеству отношений  $R_i$ :  $\text{CC}(R_i) = \langle \text{INT}(R_i), \text{EXT}(R_i) \rangle$ ,  $i = 1, 2, 3, \dots, n$ .

Графически атрибутивную СС можно представить в виде звездчатых графов (рис. 7).

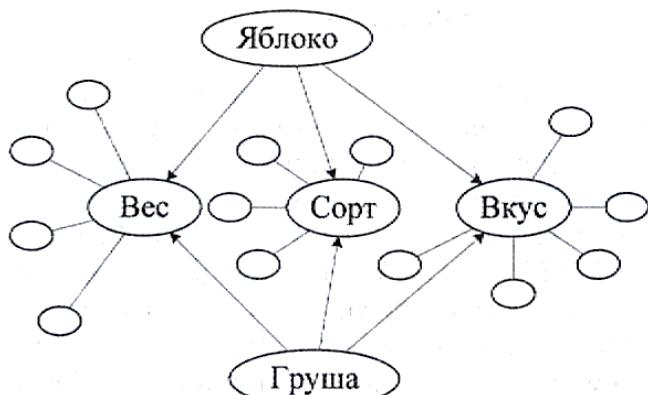


Рис. 7. Пример графа атрибутивной СС (стрелки означают отношение ИМЕЕТ)

## Вывод на семантических сетях

Многообразие СС определяет сложность процессов вывода решений, поскольку каждому типу СС (однородная, иерархическая, сценарная, функциональная и т. п.) соответствует своя модель вывода в зависимости от свойств используемых отношений (рефлексивность, симметричность, транзитивность). В практических реализациях наиболее часто используется механизм вывода, основанный на наследовании свойств от класса объектов к его конкретным представителям (класс - подкласс, класс - экземпляр класса). Более строго, наследование - это способ, которым происходит передача значений свойств объектов друг другу.

Механизм наследования реализуется на иерархии понятий, выстроенной по отношениям IsA и АКО, или Part-of-whole (часть - целое). Иерархия показывает отношение включения понятий более низкого уровня в более высокий. При этом совокупность понятий, вносимых в конкретную СС, зависит от целевого назначения системы и целей ее конкретного использования. Выделяются объекты: классы, подклассы, экземпляры, свойства экземпляров, свойства значений экземпляров.

Класс - это множество объектов, имеющих несколько или большинство общих свойств, т.е. класс является обобщением или концептом (понятием). Класс хранит информацию, присущую всем его объектам. Объекты, когда нужно, могут наследовать от класса эту информацию.

Подкласс - это класс объектов, составляющих подмножество класса более высокого уровня. Например, класс «пассажирский самолет» является подклассом класса «самолет», а «самолет» - подклассом класса «летательные аппараты».

Классы и подклассы также состоят из объектов - экземпляров, наследующих свойства объектов уровнем выше.

Рисунок 8 описывает иерархию отношений между классом ЯБЛОКО и его подклассами БЕЛЬ, АНИС, АПОРТ и отношения подкласс - экземпляр класса.

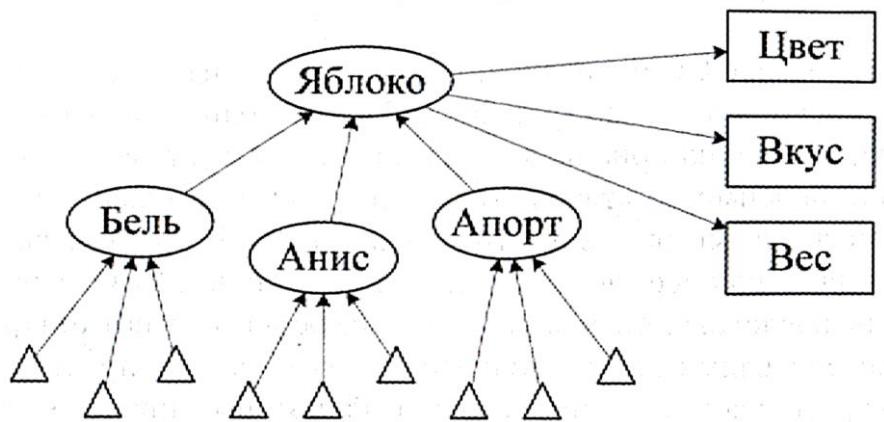


Рис. 8. Пример семантической сети

Наследование свойств означает, что свойства, приписываемые объекту (классу) высокого уровня, автоматически присваиваются объектами более низкого уровня по умолчанию. Но тут есть особенность. Рассмотрим процесс подробнее.

1. Отношение класс - подкласс. Это отношение специализации.

Здесь свойства и значения свойств наследуются по умолчанию по связям специализации (АКО). Пример: **САМОЛЕТ** - *Ту-104* (*Ту-104* наследует многие свойства класса **САМОЛЕТ**). Обратное отношение (Part-of) - отношение обобщения, т.е. подкласс специализирует, а класс - обобщает.

2. Отношение класс - объект. Это - отношение IsA. Свойства наследуются вниз по умолчанию. Это отношение классификации. Обратное отношение - отношение интеграции.

3. Отношение объект (экземпляр) - подобъект (свойства). Это отношение декомпозиции. Объект разлагается на составляющие. Обратите внимание - свойства по этой связи не наследуются! В самом деле, если на линии IsA **МЛЕКОПИТАЮЩИЕ** - **КОШКИ**, **КОШКИ** - экземпляр, то дальнейшее движение по этой линии уже приводит к разделению объекта **КОШКИ** на составляющие: **ФОРМА УШЕЙ**, **ХВОСТ**, **ЛАПЫ** и т.д. Свойства класса **МЛЕКОПИТАЮЩИЕ** здесь уже не наследуются. Обратное отношение - агрегирование. Что еще интересно: если по линии IsA число экземпляров может варьироваться и не влияет на целостность класса, то в обратном отношении Part-of должны присутствовать все части или какой-то обязательный минимум этих частей. Как уже говорилось, одним из наиболее распространенных методов вывода на сетях является метод сопоставления частей сетевой структуры.

Рассмотрим пример сопоставления.

Имеется участок сети базы данных относительно класса **САМОЛЕТЫ** (рис. 9).

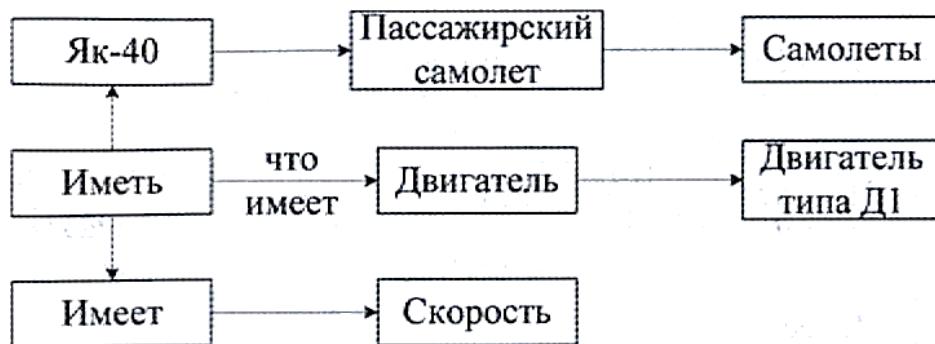


Рис. 9. Участок СС САМОЛЕТЫ

Требуется для начала ответить на вопрос: «Что имеет Як-40?». Участок сети, соответствующий этому вопросу, показан на рис. 10. Для ответа на этот вопрос проводится сопоставление. Сначала ищется вершина **ИМЕТЬ** такая, которая имеет соединение с **Як-40**. Теперь три левые вершины совпали, и по линии «Что имеет» мы делаем вывод: «Як-40 имеет двигатель Д1».

Рассмотрим процесс вывода подробнее. Пусть требуется ответить на вопрос: «Существует ли самолет, имеющий двигатель типа Д1?». Сеть, соответствующая этому вопросу, показана на рис.11. Однако в таком виде она не годится для сопоставления с базой данных. Если же ввести вершину **Як-40** и линию, показывающую, что **Як-40** - **ПАССАЖИРСКИЙ**

**САМОЛЕТ** то уже можно сопоставить вершину **ИМЕТЬ** с вершиной **ИМЕЕТ** а вершину **ДВИГАТЕЛЬ** с **ДВИГАТЕЛЬ типа Д1?** Отсюда следует вывод: «Да, это Як-40».

В процессе вывода модификация базы знаний на СС сводится к удалению и добавлению новых вершин и ребер. Базовые операции поиска информации в сети обеспечивают поиск вершины или ребра по имени, переходы от одной вершины к другой по связям и от одной связи к другой через смежные вершины. Цель поиска - получение знаний, представленных в сети и требуемых для решения задачи.

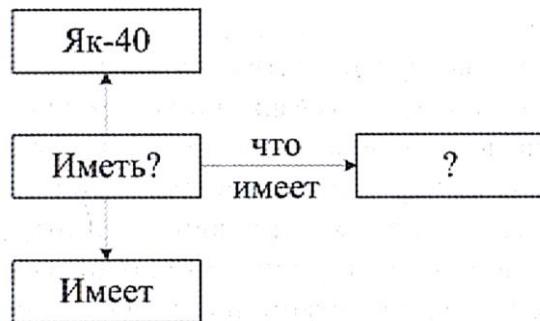


Рис. 10. Участок сети «Что имеет Як-40?»

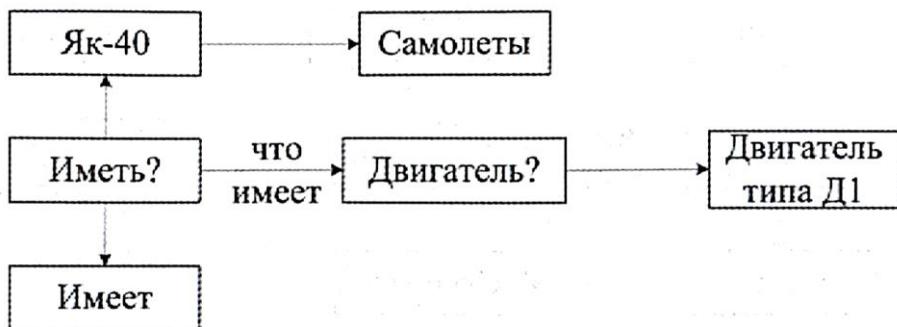


Рис. 11. Подсеть для вопроса «Существует ли самолет, имеющий двигатель типа Д1?»

Одним из мощных средств вывода является операция сопоставления с образцом. Сущность ее состоит в следующем. Информационная потребность, определяющая содержание и цель запроса к базе знаний, описывается автономной семантической сетью -сетью запроса, построенной по тем же правилам и отображающей те же объекты и отношения, которые представлены в системе знаний семантической сетью. Поиск ответа на запрос реализуется сопоставлением сети запроса с фрагментами семантической сети, т.е. путём установления отношения эквивалентности между СС запроса и СС фрагмента общей СС, представляющей систему знаний. Положительный результат отождествления позволяет получить один из ответов на запрос. Все ответы можно получить путем обнаружения всех сопоставимых с сетью запроса фрагментов.

В общем виде операцию сопоставления для СС можно описать следующим образом. Для сети, представляющей систему знаний, задается набор допустимых преобразований, переводящий исходную сеть (или ее фрагменты) в логически ей эквивалентную. Операция

сопоставления выявляет все фрагменты исходной или эквивалентных ей сетей, изоморфные сети запроса. Набор допустимых преобразований для семантических сетей дополняет сети новыми связями, полученными из транзитивности фундаментальных отношений и наследования свойств, но не ограничивается этим. В зависимости от специфики решаемых задач и особенностей того или иного конкретного средства набор эквивалентных преобразований может существенно расширяться. Поскольку теория СС не дает универсальных средств, позволяющих описывать допустимые преобразования, операция сопоставления может рассматриваться как базовая лишь методически (в том смысле, что поиск в СС всегда есть какое-то сопоставление), но не технически (в том смысле, что любой требуемый поиск может быть выражен операцией сопоставления в некотором универсальном языке представления знаний). Именно поэтому базовыми для СС здесь названы не операции сопоставления с образцом, а гораздо более примитивные операции перехода по сети. Фактически совокупность последних в каждом конкретном случае поиска реализует то или иное требуемое сопоставление.

## **Достоинства и недостатки семантических сетей**

1. Простота и прозрачность описания. Однако с увеличением размеров сети существенно увеличивается время поиска, теряется наглядность.
2. Открытость, позволяющая дополнять и модифицировать СС при необходимости.
3. По сравнению с логикой предикатов СС имеют важное преимущество: вся точно известная информация расположена в базе знаний вокруг соответствующей вершины, т.е. фокусируется в одном месте.

Это все были достоинства. В качестве недостатков отметим.

1. Неоднозначность описания.
2. Отсутствие формального аппарата установления противоречивости описания.
3. Сложность внесения изменений.

Основное применение семантические сети находят в системах обработки естественных языков, а также в системах распознавания образов, в которых СС используются для хранения знаний о структуре, форме и свойствах физических объектов. Могут создаваться СС и более сложной структуры, как мы увидим далее.

## Л 6-2. Представление знаний в виде фреймов (дополнительная лекция 2)

Фрейм, его структура и свойства. Вывод на фреймах. Достоинства и недостатки фреймового представления. Гибридные модели представления знаний.

Из предложенных ранее моделей представления знаний в виде системы продукции, семантических сетей, а еще ранее - в первой части - в виде формальных моделей мы видели, каким образом характер предметной области, типы задач для решения в СИИ влияют на выбор конкретной модели организации знаний и вывода результатов. Так, в производственных моделях знания фокусируются на множестве характерных, или ключевых состояний, требующих принятия решения. Соответственно, модель предметной области или БЗ, в конце концов, представляется в виде множества правил типа:

<прототип состояния> → <решение + действия по изменению состояния ПрО, соответствующие данному решению>.

Каждое правило представляет собой элементарный кирпичик знания о целом. Очевидно, что для достаточно сложных предметных областей БЗ могут быть настолько сложными и запутанными, что их разработка и использование становятся проблематичным.

В семантических сетях - другой недостаток. Здесь фокус внимания сосредоточен на понятиях предметной области, отождествляемых с реальными объектами (сущностями, процессами). Однако представление в семантических сетях знаний недекларативного характера, например, процедурного, затруднено так же, как и информации о решениях. Поэтому естественно, что следующим началом в развитии СИИ стал поиск такого представления, который был бы, с одной стороны, достаточно крупным фрагментом знания о мире, а с другой - сохранял бы наиболее достоинства других представлений, такие как возможность использования процедурных знаний (как в производственных моделях), наглядность и иерархичность (как в семантических сетях), возможность моделирования правдоподобного вывода (как в формальных системах). Поиски привели к модели представления знаний в виде фреймов.

### Фрейм, его структура и свойства

Впервые понятие фрейма (frame - рамка, каркас, структура) было введено М. Минским в 1975 году. В основу его положены представления гештальтпсихологии, занимающейся изучением восприятия человеком внешнего мира в форме целостных фрагментов. Таким фрагментом может быть объект внешнего мира с его наиболее характерными свойствами, ситуация, процесс и т.п. **По Минскому**, фрейм - это структура данных, содержащая минимально необходимую информацию для представления класса объектов (явлений или процессов), которая однозначно определяет эти объекты.

**По определению Постелова Д.А.**, фрейм - это термин для обозначения описания какого-либо объекта или явления, обладающего тем свойством, что удаление из этого описания любой его части приводит к потере определяющих суть данного объекта или явления свойств.

**Еще одно определение:** фрейм - это структура данных для представления стереотипных ситуаций. Различают **фрейм-прототипы** (protoфреймы) и **фрейм-примеры** (или фрейм-экземпляры).

**Фрейм-прототип** содержит знания, общие для всех частных случаев, т.е. примеров, а фрейм-пример содержит знания, отличающие частный случай от общего.

В наиболее простом случае фрейм представляется в следующем виде:

$$F = (< I >, < r_1, f_1, \text{ПП}_1 >, < r_2, f_2, \text{ПП}_2 >, \dots, < r_n, f_n, \text{ПП}_m >),$$

где  $I$  - имя фрейма;  $r_i$  - имя слота;  $f_i$  - значение слота ( $i = 1, \dots, n$ );  $\text{ПП}_j$  - имя присоединенной процедуры ( $j = 1, \dots, m$ ).

В качестве **имен** фреймов могут фигурировать имена объектов, событий процессов и т.п. **Слотами** выступают характерные свойства или атрибуты описываемых объектов с именем  $I$ .

В качестве **значений** слотов  $f_i$  могут выступать:

- имена других фреймов,
- имена процедур,
- значения слотов могут задаваться массивом, матрицей, списком, любым типом данных.

Из одного фрейма можно сослаться на один или несколько других, образуя сложные сетевые структуры. Таким образом, существует возможность ссылки из одного фрейма в один или несколько других, что позволяет вводить структурированные связи между фреймами. При этом возникает однородная СС со сложными вершинами, допускающими дальнейшую развертку. Семантическую сеть можно рассматривать как частный случай фрейма, так как очевидно, что любая СС может быть переведена во фреймовое описание и наоборот. В случае рекурсивного вложения фреймов друг в друга порождается иерархическая система фреймов.

Рассмотрим пример, представленный на рис. 1 и описывающий в виде фреймов понятие **ДЕЛОВАЯ ПОЕЗДКА**.

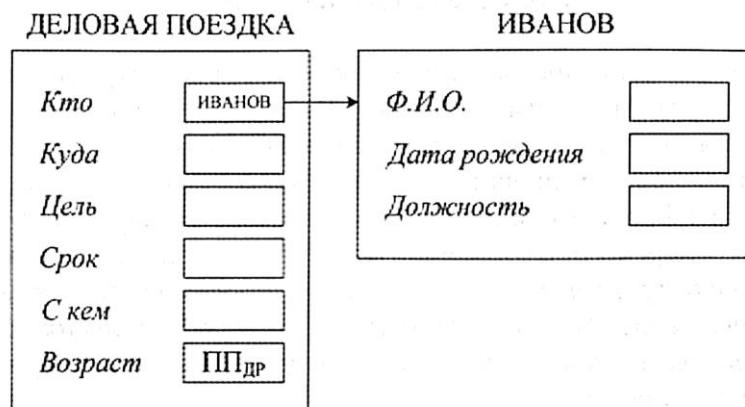


Рис.1. Структура иерархической системы фреймов, описывающих понятие ДЕЛОВАЯ ПОЕЗДКА

Очевидно, что на рис. 1 показаны не все вложения, но принцип ясен. В общем случае слотами могут быть незаполненные подструктуры фрейма, заполнение которых приводит к тому, что данному фрейму ставится в соответствие некоторая конкретная ситуация, объект или явление.

Фрейм с не полностью заполненными слотами соответствует протофрейму. Заполнение слотов формирует соответственно фрейм-пример.

Слоты могут быть **терминальными** или **нетерминальными**.

**Нетерминальные** слоты служат именами других фреймов, составляющих содержание данного слота. **Терминальные** слоты задают список соответствующих значений. Так, слоты КУДА, ЦЕЛЬ, СРОК и др. являются терминальными, а слот КТО - нетерминальный.

Фреймы типа ДЕЛОВАЯ ПОЕЗДКА называют еще ролевыми, т.к. в качестве слотов служат указатели ролей типа: кто, что, куда, когда, где, почему и т.п.

При этом выделяются слоты **обязательные** и **необязательные** для заполнения. Например, слот С КЕМ может быть необязательным. Значениями слотов могут быть также присоединенные процедуры (ПП). Необходимость в них возникает, когда активизация фрейма возможна лишь при выполнении некоторого предусловия. Проверка предусловия осуществляется при наличии ссылки на него во фрейме.

Например, слот ВОЗРАСТ заполняется в результате обращения к присоединенной процедуре (ППдр) и вычисляется в результате обращения к значению слота ДАТА РОЖДЕНИЯ соответствующего фрейма. Примерами ПП могут быть, например, процедуры заполнения слотов, активизации других фреймов, выполнения некоторых дополнительных действий, проверка условий, введение новых слотов или прототипов и т.д. Таким образом, слоты могут заполняться значениями, символами, множествами, а также управляющей информацией, благодаря которой фреймы и организуются в структуру.

Если сравнить фрейм с СС, то очевидно, что **протофрейм** соответствует **интенсиональному** описанию СС, а **фрейм-пример** - **экстенсиональному**. Отсюда следует основной механизм соотнесения фрейм-прототипов и фрейм-примеров. Извлекается фрейм-пример и сравнивается с множеством фрейм-прототипов. Если один из них «накладывается» на фрейм-прототип, то соответствующий образец активизируется. Условие перехода к другому фрейму может записываться как в декларативной, так и в процедурной формах. Можно выделить два типа процедур:

- проверяющие некоторые условия,
- разрешающие выполнять некоторые действия (например, вычисления).

Первый тип присоединенных процедур называют **процедурами-слугами**, вторые - **процедурами-демонами**. Последние активизируются при каждой попытке добавления или удаления данных из слота (по умолчанию). Очевидно, что формирование множества присоединенных процедур (ПП) связано с поддержкой основных типов процессов обработки фреймовых структур. Используются следующие основные типы процессов:

- 1) создание нового экземпляра фрейма;
- 2) активизация фреймов;
- 3) организация вывода путем последовательного поиска и активизации сети фреймов до нахождения наиболее соответствующего и построения на его основе экземпляра фрейма.

ПП могут также реализовывать эвристики, направленные на поиск необходимой для заполнения слотов информации.

## **Вывод на фреймах**

Для организации процесса вывода используются механизмы наследования информации и присоединенных процедур. Наиболее системная структура фрейма, удобная для описания, приведена на рис. 2.

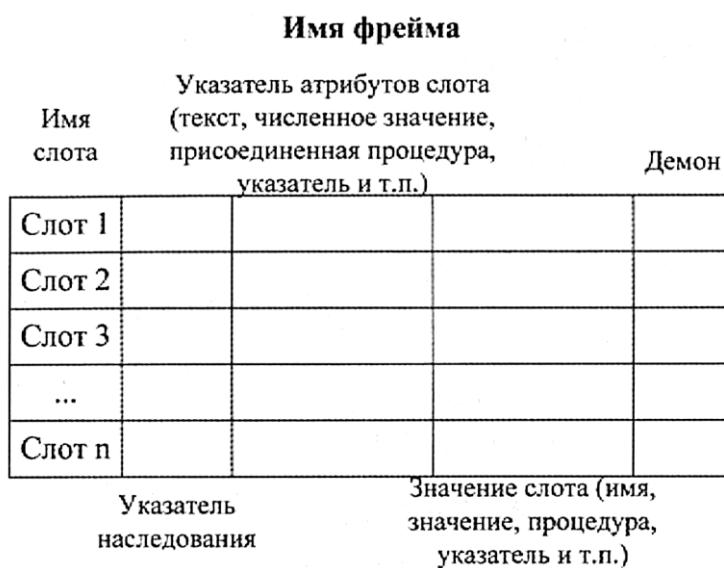


Рис. 2. Структура данных слота

Здесь:

- 1) *имя фрейма* - уникальный идентификатор. Для каждого фрейма определяется слот IsA - родитель данного фрейма - и слот дочерних фреймов, задаваемый списком имен (указателей). В список слотов включается ряд слотов, необходимых для организации интерфейса с пользователем, для организации внутрисистемных процессов и т.п. В их число входят, например, слоты для определения имени пользователя, даты определения, даты изменения текста, комментария и т.п.;
- 2) *имя слота* также служит его идентификатором, т.е. уникальным именем во фрейме, к которому он принадлежит. Могут вводиться специальные идентификаторы, отождествляемые с отношениями IsA, Part-of, причина-следствие и др. Кстати, в качестве слота может выступать слот «Условие», а его значением может быть продукционное правило. Таким образом, реализуется возможность использования продукции для организации вывода. Вводятся также слоты типа: «Дата модификации фрейма», «Комментарий», «Отношение» и др. Такие слоты называются системными и используются для редактирования БЗ и управления выводом;

3) указатели наследования необходимы только для фреймовых систем иерархического типа. Типичными указателями могут служить: «уникальный» (*unique*: *U*), «игнорировать» (*override*: *O*), «такой же» (*same*: *S*), «установить» (*range*: *R*) и т.п. Метка *U* показывает, что каждый фрейм может иметь слоты с различными значениями; *S* - все слоты должны иметь одинаковое значение; *R* - значения слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фреймов верхнего уровня; *O* - при отсутствии указания значение слотов фрейма верхнего уровня становится значением слота фрейма нижнего уровня;

4) *указатель атрибутов слота* показывает тип данных - число, указатель другого фрейма, т.е. является именем (FRAME (указатель на другой фрейм), INTEGER (целый), REAL (действительный), BOOL (логический), TEXT (текстовый), LIST (список), TABLE (таблица), EXPRESSION (выражение) и др.);

5) *значение слота*, очевидно, должно совпадать с указанным типом данных этого слота, а также должно выполняться условие наследования;

6) *демон* - присоединенная процедура. Примеры таких ПП: If-needed, If-added, If-removed и т.д. Эти процедуры запускаются автоматически при выполнении некоторого условия. ПП-слуга - обычно программы процедурного типа, называемые служебными (в языке LISP), методами (в языке Smalltalk). Процедура запускается по сообщению из другого фрейма.

**Как таковой формальный механизм вывода отсутствует и реализуется с помощью механизма наследования и присоединенных процедур. Это позволяет объединять возможности сетевого и иерархического представления знаний.**

Из данного примера видно, насколько разнообразными могут быть конкретные реализации СИИ фреймового типа, а также их достоинства.

Приведем пример записи иерархии фреймов по отношению ISA с указанием фрейма более высокого уровня, из которого наследуются свойства (рис. 3).

ПТИЦА (ЛЕТАЕТ по умолчанию ДА) (ИМЕЕТ КРЫЛЬЯ) (ИМЕЕТ ПЕРЬЯ)
ОРЕЛ (IsA ПТИЦА) СОКОЛ (IsA ПТИЦА) (УМЕЕТ ОХОТА)
СТРАУС (IsA ПТИЦА) (ЛЕТАЕТ НЕТ)
ПИНГВИН (IsA ПТИЦА) (ЛЕТАЕТ НЕТ)
ДЖО (IsA СОКОЛ)
ПИККОЛО (IsA ПИНГВИН)

Рис. 3. Структура данных слота

Значение слота ЛЕТАЕТ по умолчанию означает, что если нет специальной оговорки, все подмножества ПТИЦ и отдельные ПТИЦЫ - летают.

Остальные свойства (ИМЕЕТ КРЫЛЬЯ), (ИМЕЕТ ПЕРЬЯ) наследуются всеми элементами, связанными отношением IsA.

Функция значений «по умолчанию» очень популярна в представлении фреймов, поскольку позволяет однозначно определять значения слотов на иерархии.

Например, если задать вопрос к базе фреймов: способен ли к охоте Джо? Система обращается к фрейму СОКОЛ и выясняет, что сокол умеет охотиться. То есть это свойство наследуется из фрейма более высокого уровня. Если мы хотим получить ответ на вопрос: летает ли ПИККОЛО, то системе приходится разрешать коллизию, возникающую при анализе фреймов ПТИЦА (ПИНГВИН есть ПТИЦА) и ПИНГВИН. Свойство ЛЕТАТЬ не определено во фрейме ПИККОЛО, поэтому система обращается к фрейму ПИНГВИН по указателю IsA. И здесь есть прямое указание, что пингвин не летает. Это свойство наследуется во фрейме ПИККОЛО. И хотя указатель нас отправляет к фрейму ПТИЦА, где по умолчанию стоит значение ДА, предпочтение отдается значению в слоте ПИНГВИН, откуда была последняя ссылка.

Часто языки фреймового типа называют объектно-ориентированными. Таким языком является Smalltalk. Другими примерами языков программирования фреймового типа являются языки FML (Frame Manipulation Language), FRL (Frame Representation Language), KRL (Knowledge Representation Language), являющиеся надстройками над LISP-системами. Очень часто такие языки также называют объектно-ориентированными, но, для предотвращения путаницы с языком Smalltalk, обычно сохраняют название «язык фреймового типа».

## **Достоинства и недостатки фреймового представления**

К достоинствам фреймового представления знаний следует отнести то, что оно:

- 1) обеспечивает эффективную реализацию процедур вывода;
- 2) имеет возможность логических скачков, т.е. немонотонного вывода;
- 3) обеспечивает возможность образования СС фреймов, что дает большую экономию памяти при представлении информации за счет наследования свойств фреймов более высоких уровней во фреймах более низких уровней;
- 4) обеспечивает хорошее соответствие реальной действительности;
- 5) позволяет комбинировать различные модели представления знаний, объединяя их достоинства и компенсируя их недостатки.

К недостаткам отнесем:

- 1) каждый фрейм представляет собой достаточно сложный фрагмент знаний. Поэтому удаление или включение нового фрейма - весьма болезненная процедура, так как должна предусматривать и удаление всех составляющих элементов, которые могут быть составными частями других фреймов;

- 2) достаточно сложно осуществлять на фреймах представление временных процессов;
- 3) отсутствует формальная теория вывода на фреймах. Поэтому на инженере по знаниям целиком лежит ответственность за корректность организации иерархии фреймов и их заполнения.

## Гибридные модели представления знаний

При проектировании баз знаний сложных предметных областей, в которых необходимо представление знаний всех известных типов - иерархического, декларативного, процедурного, разрабатываются гибридные модели, сочетающие продукционную, семантическую и фреймовую модели.

Наиболее популярными являются следующие **гибридные модели**: семантико-продукционная, семантико-фреймовая, фреймо-продукционная, семантико-фреймо-продукционная. В таких моделях разработчики стремятся компенсировать недостатки моделей одного типа достоинствами других. Например, недостатком продукционной модели является неэффективность вывода и нечитабельность (т.е. трудность понимания того, что делает система в процессе вывода) при больших БЗ. Этот недостаток хорошо компенсируется фреймовой моделью, когда декларативные знания представляются в виде фреймов иерархического типа, а процедурные знания - в виде слотов обращения к присоединенным процедурам. В итоге формируется иерархия, вершинами которой являются фреймы, моделирующие левую часть правил-продукций, а правая часть правил моделируется слотами - присоединенными процедурами. Если знания имеют слабую иерархичность или вложенность, то удобнее семантико-фреймовая модель, в которой вершины задаются фреймами, а связи между ними описываются семантической сетью.

Во фреймо-продукционной модели база знаний состоит из продукции, задающих причинно-следственные отношения между простыми и сложными объектами (любыми сущностями) моделируемой предметной области. При этом в качестве сложных объектов выступают фреймы. На основе объектов-условий определяется значение выделенного объекта-цели, играющего роль одной из гипотез решения. Вывод истинной (т.е. наилучшей с точки зрения имеющейся БЗ) гипотезы-решения может осуществляться как по прямой, так и по обратной стратегиям. При прямом выводе обход дерева условий выполняется от корня до соответствующей заданным условиям гипотезы. При обратном выводе выполняется проверка истинности условий, при которых предполагаемая гипотеза оказывается истинной. Возможно применение и смешанной стратегии вывода.

## **Лекция 7. Продукционная модель представления знаний**

Форма представления знаний. Продукционная модель знаний. Механизм вывода в производственной системе знаний. Стратегии управления выводом решения.

Эвристические принципы управления правилами. Достоинства и недостатки производственной модели.

Если проследить за речью человека (не ее содержанием, а структурным построением фраз), то легко заметить, что значительная часть наших мыслей оформляется в виде правил типа «ЕСЛИ (далее следует ряд условий), ТО (далее следует что надо сделать)».

Примеры: «Если ты будешь хорошо учиться, то папа купит тебе велосипед», «Закалаясь, если хочешь быть здоров», «(Если) Не продается вдохновение, но (то) можно рукопись продать» и т.п.

Несмотря на то, что такая форма представления знаний чрезвычайно популярна для человека, она является достаточно сложной для реализации в виде жестких алгоритмических схем. Проблемы появляются в связи с тем, что знания многообразны как по уровням, так и по спектру, допускают неоднозначное толкование, чаще всего имеют ассоциативный характер использования. Трудно четко разложить их на причины и следствия и описывать их, например, на языке исчисления высказываний и предикатов. Добавим еще, что знания постоянно меняются, пополняются, модифицируются и приспособливаются к условиям среды.

### **Форма представления знаний**

Поиск удобной модели для представления знаний типа «Если <условия>, то <действия>» привел к идеи введения продукции в качестве эквивалента представления знаний и механизма работы СИИ.

### **Продукционная модель знаний**

Термин «продукция» был введен американским логиком Е. Постом в 40-х годах в работах по обоснованию и формализации алгоритмических систем. Продукцией называется правило вида  $\alpha \rightarrow \beta$ , где  $\alpha, \beta$  - слова над некоторым алфавитом А. Совокупность правил типа  $\alpha \rightarrow \beta$  получила название системы продукции.

Рассмотрим этот механизм представления, интерпретируя левую и правую части правил продукции как логические функции. Наиболее популярная форма такая:

$$P(x, y) \rightarrow A \& D. \quad (1)$$

Здесь  $P(x, y)$  - логическое выражение;  $A$  - заключение;  $D$  - действие, то есть продукция - это пара <антecedent, consequent>, а « $\rightarrow$ » - знак секвенции. Для простоты изложения кванторы опущены. Выход за рамки формальной системы осуществляется путем добавления такой понятийной категории, как действия.

Если при замене  $x$  и  $y$  конкретными значениями  $a \in X, b \in Y$   $P(a, b)$  принимает значение И, то выдается заключение  $A$  и выполняется действие  $D$ . Формы заключения могут быть различными:

$P(x, y) \rightarrow A$  (здесь  $A$  - статическое высказывание, истинное всегда);

$P(x, y) \rightarrow D$  (здесь  $D$  - одно или несколько действий, не требующих интерпретации);

$P(x, y) \rightarrow A(x, y) \& D(x, y)$  (здесь  $A(x, y)$  - логическая функция, истинная при  $P(x, y) = \text{И}$ ,  $D(x, y)$  - действие, которое доопределяется путем замены  $x, y$  на  $a, b$  соответственно).

В отличие от импликации случай  $\text{И} \rightarrow \text{Л}$  здесь не рассматривается. Это значит, что при истинности левой части реализуется все, что рекомендуется в правой, то есть продукция истинна всегда.

В простейшем случае правая и левая части продукции могут быть простыми высказываниями. Главное отличие продукции от логических формул - они допускают более широкий спектр интерпретации.

В наиболее общем виде под продукцией понимается выражение:

$$<(i), Q, N, P \rightarrow A, D>.$$

$i$  - уникальное имя продукции, с помощью которого продукция выделяется из всего множества. В качестве имени может выступать любая цепочка символов или цифр, но чаще - лексема, отражающая суть (смысл) данной продукции.

$Q$  - характеризует сферу применения данной продукции. Такие сферы легко выделяются в системах знаний человека по принципу «полки». На одной «полочке» хранятся знания о том, как добраться до работы, на другой - как варить кашу и т.п. Разделение знаний на отдельные сферы экономит время на поиск нужных в конкретных условиях знаний.

$P \rightarrow A$  - называется ядром продукции. Это основной элемент. Допускается форма: если  $P$ , то  $A$ , иначе  $B$ . При этом секвенция ( $\rightarrow$ ) истолковывается в обычном, логическом смысле как знак логического следования  $A$  из истинного  $P$  (в случае, если  $P$  не является истинным, как мы уже говорили, то об  $A$  ничего сказать нельзя).

$N$  - есть условие применимости ядра продукции. Обычно, если  $N$  принимает значение И, то ядро продукции активизируется. В противном случае продукция не может быть применена.

Элемент  $D$  описывает постусловия продукции. Они активизируются лишь в случае, если ядро продукции реализовалось. Постусловия описывают действия и процедуры, которые необходимо выполнить после реализации ядра.

Ядра продукции можно классифицировать по различным основаниям. Прежде всего все ядро делится на два больших типа: детерминированные (точно определенные) и недетерминированные (вероятностные). В последнем случае  $A$  может не выполняться и при истинности  $D$ . Ядро продукции интерпретируется тогда так: если  $P$ , то, возможно,  $A$ . Сама возможность определяется некоторыми оценками реализации ядра. Для формирования оценок использует-

ся вероятностный или какие-либо другие подходы. В простейших случаях продукции могут быть такими:

- если  $P$ , то с вероятностью  $p$  реализовать  $A$ , или,
- если  $P$ , то с большой долей уверенности  $A$ .

Мы столкнулись здесь с нечеткостью определения оценок, что и стало предметом изучения нечеткой логики.

Детерминированные продукции могут быть однозначными и альтернативными (связка «или»). Здесь тоже нужен свой механизм выбора альтернативы из нескольких возможных. Это может быть, например, механизм присваивания весов, вероятностных или экспертных оценок. Например: если  $P$ , то чаще надо делать  $A_1$ , реже  $A_2$ . Для решения задач прогнозирования могут вводиться оценки типа «с вероятностью  $p$  можно ожидать  $A_1$ ».

Способности выбора и конструирования форм продукции в каждом конкретном случае определяют культуру и возможности специалиста по проектированию интеллектуальных систем (инженера по знаниям).

Механизм вывода в производственной системе знаний

Для того чтобы «запустить» машину вывода при наличии системы производственных правил, очевидно, необходимы следующие компоненты:

- задание значений переменных, используемых в продукции;
- сами правила-продукции;
- механизм интерпретации правил.

Значения переменных обычно задаются в так называемой **рабочей базе данных** - РБД (рабочая память) в виде набора конкретных фактов, формат которых согласуется с форматом правил. Совокупность правил-продукций является сердцем системы и называется **базой знаний или правил** (БЗ). Механизм интерпретации правил как раз и является **механизмом вывода** (МВ).

Рассмотрим механизм вывода в производственной системе. Как следует из сказанного, под производственной системой (ПС) понимается совокупность:

$$\text{ПС} = \langle \text{РБД}, \text{БЗ}, \text{МВ} \rangle. \quad (2)$$

В каждом цикле интерпретации выполняются следующие процессы:

- определяется множество означиваний, то есть выделяется набор продукции БЗ и фактов из РБД, согласующихся между собой;
- выполняются сами означивания (сопоставления), то есть реализация указаний из правой части продукции.

В производственной модели легко просматриваются аналогии: РБД - наборы фактов, описывающих текущее состояние предметной области;

БЗ - модель предметной области;

МВ - модель механизма принятия решений (вывода) - прямой аналог человеческим рассуждениям.

Таким образом, обобщенный алгоритм работы ПС состоит из:

- задания модели текущего состояния предметной области (прямой диалог, моделирование, генерация, интерполяция и т.д.);
- интерпретации текущего состояния предметной области (то есть РБД) на БЗ и выработки вариантов решения;
- выбора варианта решения и выдачи его на выход системы для пользователя;
- изменения состояния модели предметной области (РБД) путем выполнения действий и процедур, рекомендованных в правых частях правил.

Самое замечательное в этом механизме то, что управление процессом вывода осуществляется путем изменения состояния РБД и не затрагивает БЗ.

### **Стратегии управления выводом решения**

Это управляющий компонент ПС. Он определяет порядок применения правил и устанавливает, имеются ли еще факты в РБД, которые могут быть изменены для продолжения работы. Основных стратегий вывода - две: прямая - «ЛЧП → ПЧП» (левая часть правила → правая часть правила) и обратная (ПЧП → ЛЧП). В каждом цикле может срабатывать только одно правило. Процесс управления состоит из выполнения четырех основных операций:

- сопоставления (образец правила сопоставляется с имеющимися фактами в РБД);
- выбора (если в конкретной ситуации могут быть применены сразу несколько правил, то есть правила конфликтуют за возможность первого применения, то из них выбирается наиболее подходящее);
- срабатывания (если образец правила при сопоставлении совпал с какими-либо фактами из РБД, то соответствующее правило срабатывает);
- изменения состояния РБД на новое (РБД изменяется путем добавления в нее заключения сработавшего правила. Если в правой части содержится еще и указание на какие-либо действия, то они выполняются настолько, насколько это возможно). Прямая стратегия «ЛЧП → ПЧП» соответствует стратегии вывода **«снизу-вверх»** и осуществляется от ситуации к цели. Алгоритм вывода при этом следующий:

1) определяется ситуация, в которой выполняется условие в левой части (если), и делается логический вывод согласно констатирующей части (то);

2) полученный вывод понимается как условие, выполнение которого следует определить по условной части другого правила и т.д.

Пример 1. Рыбак собирается на рыбалку и рассуждает по следующей схеме.

1. Если вода в реке поднимается, то рыба выходит на мель.
2. Если вода в реке опускается, то рыба уходит в глубину.

3. Если лето влажное, то вода в реке поднимается.
4. Если лето сухое, то вода в реке опускается.
5. Если рыба выходит на мель, то следует применять легкую блесну.
6. Если рыба уходит в глубину, то следует применять тяжелую блесну.

Лето выдалось влажное - какую блесну брать на рыбалку? Мы будем решать эту задачу по алгоритму «снизу-вверх».

- 1) В условной части заданных правил ищем условие, совпадающее с исходной ситуацией «лето влажное» (правило 3). Делаем вывод: «вода в реке поднимается».
- 2) Принимаем этот вывод как новое условие, следствие которого следует определить, и ищем соответствующее правило (правило 1). Делаем вывод: «рыба выходит на мель».
- 3) В условной части правил ищем совпадение для ситуации «рыба выходит на мель» (правило 5). Окончательный вывод: блесну следует брать легкую.

Если представить отдельное правило в виде прямоугольника, темная часть которого означает условие, а светлая - заключение, то прямая стратегия отобразится в виде рис. 1.



Рис. 1. Стратегия «снизу-вверх» для приведенного примера

В общем случае правило удобно изображать в виде графа-дерева. Если оно имеет несколько условий ( $P_i$ ) и одно заключение ( $A$ ), то график будет иметь вид рис. 2. Но правило может содержать и несколько заключений, и несколько действий.

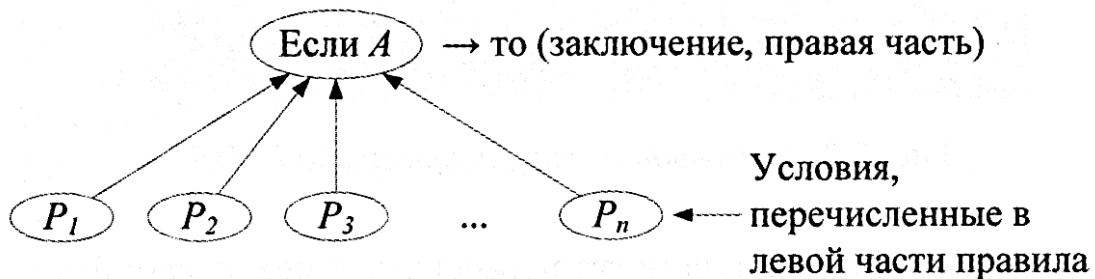


Рис. 2. Структура правила

Очевидно, что правила могут пересекаться между собой:

- по условиям:  $P_i \cap P_j = \emptyset$ ;
- по заключениям:  $A_i \cap A_j = A$ ;
- по действиям:  $D_i \cap D_j = D$  и т.д.

С учетом возможных пересечений база знаний может быть представлена древовидной структурой типа И/ИЛИ-графа, которая хорошо показывает связи между результатами вывода по отдельным правилам.

Вывод происходит «снизу-вверх» от состояния к цели. В случае подтверждений условий фактами БД осуществляется подъем к некоторому заключению, которое является условием вместе с заключениями по другим правилам для получения другого заключения более высокого уровня. В случае альтернативного развития вывода («ИЛИ») выполняются генерация конфликтного набора правил, его разрешение с помощью метаправил и «подъем» далее. Данная стратегия вывода называется прямой или стратегией «от состояния к цели». Могут быть варианты этой стратегии:

- а) возвратная, когда возврат осуществляется опять к первому правилу, то есть одна и та же продукция применяется до тех пор, пока это требуется;
- б) безвозвратная, когда каждая продукция применяется один раз.

Вторая стратегия вывода - обратная - «сверху-вниз» (или от цели к состоянию). Обратный вывод аналогичен прямому, но возникают дополнительные проблемы ограничения конфликтных наборов правил и выбора алгоритма разрешения конфликтов. Есть и другие проблемы, например, оценки условий в условной части. Все они решаются по-разному в рамках соответствующих инструментальных систем производственного типа.

Согласно алгоритму вывода «сверху-вниз» вначале ищут совпадения имени цели с частью то какого-либо правила. Соответствующая ей часть если определяет следующее условие, по которому ищут совпадения с частью то другого правила и т.д. Процесс, таким образом, идет от цели к ситуации. Для простоты понимания рассмотрим процедуру обратного вывода на примере правил, записанных на языке исчисления высказываний.

Пример. Допустим, что в базе знаний имеются правила:

П1  $A \wedge B \wedge C \rightarrow D$ ,

П2  $D \wedge F \rightarrow G$ ,

П3  $A \wedge J \rightarrow G$ ,

П4  $B \rightarrow C$ ,

П5  $F \rightarrow B$ ,

П6  $L \rightarrow J$ ,

П7  $G \rightarrow H$ .

Дано:  $A$  и  $F$  - истинны. Доказать, что  $H$  - истинно.

Алгоритм:

1. Ищем правило, где  $H$  в правой части - П7.
2.  $H$  будет истинным, если  $G$  - истинно; ищем продукцию, где  $G$  в правой части, их две - П2 и П3.  $G$ , таким образом, будет истинным, если истинна хотя бы одна из конъюнкций в П2 и П3. В П2:  $F =$  И по условию, но неясно  $D$ ; в П3:  $A =$  И по условию, неясно  $J$ .
3. Ищем продукцию, в которой  $D$  является целью. Это П1. Здесь  $A$  истинно по условию,  $B$  и  $C$  неизвестны.

4. Из П5:  $B = И$ , так как  $F = И$  по условию.
5. Из П4: так как  $B = И$ , то и  $C = И$ .
6. Обратимся к П1. Все высказывания в левой части здесь истинны, следовательно,  $D = И$ .
7. Так как  $D = И$ , то из П2:  $G = И$ .
8. Из П7:  $H = И$ , что и требовалось доказать.

Теперь можно обобщить обе стратегии вывода на продукциях. В обоих случаях алгоритм начинается с сопоставления исходной ситуации: в первом случае с условием (Если), во втором - с целью (то). В результате могут быть активизированы сразу несколько правил (в нашем примере П2 и П3). Образуется «конфликтное множество», оно разрешается различными способами по заданному критерию выбора. Выбранное правило выполняется, что позволяет осуществить необходимые действия (о действиях - ниже). Обобщенная стратегия показана на рис. 4.



Рис. 4. Стратегия управления выводом на продукциях

Процесс интерпретации очень хорошо может быть проиллюстрирован как поиск решения в пространстве состояний. Поскольку невозможно построить сразу все пути решения задачи, движение в пространстве состояний осуществляется в соответствии с правилами-эвристиками, составляющими содержание критериев.

Каждое правило легко представить в виде отдельного «кирпичика» (модуля, блока и т.д.), управляемого данными. Правила не зависят друг от друга, что позволяет наращивать и изменять БЗ без изменения самого механизма функционирования и БЗ.

### Эвристические принципы управления правилами

Несмотря на «прозрачность» механизма вывода, в продукционных системах (ПС) возникает проблема выбора, то есть проблема активизации правил. Этим занимается система управления, являющаяся частью всей машины вывода. Подходящих продукции в каждый момент времени может быть несколько, и в принципе здесь возможен параллельный вывод. Эта

возможность зависит от имеющегося ресурса и архитектуры вычислительной техники. Используется несколько эвристических принципов «упаковки» продукции в БЗ. Перечислим наиболее популярные из них.

1. Принцип «стопки книг». Наиболее часто используемые продукции актуализируются первыми. Этот принцип целесообразно применять, если продукции относительно независимы друг от друга.

2. Принцип наиболее длинного условия. Первой выбирается продукция, у которой наиболее «длинное» условие выполнимости ядра. Этот принцип целесообразен, когда продукции хорошо структурированы по отношению «частное-общее».

3. Принцип метапродукций. В систему продуктов вводятся специальные правила-метапродукции, задачей которых является разрешение ситуаций в условиях конфликтного набора правил.

4. Принцип приоритетного выбора. На продукцию вводятся статические или динамические приоритеты. Статические формируются априори, чаще всего на основе экспертных оценок. Динамические приоритетырабатываются в процессе функционирования системы продукции и могут отражать, например, такой параметр, как время нахождения продукции во фронте готовых продукции. К данному типу управления относится задание последовательности приоритетов с помощью специальной каузальной (причинной) семантической сети. В этом случае задается некоторый каузальный сценарий, движение по которому определяется складывающимися ситуациями и в каждой вершине которого задана функция выбора очередной продукции из фронта готовых продукции. Если продукции работают над семантической сетью, приоритеты могут формироваться не в системе продукции, а в самой сети.

5. Принцип «классной доски». Основан на идее спусковых функций. При реализации принципа «классной доски» выделяется специальное рабочее поле памяти - аналог классной доски, на которой мелом пишут объявления и стирают их при необходимости. На этой «доске» параллельно выполняющиеся процессы находят информацию, инициирующую их запуск, на нее же они и выносят информацию о своей работе, которая может оказаться полезной для других процессов. Как правило, на «классной доске» выделяются специальные поля для формирования условий применимости ядер продукции, различные для разных сфер применения продукции, специальные поля для записи результатов срабатывания продукции и для записи постусловий, если они адресованы другим продукциям. С принципом «классной доски» может комбинироваться принцип управления с помощью метапродукций, т.к. он требует проверки некоторых условий, которые фиксируются в рабочем поле памяти, а также другие принципы управления.

Если система продукции работает над некоторой сетевой моделью в базе знаний, то необходимо (особенно при альтернативных продукциях) принимать специальные меры защиты от «порчи знаний» работающими продукциями. Для этого также используется рабочее поле

памяти, куда временно переносится фрагмент знаний, которым оперируют продукции. Этот фрагмент фиксируется на специальном поле «классной доски».

### **Достоинства и недостатки производационной модели**

Производационным моделям не хватает строгой теории. Пока на практике при работе с производационными моделями в них используют эвристики. При задании модели проблемной области в виде совокупности продуктов нельзя быть уверенным в ее **полноте и непротиворечивости**. Причина неудач создания теории кроется в расплывчатости понятия продукции из той интерпретации, которая приписывается ядру, а также в различных способах управления системой продукции. Переход к алгоритмической схеме мало что дает, так как при этом исключается основное свойство производационных систем - их модульность и проистекающие из этого асинхронность и параллельность выполнения продукции в системе.

Популярность производационных моделей определяется несколькими факторами.

- 1) Подавляющая часть человеческих знаний может быть записана в виде продукции.
- 2) Системы продукции являются модульными. За небольшим исключением удаление и добавление продукции не приводит к изменениям в остальных продукциях.
- 3) При необходимости системы продукции могут реализовывать любые алгоритмы и, следовательно, способны отражать любое процедурное знание, доступное ЭВМ.
- 4) Наличие в продукциях указателей на сферу применений продукции позволяет наиболее эффективно организовать память, сократив время поиска в ней необходимой информации. Классификация сфер может быть многоуровневой, что еще более повышает эффективность поиска знаний, так как позволяет наследовать информацию в базе знаний.
- 5) При объединении систем продукции и сетевых представлений получаются средства, обладающие большой вычислительной мощностью.
- 6) Естественный параллелизм в системе продукции, асинхронность их реализации делают производационные системы удобной моделью вычислений для ЭВМ новой архитектуры, в которой идея параллельности и асинхронности является центральной.

**Производационные модели имеют, по крайней мере, два недостатка.**

- 1) При большом числе продукции становится сложной проверка непротиворечивости системы продукции. Это заставляет при добавлении новых продукции тратить много времени на проверку непротиворечивости новой системы.
- 2) Из-за присущей системе недетерминированности (неоднозначного выбора выполняемой продукции из множества активизированных продукции) возникают принципиальные трудности при проверке корректности работы системы.

Считается, что если в ИС число продукции достигает тысячи, то мало шансов, что система продукции во всех случаях будет правильно функционировать. Именно поэтому число продукции, с которым, как правило, работают современные ИС, не превышает тысячи.

## ВОПРОСЫ.

1. Крестьянину необходимо перевести через реку волка, козу и капусту. Рядом с крестьянином в лодке может поместиться либо только волк, либо только коза, либо только капуста. Очевидно, что на берегу нельзя оставлять волка с козой, а козу с капустой. Построить БЗ и РБД для случая, когда крестьянин все-таки перевез все, что хотел, на другой берег. Найти решение задачи.

2. К реке подошли три рыцаря с оруженосцами. У берега оказалась одна двухместная лодка. Как организовать переправу при условии, что ни один оруженосец нигде не должен оставаться в обществе других рыцарей без хозяина? Построить БЗ и РБД успешного решения задачи. Найти решение задачи.

3. Тот же случай, но рыцарей с оруженосцами - четыре.

4. Восьмиведерный бочонок полон кваса. Требуется разделить этот квас пополам при помощи двух других бочонков объемом в 3 и 5 ведер. Построить БЗ и РБД. Найти решение задачи.

5. Имеется равнобедренный треугольник. Доказать, что углы при основании равны. Построить БЗ и РБД.

6. Задача о вождении автомобиля. Вождение (увеличение/уменьшение скорости или останов, пропуск сзади идущего транспорта) осуществляются в зависимости от качества дороги (главная, второстепенная), указателей на дороге (ограничение на скорость, переход). Смоделировать процесс управления автомобилем.